Symbolic Design Optimization: A Computer Aided Method to Increase Monotonidty Through Variable Reformulation

by
John D. Watton, James R. Rinderle

EDRC 24-56-91

Computer Aided Applied Artificial
Intelligence, **(in review), December 1990. EDRC-^.** *dJf - JT£ - 9/*

Symbolic Design Optimization: A Computer Aided Method to Increase Monotonicity Through Variable Reformulation

John D. Watton
Applied Mathematics and Computer Technology Division
Alcoa Laboratories
Alcoa Center, PA 15069

phone: (412) 337-2165
fax: (412) 337-5436
e-mail: JDW@jazz.al.alcoa.com

James R. Rinderle
Mechanical Engineering Department
Carnegie **Mellon** University
Pittsburgh, PA 15213

phone: (412) 268-3677
fax: (412) 268-6421
e-mail:rinderle@cs.cmu.edu

DRAFT of 13 November 1990 at 09:52

Symbolic Design Optimization: A Computer Aided Method to Increase Monotonicity Through Variable Reformulation

Abstract

Monotonicity analysis, developed by [Wilde 78] and [Papalambros 79], is an approach to simplifying and solving some nonlinear, constrained global optimization problems without iterative numerical calculations. Unfortunately, monotonicity analysis is limited to problems in which the objective function and constraints vary roonotonically with the design variables. To alleviate this limitation it is sometimes possible to reformulate the design problem to increase the degree of monotonicity and thereby facilitate the complete or partial application of monotonicity analysis procedures. These useful reformulations are accomplished by a transformation to alternative design parameters, such as a critical ratio, a nondimensional parameter, or a simple difference; e.g. the ratio of surface area to volume for heat transfer loss, the Reynold's number in fluid mechanics, or the velocity difference across a fluid coupling. We have developed a method by which the alternative parameters are chosen for physical significance and for the ability to reduce the number of nonmonotonic variables in a system of constraints. Rules have been developed for the creation of physically significant new parameters from the algebraic combination of the original parameters. The rules are based on engineering principles and rely on knowledge about what a parameter physically represents rather than other qualities such as dimensions. A computer based system, called EUDOXUS, has been developed to automate this procedure. The method and its implementation have demonstrated successful results for highly nonlinear, nonmonotonic, and coupled parameterized designs in various mechanical engineering domains.

Introduction

In this paper we ait concerned with parametric design problems that can be expressed in tenns of *n* design variables, *V* = [u_v *w^* ...» u_m]. The optimization problem is to minimize or maximize an objective function:

/o(0> (1)

subject to *k* inequality constraints and *r* equality constraints:

f^U) <≥ 0   i = l , * (2)

fj(U) = 0   y = *+l,*+r (3)

In most cases the number of unknowns, n, is greater than the number of equality constraints, r, so that the problem is underconstrained.

Optimal, rather than merely satisfactory solutions to design problems are often identified intuitively by experienced and insightful engineers. This is true even when the design constraints are very complex because optimal designs are frequently determined by the inequality constraints which delimit the feasible design space. In these cases, finding an optimal solution requires the identification of the inequality constraints which are satisfied as equalities at the optimum. This is often possible when the objective and the constraints vary monotonically with respea to some or all of the unknowns. A procedure known as monotonicity analysis facilitates the identification of the critical constraints at the optimum. The necessary conditions for the optimality of monotonic systems, first explicated by Wilde and Papalambros [Wilde 78, Wilde 86, Papalambros 79, Papalambros 88], can be summarized in the following two rules paraphrased from [Agogino 87]:

> Rule 1: When an objective function is monotonic wkh respect to a variable parameter then there exists at least one active constraint at the optimum which bounds the variable in a direction opposite of the objective.
>
> Rule 2: When a variable is not contained in the objective function then it must either be bounded from both above and below by active constraints at the optimum or not bounded at all such that all constraints which are monotonic with respect to that variable are inactive and can be removed from the problem definitioa

These rules can be used to solve some optimization problems symbolically (and automatically [Agogino 87]). The active constraints which limit the degree to which the objective function can be minimized or maximized become explicit. Some of the drawbacks of numerical techniques, e.g. convergence and local optima, are avoided and more importantly valuable qualitative information is obtained. Unfortunately, monotonicity analysis is limited to design optimization problems where the objective function and constraints are monotonic either globally or over a predefined regional domain. In the next section we illustrate, using a simple truss design problem, how reformulations of design problems involving a transformation of variable can increase the number and degree of monotonicity assignments thereby improving symbolic optimization procedures. Although an unlimited number of reformulations are possible, those which are most useful exhibit three characteristics:

1. a reduction in the number of variables in some of the constraints and the objective function,
2. an increase in the degree of constraint and objective monotonicity,
3. and are based on new variables which are physically meaningful.

New variables with physical significance are especially important for establishing qualitative insights and for making meaningful estimates when numerical methods are used.

Following the details of the example we concentrate on the methods used to identify reformulations as driven by these attributes. In short, we take a two step approach. First, we discuss a method to create physically meaningful new parameters based on the physical meaning of the original parameters. The candidate parameters must then be grouped into transformation sets that produce a more monotonic foimulatioa The second step is to assemble the new parameters into basis sets, perform transformations of variable, and evaluate the utility of the resulting problem reformulations. Although these techniques are independent of any computer implementation they are, in general, computationally intensive. These ideas are implemented in a computer program called EUDOXUS.[1] The final part of this paper discusses the program implementation and effectiveness.

