Taylor & Francis
Taylor & Francis Group

Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: http://www.tandfonline.com/loi/geno20

TECHNIQUES FOR INTEGRATING QUALITATIVE REASONING AND SYMBOLIC COMPUTATION IN ENGINEERING OPTIMIZATION

ALICE M. AGOGINO & ANN S. ALMGREN

Eng. Opt., 1987, Vol. 12, pp. 117–135
Photocopying permitted by license only

TECHNIQUES FOR INTEGRATING QUALITATIVE REASONING AND SYMBOLIC COMPUTATION IN ENGINEERING OPTIMIZATION

ALICE M. AGOGINO and ANN S. ALMGREN

Department of Mechanical Engineering,
University of California, Berkeley, CA 94720 *USA*
aagogino@euler.berkeley.edu

This paper presents techniques that extend theories from the fields of artificial intelligence (AI) and engineering optimization for automating design decision making at three levels of reasoning: *qualitative, functional, and numerical.* Qualitative reasoning about constraint activity is implemented in SYMON (SYmbolic MONotonicity analyzer) through monotonicity analysis and the maximal activity principle. Functional reasoning is employed in SYMFUNE (SYMbolic FUNctional Evaluator) in the form of algebraic manipulations of the constraint functions and the Karush-Kuhn-Tucker optimality conditions. The techniques are applied to a parametric multiobjective optimal design problem from the literature

KEYWORDS: Qualitative reasoning, artificial intelligence, symbolic computation, parametric and multiobjective optimization.

1 INTRODUCTION

This paper focuses on three kinds of human and computer-assisted reasoning: *qualitative, functional, and numerical.* AI approaches in qualitative reasoning and symbolic computation are proposed as the mechanisms for integrating all three levels into the SYMON and SYMFUNE computer-aided design tools.

1.1 Qualitative Reasoning

Reasoning at the *qualitative level* is defined as "reasoning about objects and their qualities or parameters in a way that does not rely on specific numerical values". Programs utilizing artificial intelligence (AI) techniques have concentrated on the qualitative level by means of pattern matching algorithms that operate on list structures. Production rules in knowledge-based systems have recently demonstrated potential in applying AI to engineering design[1–4]. The "knowledge" is written in a declarative format and can be read directly from the data base, making it easy to understand the underlying knowledge behind the program. (See Gero *et al.*[5] for a comparison of procedural and declarative programming languages for optimization problems.) Although object-oriented programming techniques and frame representations do much to capture the fundamental physical relationships between objects in rule-based systems[4,6], this knowledge, for the most part, is shallow with little apparent relationship to the underlying mathematical, engineering, or physical principles involved.

Recent work in qualitative process theory extends these rule-based techniques to take into account monotonic information of the behavior of system variables and to qualitatively reason about them[7–8]. For continuous functions, qualitative trends can be defined by the sign of the gradient terms of system interrelationships. For example, an engineer might reason that the potential energy of a spring, E, increases with the displacement from equilibrium, x ($\partial E/\partial x \geq 0$) or that the tangential stress, s_t in a thin-walled cylinder decreases as the thickness t of the cylinder increases ($\partial s_t/\partial t \leq 0$). In this context, qualitative information does not depend on knowing the exact functional form of the system interrelationships, only the qualitative trends. Thus one could still make the same qualitative statement given previously about the interrelationship between potential energy of a spring and displacement even for a nonlinear spring in which the functional relationship was not precisely known. A few researchers have succeeded in combining qualitative reasoning in a rule-based approach with numerical optimization codes[9–10]. Written in procedural languages in order to communicate directly with the numerical algorithms involved, these programs do not have the ability to explicitly reason at the *functional* level except to the extent that FORTRAN or PASCAL statements may be used to provide a mapping between input and output at a numerical level.

1.2 Reasoning with Functional Information

Often the strongest information designers have about a physical system is at the *functional level*. A designer might reason that the potential energy associated with most springs varies with the *square* of the displacement from equilibrium ($E \sim x^2$) and that the tangential stress in thin-walled cylinders varies *inversely* with the thickness ($s_t \sim 1/t$). Computer programs developed to perform symbolic computation in the early 1960s are examples of the first expert systems ever developed. Originally based on heuristics, these programs now employ intricate mathematical algorithms to perform symbolic computation at an impressive level of technical proficiency. Although symbolic computation allows computer-based reasoning at the functional level, very few applications can be found in the engineering literature. Notable exceptions are in the fields of mechanical dynamics and control theory[11–14].

