3

Model reduction and verification

PANOS Y. PAPALAMBROS

3.1 INTRODUCTION

A successful optimization study depends strongly on the appropriateness of the underlying mathematical model used for performing optimization. Indeed in most cases the model will determine whether a solution may or may not be achieved. Numerical algorithms are very sensitive to the precise mathematical form in which the model is cast. Moreover, preliminary models resulting from early modeling efforts may be poorly bounded so that either no finite optimum exists or an optimum is located by artificially forcing activity of constraints that have little or no engineering meaning. Even when a model is properly bounded the solution may be obtained, some times trivially, by forcing activity of as many constraints as the number of degrees of freedom in the model, thus making further optimization unnecessary. This tends to happen in preliminary model formulations with objective and constraint functions depending monotonically on most of the design variables.

Proper modeling requires experience and good understanding of the numerical algorithms that will be used for computing solutions. There are also some simple and rigorous principles that can be applied to exploit monotonicity properties of the model. This chapter reviews some of the basic ideas for proper model formulation and illustrates by example how they may be applied during an optimization study. Modeling considerations prior to embarking on numerical computations are examined first. Next, an extended example illustrates how a model may be reduced and verified systematically. Finally some automation efforts for such modeling considerations are discussed.

3.2 MODELING CONSIDERATIONS PRIOR TO COMPUTATION

At some point during the design process, it becomes possible to represent design decisions by a precise mathematical optimization statement. Usually a nonlinear programming (NLP) model formulation, such as

\[
\begin{align*}
\text{minimize} & \quad f(x) \quad x \in X \\
\text{subject to} & \quad h_j(x) = 0 \quad j = 1, 2, \ldots, m_1 \\
& \quad g_j(x) \leq 0 \quad j = m_1 + 1, m_1 + 2, \ldots, m_1 + m_2
\end{align*}
\]  

(3.1)
Model reduction and verification

is adequate. In equation (3.1), \( f, h \) and \( g \) are the scalar objective, equality and inequality constraints respectively, \( x \) is the vector of design variables \( x_1, x_2, \ldots, x_n \), and the set \( X \) is a subset of the \( n \)-dimensional real space \( R^n \), included in the statement to indicate that there may exist other restrictions on the variables that are not explicit in the constraints, for example, integer values. (Note that boldface characters indicate vector quantities.) Here the objective is shown as a single one, since multiplicative formulations are usually transformed to a single objective. The above mathematical statement will be referred to as the design optimization model cast into the so-called negative null form, a standard form where the objective is to minimize, while equality and inequality constraints are written with zero and less than or equal to zero, respectively, placed at the right-hand side of the equation.

3.2.1 Explicit vs implicit models

The design model implicitly assumes the existence of an analysis model used to evaluate the functions \( f, h, g \). For example, the analysis models may be finite element analyses of structural behavior, system simulation, or curve-fitted experimental data. Analysis models may be explicit or implicit. In an explicit model, the model function forms are explicitly represented by mathematical, usually algebraic, functions; in an implicit model, the model functions are usually represented by a procedure or subroutine, for example, an iterative numerical solution of a set of differential equations.

It is generally preferable to deal with explicit models since then it is easier to make assertions about optimality, for example, by detecting mathematical properties such as monotonicity or convexity (a property that guarantees a unique minimum); also, it is often possible to manipulate the model in order to extract additional information or properties, or to recast it in a more convenient form, as we will see below. From a practical viewpoint, the models in many important classes of design optimization problems are implicit, which poses a substantially increased burden on ascertaining reliability of the optimization results. In practice one seldom seeks a mathematical optimum. Rather, one aims only at so-called satisfying designs (Wilde, 1978), namely, design solutions with a value of the objective function within a known acceptable distance from the (unknown) true optimal value, and frequently only at designs improved by a certain percentage over current ones. Thus the disadvantage of implicit models is significantly tempered.

From an operational viewpoint, it is useful to distinguish between two phases in the optimization process. At the point where the optimal design model has been precisely formulated, the modeling phase of the optimization procedure is concluded. What follows constitutes the solution phase of the procedure. For most realistic models, the solution phase will include an iterative procedure that should converge to the optimum. An iteration begins by selecting an initial design corresponding to a starting value for the design variables and executing the calculation represented by the analysis model(s), so the values for objective and constraint functions are obtained. The cost of subsequent calculations for evaluating the next iterant design increases with the size of the problem, as measured by the number of variables and constraints.

When an implicit analysis model is computationally expensive, the iterative nature of numerical solution strategies makes the use of such a model unattractive. In a preliminary design effort it may be possible to use a simplified explicit model instead of the more accurate implicit one. The process could be similar to the design of physical experiments, such as Taguchi methods. These methods use a carefully planned efficient set of experiments to discover how design variables relate to each other. In the present context the experiments are performed computationally using the 'full' model and the results are curve fitted. After an initial optimization study has been successfully completed with the simplified model, a few iterations using the full model and the identified optimum as a starting point may complete the study.

3.2.2 Design parameters vs variables

The optimization model for any given design problem is more accurately described as

\[
\begin{align*}
\text{minimize} & \quad f(x,p) \\
\text{subject to} & \quad h_j(x,p) = 0, \quad j = 1, 2, \ldots, m_1 \\
& \quad g_j(x,p) \leq 0, \quad j = m_1 + 1, m_1 + 2, \ldots, m_1 + m_2
\end{align*}
\]

where \( p \) is a vector of parameters, \( p = (p_1, p_2, \ldots, p_m) \), and \( P \) is a subset of \( R^m \). The quantities \( p_1, p_2, \ldots, p_m \) are the design parameters and they are considered fixed to specific values during the optimization process. Variables and parameters together give a complete description of the design, and the constraint functions expressed as equalities and inequalities in terms of variables and parameters give a complete description of the design space, i.e. the space that contains all acceptable solutions. The selection of variables and parameters is generally a modeling decision that depends on the position of the designer in the hierarchy of decision-making and on expediency for actually solving eventually the underlying mathematical problem.

When posing the design problem one must consider the position one occupies in the decision-making hierarchy. The higher is the position the fewer are the quantities that one considers as fixed; hence more quantities will be considered as variables than as parameters. A more subtle consideration in classifying parameters and variables comes from the relative ease of solution that results from this classification. For example, material selection is a typical key decision in design. In an optimization model, one material will correspond to several quantities in the model, e.g. density, yield strength, modulus of elasticity. Treating the material as a variable will require inclusion of additional relations in the model that link these quantities, such relations usually given at best in some tabular form. In subsequent numerical processing table look-up or curve-fitting will be then required. Table look-up should not be used as it may introduce discontinuities and nondifferentiabilities that could defy many NLP algorithms. Even with curve-fitting the problem will still contain discrete variables corresponding to the different materials, so continuous NLP algorithms cannot be used and the problem becomes considerably more difficult. Thus the best choice is to treat material as a parameter and to find the optimal solutions for
3.2.3 Constraint activity

The concept of constraint activity is very important in design optimization. Loosely speaking, an active constraint corresponds to a "critical" design requirement, i.e., one whose presence determines where the optimum will be. In traditional structural design, 'fully stressed' designs would correspond to finding minimum weight designs with all stress constraints active. Formally, a constraint (equality or inequality) is called (globally) active if and only if removing it from the model changes the set of (globally) optimal solutions; it is inactive otherwise. An optimal solution is called constraint bound if and only if there are exactly as many active constraints as there are design variables. A constraint is called tight if and only if it is satisfied as a strict equality at the optimum. For continuous variables, active constraints will also be tight. However, not all tight constraints are active. A feasible point is called constraint bound if and only if at this point there are as many tight constraints as there are design variables. As mentioned earlier, in a constraint-bound optimum one needs only to find the active constraints. Thus, constraint-bound points are defined by solving a system of n equations in n unknowns, these equations constituting a working (active) set. Of course, not all such solutions are constraint-bound optima. For some sufficiency criteria see Papalambros (1988a). If the point is optimal, then the working set is the (true) active set. The term 'active' is sometimes used for local minima, but then all relevant properties must be considered valid within an appropriate neighborhood or region. A more comprehensive discussion of activity is given in Pomrehn and Papalambros (1992). Identified active inequalities are indicated with the symbol \( \leq \) instead of \( \leq \).