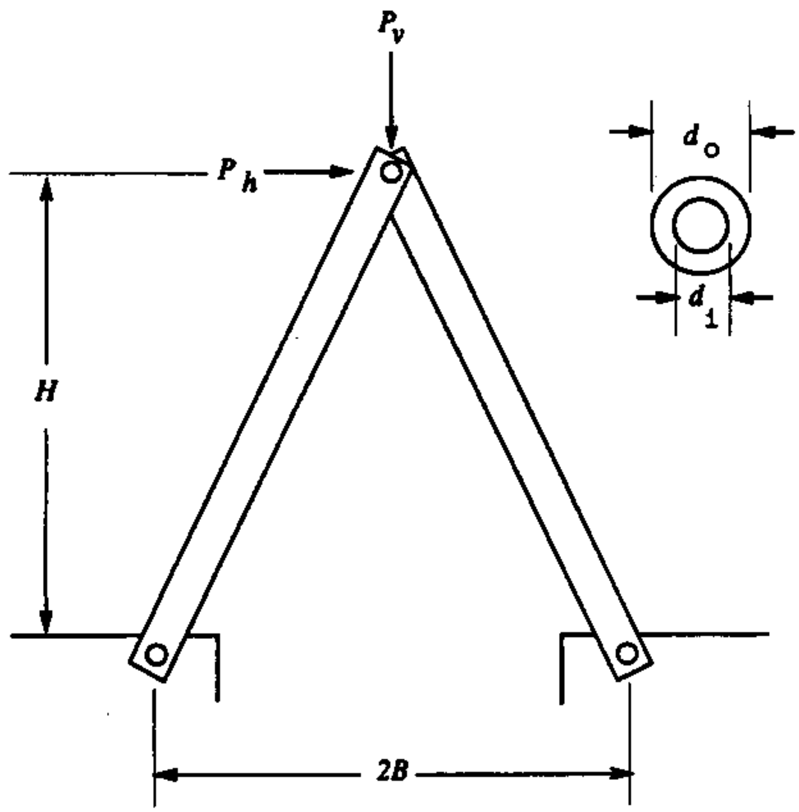

Figure 1: Two Bar Truss Configuration

'Eudoxus of Cnidus (b. 408 BC, d. 355 BC) was a Greek scholar who made contributions in mathematics, astronomy, geography, philosophy, and law. His theory of proportions solved the crisis of the Pythagorean discovery of irrational numbers and his method of exhaustion was a forerunner to modern calculus.

Two Bar Truss Example

A stnictuie must be designed to suppoit horizontal and veitical loads at a minimum height. A simple truss **of the type** shown in Figure 1 is to be considered [Fox 71, Parkinson 85]. The truss consists of a pair **of tubes pinned** together and to ground supports. The structure must withstand a veitical load, P_v, and **a** horizontal load, $P_{h\%}$ without failing by yielding or buckling, and without excessive veitical deflection, 5^^. Additional constraints enforce the requirement on minimum height, $H_{min\$}$ and maintain an outside diameter greater than the inside diameter. The parameters describing the truss are the half-span, B, height, //, tube outside tube diameter, d_o, and the inside diameter, d_V The material properties, modulus of elasticity, £, yield stress, $o_{yig[d9}$ and density, p, are known. Mathematically, the problem is to minimize truss mass:

$$f_Q = Mass = \rho\frac{\pi}{2}(d_o^2 - d_i^2)\sqrt{B^2 + H^2} \qquad \text{Objective} \qquad \textbf{(4)}$$

Subject to:

$$I, - \sigma_{yi\pounds ld} - \frac{2<\overline{B^2 + H^2}}{\pi(d_o^2 - d} \frac{P_v}{H} + \frac{P_h}{B} \pounds 0 \qquad \text{Yielding} \qquad (5)$$

$$f_2 = \delta_{max} - \frac{1P_V\{B^2 + H^2)^{1.5}}{\pi EH^2(d_o^2 - d_i^2)} \geq 0 \qquad \text{Deflection} \qquad (6)$$

$$h \ll \frac{\pi^2 E(d_o^2 + d?)}{16(B^2 + ffi)} - \frac{2\sqrt{B^2 + H^2}}{\pi(d_o^2 - d_i^2)}\left(\frac{P_v}{H} + \frac{P_h}{B}\right) \geq 0 \qquad \textbf{BuckUng} \qquad \textbf{(7)}$$

$$U = d_o - d_i \geq 0 \qquad \text{Tube limitation} \qquad (8)$$

Minimum height (9)

The optimization problem can be summarized as:

Minimize:	$f_0(.d_o\backslash d_r,B\backslash H+)$	from 4, Objective
Subject to:	$f\wedge d_o\backslash d_-,B\backslash\}T)$ t 0	from S, Yielding
	$f_2(d_o\backslash d_-,\beta -,H')$ 2>. 0	from 6, Deflection
	$f_z\{d_o\backslash d^*,\beta\backslash ir)$ Z 0	from 7, Buckling
	$Md_o\backslash d\sim)$ 2 0	from 8, Tube limitation
	$f_5(H^+) \geq 0$	from 9, Height

Monotonically increasing trends are indicated by a (-1-) superscript and decreasing trends by a (-) superscript. Nonmonotonicities are indicated by a (*). The objective function for mass can be decreased by reducing B_9 H_9 or d_o or by increasing d_v Unfortunately, a value change in any one of these variables

necessitates an unwanted change in competing variable values. This coupling effect in addition to the nonmonotonicities makes qualitative conclusions about active constraints very difficult The application of monotonicity analysis to this problem results in little information. Although the objective is monotonic in all four of the design variables, die constraints are largely nonmonotonic. We can conclude *only* that at least one of die yielding, buckling, deflection or tube constraints are active.

Alternatively, consider a reformulation based on the following parameter transformation:

$$
\begin{aligned}
H &= H \\
R &= B/H \\
A_m &= \frac{\pi}{4}(d^2 - d_i^2) \\
A_i &= \frac{\pi}{4} d_i^2
\end{aligned}
\qquad (10)
$$

where the aspect ratio, *R*, is the ratio of half width to height, A_m is the metal cross section area, and A_i is the inside hollow cross section area of the truss members. The design objective and constraints become:

$$g_0 = Mass = 2\rho A_m H\sqrt{R^2 + 1} \qquad \text{Objective} \qquad (11)$$

Subject to:

$$g_1 = \sigma_{yield} - \frac{\sqrt{R^2 + 1}}{2A_m}\left(P_v + \frac{P_h}{R}\right) \geq 0 \qquad \text{Yielding} \qquad (12)$$

$$g_2 = \delta_{max} - \frac{P_v H (R^2 + 1)^{1.5}}{2A_m E R^2} \geq 0 \qquad \text{Deflection} \qquad (13)$$

$$g_3 = \frac{\pi E (A_m + 2A_i)}{4H^2(R^2 + 1)} - \frac{\sqrt{R^2 + 1}}{2A_m}\left(P_v + \frac{P_h}{R}\right) \geq 0 \qquad \text{Buckling} \qquad (14)$$

$$g_4 = \sqrt{1 + A_i/A_m} - 1 \geq 0 \qquad \text{Tube limitation} \qquad (15)$$

$$g_5 = H - H_{min} \geq 0 \qquad \text{Minimum height} \qquad (16)$$

The reformulated optimization problem can be summarized as:

Minimize:	$g_0(R^+, A_m^+, H^+)$	from 11,	Objective
Subject to:	$g_1(R^-, A_m^+) \geq 0$	from 12,	Yielding
	$g_2(R^-, A_m^+, H^-) \geq 0$	from 13,	Deflection
	$g_3(R^{\bullet}, A_m^{\bullet}, H^-, A_i^+) \geq 0$	from 14,	Buckling

$g_4(A_m+,Ar)$ Z 0 . from 15, Tube limitation

$g_5(H+)$ * 0 from 16, Height

In the original formulation, the minimum mass objective depended monotonically on all four design variables. In the reformulation, the objective function monotonicity is preserved but only three variables appear in the objective function. The reformulation similarly improves the constraints. In the case of the yielding constraint the number of design parameters is reduced from four to two and the number of nonmonotonic variables is reduced from two to one. Simplifications of this type in all of the constraints make it possible to systematically apply the monotonicity rules to conclude that:

- The constraint on buckling does not {day a part in minimizing mass and the value of A_i only needs to be chosen to satisfy buckling (constraint g_3) and the tube limitation (constraint *gj.*
- An unconditionally active constraint is a minimum//value (// = H_{min} from constraint g_3).
- The constraint on yielding (constraint g_x) is always active at minimum mass.
- The constraint on maximum deflection (constraint g_2) may or may not be active at minimum mass depending on the specific parameter values.

To reach these conclusions we begin with the reformulated mass expression. Since A_t monotonically influences all constraints in which it appears but does not appear in the objective, Rule 2 requires that A_i is either bounded from both above and below by active constraints at the optimum or A_i is irrelevant. The constraint monotonicities show that the only constraint which can bound A_i from above[2] is the tube constraint g_4. Closer examination of that constraint, however, shows that it is unconditionally satisfied for non-negative values of A_i and A_m. The constraint is therefore irrelevant and can be deleted. Because no other constraint can bound A_i from above, Rule 2 states that all constraints which are monotonic in A_i can be deleted. In the present case we may therefore delete the buckling constraint, g_4. Thus, the constraints g_3 and g_4 are not involved in optimizing for mass. Physically, this means that the tube hollow area, A,, can be set arbitrarily large to insure the satisfaction of the buckling constraint.[3]

The application of Rule 1 to H now reveals that g_4 must be constrained from below and therefore the minimum height constraint, g_5, must be active. Physically this means that H takes on the minimum allowable value, H_{min}.

Only two constraints, yielding and buckling, remain and with H now determined, the design problem involves only two variables, the aspect ratio, R_f and the metal area, A_m. Both variables influence the two remaining constraints and the objective. However, the objective and deflection constraint are monotonic in both and the yielding constraint is monotonic in A_m. Rule 1 requires that at least one of the two constraints must be active to constrain A_m from below. There are two possibilities. In the first case g_x is active to bound A_m from below and g_2 need not be active. Physically, this corresponds to a design

*The implicit constraint requiring a non-negative value for A_t bounds A_i from below. The parameter value constraints were left implicit so as not to unnecessarily complicate the problem.

[3]Although the buckling stress margin improves monotonically with $A_{i\%}$ the tube thickness decreases and other unmodeled failure modes, e.g. skin buckling may become important.

which is limited by yielding. The other possibility is that g_2 is active to bound A_m from below. Since g_2 cannot also bound R from below then g_x must also be active. Physically, this constraint activity corresponds to t truss design which is simultaneously limited by both yielding and deflectioa The yielding constraint is therefore unconditionally active. The deflection constraint may or may not be active depending on the truss loads, material and allowable deflectioa Because the yielding constraint is active, i.e. satisfied as an equality, we may use the constraint to eliminate A_m from the objective and deflection constraint to obtain a one degree of freedom constrained optimization. Figure 2 shows how the truss mass and deflection depend on R for one set of specifications. In the absence of a restrictive deflection constraint, R is chosen to minimize mass resulting in a truss mass of 58.1 kg. If deflection, however, must remain less than approximately 1 mm the solution is forced away from the minimum mass result as shown in die figure. Once R is determined, the value ofA_m is calculated from the yielding constraint and the remaining original variables, B, $d_{i\$}$ and d_o, follow from Equations 10. The reformulated design optimization problem was significantly easier to solve than the original problem. It may be more important, however, that the reformulation facilitates reasoning by the designer about critical constraints and tradeoffs.