1.3 Numerical Analysis

By far the most prevalent use of computers in the engineering community is at the *numerical level*, e.g., numerical optimization (MINOS[15] and GRG2[16]), finite element and difference programs (ANSYS[17] and NASTRAN[18]), and numerical simulation packages (Parasol[19]). For the most part, these programs map one set of input numbers to another set of output numbers. The underlying knowledge embedded in these programs is hidden within the structure of the programs themselves. The source code is often not available for the majority of users and even if it were it would be difficult to directly appreciate the underlying knowledge behind the programs by reading the code.

In what follows, techniques to integrate all three levels of human reasoning in a computer-aided optimal design system are described. SYMON reasons qualitatively, SYMFUNE performs functional evaluations, and FORTRANIZER translates to standard FORTRAN code if further numerical analysis is desired. The purpose of these programs is to provide insight, in addition to solutions, in order to complement and enhance the expertise of the human designer.

2 ENGINEERING OPTIMAL DESIGN

Analytical and numerical optimization techniques have significantly contributed to recent advances in computer-aided design. Radford and Gero[20] refer to design as a goal-oriented problem solving activity. The goal is to find a design that is feasible (satisfying the imposed equality and inequality constraints) and optimal (ranking highest in design objectives relative to alternate feasible designs).

As with other applications of computer-aided design, optimization has often been considered primarily a numerical process. Yet human designers who use qualitative and functional levels of reasoning, rather than numerical processing, achieve added insight into the important aspects of the problem's structure and solutions in parametric form. Unfortunately, it does not take a large increase in the complexity of a design problem to confound even the most experienced designer, and numerical schemes are the alternative. Although the literature reflects interesting work in numerical sensitivity analysis, most techniques only apply locally and, in general, the problem must be numerically re-solved for large changes of parameters because of the nonlinearities present in most engineering design problems and difficulties in predicting changes in constraint activity.

Computer techniques that can reduce optimization problems at a non-numeric level are proposed. The AI language FranzLISP and the symbolic computation language VAXIMA are used to implement these techniques into a set of programs that integrate qualitative, functional and numerical reasoning in a computer-aided optimal design system called SYMON-SYMFUNE (Figure 1).

The input to SYMON is a description of the optimal design problem in either functional or qualitative form. SYMON performs symbolic monotonicity analysis on this information. The output from SYMON is useful in providing guidance about qualitative trends for design optimization and constraint activity and is input to the SYMFUNE program. If not already provided, this should be supplemented with functional equations when available. SYMFUNE performs symbolic optimization using a combination of augmented Lagrangian methods, symbolic monotonicity

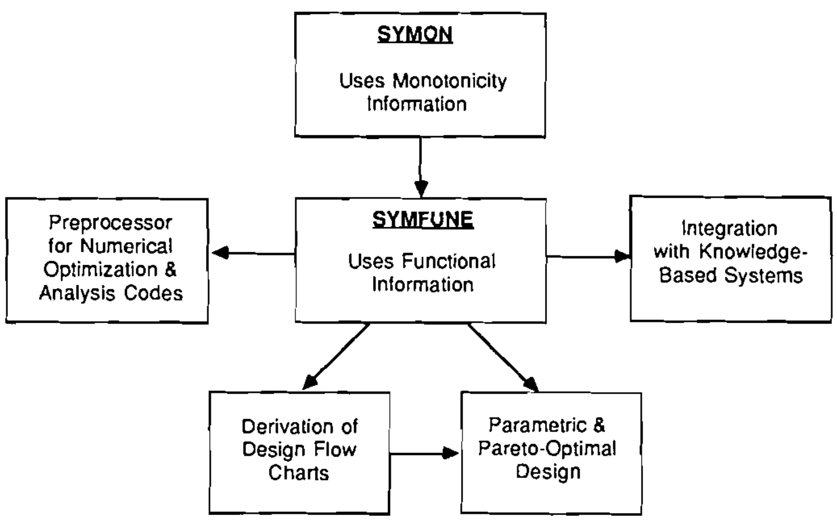

Figure 1 SYMON-SYMFUNE flow chart.

analysis and application of the Karush–Kuhn–Tucker optimality conditions. Although SYMON and SYMFUNE programs stand effectively on their own, they can also be used as preprocessors for integration with numerical-oriented optimization codes. Their use in the derivation of design flow charts and in parametric and multicriteria design will be demonstrated on the multiobjective hydraulic cylinder design problem. Integration with knowledge-based systems is suggested in the Discussion.