An active constraint is a direct contributor to optimality and corresponds either to an equality constraint (such as equilibrium) or to a critical design requirement (such as setting the maximum stress equal to the strength of the material used). When a subset of the original constraints is selected for further processing, these constraints together with the objective function constitute a submodel of the original problem. The submodel contains only those constraints that are known to be active or that have been judiciously selected as having a high likelihood to become active. An active set strategy is a set of rules which are used to decide which constraints may be active at the optimum. An active set (at a given point) is a set that contains only those constraints that are considered as being definitely active at the current iteration, which usually means that they are satisfied as equalities (at that point). An extended active set is a set that contains both the currently active constraints and additional ones that are considered candidates for activity in one or more subsequent iterations. Use of such sets is often necessary in order to control computational costs associated with having a small number of variables but a very large number of inequality constraints (only few of which can possibly be active).

Clearly, at a given point the active set is a subset of the extended one. The rules in an active set strategy will in general apply to both an active set and an extended one. In the mathematical programming literature the term 'active set strategy' applies only to the active set, while in the structural optimization literature the term usually applies to the extended active set.

The main idea here is to emphasize that knowing the active constraints as early as possible may be very advantageous both in locating and in verifying an optimal solution.

3.2.4 Monotonicity principles

In many preliminary design models, usually because of simplification of trade-offs, the objective and constraint functions are monotonic with respect to the design variables. A continuous differentiable function \( f(x) \) is increasing (decreasing) with respect to (wrt) a design variable \( x_i \) if \( \frac{df}{dx_i} > 0 \) (\( \frac{df}{dx_i} < 0 \)). We say that \( f \) is coordinate-wise monotonic wrt \( x_i \), or that \( x_i \) is a monotonic variable (in \( f \)). This concept can be extended to other functions but this is not necessary here. We will also assume that all design variables are strictly positive. This is usually the case in design problems; if not, a simple shift in the datum can make the variables positive, e.g. using an absolute temperature scale rather than Fahrenheit or Celsius. Monotonicity analysis is based on two simple principles (see Papalambros and Wilde (1988) for further details).

First monotonicity principle (MP1) In a well-constrained objective function every increasing (decreasing) variable is bounded below (above) by at least one active constraint.

Second monotonicity principle (MP2) Every monotonic variable not occurring in a well-constrained objective function is either irrelevant and can be deleted from the problem together with all constraints in which it occurs, or is relevant and bounded by two active constraints, one from above and one from below.

We will now illustrate how these principles can be used to identify a model that is not well constrained or even to obtain an optimal solution in simple cases. The modeling example of a linear actuator will later demonstrate the approach in a more complicated situation.

Functional relations in a model usually must be rearranged in order to cast them into a standard form, such as equation (3.1). In general, a given relation may be 'equivalent' to several standard forms, some of which may appear monotonic and some not. Therefore, monotonicity should be examined in the original formulation of the problem and, if algebraic manipulations are employed, care should be taken not to disguise monotone properties. However, once a relation has identified monotonicities, these are invariant in the sense that any standard form resulting in identified monotonicities will have exactly the same types of monotonicities. The following implicit function theorem specifically stated for monotone functions is useful in studying how monotonicity properties may be inherited when equalities are eliminated through variable substitution.
Model reduction and verification

Implicit function theorem  Let \( X_i \subseteq R, i = 1, \ldots, n, \) be \( n \) subsets (finite or infinite) of \( R \) and let \( X = \{x | x = (x_1, \ldots, x_n), x_i \in X_i, i = 1, \ldots, n\}. \) Let \( F:X \rightarrow R \) be a function coordinate-wise monotonic on \( X. \) Then, for each \( s \) in the range of \( F \) and for each \( i = 1, \ldots, n, \) there exists a function \( \phi(i, x, x') \) of the variable vector \( x'_i = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \) such that \( \phi(i, x, x') \) is (coordinate-wise) monotonic wrt \( x_i. \) Furthermore, for \( 1 \leq j \leq n \) and \( i \neq j, \) if \( F \) is monotonic in the same (opposite) sense wrt \( x_i \) and \( x_j, \) then \( \phi(i, x, x') \) is decreasing (increasing) wrt \( x_j. \)

Monotonicity principles can be applied directly when the model contains only inequalities, i.e.

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad \phi_j(x) \leq 0 \\
& \quad j = 1, 2, \ldots, m \tag{3.3}
\end{align*}
\]

When active constraints (equalities or inequalities) have been identified, a new round of monotonicity analysis can be performed if the active constraints are eliminated through variable substitution. In the new reduced model the exact functions may not be known but the relevant monotonicities may be implicitly defined using the above theorem.

Before proceeding with some examples, it is interesting to recall that in the presence of differentiability and the usual constraint qualifications the Karush–Kuhn–Tucker (KKT) necessary optimality conditions for the model in equation (3.3) are

\[
\nabla f + \mu^* \nabla g = 0, g \leq 0
\]

\[
\mu^* g = 0, \mu \geq 0 \tag{3.4}
\]

where \( \nabla g \) is the Jacobian matrix of the vector function \( g(x). \) For this case, the two monotonicity principles can be derived trivially from equation (3.4). The principles are more general necessary conditions than the KKT conditions, as they apply globally and do not assume continuity or differentiability. On the other hand, they assume (global or regional) monotonicity, a property that could be considered more restrictive. Their utility is based on the argument that many early design optimization models do have extensive monotonicity properties (see for example, Papalambros and Wilde (1988)). The principles therefore allow a drastic reduction of the number of cases needed to be examined under the complementary slackness conditions of equation (3.4). This can be seen in some simple examples (Papalambros and Li, 1983). Note that when an implicit function is used, a positive (negative) superscript sign for a variable indicates increasing (decreasing) function wrt that variable.

EXAMPLE 3.1

Consider the problem

\[
\begin{align*}
\text{minimize} & \quad f = (x_1 - 3)^2 + (x_2 - 3)^2 \\
\text{subject to} & \quad g_1 = -x_1 \leq 0 \\
& \quad g_2 = -x_2 \leq 0 \\
& \quad g_3 = x_1 + x_2 - 4 \leq 0 \tag{3.5}
\end{align*}
\]

EXAMPLE 3.2

Consider the problem

\[
\begin{align*}
\text{minimize} & \quad f = \frac{1}{2}(x_1^2 + x_2^2 + x_3^2) - (x_1 + x_2 + x_3) \\
\text{subject to} & \quad g_1 = x_1 + x_2 + x_3 - 1 \leq 0 \\
& \quad g_2 = 4x_1 + 2x_2 - 7/3 \leq 0 \\
& \quad x_i \geq 0, i = 1, 2, 3 \tag{3.7}
\end{align*}
\]

From the KKT conditions there are 32 cases to be examined. The objective appears nonmonotonic, but in fact \( \frac{\partial f}{\partial x_i} = x_i - 1 \leq 0 \) because \( g_1 \) implies \( x_i \leq 1, i = 1, 2, 3. \) Hence, by MPI wrt \( x_3 \) we have two cases: if \( x_3 = 1 \) then \( x_2 = x_1 = 0 \) (tight) and \( f = -1/2; \) if \( x_3 < 1, \) then \( g_1 \) is active. Defining \( x_1 = 1 - x_2 - x_3 = \phi_1(x_1^*, x_3^*) \) we can calculate the constrained derivative of the reduced objective \( f_1 \) wrt one of the remaining variables, say \( x_i. \)

\[
\frac{\partial f_1}{\partial x_i} = \frac{\partial f}{\partial x_3}(\frac{\partial \phi_1}{\partial x_1}) + \frac{\partial f}{\partial x_1} = (1/2)(4x_1 + 2x_2 - 2) \tag{3.8}
\]