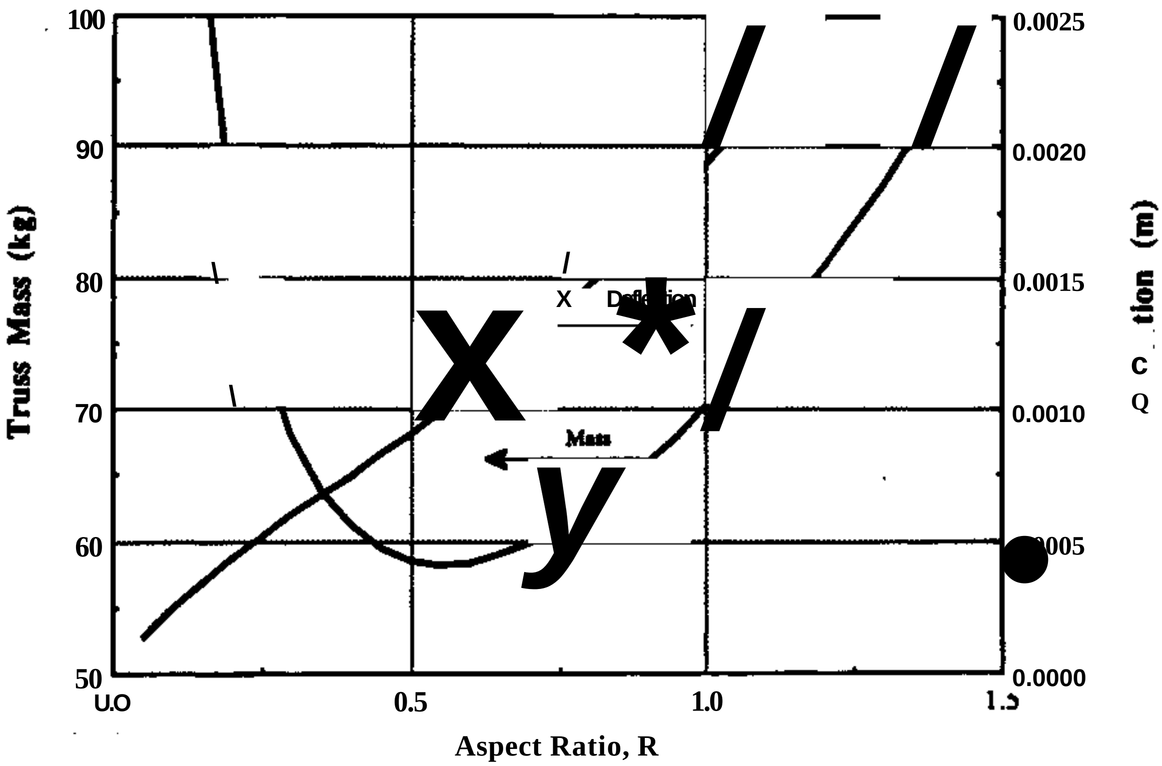

Figure 2: Trass mass and deflection as a function of aspect ratio, *R.* for the case when /»„ * 400 kN, P_h= 200 kN, q ^ • 200 MPa, p = 7800 kg/hi^3, *H^ = 15* m, and £= 207,000 MPa

Identifying Reformulations for Increased Monotonicity

In general a transformation requires two steps

1. Generating new parameters and
2. Selecting useful sets of new parameters

These two steps in principle could be approached with exhaustive combinatorial methods. For life size problems this approach is not practical. Instead we focus on generating new parameters with physical meaning and on grouping them into basis sets using an incremental search technique that seeks to continuously reduce the number of nonmonotonic assignments.

Generation of Parameters According to Physical Meaning

To insure that the new parameters maintain a strong element of physical meaning we use the parameters of the original problem to algebraically construct new parameters via basic physical meaning rules. These rules rely on knowledge about what a parameter physically represents rather than other qualities such as dimensions.

The basic approach is to construct new parameters as algebraic combinations of the original parameters. A similar concept is employed in dimensional analysis to nondimensionalize physical relationships. In many engineering disciplines dimensional analysis has been found useful typically because of the reduction in the number of parameters which have to be correlated. The goal of dimensional analysis is to transform the parameters considered important in some physical phenomena, to a set of dimensionless parameters often called a pi group [Buckingham 14]. The pi parameters are constructed algebraically as products of the original parameters raised to various integer powers. The easiest and most well known mathematical technique, but perhaps the least useful one, is based strictly on the dimensions of the original parameters. This approach involves the choice of a repeating group of the original parameters which can be used to algorithmically nondimensionalize each nonrepeating parameter. This technique is described in many texts, e.g. [Streeter 79], despite the fact that it often results in parameters with little physical meaning.

More successful approaches to dimensional analysis employ the principle of *similitude*, i.e. that two systems will exhibit similar behavior if geometric, kinematic, and dynamic similarity are maintained. Similitude approaches are based on the physical meaning of the parameters of the problem. Similitude conditions are satisfied if two systems are geometrically similar and if the ratio of all the pertinent forces are made the same in the two problems. The fact that almost all of the nondimensional parameters commonly used are one of four types, ratios of lengths, forces, energies, or properties [Kline 86], follows from the fact that all of the fundamental equations of continuum analysis can be nondimensionalized with these four types of nondimensional parameters. In many cases the forces and energies are constructed from more fundamental parameters. Reynold's number, for example, is the ratio of fluid inertial force to fluid viscous force. In turn these forces are constructed from the lower level parameters such as fluid velocity, viscosity, length, and mass density. Similarly, all of the common nondimensional parameters in fluid mechanics can be constructed from fluid force ratios.

Similitude considerations provide a few specific rules to facilitate the identification of