2.1 SYmbolic MONotonicity Analysis (SYMON)

Monotonicity analysis is an iterative partial optimization technique to reduce the dimensionality of the optimization problem and detect flaws in the problem formulation. The unconditionally active constraint sets and the combinations of active constraints which yield potentially feasible and bounded solutions are found based solely on the monotonicities (qualitative level) of the objective function and constraints. In order to formalize the methodology, the following terms concerning monotonic properties and constraint activity in optimization are introduced for future reference. Without loss of generality, positive systems are assumed.

Definitions:

a. The *monotonicity* of a continuously differentiable function $f(x)$ with respect to (w.r.t.) variable x_k is the algebraic sign of $\partial f/\partial x_k$. A discrete or continuous function $f(x)$ is *strictly globally monotonically increasing* over a domain Ω if and only if (iff) $f(x_1) < f(x_2) \forall \{x_1, x_2 \in \Omega: x_2 > x_1\}$ and *strictly globally monotonically decreasing* iff $f(x_1) > (x_2) \forall \{x_1, x_2 \in \Omega: x_2 > x_1\}$. The monotonicity of a function w.r.t. a variable is designated by a positive or negative superscript for the variable in the argument list of the function. For example, the term $f(x_1^+, x_2^-)$ implies that the function is monotonically increasing with respect to x_1 and decreasing with respect to x_2.

b. A positive variable x_k is said to be *bounded above* by a constraint $g_i(x) \leq 0$ if it achieves its maximum value at strict equality, i.e., when the constraint is active. A positive variable x_k is *bounded below* by $g_i(x) \leq 0$ if it achieves its minimum value at strict equality. Thus if $g_i(x)$ is monotonically increasing with respect to a variable x_j, then the inequality constraint $g_i(x) \leq 0$ bounds x_j from above. If $g_i(x)$ is monotonically decreasing with respect to a variable x_k, then the inequality constraint $g_i(x) \leq 0$ bounds x_k from below.

c. An inequality constraint $g_i(x) \leq 0$ is *active* at x_0 if $g_i(x_0) = 0$. An inequality constraint $g_i(x) \leq 0$ is *inactive* at x_0 if $g_i(x_0) < 0$.

d. A problem is *bounded at optimality* by a set of J active constraints $\{g_j = 0$ with indices $j \in J \in M$, where M is the set of indices for all of the constraints$\}$ if, for each variable, optimization does not drive that variable to plus or minus infinity (or degeneracy for positive systems).

e. A constraint $g_i(x) \leq 0$ is said to be *unconditionally active* at optimality if elimination of that constraint will lead to an unbounded or degenerate solution at optimality. An equality constraint $h_i(x) = 0$ is active and *relevant* at optimality if elimination of the constraint will lead to an unbounded or degenerate solution at optimality. In this sense, the equality can be changed to an active inequality constraint, the direction of which defines the *directionality* of the equality. An *irrelevant* equality constraint is one that does not bound any variable in the objective function to be minimized (in numerical terms the associated Lagrange multiplier is zero). An irrelevant equality may be used to define a system variable, but the value of that variable has no influence on the objective function at optimality. For convenience, an irrelevant equality constraint will be called "inactive" at optimality.

f. A set of J constraints $\{g_j : j \in J \in M\}$ is said to be *conditionally active* at optimality if elimination of *all* of the constraints in the set will lead to an unbounded or degenerate solution at optimality.
g. The *dimensionality of an active constraint set* is the number of degrees of freedom in the solution of the optimization problem, as determined by the number of variables minus the number of nonredundant active constraints.
h. An *optimal solution candidate* is a sufficient set of equations representing enough constraints to bind every relevant variable so that optimization of the objective will be bounded at optimality.
i. The *domain of optimality* is the set of active inequalities which define the range of parameter values where a certain solution is optimal and bounded.

The foundation for monotonicity analysis is two rules for defining well-constrained optimization problems[21].