Now we observe the following: (a) if \( 4x_1 + 2x_2 < 2, \) then \( g_2 \) is inactive with \( f_1(x_1^*, x_2^*) \) and no solution exists; (b) if \( 4x_1 + 2x_2 = 2, \) then \( g_2 \) is inactive and the remaining one-degree-of-freedom problem has the solution \( x_3 = x_1 = x_2 = \frac{1}{2}, \) \( f = -\frac{3}{2}; \) (c) if \( 4x_1 + 2x_2 > 2, \) then the problem becomes \( \min f_1(x_1^*, x_2^*) \) subject to \( g_1(x_1^*, x_2^*) \leq 0 \) with \( x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 1 \) and the solution is \( x_1 = 0, x_2 > 1 \) which is not permitted. The conclusion is a global optimum at \( x_1 = \frac{1}{2}, i = 1, 2, 3. \)

EXAMPLE 3.3

Consider the problem

\[
\begin{align*}
\text{maximize} & \quad f = 0.0201 d^4 w^2 \\
\text{subject to} & \quad g_1 = d^4 w - 675 \leq 0 \\
& \quad g_2 = 3d - 36 \leq 0 \\
& \quad g_3 = n - 125 \leq 0 \\
& \quad g_4 = n^2 d^2 - (0.419)(10^7) \leq 0 \\
& \quad d, n, w \geq 0 \tag{3.9}
\end{align*}
\]

There are eight possible cases dictated by the complementary slackness conditions. The objective is nonmonotonic but we observe that if \( d_1 \geq 3, i = 1, 2, \) there is no solution since \( g_2 \) is violated. Thus \( x_1 < 3 \) for at least one \( i, i = 1, 2, \) and the objective is decreasing wrt at least one \( i, \) which then makes \( g_2 \) always active by the first monotonicity principle (MPI). The problem has now one degree of freedom and the solution can be obtained quickly using constrained derivatives (Wilde and Beightler, 1967). If \( f_1 \) is the reduced objective resulting from elimination of \( x_1, \) and \( g_2 \) we have

\[
\frac{df_1}{dx_2} = (\frac{\partial f}{\partial x_1})(\frac{\partial \phi}{\partial x_2}) + \frac{\partial f}{\partial x_2} = 0 \tag{3.6}
\]

where \( \phi(x_2) = x_1 = 4 - x_2. \) Evaluation of equation (3.6) gives \( x_2^* = 2, x_3^* = 2, f^* = 2 \)
Model reduction and verification

This problem involves maximizing the stored energy in a flywheel and is reduced to the above-mentioned form after some manipulations (Siddall, 1972). Constraint \( g_1 \) is active by MP1 wrt \( w \). Note that the standard negative null form is obtained by using the objective (minimize \(-f\)). Elimination of \( w \) from the objective gives \( f = (0.0201)(675d^2h^2) \). Clearly \( g_4 \) will be active except if \( g_2, g_3 \) combined impose a stricter bound on \( d^2h^2 \). For the given numbers this is not the case, so there are infinite solutions:

\[
\max f^* = (5.68)(10^3), w_0 = 675d^2h^2, n_e = 2047d_e^{-1} \\
16.4 \leq d_e \leq 36 \quad (3.10)
\]

EXAMPLE 3.4

Consider now the problem

\[
\begin{align*}
\text{maximize} & \quad f = x_1^{-2} + x_2^{-2} + x_3^{-2} \\
\text{subject to} & \quad g_1 = 1 - x_1 - x_2 - x_3 \leq 0 \\
& \quad g_2 = x_1^2 + x_2^2 - 2 \leq 0 \\
& \quad g_3 = 2 - x_1x_2x_3 \leq 0 \\
& \quad x_i \geq 0, i = 1, 2, 3
\end{align*}
\]

(3.11)

By MP1 constraint \( g_2 \) must be active providing upper bounds on \( x_1 \) and \( x_2 \). However, \( x_3 \) is unbounded from above since \( f, g_1, \) and \( g_3 \) are all decreasing wrt \( x_1 \). The problem has no solution unless the objective and/or constraint functions are appropriately modified by remodeling. Any nonredundant equality constraint would also serve. The usual practice in such cases is to add a simple inequality constraint, e.g. \( x_3 \leq a, a > 0 \). However, this constraint will be always active by construction, which means that the optimum is in fact determined by this artificial bound. In real problems, such information must be consciously considered because the optimum is essentially arbitrarily fixed by the modeler. We will explore this further in the linear actuator example.

3.2.5 Modeling decisions

It is good practice as one embarks on an optimization study to make a checklist of all the decisions that are required for developing a model. As an example, Table 3.1 shows what this checklist might look like for a structural optimization problem using finite elements for the analysis model (Papalambros, 1988b). This list is certainly not complete and it may change depending on the problem. With experience, the need for such lists may diminish. Yet even experienced users will benefit by going through a detailed itemization of all the decisions they make as they go. It is not uncommon for some overlooked decision to have a strong bearing on the optimization results, particularly when something goes wrong. Furthermore, such decisions can be increasingly automated with the use of artificial intelligence techniques; see, for example, Thomas (1990). This is a particularly attractive approach for developing quickly and accurately optimization
models within a specific domain, for example, optimal trusses or plate girders in bridge construction (Adeli and Balasubramaniam, 1988a; Adeli and Mak, 1988). For further discussion on the use of expert systems in structural design see Adeli and Balasubramaniam (1988b).

### 3.3 Modeling Example: A Linear Actuator

In this extensive example we will go through various modeling steps using several of the ideas mentioned earlier and some new ones. The procedure followed is typical for models with relatively small number of variables and explicit functions, but can be used in other situations to advantage.

#### 3.3.1 Model Setup

The example presented here is part of a larger classroom project dealing with the optimal design of a drive screw linear actuator (Alexander and Rycenga, 1990). The drive screw is part of a power module assembly that converts rotational to linear motion, so that a given load is linearly oscillated at a specified rate. Such a device is used in some household appliances. The assembly consists of an electric motor, drive gear, pinion (driven) gear, drive screw, load-carrying nut, and a chassis providing bearing surfaces and support. The present example addresses only the design of the drive screw, schematically shown in Fig. 3.1.

The objective function in the design model is to minimize product cost consisting of material and manufacturing costs. Machining costs for a metal drive screw or injection molding costs for a plastic one are considered fixed for relatively small changes in the design, hence only material cost is taken as the objective to minimize, namely

$$ f_o = (C_m \pi r' (d_2^2 L_1 + d_2^2 L_2 + d_2^2 L_3)) $$  \hfill (3.12)

Here $C_m$ is the material cost ($\$/lb$) , $d_1$, $d_2$ and $d_3$ are the diameters of gear–drive screw interface, threaded and bearing surface segments, respectively, $L_1$, $L_2$ and $L_3$ being the respective segment lengths.

There are operational, assembly, and packaging constraints. For strength against bending during assembly we set

$$ Mc_o/I \leq \sigma_{all} $$ \hfill (3.13)

**Fig. 3.1** Schematic of drive screw design.

where the bending moment $M = F_s L/2$, $F_s$ being the force required to snap the drive screw into the chassis during assembly, and $L$ being the total length of the component:

$$ L = L_1 + L_2 + L_3 $$ \hfill (3.14)

Furthermore, $c_1 = d_1/2$, $I = nd_1^4/64$ is the moment of inertia, and $\sigma_{all}$ is the maximum allowable bending stress for a given material.

During operation, a constraint against fatigue failure in shear must be imposed:

$$ KT c_3/J \leq \tau_{all} $$ \hfill (3.15)

Here, $K$ is a stress concentration factor, $T$ is the applied torque, $C_3 = d_1/2$, $J = \pi d_1^4/32$ is the polar moment of inertia, and $\tau_{all}$ is the maximum allowable shear stress. The torque is computed from the equation

$$ T = T_m C_2 N_s/N_m $$ \hfill (3.16)

where $T_m$ is the motor torque, $c_2 = 1/16$(lb oz$^{-1}$) is a conversion factor, and $N_s$ and $N_m$ are the number of teeth on the screw (driven) and motor (drive) gear, respectively.