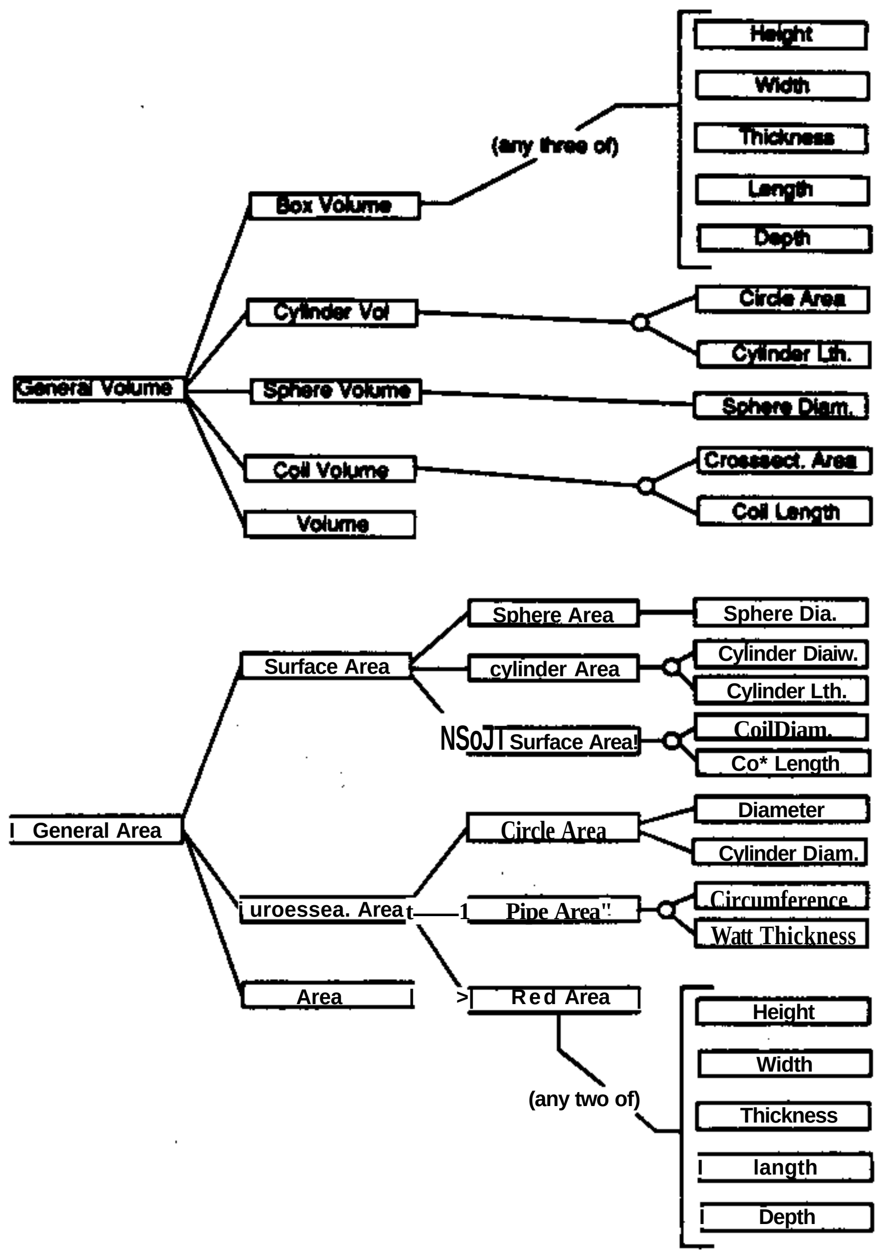

Figure 3: General Expansion of Some Geometry Rules

nondimensional parameters. For the construction of dimensional parameters, the rules are more varied and plentiftil. For example, dynamic forces can be inertial or viscous which in tum can be constructed from lower level parameters. An ineitial force may consist of the product of parameters representing a mass and an acceleration. In turn, an acceleration can be constructed according to other rules. Caution is required, however, because parameters may have the same dimensions yet represent distinct quantities. For example, both *torque* and *energy* have units of newton-meters.

We generate physically significant alternative parameters in a similar fashion: New parameters are constructed from algebraic combinations of existing parameters and they are constructed according to commonly used and understood engineering physical principles. Furthermore, we explicitly represent the physical significance of new parameters so that these new parameters can be used to generate additional parameters. The parameters constructed will then rely on previous engineering knowledge and experience. We have systematically collected physical meaning rules for several mechanical engineering domains. A compilation which can be found in [Watton 89] is organized into rules for geometry, kinematics, dynamics, solid mechanics, heat transfer, fluid mechanics, and electricity and magnetism. The rules are of two different types: product type and linear type. Product type rules define new parameters that are the product of constituent parameters to various exponential powers. The construction of Reynold's number is of this type. Linear type rules define new parameters as a linear combination of constituent parameters. In Figure 3 the rules in the geometry domain for constructing volumes and areas are expanded in a graphical structure. The small circle nodes represent rules requiring two or more constituent components. A direct connection between two boxes is interpreted as using a single parameter in the construction of a new one through a simple renaming, reciprocal, or squaring. For example, any circle area is also a cross section area. The parameters increase in generality from right to left across the page. Although some of the rules are quite specific, such as the construction of the coil volume, the set is not complete. The rules have been structured, however, to facilitate the addition of new parameter construction rules. The following example demonstrates how alternative parameters are created from the constants and variables in the original formulation of the two bar truss problem.

Example: Two Bar Truss. Consider some of the parameters and constants important in the design of a two bar truss:

d_o	: diameter, cylinder diameter
$d_{\{}$	: diameter, cylinder diameter
B	: width
H	: height

Note that a parameter may have multiple descriptions or names. The new parameters generated using the rules of geometry are:

$V_{x1} = \pi d_o$	: circumference
V ^ s $nd.$	: circumference
V_x^{ids} * BH	: rectangular area, cross section area
V_2 » $0.25nd_o^2$ » A_o	: circle area, cross section area
V_3 » $O.lSndj^1$ m A_i	: circle area, cross section area
V_4 a $d_o ld_i$	: ratio of diameters
V_5 = dJH	: ratio of lengths
V_6 « dJB	: ratio of lengths
$V,$ « dJH	: ratio of lengths
$V*$ = dJB	: ratio of lengths
V_9 m BIH m R	: ratio of lengths

Note that the first two new parameters, V_{xl} and $V\&$ are slightly disguised original parameters. They are not useful for transformations of variable but they are useful building blocks for other new parameters. In the final list they will be discarded. With the new elements other parameters of the linear type can be constructed:

$V_{10} = V_2 - V_3 = A_o - A_i = A_n$: area difference
$V_{11} = V_2 - V_1 = A_O\text{-}BH$: area difference
$V_{12} = V_3 - V_1 = A_iBH$: area difference
$V_{13} = 2H + 2B$: rectangular circumference

These constructions are straightforward to perform but some errors in assigning physical meaning have been made. The parameters V_n and V_{l2} are not really meaningful area difference since tube inside and outside areas are measured in planes perpendicular to the rectangular area spanned by H and S. In addition, the variables K_5, V_6, K_7, and V_g are also not very meaningful. Other candidate parameters were created but eliminated because they were reciprocals of others or similar to others except for a constant multiplier, such as V_{xX} and $V\&$

There are a number of limitations to the approach. First, is the impracticality of a complete collection of rules. It is straightforward to create and use high level general rules, such as creating a General Area from a Surface Area, but it is more difficult to create specific rules for complicated geometries which can be applied successfully. Having a rule for every configuration is impossible, nevertheless, the ability to add more rules exists in the organizational structure. What can be accomplished with a small collection of rules, even if quite general, is considerable. The second limitation is the possibility of making errors in the assignment of physical meaning. For example, the area of a rectangle can be constructed as the product of a height and a width, but if there are several of each of these quantities, then how do we construct correa rectangular areas? There are many possibilities to consider - some of them may create areas and some of them may not. In the next section we employ the idea of spatial proximity to help reduce the number of new parameters generated to a reasonable level and avoid some of the more obvious errors.

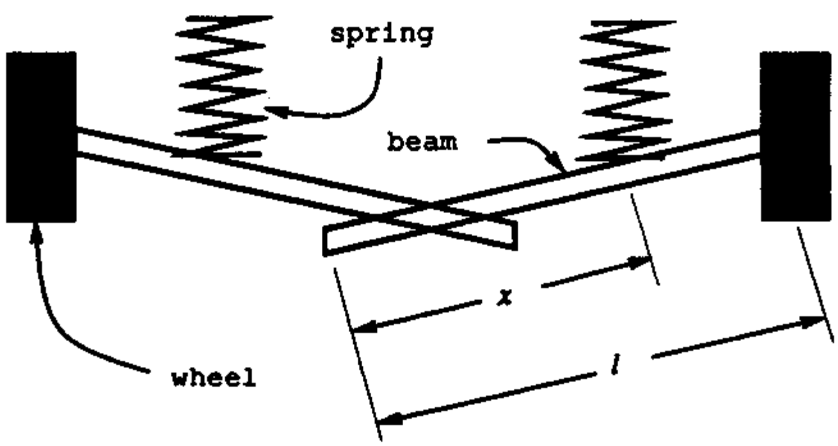

Figure 4: A Simple Suspension System

Spatial or Component Proximity Considerations

A mechanical system generally consists of many components. For example, a helical coil spring may be considered a single component but if it is used in a suspension system then it would be one of several components. Figure 4 shows an example of a simple suspension system with two basic components, a spring and a beam. The proximity with which components are connected often indicates the "associative" importance of the parameters "measured on" or "belonging to" each component. Parameters "measured on" the same component may form, when grouped among themselves using the physical meaning rules, new parameters with the greatest likelihood of being important and without errors in assigning physical meaning. The parameters measured on two spatially adjacent components may be grouped together, as well, but with an increasing likelihood of errors and so on as the components spatially depart from each other. This principle is similar to Huntley's addition [Huntley S3] to dimensional analysis where he notes that there is a distinction between a length measured in the x direction and one in the *y* direction.