Rule One: *If the objective function is monotonic with respect to a variable, then there exists at least one active constraint which bounds the variable in the direction opposite of the objective.*

Rule Two: *If a variable is not contained in the objective function then it must be either bounded from both above and below by active constraints or not actively bounded at all* (i.e. all constraints monotonic w.r.t that variable must be inactive).

Wilde[22] has proposed a third rule (the maximal activity principle) to eliminate overconstrained cases. It restricts the dimensionality of any set of active inequalities to be non-negative.

Maximal Activity Principle: *The number of nonredundant active constraints cannot exceed the total number of variables.*

Although the above rules are qualitative in nature and do not appear to be consistent with the precise mathematics of optimization theory, they in fact, represent necessary conditions for optimality of monotonic systems, a special case of the Karush–Kuhn–Tucker optimality conditions in nonlinear programming. The first two rules provide logical procedures for identifying unconditionally active inequality constraints and sets of conditionally active constraints. They also help determine relevance and directionality of equality constraints (the direction of the inequality that could replace an equality constraint and lead to the equivalent optimal solution). Proofs of these rules can be derived by constructing an optimization problem with monotonic functions and applying the Karush–Kuhn–Tucker optimality conditions. Differentiation of the Lagrangian w.r.t. variables in the objective function gives the conditions in rule one, and differentiation with respect to variables that are not in the objective leads to rule two. The maximal activity principle is used to eliminate infeasible subsets of constraints that would lead to overconstrained cases.

Additional Optimization Heuristics. Heuristics are often adopted on a problem by problem basis in optimization[23]. In fact, the Karush-Kuhn-Tucker optimality conditions can be traced to a heuristic principle often stressed by Richard Courant ... *in a variational problem where an inequality is a constraint, a solution always behaves as if the inequality were absent, or satisfies strict equality*[24]. The following heuristic is useful in eliminating constraints whose monotonicities are redundant to those in the objective. Application of the monotonic redundancy principle is a form of *default reasoning*. The simplifications obtained are considered valid until the resulting solution violates a constraint assumed inactive from application of the heuristic.

Monotonic Redundancy Principle: *If a minimizing (maximizing) objective function is monotonic w.r.t. a variable x_i, then it is unlikely to be limited by a constraint $g_j \leq 0$ which has the same (opposite) monotonicity w.r.t x_i. If the constraint g_j has the same (opposite) monotonicities as those in the minimizing (maximizing) objective function w.r.t all of its variables, then as a first assumption consider the constraint to be inactive.*

These rules have been automated by Choy and Agogino[25] in the program SYMON (SYmbolic MONotonicity analyzer), written in VAXIMA language running on DEC Vax minicomputers under the Unix™ 4.3 BSD operating system. VAXIMA[12,26] is a symbolic math language written in FranzLISP[27] which supports the symbolic manipulation necessary to perform analysis at the qualitative and functional levels. The input to SYMON is the problem statement: the objective function to be minimized and the inequality and equality constraints. This input can be expressed either in functional form or qualitatively in terms of the monotonicities of each variable. SYMON differentiates the functional forms symbolically in order to obtain monotonicity information when necessary.

The output from SYMON is a list of combinations of active and inactive constraints (a superset of the final optimal solution candidates) which yield *potentially* feasible and bounded solutions. SYMON also outputs the solutions in final functional form, when possible, for the constraint-bound solutions, and in reduced functional form for those solutions with positive degrees of freedom.

Process for Automating Monotonicity Analysis. The process of applying these rules exhaustively is shown in Figure 2. First the monotonicities of the terms are determined where possible and stored in list structures for further analysis. Unresolved monotonicities are recorded as question marks. A graphical monotonicity table is constructed for the benefit of the user (an example of this display is given in Section 3). The formal statement of the problem in terms of the relevant monotonic functions is given below.

$$\text{Minimize } Z(\mathbf{x}^{\pm}) \text{ subject to } \mathbf{g}(\mathbf{x}^{\pm}) \leq \mathbf{0},\ \mathbf{h}(\mathbf{x}^{\pm}) = \mathbf{0}, \text{ and } \mathbf{x} > \mathbf{0}$$

The monotonic redundancy principle is applied to the list structures next, followed by rules one and two. Unconditionally active constraints are found by refutation. Each constraint is assumed inactive. If no potential optimal solution candidates can be found, then the constraint must be active. Finally the maximal activity principle is applied to all subsets determined previously. During all operations checks on the validity of the default assumptions are made and adjustments are made if necessary.