To meet the specified linear cycle rate of oscillation, a speed constraint is imposed:

$$ c_4 N_m S_m/N_s N_T \leq S $$ \hfill (3.17)

where $c_4 = 60$ (number of threads rev$^{-1}$(min s$^{-1}$) is a conversion factor, $S_m$ is the motor speed (rev min$^{-1}$), $N_T$ is the number of threads per inch, and $S$ is the specified linear cycle rate (in s$^{-1}$).

In order for the screw to operate in a drive mode the following constraint must be satisfied (Juvinall, 1983):

$$ \frac{W d_2 \pi f d_2 + N_T^{-1} \cos \alpha}{2 \pi d_2 \cos \alpha - N_T^{-1} f} \leq T $$ \hfill (3.18)

Here $W$ is the drive screw load, $f$ is the friction coefficient, $N_T^{-1}$ is the lead of screw threads, and $\alpha$ is the thread angle measured in the normal plane. There is also an upper bound on the number of threads per inch imposed by mass production considerations,

$$ N_T \leq 24 $$ \hfill (3.19)

From gear design considerations, particularly avoidance of interference, limits on the numbers of gear teeth are imposed:

$$ N_m \geq 8 \quad N_s \leq 52 $$ \hfill (3.20)

Finally, there are some packaging and geometric considerations that impose constraints:

$$ 8.75 \leq L_1 + L_2 + L_3 \leq 10.0 $$ \hfill (3.21)

$$ 7.023 \leq L_2 \leq 7.523 $$ \hfill (3.22)

$$ 1.1525 \leq L_3 \leq 1.6525 $$ \hfill (3.23)

$$ d_1 \leq d_2, d_1 \leq d_3, d_2 \leq 0.625 $$ \hfill (3.24)
Model reduction and verification

Note that several assumptions were invoked in the model above: manufacturing costs remain fixed; a high volume production is planned; Standard Unified Threads are used; the assembly force for the drive screw is concentrated at the mid-point; frictional forces are considered only between threads and load nut, and all others are assumed negligible.

3.3.2 Model validity constraints

During the early stages of developing a mathematical optimization model many assumptions are made in order to obtain reasonably simple expressions for the objective and constraint functions. One must always check whether subsequent results from optimization conform to these assumptions, lest they are violated. This would indicate that the model used is inappropriate for the optimal design obtained and the optimization results are at least suspect and possibly erroneous. The remedy is usually a more accurate, probably also more complicated, mathematical model of the phenomenon under question. For example, equation (3.13) is valid only if the length/diameter ratio is more than 10.

3.3.3 Material choice as a parameter

A final observation on the initial drive screw model is that a significant trade-off exists on the choice of material: stainless steel vs plastic. A steel screw will have higher strength and be smaller in size but will require secondary processing, such as rolling of threads and finishing of bearing surfaces. A plastic screw would be made by injection molding in a one-step process that is cheaper but more material would be used because of lower strength, plastic having a higher cost per pound than steel. Specialty plastics with high strength would be even more expensive and less moldable. Thus the choice of material must be based on a model that contains more information than the current one. The constant term representing manufacturing costs should be included in the objective. Indeed a more accurate cost objective should include capital investment costs for manufacturing.

Nevertheless, substantial insight can be gained from the present model if we include material as a parameter; in fact each material is represented by four parameters $C_w$ $\sigma_{all}$ $\tau_{all}$ $f$. In the model analysis that follows we keep these parameters in the model with their symbols, rather than giving numerical values insofar as possible. The goal is to derive as many additional results as possible independently of the material used. This will substantially facilitate a post-optimal parametric study on the material. It would be much more difficult to treat material as a variable, because then we would have four additional variables with discrete values and implicitly linked, perhaps through a table of material properties. As mentioned earlier, this would destroy the continuity assumed in nonlinear programming formulations.

3.3.4 Standard null form

The model is now summarized in the negative null form, all parameters represented by their numerical values (Table 3.2), except for material parameters.

### Table 3.2 List of parameters (material values for stainless steel)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_w$</td>
<td>material cost ($/lb$)</td>
</tr>
<tr>
<td>$f$</td>
<td>friction coefficient (0.35 for steel on plastic)</td>
</tr>
<tr>
<td>$F_a$</td>
<td>force required to snap drive screw into chassis during assembly (6 lbf)</td>
</tr>
<tr>
<td>$K$</td>
<td>stress concentration factor (3)</td>
</tr>
<tr>
<td>$L_1$</td>
<td>length of gear/drive screw interface segment (0.405 in)</td>
</tr>
<tr>
<td>$S$</td>
<td>linear cycle rate (0.0583 in $^{-1}$)</td>
</tr>
<tr>
<td>$S_m$</td>
<td>motor speed (300 rpm)</td>
</tr>
<tr>
<td>$T_m$</td>
<td>motor torque (2 in oz)</td>
</tr>
<tr>
<td>$W$</td>
<td>drive screw load (3 lb)</td>
</tr>
<tr>
<td>$\alpha_n$</td>
<td>thread angle in normal plane (60°)</td>
</tr>
<tr>
<td>$\sigma_{all}$</td>
<td>maximum allowable bending stress (20,000 lbf/in $^{-2}$)</td>
</tr>
<tr>
<td>$\tau_{all}$</td>
<td>maximum allowable shear stress (22,000 lbf/in $^{-2}$)</td>
</tr>
</tbody>
</table>

All 'intermediate' variables defined through equalities are eliminated together with the associated equality constraints by direct substitution. This should be always done when possible, in order to arrive at a model with only inequality constraints, thus facilitating subsequent monotonicity analysis. This is a model reduction step, since the number of design variables is reduced. Note, however, that this may not be always a model simplification step, as the resulting expressions may become more complex with undetermined monotonicities. Some judgement must be exercised here. Occasionally, 'directing' an equality may be useful (see further below) in avoiding direct elimination. In the model below the intermediate variables $M, L, T, I, c_1, c_3$ and $J$ together with the corresponding defining equalities have been eliminated. The variables are $d_1, d_2, d_3, L_2, L_3, N_m, N_m, N_T$.

**Model 1**

minimize $f_0 = (C_w \pi/4)(0.405d_2^2 + L_3 d_1^2 + L_2 d_3^2$ subject to:

- $g_1 = 38.88 + 96L_3 + 96L_3 - \pi \sigma_{all} d_2^2 \leq 0$
- $g_2 = 6N_m/N_m - \pi \sigma_{all} d_1^2 \leq 0$
- $g_3 = 8.345 - L_2 - L_2 \leq 0$
- $g_4 = -9.595 + L_4 + L_4 \leq 0$
- $g_5 = L_2 - 7.523 \leq 0$
- $g_6 = 7.023 - L_2 \leq 0$
- $g_7 = L_4 - 1.6525 \leq 0$
- $g_8 = 1.1525 - L_4 \leq 0$
- $g_9 = d_3^2 - 0.625 \leq 0$
- $g_{10} = d_3 - d_1 \leq 0$
- $g_{11} = d_1 - d_2 \leq 0$
- $g_{12} = 5N_m/N_m - 0.0583 N_T \leq 0$
- $g_{13} = 1.5d_2 \sqrt{f_{nd_2} + 0.5N_T^{-1}} - 0.5d_2 - f_{nd_2} \leq 0$
- $g_{14} = N_T - 24 \leq 0$
- $g_{15} = 8 - N_m \leq 0$
- $g_{16} = N_T - 52 \leq 0$

(3.25)

As there are no equality constraints, there are eight degrees of freedom.
Model reduction and verification

corresponding to the eight design variables. Three of these variables, \(N_m\) and \(N_g\), must take integer values, so this problem is in fact a mixed continuous-integer variable nonlinear programming problem and standard numerical NLP methods will not work. We will see later how this is dealt with in the particular example.