Consider the following example. For the suspension system of Figure 4 we can, for the purpose of this example, design for wheel deflection, 8 ^ ^ , and stiffness, k_w, disregarding what other important functions might be desired:

$$6_{maxw} = \frac{8\pi\tau_{max} D^2 nl}{9Gdx} = h_x(d_9 D_f n,x,t) \tag{17}$$

$$k_w = \frac{Gd^4x^3}{8/iD^3/^3} = h_2(d,D,n,x,l)$$

The original design parameters are {n,d,D,x,/}. The parameters $\{n_f d,D)$ are all strictly spring design parameters, while / and x are beam design parameters. The following table indicates which design parameters belong to which components, where X indicates ownership, and *o* no ownership:

parameter	spring	beam
n -No.coils	X	*0*
d -Wirediameter	X	*0*
D -Coildiameter	X	*0*
x -Position	*0*	X
/ -BeamLength	*0*	X

If we include all design parameters despite ownership then the set used to generate new design parameters is {n,D,d,x,/} and 16 new product type design parameters can be generated with the present set of rules. If only X relationships are used then we have two separate sets $\{n_9 d_9 D\}$ and {x,/} and mixing between the two is not permitted. With this restriction 12 new product type parameters can be generated. The second set of groupings produces fewer new parameters to consider and helps avoid some obvious errors in physical meaning.

In the computer implementation we assign both a primary and secondary ownership of parameters to the components of the mechanical configuration. The most restrictive criterion is to group only members of common primary ownership. If this fails to generate enough candidate new parameters the condition can be relaxed by including parameters of secondary ownership as welL Finally, all parameters can be grouped together ignoring component ownerships for those designs where generating sufficient candidates is difficult (usually not the case). This approach to component proximity considerations is rather primitive because all the rules of candidate parameter construction are treated the same. A refinement would be to use a set of meta-rules to specify the degree to which a spatial or component proximity condition should be applied.

Methods to Generate Transformations

So far we have discussed only how to create new parameters which have the strong possibility of being physically significant and meaningful to a designer for a specific mechanical design problem. The result of this approach is to produce a list of possible new parameters algebraically constructed from parameters in the original problem. Needless to say, such a list can be of significant length and not all of the new parameters can be used in any one transformation. Usually there are many possible subgroupings of the new parameters, (in addition to some original parameters) which will "span" the original design space. In this section, we are concerned with creating and using spanning sets to perform transformations, including techniques to algebraically perform the transformation and to explore as many spanning sets as possible. The choice of transformation depends fundamentally on the desirable characteristics of the transformed design equations and measures of those characteristics.

Monotonicity Measure of Improved Formulations

Any measure of design equation structure can be used to select among alternative transformations of variable. In this paper we focus on improvements which facilitate the application of monotonicity analysis.[4] A refined measure could be based on the actual application of monotonicity analysis to the transformed design equations. The measure which we employed and implemented in EUDOXUS is an expedient alternative: It is a simple sum of the number of nonmonotonic variable occur usually rences in the design equations. In the original formulation of the truss problem there were six occurrences of nonmonotonicities. The reformulated equations exhibited only two, a net improvement of four in the measure of degree of monotonicity. Counting the nonmonotonicities programmatically for symbolic mathematical representations requires certain restrictions. If the constraints are polynomials of any order (including variables to negative exponents) and the ranges on all variables and constants are nonnegative (as is often the case in mechanical design) then conclusions about global monotonicities can be drawn. When these restrictions are not present other means can be employed, e.g. applying interval arithmetic to the symbolic derivative of a constraint or objective. In the case of the truss, the simple restrictions apply and the measure of monotonicity is easily implemented and quickly computed. A more refined measure based on a full implementation of the monotonicity analysis reasoning could be tailored specifically to reward the identification of irrelevant variables, the elimination of constraints, or a reduction in the number of objective variables, however the measure would be significantly more difficult to implement and compute.

[4]We have used a similar approach in previous work which was focussed on design equation coupling [Walton 89, Walton 90]

Combinatorial and Incremental Approaches to Forming Basis Sets

How ait die basis sets of the successful reformulations to be formed from the large number of new **parameters generated?** Two methods have been explored, namely the combinatorial algorithm and the incremental heuristic. Prior to describing these methods we first discuss what constitutes spanning and basis sets. **We have** used the term *spanning set* to refer to an alternative set of variable parameters which will **"span" the same** design space as the original variable parameters. We want each variable to be independent from the others in the set so that it forms a *basis* or *basis set* for describing the design space. A basis set is a spanning set where the elements are independent In this sense, when we refer to sets of new variable parameters which span the original design space, we mean a minimal set which can act as an alternative coordinate system for the design description. For a parameterized design description the variable parameters form a basis for the solution space. We explore a change of basis by symbolically transforming the design equations. When the transformation of variables involves exclusively either linear or product type new parameters, the transformation is straightforward (for the details see [Watton 90, Watton 89]). More generally, however, a new basis might involve a combination of new parameter types such as polynomials or variables based on transcendental functions. These types of new parameters may be useful, but there is difficulty in creating algorithms to handle the algebra symbolically. Nevertheless, many, if not most new parameters of interest are of the linear type or of die piodua type and a transformation that involves a combination of both types can be achieved incrementally. In fact, using the incremental technique it is possible to obtain some polynomial type new parameters [Watton 89]. This is **the** case for the two bar truss example.