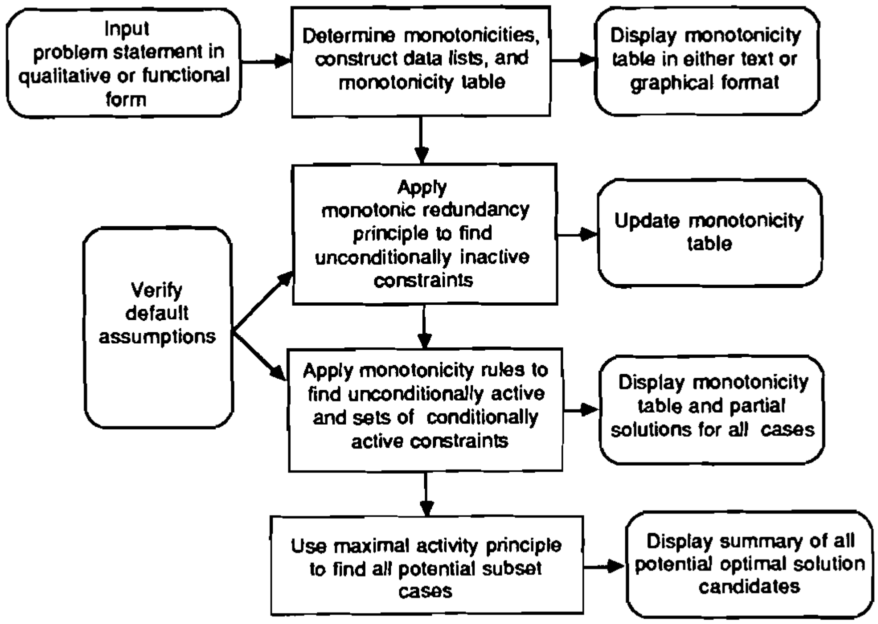

Figure 2 SYMON's monotonicity analysis process.

2.2 SYMbolic FUNctional Evaluation (SYMFUNE)

At this point the search space for feasible solutions has been reduced, and those cases not listed by SYMON will never yield feasible and bounded solutions. Numerical optimization routines could be employed to complete the solution. However, it is advantageous to complete the problem at the functional level where possible. Monotonicity analysis exploits the qualitative information in the problem, but functional evaluations can extend the analysis of the problem one step further before numerical processing is required.

Once the list of potential solutions has been derived by SYMON, SYMFUNE (pronounced "symphony") extends the analytical solutions of the optimization problem, first deciding which of the potential solutions are, in fact, feasible and bounded. If the solution is constraint-bound, then it is assumed to be bounded at the parametric level. If the solution is not constraint-bound, the objective function is re-evaluated by substituting in all active constraints, and the derivative of the objective function with respect to each remaining degree of freedom is set equal to zero (first-order necessary conditions for optimality). In theory, back-substitution would not be necessary if implicit differentiation was used instead. If the revised objective function is independent of the remaining free variables, then there are an infinite number of solutions, which must be expressed in parametric form, and limited only by the restrictions that the free variables satisfy the inactive inequality constraints. This unusual situation is illustrated in the following simple example:

$$\begin{aligned} &\text{Minimize} && f(x_1^+, x_2^+) = (x_1)^2 + x_2 \\ &\text{subject to} && g_1(x_1^-, x_2^-) = C - (x_1)^2 - x_2 \le 0 \\ & && \boldsymbol{x} > 0 \end{aligned}$$

Application of rule one by SYMON reveals that the constraint g_1 is unconditionally active in order to bound the variables x_1 and x_2 from below. SYMFUNE substitutes the resulting equality constraint into the objective function to get the following reduced optimization problem: Minimize $f(x) = C$, *subject to* $x > 0$. Any non-negative value of x that satisfies constraint g_1 will yield the same constant value, C, for the objective function to be minimized. SYMON, using only monotonic information, does not detect this unusual feature for this problem. SYMFUNE, using functional information, can detect that the constraint is of the same functional form as the objective, revealing that the optimal solution is not unique.