3.3.5 Feasibility checking

Before embarking on analyzing the model, it is a good idea to check that the feasible domain is not empty, i.e. there exists at least one feasible point. From a mathematical viewpoint this may be a hard problem (possibly as hard as the optimization itself), but from an engineering viewpoint past experience can be a guide. In the drive screw example, an existing design using stainless steel has the following values: \(d_1 = 0.1875\) in, \(d_2 = 0.3125\) in, \(d_3 = 0.2443\) in, \(L_2 = 7.273\) in, \(L_3 = 1.4025\) in, \(N_m = 8\), \(N_g = 48\), \(N_T = 18\). The design is feasible with an objective function value of \(f_0 = 0.635C_m\). Now we can proceed knowing that an optimization attempt is possible.

3.3.6 Monotonicity

Looking at model 1 one notes that most constraints were written in a form that requires no divisions. This is always advisable, since in subsequent numerical processing a denominator may become zero and cause an abrupt termination by overflow error. This can happen even if the imposed constraints exclude the relevant variable values, because many numerical algorithms will temporarily operate in the infeasible domain. In model 1, constraint \(g_{13}\) has not been rewritten yet because of concern that this might obscure its monotonicity wrt \(d_2\) and \(N_T\). Let us examine this more carefully. Assuming a strictly positive denominator of the first term in \(g_{13}\), we multiply both sides by it and collect terms:

\[
g_{13} : 1.5 \pi f d_2^2 + 0.75 N_T^{-1} d_3 - 0.0625 (N_g/N_m) d_4 + 0.125 f (N_g/N_m) N_T^{-1} \leq 0
\]

(3.26)

Clearly \(g_{13}\) decreases wrt \(N_T\) but is nonmonotonic wrt \(d_2\). In fact,

\[
\frac{\partial g_{13}}{\partial d_3} = 3 \pi f d_2 + 0.75 N_T^{-1} - 0.0625 N_g/N_m
\]

(3.27)

which can be positive or negative depending on the variable values. All remaining functions in the constraints have obvious monotonicities.

3.3.7 Variable transformation

The model can be further simplified by a variable transformation. We observe that the two variables \(N_g, N_m\) appear together as a ratio everywhere except in the simple bounds \(g_{13}, g_{16}\). We can define a new variable \(R\),

\[
R = \frac{N_m}{N_g}
\]

(3.28)

which indeed is the reduction ratio of the gear drive, and eliminate variable \(N_m\). Using \(N_m = RN_g\). The new model, including the reformulated constraint \(g_{13}\), is as follows.

Model 2

\[
\begin{align*}
\min f_0 & = (C_m \pi/4)(0.405 d_1^2 + L_2 d_2^2 + L_3 d_3^2) \\
\text{subject to} & \\
g_1 & = 38.88 + 96 L_2 + 96 L_3 - \pi \sigma \pi d_2^2 \leq 0 \\
g_2 & = 6 - \pi \sigma \pi R d_1^2 \leq 0 \\
g_3 & = 8.345 - L_2 - L_3 \leq 0 \\
g_4 & = -9.595 + L_2 + L_3 \leq 0 \\
g_5 & = L_2 - 7.523 \leq 0 \\
g_6 & = 7.023 - L_3 \leq 0 \\
g_7 & = L_3 - 1.6525 \leq 0 \\
g_8 & = 1.1525 - L_3 \leq 0 \\
g_9 & = d_2 - 0.625 \leq 0 \\
g_{10} & = d_3 - d_4 \leq 0 \\
g_{11} & = d_1 - d_4 \leq 0 \\
g_{12} & = 5 R - 0.0583 N_T \leq 0 \\
g_{13} & = 1.5 \pi f R d_2^2 + 0.75 N_T^{-1} d_3 - 0.0625 \pi d_4 + 0.125 f N_T^{-1} \leq 0 \\
g_{14} & = N_T - 24 \leq 0 \\
g_{15} & = 8 - R N_T \leq 0 \quad (3.29)
\end{align*}
\]

Note that the requirement of integer values for \(N_m\) is now converted to one of rational values for \(R\).

3.3.8 Repairing a model

Model 2 is now used to perform the first cycle of monotonicity analysis. The monotonicity table is a convenient tool to do this: Table 3.3. The columns are the design variables and the rows are the objective and constraint functions, the entries in the table being the monotonicities of each function with respect to each variable. Positive (negative) sign indicates increasing (decreasing) function, \(U\) indicates undetermined or unknown monotonicity. An empty entry indicates that the function does not depend on the respective variable, so the table acts also as an incidence table. (Items in parentheses will be explained in the next subsection.)

Monotonicity principles can be quickly applied by inspection using the monotonicity table. Looking at Table 3.3, by MP1 wrt \(d_3\) we see that model 2 is not well constrained because no lower bound exists for \(d_1\). Note that \(d_3 > 0\) is not an appropriate bound because of the strict inequality. If the model were treated numerically as is, no convergence would occur if the algorithm was successful or an erroneous result would be found if the algorithm was led astray. Examining the engineering meaning of this model deficiency we see that an adequate thrust surface must be provided to keep the shaft from wearing into the bearing support, so we accept the simple remedy of adding a new constraint

\[
g_{17} = 0.1875 - d_3 \leq 0 \quad (3.30)
\]

Poor boundedness is cause for concern, so the above deficiency triggers also
### Model reduction and verification

#### Table 3.3 Monotonicity table for model 2 (with model repairs in parentheses)

<table>
<thead>
<tr>
<th>Functions</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_0$</td>
<td>$d_1$, $d_2$, $d_3$, $L_2$, $L_3$, $R$, $N_T$, $N_T$</td>
</tr>
<tr>
<td>$g_1$</td>
<td>+</td>
</tr>
<tr>
<td>$g_2$</td>
<td>-</td>
</tr>
<tr>
<td>$g_3$</td>
<td>-</td>
</tr>
<tr>
<td>$g_4$</td>
<td>+</td>
</tr>
<tr>
<td>$g_5$</td>
<td>-</td>
</tr>
<tr>
<td>$g_6$</td>
<td>-</td>
</tr>
<tr>
<td>$g_7$</td>
<td>+</td>
</tr>
<tr>
<td>$g_8$</td>
<td>-</td>
</tr>
<tr>
<td>$(g_{10})$</td>
<td>+</td>
</tr>
<tr>
<td>$g_{11}$</td>
<td>-</td>
</tr>
<tr>
<td>$g_{12}$</td>
<td>U</td>
</tr>
<tr>
<td>$g_{13}$</td>
<td>-</td>
</tr>
<tr>
<td>$g_{16}$</td>
<td>-</td>
</tr>
<tr>
<td>$(g_{17})$</td>
<td>-</td>
</tr>
<tr>
<td>$(g_{18})$</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functions</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_{18}$</td>
<td>$d_2 \geq d_2 + 1.2990/N_T$</td>
</tr>
</tbody>
</table>

#### 3.3.9 Optimality rules

At this point we decide to add constraints $g_{17}(d_2^-$) and $g_{18}(d_2^+, d_3^+, N_T^-)$ to the model and delete $g_{16}$ as redundant. These changes are shown in parentheses in Table 3.2 and applying monotonicity analysis to this model we now obtain the following results, which represent necessary rules for optimality.

- (R1) By MP1 wrt $d_1$, $g_2$ is active.
- (R2) By MP1 wrt $d_2$, at least one constraint from the set $\{g_1, g_{11}, g_{13}\}$ is active.
- (R3) By MP1 wrt $L_2$, at least one constraint from the set $\{g_3, g_4\}$ is active.
- (R4) By MP1 wrt $L_3$, at least one constraint from the set $\{g_5, g_8\}$ is active.
- (R5) By MP2 wrt $R$, either all constraints $g_2, g_{12}, g_{13}$ and $g_{15}$ are inactive or at least one from each of the sets $\{g_2, g_{15}\}, \{g_{12}, g_{13}\}$ is active.
- (R6) By MP2 wrt $N_T$, either $g_{15}$ and $g_{16}$ are both active or they are both inactive.
- (R7) By MP2 wrt $R$, if $g_2$ is active then at least one of $\{g_{12}, g_{13}\}$ is active. Then, by MP2 wrt $N_T$ and (R1), $g_{18}$ is active.