We first discuss a *combinatorial algorithm* which evaluates all basis sets which can be constructed from some combination of the available parameters. This method is most useful on small problems and as a benchmark against which other methods may be evaluated. The computational expense of this method depends on the total number of candidate variables generated which depends on the number of variables in the original formulation. The algorithm is straight forward. For example, if the original system of design equations has a basis set of five design parameters, then all combinations of groups of five from die total list of candidates and the original five design parameters are generated. Original parameters are included because many useful reformulations include mostly the originals parameters with the addition of a few new parameters. Each set generated is then tested to determine if it is a basis set One quick test is to check that each original parameter of the basis set is either contained in the new set or is used in the construction of at least one new parameter in the set The final test is when the proposed basis is solved to isolate the original parameters in order to execute the transformation. If this inverting procedure is possible then a valid basis exists. The final step is to perform the substitutions and check the resulting reformulations for measured improvements over the original formulation. This method becomes impractical for most realistic design problems since the number of proposed basis sets to check increases with the size of the list of candidates, n, and the size of the basis set, m, according to:

$$\text{combinations} = \frac{n!}{m!(n-m)!} \qquad (18)$$

Another drawback is that of the two types of new parameters generated, namely product type and linear type, a proposed basis set that includes both types cannot be used unless an incremental approach is employed due to the limitations of the specific transformation methods we employ. Probably, the most useful property of die combinatorial method is the fact that it can be used to compare the effectiveness of less powerful methods.

Alternatively, we consider an *incremental heuristic* which intelligently focuses the search for basis sets that lead to improved formulations. This heuristic involves replacing one of the original members of a basis set with a new parameter to form a new basis set After applying the transformation the new set of equations is evaluated for improvements in the measure of monotonicity. For all improved reformulations, the same procedure is applied again. When none of the remaining new parameters can be used to create a reformulation that has an improved measure then the bottom of a branch is reached. The basis set along with the design equations it produces is offered as an answer. The bottom of a branch is also reached when the list of new parameters is exhausted. The incremental method will not explore all combinations, but it should focus on sets of new variables which will offer improvements. The incremental method suffers from what is called the horizon problem. The algorithm, as described, does not explore paths where the measure does not improve continuously from node to node. A useful reformulation may be overlooked because the measure had to get worse before getting better. This problem is somewhat compensated by the fact that multiple paths to the same new basis set generally exist. One advantage of the incremental method over the combinatorial method is that it can handle basis sets which contain both linear and product type new parameters. This is because it seeks only to add me new parameter at a time and thus at any point can use either the linear variable or produa variable method of transformation - whichever is appropriate. To help overcome the horizon problem relaxed conditions of incremental search are pennitted. In this situation the search is permitted to continue along paths where the measure of monotonicity may not initially improve. This allows one to find more reformulations (at a time penalty) and also allows the linear type parameters to be more fully utilized.

Implementation and Execution Details

The techniques discussed in this paper were implemented and tested on a Symbolics Maclvory computer. Common Lisp was chosen for the implementation because it is well suited for the symbolic computations needed to transform systems of algebraic expressions. In this section, we will discuss the basic structure and flow control of the EUDOXUS implementation, including its limitations. The program flow chart is shown in Figure 5. An example of the input and output of the program can be seen in [Watton89].

The flow chart shows that the initial stage involves defining the appropriate description of the design problem to be considered. This is done in the form of an input file. The main aspects to be included in this file are the design equations in algebraic symbolic format with an indication of which parameters are the variables and the constants. Information on each parameter must include units, and a physical meaning name or type (e.g. area, diameter, torque, velocity). The constraints are limited to algebraic expressions constructed with basic operations (e.g. addition, subtraction, division, multiplication, exponentiation) and function calls (e.g. sine, cosine, log, etc.). Differential and integral equations are not admissible. The user must specify which design parameters are permitted to be used in the construction of candidate new parameters. Extensive checking routines insure that the problem is well posed. The implementation is not limited by the size of the design problem as measured by the number of expressions and parameters except by execution time and virtual memory space.

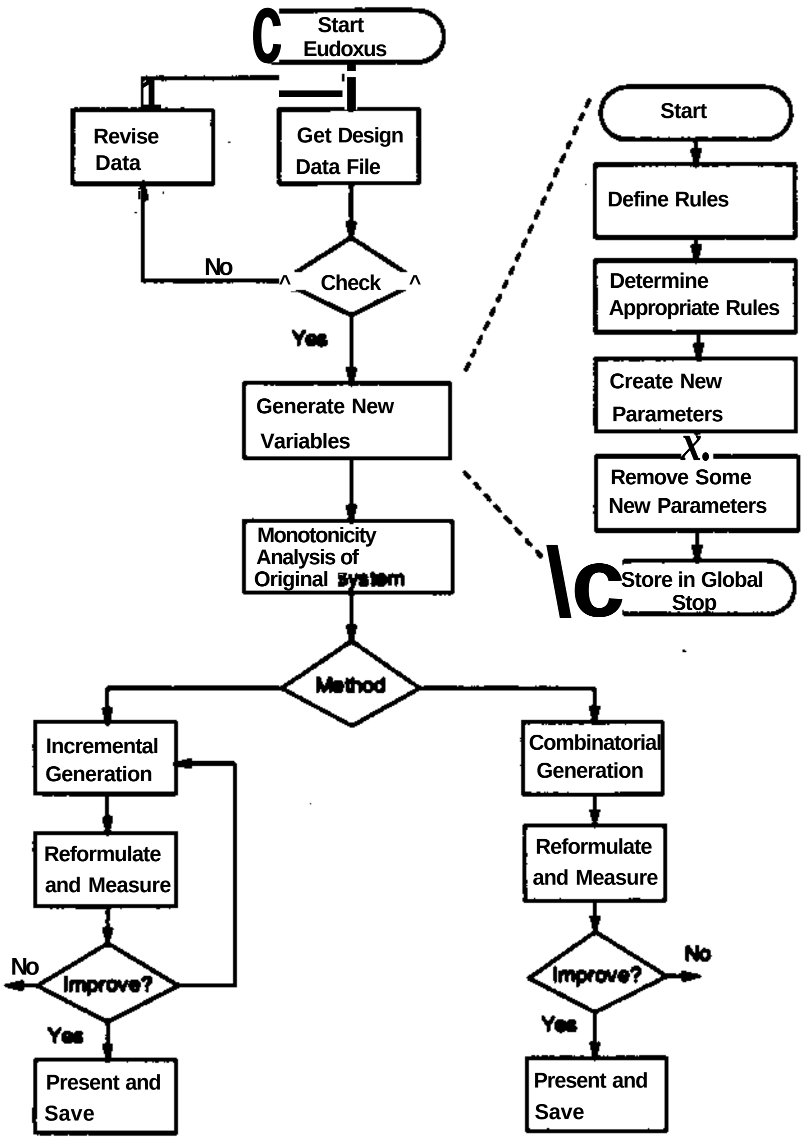

Figure 5: Flowchart of the EUDOXUS Implementation

The implementation is limited, however, by the type and quantity of routines used in the mathematical manipulations. Such basic functions as collecting like terms and canceling terms in division are limited to the simplest of cases. In addition, because the methods implemented in EUDOXUS are not complete and algorithmic but based on a generate and test scheme it is not possible to defend them strictly on a theoretical basis. A summary of the results for the truss problem is presented in Table 1. The first row indicates whether the run used the incremental or combinatorial method, the second row indicates if the incremental method was relaxed, and the third row indicates if the incremental method was verbose or not. A relaxed incremental mode permits a noncontinuous improvement in the