If the function does depend on the free variables, and the derivative can be set equal to zero then the gradient equations are used to eliminate the remaining degrees of freedom and the solution is completed. If the derivative cannot be set equal to zero for any set of non-negative design variables then this case is unbounded or degenerate, revealing a *hidden monotonicity*. In this event, substitution of the unconditionally active constraints into the objective function has caused the objective to become monotonically increasing or decreasing and one or more inequality constraints must be made active. This case is illustrated in the following simple example:

$$\begin{aligned} \text{Minimize} \quad & f(x_1^+, x_2^-) = x_1/x_2 \\ \text{subject to} \quad & g_1(x_1^-, x_2^+) = x_2/(x_1)^2 - 1 \leq 0 \\ & x > 0 \end{aligned}$$

Application of rule one by SYMON reveals that constraint g_1 is unconditionally active in order to bound x_1 from below and x_2 from above. After SYMFUNE substitutes this equality into the objective, the reduced optimization problem is unbounded Minimize $1/x_1$, subject to $x_1 > 0$. Again it took functional information to reveal a hidden monotonicity not detected at the monotonic level in the original problem formulation. This example illustrates that the rules of monotonicity analysis are necessary but not sufficient conditions for a well-constrained optimization problem.

Once it is known which solutions are bounded, SYMFUNE must decide under what conditions, in terms of the given parameters and weighting factors, each solution is feasible and optimal. At this point, a solution is considered feasible if it satisfies the inactive inequality constraints and it is these parameter restrictions that comprise the *feasibility conditions.*

The Lagrange multiplier method with the Karush-Kuhn-Tucker (KKT) conditions is used in SYMFUNE to decide when each solution is optimal. The inequalities governing when each solution is feasible and optimal are referred to collectively as the *domain of optimality*. With this method, a reduced Lagrangian can be constructed on a case-by-case basis, since the number of feasible and bounded cases and their associated active constraints are known. The reduced Lagrangian is constructed by first expressing the objective function in terms of as few free variables as possible, by substituting in the relevant equality and unconditionally active inequality constraints, then adding to the objective function a term corresponding to each constraint active in this specific case. The term is constructed by multiplying the i^{th} inequality constraint (in the form $g_i(x) \leq 0$) by a (Lagrange) multiplier μ_i. Once the reduced Lagrangian is constructed SYMFUNE sets the derivative with respect to each of the free variables equal to zero. The resulting equations are used to solve for each of the multipliers, which are used to determine the optimality of a case. The mathematical optimization problem solved by SYMFUNE can be expressed formally as follows:

Given an objective function $f(x)$, $x \varepsilon R^n$, to be minimized, subject to

$$\boldsymbol{h}(x) = \mathbf{0} \tag{1}$$

$$\boldsymbol{g}(x) \leq \mathbf{0} \tag{2}$$

The augmented Lagrangian is defined as

$$L = f(x) + \boldsymbol{\lambda}^t \mathbf{h}(x) + \boldsymbol{\mu}^t \mathbf{g}(x) \tag{3}$$

Assuming that the constraints are independent at point x^*, the Karush-Kuhn-Tucker optimality conditions specify that x^* is a relative optima of $f(x)$ and satisfies the constraints $\boldsymbol{g}(x) \leq \mathbf{0}$ and $\boldsymbol{h}(x) = \mathbf{0}$ if and only if [28,29]:

$$\nabla \mathrm{L}(x^*) = \mathbf{0} \tag{4}$$

$$\boldsymbol{\mu}^t \boldsymbol{g}(x^*) = \mathbf{0} \tag{5}$$

$$\mu(x^*) \geq \mathbf{0} \tag{6}$$

where

$$\nabla \mathbf{L} = \nabla \mathrm{f}(x) + \boldsymbol{\lambda}^t \nabla \mathbf{h}(x) + \boldsymbol{\mu}^t \nabla \mathbf{g}(x) \tag{7}$$

Note that the complete Lagrangian contains terms for the equality constraints, while the reduced Lagrangian used in SYMFUNE is evaluated using those constraints and hence does not contain them explicitly.

If any of the Lagrange multipliers for a specific case is unconditionally negative then that case is never optimal (i.e. condition (6) cannot be satisfied). If any of the multipliers is unconditionally positive (regardless of the value of the parameters) then no new restriction is added for the domain of optimality. Otherwise, setting each multiplier greater than zero provides the additional restrictions on the domain of optimality. This domain is defined in full by the feasibility conditions: that the inactive inequality constraints be satisfied by the completed solution; and the optimality conditions: that the multipliers be non-negative.