#### 3.3.10 Active constraint elimination

The three active constraints, equations (3.32), are used to eliminate three variables from model 2, namely $d_1, d_3$, and $N_T$. There are two reasons for this. One is that monotonicity analysis on the new reduced model may reveal additional activity requirements. Another is that dominance arguments will be simpler in the reduced model. Which variables to eliminate is a judicious choice, based on what may be algebraically simpler and what may be a desirable form of the reduced model. The new model is as follows.

**Model 3**

Minimize $f_0 = 0.25 \pi C_m \left( \frac{0.405(6/\pi \tau_{all} R)^{2/3} + L_2 d_2^2 + L_2 (0.1875)^2}{2} \right)$

Subject to

- $g_1 = 38.88 + 96L_2 + 96L_3 - \pi \sigma_{all} d_2^2 \leq 0$
- $g_2 = 8.345 - L_2 - L_3 \leq 0$
- $g_3 = -9.595 + L_2 + L_3 \leq 0$
- $g_4 = L_2 - 7.523 \leq 0$
- $g_5 = 7.023 - L_2 \leq 0$
- $g_6 = L_3 - 1.6525 \leq 0$
- $g_7 = 1.1525 - L_3 \leq 0$
- $g_8 = d_2 - 0.625 \leq 0$
- $g_{11} = (6/\pi \tau_{all} R)^{1/3} - d_2 \leq 0$
- $g_{12} = R - 0.2798 \leq 0$
- $g_{13} = 1.5\pi R d_2^2 + 0.0313 R d_4 - 0.0625 \pi d_3 + 0.0052 f \leq 0$
- $g_{15} = 8 - R N_T \leq 0$
- $g_{16} = N_T - 52 \leq 0$
- $g_{18} = 0.2416 - d_2 \leq 0$

The monotonicity table for model 3 is shown in Table 3.4. No new results are obtained from this table. So dominance arguments must be sought in order to clarify the previously stated conditional activities and to obtain further model reduction.
Modeling example: a linear actuator

Figure 3.2 Activity map for constraints $g_3, g_6$, and $g_8$.

In the objective, the monotonicity wrt $L_3$ is determined by the sign of the quantity

$$ (0.1875)^2 - d_2^2 $$

which is negative since $d_2 \geq 0.2416$ from $g_{18}$. Hence, the objective is decreasing wrt $L_3$ and an upper bound is required by MP1. Constraint $g_6$ is obviously the dominant and active one, so

$$ L_{3e} = 1.322, L_{2e} = 7.023 $$

and $g_3, g_7, g_8$ are inactive.

The above results lead to yet another further reduced model with only three degrees of freedom.

Model 4

minimize $f_0 = 0.25\pi C_m \left[ 0.405(6/\pi \sigma_{all} R)^{1/3} + 7.023 d_2^2 + 1.322(0.1875)^2 \right]$

subject to

$$ g_1 = 840 - \pi \sigma_{all} d_2^2 \leq 0 $$
$$ g_2 = d_2 - 0.625 \leq 0 $$
$$ g_{11} = (6/\pi \sigma_{all} R)^{1/3} - d_2 \leq 0 $$
$$ g_{12} = R - 0.2798 \leq 0 $$
$$ g_{13} = 1.5\pi f R d_2^2 + 0.0313 R d_2 - 0.0625 \pi d_2 + 0.0052 f \leq 0 $$
$$ g_{15} = 8 - R N_3 \leq 0 $$
$$ g_{16} = N_3 - 52 \leq 0 $$
$$ g_{18} = 0.2416 - d_2 \leq 0 $$

The monotonicity table for this model is shown in Table 3.5. Rules (R2), (R5') and (R6) are still the only results derived from monotonicity principles.

At this point, the unknown monotonicity of $g_{13}$ wrt $d_2$ prevents us from continuing the reduction process. Indeed,

$$ \frac{\partial g_{13}}{\partial d_2} = 3\pi f R d_2 + 0.0313 R - 0.0625 \pi $$
and for $g_{13}$ to be increasing wrt $d_2$ we would need to have (for $f = 0.34$)

$$d_2 \geq 0.0595 R^{-1} - 0.0033 \approx D_2(R^-)$$

for the entire feasible range of $d_2, R$. We note that

$$\max D_2(R^-) = 0.0595 R^{-1} - 0.0033 = 0.0595 N_{\text{max}}/8 - 0.0033 = 0.0595(52)/8 - 0.0033 = 0.3983$$

This last number falls within the known feasible range of $d_2, 0.2416 \leq d_2 \leq 0.625$, so $g_{13}$ appears really nonmonotonic in the feasible domain.

### 3.3.12 Parametric models and case decomposition

The idea of regional monotonicity could be used here, i.e. try to identify in what interval of values of $d_2$ is $g_{13}$ monotonic and examine each case separately, comparing the results at the end. This would be unnecessarily complicated here. Instead, a simple problem decomposition can be applied: case A with $g_{13}$ inactive, and case B with $g_{13}$ active. We can examine these two cases separately and compare the results.

Before we proceed with the cases, it is instructive to recast model 4 in a simplified parametric form. It is the parameters $K_i, K_{ji}, i = 0, \ldots, 4$,  and rearranging (Table 3.6). The revised model is as follows. Note that all parameters $K_i$ relate to material properties.

**Model 5**

minimize $f_0 = K_0 R^{2/3} + K_6[7.023d_2^2 + 1.322(0.1875)^2]$

subject to

$$g_1 = K_1 - d_2 \leq 0 \quad g_5 = d_2 - 0.625 \leq 0$$

$$g_{11} = K_2 R^{-1/3} - d_2 \leq 0 \quad g_{12} = R - 0.2798 \leq 0$$

$$g_{13} = K_3 R d_2^2 + 0.0313 R d_2 - 0.1963 d_2 + K_4 \leq 0$$

Table 3.5 Monotonicity table for model 4

<table>
<thead>
<tr>
<th>Functions</th>
<th>Variables</th>
<th>$d_2$</th>
<th>$R$</th>
<th>$N_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_0$</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$g_1$</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$g_{11}$</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>$g_{12}$</td>
<td>+</td>
<td>U</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>$g_{13}$</td>
<td>U</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>$g_{14}$</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$g_{15}$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

This table provides the necessary conditions for monotonicity of each function with respect to the variables $d_2, R, N_8$.

Consider now case A with $g_{13}$ inactive. Then $g_{12}$ is active from rule (R5)' and $R_*=0.2798$. From rule (R2) we now have

$$d_2 = \max(K_1, K_{12} R_{12}^{1/3} 0.2416)$$

and from rule (R5) we have

$$8/R_* \leq N_{8g} \leq 52$$

Note that one degree of freedom remains at the optimum, as $N_{8g} N_5$ can be selected to give the largest rational number not exceeding 0.2798 and satisfying the range in equation (3.43). Any solution thus obtained must be checked that it satisfies the remaining inactive constraints. This feasibility check is frequently overlooked in the application of monotonicity analysis, leading to erroneous conclusions.

Next, consider case B with $g_{13}$ active. Rules (R2) and (R5) are satisfied and no new activity results can be obtained. Note that now, locally, $g_{13}$ must be decreasing wrt $d_2$, i.e. $g_{13}(d_2^*, R^*) \leq 0$, while $f_2(d_2^*, R^-)$. An implicit solution of $g_{13}$ gives $d_2 = \varphi_{13}(R^*)$ and substitution in the objective gives $f_3(d_2^*, R^-) = f_3(\varphi_{13}(R^*), R^-) = f_3(R)$ with $R$ having unclear monotonicity. There are two degrees of freedom left, but a one-dimensional search in $d_2$ would suffice if $g_{13}$ is solved explicitly for $R$ and the objective is expressed as a function of $d_2$ only. Constraints $g_{15}$ and $g_{16}$ could be replaced by

$$R \geq 8/N_{8g}^{-1} \geq 8/52 = 0.1538$$

There is a lingering concern, though, regarding the physical meaning of $g_{13}$ being active. Essentially, friction forces and applied forces on an equivalent inclined plane would be equal and motion would be impending. This would not represent a stable design for the lead screw, albeit possibly an optimal one. The designer must then examine the implications on the appropriateness of the model and/or the parameter values selected. Also, satisfaction of model validity for the beam stress formula (3.13) should be checked. The results obtained numerically using SOP code NLPO (Schmitzowski, 1984) with $C_p = 10.0$ and continuous values for the integer variables indicate constraints $(g_{12}, g_{13})$ and $(g_{12}, g_{13})$ are active. The activity of the first five was the one discovered a priori by analysis. Constraints $(g_1, g_2, g_3, g_4, g_5, g_{11}, g_{12}, g_{13}, g_{14})$ are inactive.
3.3.13 Directing an equality

It is interesting to note that \( g_{18} \) is active in the final solution. Indeed, according to Spotts (1985) the required relation is an equality:

\[
h_{10} \cdot d_3 = d_2 - 1.2990/N_7
\]  

(3.45)

rather than the inequality (3.31). Instead of using \( g_{18} \) in the model we could have used \( h_{10} \), which would prevent direct application of MP1.