Two **Bar** Truss - (6 constraints, 4 variables)					
	truss-1	**truss-2**	tniss-3	**truss-4**	**truss-5**
method	**incr.**	**incr.**	**incr.**	**incr.**	**comb.**
relaxed	**no**	**no**	**yes**	**no**	**n/a**
verbose	**no**	**yes**	**n/a**	**yes**	**n/a**
no. product vars.	9	9	9	9	9
no. linear vars.	4	4	4	0	0
cpu time (sec)	16	18	221	15	50
no. answers	2	16	59	16	**44**

Table 1: Computer Execution for the Two Bar Truss

monotonicity measure by one step. This helps alleviate the horizon problem, especially when working with combinations of linear and product type new parameters. A verbose run presents all improvements over the original system while the incremental mode operating in a non-verbose manner will present only those reformulations which are improvements from the previous node. The remaining rows provide details on the results. The number of answers in the last row are the number of reformulations discovered with an improved measure of monotonicity.

To compare the incremental and combinatorial method we restrict ourselves to the situation where only product type new parameters are generated (tmss-4 and tmss-5). Comparing tniss-4 and truss-5 we see that the combinatorial method takes three times as much cpu time to find almost three times as many reformulations. The 16 answers of truss-4 are a subset of the 44 answers of tmss-5. The trend for the combinatorial method is to take ever more time to find fewer additional reformulations. Only small problems are reasonable candidates for the combinatorial method. Small is measured by the number of new variables generated which is usually a function of the number of variables in the original formulation

The other advantage of the incremental method is that it can be more focused operating in a non-verbose manner that presents only critical reformulations which are improvements from the previous node. Sometimes the use of a certain new parameter is the reason for an improvement, but often that new parameter can be grouped with others that make no difference to the measure of improvement It is these essentially repeat reformulations which can be eliminated with the incremental technique. The combinatorial method is blind to which of the new parameters are the most critical ones in any particular reformulation. Comparing tiuss-1 and truss-2 we see that of the 16 incremental reformulations only 2 of them are different since the answers of truss-1 are a subset of those for truss-2. This will allow the user to choose the most critical reformulations from those discovered with the incremental technique.

The situation in truss-3 is special. The relaxed conditions of incremental search are permitted which allows the linear type parameters to be fully utilized. The result is 59 improved formulations in 221 seconds. Without relaxing the search conditions (truss-2) only about a quarter of these reformulations were found and none contained new linear type parameters. In fact, truss-3 is the only run that contained the reformulation used earlier in the two bar truss problem.

Conclusion

Mechanical design is a process of generating artifacts that will perform as required. It is generally an iterative process that involves both a synthesis of form and an analysis to determine if that form meets the required functions. If a satisfactory solution is possible the next step is optimization with respect to a particular objective. Reformulating parametric constraints that describe candidate design artifacts can be an important and useful part of this process. In this paper we have outlined a series of methods that have been implemented to automate the procedure of finding useful transformations of mechanical design constraints. We demonstrated the utility of using variable transformations for constrained nonlinear optimization problems. Increasingly monotonic representations help identify the active constraints of an optimization through the application of monotonicity analysis rules.

Acknowledgements

The authors are pleased to acknowledge the support of the Design Theory and Methodology Program of the National Science Foundation (NSF Grants DMC-84-51619 and DMC-88-14760) and the Engineering Design Research Center at Carnegie Mellon University (NSF Grant CDR-85-22616).

References

[Agogino 87]
Agogino, A.M., "AI in Computer-Aided Design: Qualitative Reasoning and Symbolic Computation," *Proceedings from the NSF Workshop on the Design Process,* Waldron, M.B. ed., Oakland, CA, February 1987.

[Buckingham 14]
Buckingham, E., "On Physically Similar Systems: Illustrations of the use of Dimensional Equations," £. *Phys. Rev.,* Vol. 4,1914, pp. 345.

[Fox 71]
Fox, R.L., *Optimization Methodsfor Engineering Design,* Addison-Wesley Pub. Co., 1971.

[Huntley S3]
Huntley, H.E., *Dimensional Analysis,* McDonald and Company, London, 19S3.

[Kline 86]
Kline, S J., *Similitude and Approximation Theory,* Springer-Verlag New York Inc., 1986.

[Papalambros 79]
Papalambros, P. and Wilde, DJ., "Global Non-iterative Design Optimization Using Monotonicity Analysis," *ASME Journal ofMechanical Design,* Vol. 101, March 1979, pp. 645-649.

[Papalambros 88]
Papalambros, P. and Wilde, D J., *Principles of Optimal Design,* Cambridge University Press, 1988.

[Parkinson 85]
Parkinson, A*R.*, Balling RJ., and Free, J.C., "Exploring Design Space," *Computers in Mechanical Engineering,* March 1985, pp. 28-36.

[Strecter79]
Strceier, VX. **and** Wylie, E.B., *Fluid Mechanics,* McGraw-Hill Book Company, 1979.

[Watton89]
Watson, John D.9 *Automatic Reformulation ofMechanical Design Constraints to Enhance Qualitative and Quantitative Design Decisions,* PhD dissertation, Carnegie Mellon University, April 1989.

[Watton90]
Watton J.D. and Rinderle, J.R., "A Method to Identify Reformulations of Mechanical Parametric Constraints to Enhance Design," *Proceedings of the Second ASME Design Theory and Methodology Conference,* September 1990.

[WikJe78]
Wilde, DJ., *Globally Optimal Design,* Wiky-Interscience, 1978.

[Wilde 86]
Wilde, DJ., "A Maximal Activity Principle for Eliminating Overconstrained Optimization Cases," *ASME Journal of Mechanisms, Transmissions, and Automation in Design,* Vol. 108, December 1986, pp. 312-314.