However, an equality constraint can be viewed as an active inequality that has been 'properly directed' (Papalambros and Wilde, 1988). One way to determine such a direction is to use the monotonicity principles. Ignoring equation (3.30) for the moment, and with equation (3.45) replacing \( g_{10} \), we see that by MP1 wrt \( d_3 \) a lower bound is required. Then \( h_{10} \) can provide this bound if directed as \( d_3 \geq d_2 - 1.2990N^{-1} \) or \( d_3 \geq 1.2990N^{-1} - d_2 \leq 0 \).

Not much inference is possible when the directed \( h_{10} \) is used in the model instead of \( g_{18} \). One could proceed by assuming \( h_{10} \) inactive and include \( g_{17} \), which would then be active. This would take us essentially through the same steps as before, checking \( h_{10} \) for violation in the final solution. Interestingly, numerical results obtained for such a scenario using NLPQL indicate \( h_{10} \) is satisfied in the final solution with a zero multiplier value.

3.4 MODELING AUTOMATION

Traditional numerical solutions rely almost exclusively on iterative use of local information, such as values of functions, gradients or composition of locally active sets. Knowledge accumulated during such an iterative process may be used heuristically to speed up the solution process, for example, as discussed in Arora and Baenzinger (1986) and Adeli and Balasubramanyam (1988a, b). Analytical techniques aiming at active constraint identification, such as monotonicity analysis, use primarily global information, which is true for all points in the design space. A production system using rules for global information processing was first introduced in Li and Papalambros (1985a). Combined qualitative and quantitative reasoning for optimization in an AI environment has also been discussed in Agogino and Almgren (1987), Hammond and Johnson (1987, 1988), Hansen, Jaumard and Lu (1988a, b), and Watton and Rinderle (1991). The need for developing new types of algorithms using both global and local knowledge was explored in Li (1985) and Li and Papalambros (1988). Artificial intelligence (AI) techniques, including symbolic manipulations, have been proposed as modeling tools; see, for example, Li and Papalambros (1985b) and Choy and Agogino (1986).

One possibility offered by the use of AI techniques in modeling is to generate automatically global information about constraint activity and model boundedness, which can then be given directly to a global or local—global strategy, as described in the above references. This will remove, at least in part, the difficulty that relatively unskilled users may encounter in discovering rigorous global knowledge. Moreover, it eliminates tedious procedures prone to errors, when manually performed. The system PRIMA (production system for implicit elimination in monotonicity analysis) was implemented in the OPS5/OPS87 production system development tool (Forgy, 1981).

In PRIMA (Rao and Papalambros, 1987) the monotonicity table forms the very basic abstract data structure in the problem domain. Rules based on the monotonicity principles are applied to obtain facts about the design mode. These facts, called state predicates, being derived from necessary conditions, must be satisfied by every optimum solution, called a state. As these facts are derived, the system automatically generates LISP-like macros in the form of predicate functions, each of which should evaluate to true at every optimal design or state. In the presence of monotonicity, additional facts can be derived after one or more active constraints have been implicitly eliminated from the model (together with one or more corresponding variables). The result is again a new derived monotonicity table that may generate new state predicates. This process continues until all active constraints are eliminated, or no useful monotonicities remain.

The primary task, then, is to search an 'implicit elimination tree' (Fig. 3.3), where (i) visiting a node, i.e. monotonicity table, consists of applying mathematically rigorous necessary conditions implemented in the form of rules, (ii) the tree arcs (connecting one node to another) correspond to the process of implicit elimination, and (iii) the choice of pivots used in elimination (specific variable and constraint that will be eliminated) corresponds to selective expansion of nodes and is based on heuristic rules. This classification of rigorous rules (at

---

**Fig. 3.3** Generic implicit elimination tree in PRIMA. The nodes \( g_1 - x_1 \) etc. are reduced monotonicity tables obtained by implicit elimination. Facts 1, 2, 3 etc. could typically be obtained at any node in this tree.
the nodes) and heuristic rules (at the arcs) can be considered a characteristic of this problem domain.

The model of the desired automation procedure is as follows. A tree data structure, called the implicit elimination tree, Fig. 3.3, is built in the space of feasible designs. Each node is a monotonicity table containing all the boundedness and constraint activity information that characterizes it. The root of the tree is the given top-level monotonicity table. The arcs of the tree structure correspond to a pivot used during the implicit elimination process. The root node has a degree of freedom (dof) greater than or equal to unity, which might decrease if more constraints are identified as active using MP1 or MP2. The terminal nodes along a depth-first search path correspond to the following cases: (i) the dof has been reduced to zero (i.e., the solution is constraint bound in a non-empty feasible domain); (ii) all active constraints have been eliminated in that branch; (iii) the heuristics do not provide a pivot, the particular path is deemed unprofitable, and backtracking must be performed. Full or partial traversal of this implicit elimination tree yields state predicates embodying global information.

The heuristic rules used in PRIMA are discussed in detail in Papalambros (1988). The main function of the rules is to guide the search through an implicit elimination tree in the most profitable generate-and-test manner, i.e., to make good pivot selections and to control backtracking. The use of an AI approach here offers the usual advantages of having a flexible representation of heuristic knowledge and an efficient data structure for repeated list processing.

An example application is repeated here from Rao and Papalambros (1991) (a revised version of Rao and Papalambros (1987)). The objective is to minimize the production cost of a ship bottom structure modeled as a structural giraffe (Fig. 3.4).

The model, based on Winkle and Baird (1985), is stated as follows.

**Mathematical model for ship bottom structure**

minimize VC = MW + k MH

subject to

\[ h_1: \quad SM_{\text{frequ}} = (4.74 c_{\text{frequ}} h_{\text{frequ}} s_{\text{floor}} l_{\text{floor}}^2)/100^3 \]

\[ h_2: \quad c_{\text{frequ}} = 0.9 \]

\[ h_3: \quad f_{\text{frequ}} = f_{\text{frame}} \]

\[ h_4: \quad A_{\text{fr}} = w_{\text{fr}} f_{\text{fr}} \]

\[ h_5: \quad A_{\text{tp}} = w_{\text{tp}} f_{\text{tp}} \]

\[ h_6: \quad f_{\text{floor}} = B - 0.750 \]

\[ h_7: \quad MH = f_l(l,w, \text{etc.}) \]

\[ h_8: \quad MW = w_m f_3(l,w, \text{etc.}) \]

\[ h_9: \quad h_{\text{floor}} = l f_{\text{floor}} \]

\[ h_{10}: \quad k_{\text{keel}} = w s_{\text{keel}} - 1 \]

\[ h_{11}: \quad SM_{\text{frequ}} = f_l(l,w, \text{etc.}) \]

\[ g_1: \quad k_{\text{keel}} \geq h_{\text{min}} + 0.0015 \]

\[ g_2: \quad c_g \geq (0.063 L + 5)/1000 \]

\[ g_3: \quad A_{\text{fr}} \geq (0.168 L^{3/2} - 8)/10000 \]

\[ g_4: \quad h_{\text{keel}} \leq 2.13 \]

\[ g_5: \quad h_g \geq h_{\text{keel}} \]

\[ g_6: \quad h_{\text{keel}} \geq h_{\text{floor}} \]

\[ g_7: \quad f_{\text{frequ}} \geq (0.063 L + 4)/1000 \]

\[ g_8: \quad A_{\text{fr}} \geq (0.038 L^{3/2} + 17)/10000 \]

\[ g_9: \quad h_{\text{frequ}} \geq d \]

\[ g_{10}: \quad f_{\text{frequ}} \geq 0.66D \]

\[ g_{11}: \quad w_c(h_{\text{floor}} + t_{\text{f}}) \geq w_c 0.0625 f_{\text{floor}} \]

\[ g_{12}: \quad f_{\text{floor}} \geq 0.0625 f_{\text{floor}} \]

\[ g_{13}: \quad w_c f_{\text{fr}} \geq w_c [(f_{\text{floor}} + t_{\text{f}})/100 + 0.003] \]

\[ g_{14}: \quad f_{\text{floor}} \geq h_{\text{floor}}/100 + 0.003 \]

\[ g_{15}: \quad f_{\text{frame}} \leq 0.0115 \]

\[ g_{16}: \quad SM_{\text{frequ}} \geq \Sigma SM_{\text{frequ}} \]

\[ g_{17}: \quad f_{\text{frame}} \leq (2.08 L + 438)/1000 \]

\[ g_{18}: \quad D \geq D \]

\[ g_{19}: \quad h_{\text{bottom}} \geq h_{\text{min}} \]

\[ g_{20}: \quad h_{\text{min}} \geq (s_{\text{frame}}/519)[(L - 19.8) \max(d/D, 0.65)]^{1/2} + 0.0025 \]

\[ g_{21}: \quad h_{\text{min}} \geq (s_{\text{frame}})(L + 45.73)/(25 L + 6082) \]

\[ g_{22}: \quad w_{\text{keel}} \geq 0.750 \]

Functions \( f_1, f_2 \), and \( f_3 \) are defined using man--hour estimates, total volume of structural elements, and plate--stiffener combination section moduli respectively. The problem has 11 equality constraints and 20 degrees of freedom. The design variables are shown in Table 3.7. Equality constraints are explicitly eliminated and a top-level monotonicity table is obtained where PRIMA deduces six different facts. The first monotonicity principle is repeatedly applied eight times and constraints \( g_1, g_3, g_4, g_5, g_6, g_7 \) are identified as being active. These six active constraints and the variables appearing in each of them provide several possibilities for selecting implicit elimination pivots, and the heuristic rules are now used. Starting with the top-level monotonicity table, the sequence of pivots \( \{g_1, x_{19}, g_3, x_{20}, g_8, x_{19}, g_5, x_{22}, g_{21}, x_{18}, g_4, x_{21}\} \) leads to the table shown in Fig. 3.5, where the following additional facts or rules are obtained.

Fact 1: At least one of \( g_{16}, g_{13}, \) and \( g_{11} \) is active.

Fact 2: At least one of \( g_{21} \) and \( g_{20} \) is active.

Fact 3: At least one of \( g_{16}, g_{14}, \) and \( g_{13} \) is active.

Fact 4: At least one of \( g_{16} \) and \( g_{19} \) is active.

Fact 5: At least one of \( g_{21}, g_{20}, g_{17} \) and \( g_{16} \) is active.
These not very obvious facts are not sufficient for identifying an optimum, but there is no generally accepted solution that does not satisfy them. A genetic model for generating a knowledge-based decision-making aid in design optimization, developed by Balachandran and Gero (1997), provides a method for determining the appropriate parameter values. The mathematical model for the system structure and the associated configurations can be used to determine the appropriate parameter values. The mathematical model for the ship bottom structure and the associated configurations can be used to determine the appropriate parameter values. The mathematical model for the ship bottom structure can be used to determine the appropriate parameter values.

<table>
<thead>
<tr>
<th>MONOTONICITY TABLE</th>
<th>obj</th>
<th>x1</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x9</th>
<th>x10</th>
<th>x11</th>
<th>x12</th>
<th>x13</th>
<th>x14</th>
<th>x15</th>
<th>x16</th>
<th>x17</th>
</tr>
</thead>
<tbody>
<tr>
<td>g2</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>g6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

THE FOLLOWING FACT IS DERIVED W.R.T. VAR x17
AT LEAST ONE OF THE FOLLOWING CONSTRAINTS IS ACTIVE: g16--g13--g11
THE FOLLOWING FACT IS DERIVED W.R.T. VAR x13
AT LEAST ONE OF THE FOLLOWING CONSTRAINTS IS ACTIVE: g21--g20
THE FOLLOWING FACT IS DERIVED W.R.T. VAR x11
AT LEAST ONE OF THE FOLLOWING CONSTRAINTS IS ACTIVE: g16--g14--g13
THE FOLLOWING FACT IS DERIVED W.R.T. VAR x9
AT LEAST ONE OF THE FOLLOWING CONSTRAINTS IS ACTIVE: g19--g16
THE FOLLOWING FACT IS DERIVED W.R.T. VAR x6
AT LEAST ONE OF THE FOLLOWING CONSTRAINTS IS ACTIVE: g21--g20--g17--g16

Fig. 3.5 Final PRIMA results for ship bottom structure problem.
models codification of domain knowledge may be possible in the form of rules or other AI data structures, which can be then manipulated to make useful deductions.

Of special interest here is also work by Schittkowski on an intelligent modeling and mathematical programming package (Schittkowski, 1985), and the available commercial package GAMS (Brooke, Kendrick and Meeraus, 1988). In Schittkowski's system a set of rules assists the user in selecting algorithms and parameters within algorithms, based on an evaluation of the type of model that needs to be solved. The rules are a result of numerical analysis knowledge and experimentation with actual problems. Each rule has a level of certainty or veracity probability associated with it. These are propagated and indicated in the deductions produced by the rules. The system keeps a record of all major decisions made and attendant results every time a new problem is being solved. This information is used to modify or sharpen the rules and their associated veracity probabilities. Thus a learning function is supported. GAMS (General Algebraic Modeling System) is a high level language that integrates features of a relational database in formulating explicit (algebraic) mathematical programming models. Originally it was conceived to facilitate easy data entry and modification of very large linear programming models. Current extensions can handle nonlinear and binary (zero–one) linear programming models. The major advantage of a GAMS environment in the present context is the ability to generate quickly modifications of models by adding, deleting or changing the expressions of constraints without worrying about FORTRAN reprogramming. Such functions for optimization computations are now beginning to appear also in general-purpose packages such as MATHEMATICA (Wolfram, 1988) and MATLAB (Grace, 1991).

3.5 CONCLUDING REMARKS

When an optimization study is undertaken, a careful model analysis will save a lot of subsequent effort in trying to obtain good and reliable results from numerical computations. An initial model must be properly cast into the standard form for numerical processing and possible unboundedness should be checked. A clear understanding of the physical meaning of each constraint is necessary, as well as an exploration of how constraints may interact with each other. Model validity limitations must be always checked to ascertain that they are indeed satisfied by the optimal values of the design variables. Although we have demonstrated these ideas on relatively small explicit models, such ideas are important also when implicit models are used. Proper modeling is somewhat of an art but the tactics involved are grounded frequently on a rigorous understanding of how optimization methods work. Many of these ideas will continue to become increasingly automated.

ACKNOWLEDGEMENTS

Thanks are due to the University of Michigan graduate students B. Alexander and B. Ryenga for posing the linear actuator problem and R.P. Krishnamachari for helping with the final form of the manuscript.

REFERENCES

Li, H.L. (1985) Design optimization strategies with global and local knowledge. PhD Dissertation, Department of Mechanical Engineering and Applied Mechanics, University of Michigan, Ann Arbor, MI.
Model reduction and verification

Winkle, L.E. and Baird, D. (1985) Towards more effective structural design through synthesis and optimization of relative fabrication costs. *Proceedings of the Joint Meetings of the Royal Institute of Naval Architecture (RINA) and the Institute of Engineers and Shipbuilders, Glasgow*.