The next step is to present these conditions in a form that will best enhance decision making in the design process. With the possible exception of boundaries and problems with infinite solutions, the KKT conditions provide mutually exclusive domains of optimality which can be used to construct a parametric design flow chart or equivalent. At its present stage of development, SYMFUNE summarizes this information in a parallel parametric chart within the degrees of freedom hierarchy established by SYMON. SYMFUNE output for the multiobjective hydraulic cylinder design example is provided in the next section.

3 APPLICATION TO MULTIOBJECTIVE HYDRAULIC CYLINDER DESIGN

SYMON and SYMFUNE will now be demonstrated on a hydraulic cylinder multiobjective design problem[30], where the goal is to design a hydraulic cylinder with minimum cross-sectional area of the wall and maximum lifetime, here expressed as minimum tangential stress. The SYMON-SYMFUNE solution to this design problem demonstrates the strength of symbolic computation in producing parametric design charts which can in turn be used to reduce the numerical complexity of creating Pareto-optimal plots of the conflicting objectives. The hydraulic cylinder to be

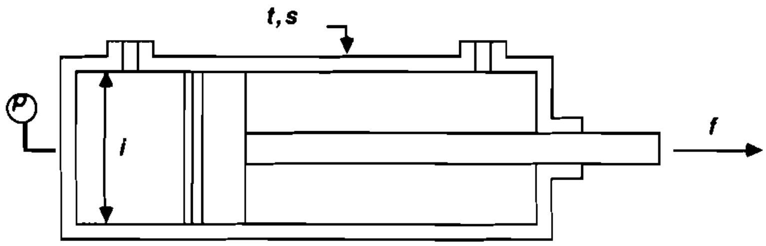

Figure 3 Hydraulic cylinder.

designed is a thin-walled pressure vessel (Figure 3) with inside diameter i and thickness t subject to a tangential stress s and using a fluid under pressure p to push a piston with output force f.

In addition to positivity constraints the design is subject to four engineering constraints. Because the cylinder is being used to support some load, the output force must be greater than or equal to the minimum load F_{min}. Due to manufacturing limitations, the cylinder thickness cannot be smaller than some minimum thickness T_{min}. There is an upper limit on the pressure source available P_{max}, and the tangential stress is to be bounded above by some maximum elastic stress S_{max}. There are two equalities relating force and stress to pressure, inside diameter and thickness. The formal qualitative statement of the problem is as follows, where the superscript $^+$ (or $^-$) means that the function is monotonically increasing (or decreasing) w.r.t. the positive variable. The multiple objective function $Z(z_1^+, z_2^+)$ is monotonically increasing with respect to each of the two single objectives: $z_1(\boldsymbol{x})$ and $z_2(\boldsymbol{x})$.

$$\underset{f,\,i,\,t,\,p,\,s}{\text{Minimize}} \quad Z[z_1(i^+, t^+)^+, z_2(s^+)^+] \tag{8}$$

$$\text{subject to} \quad g_1(f^-) \le 0 \tag{9}$$

$$g_2(t^-) \le 0 \tag{10}$$

$$g_3(p^+) \le 0 \tag{11}$$

$$g_4(s^+) \le 0 \tag{12}$$

$$h_1(f^+, i^-, p^-) = 0 \tag{13}$$

$$h_2(s^+, i^-, p^-, t^+) = 0 \tag{14}$$

$$f, p, s, i, t > 0$$

The equality constraints can be written as sets of inequality constraint doubles ($h_{i1} = h_i$ and $h_{i2} = -h_i$, $i = 1, 2$) where no more than one of each double can be active and relevant at optimality.

$$h_{11}(f^+, i^-, p^-) \le 0 \tag{15}$$

$$h_{12}(f^-, i^+, p^+) \le 0 \tag{16}$$

$$h_{21}(s^+, i^-, p^-, t^+) \le 0 \tag{17}$$

$$h_{22}(s^-, i^+, p^+, t^-) \le 0 \tag{18}$$

3.1 Symbolic Monotonicity Analysis

SYMON produces a monotonicity table for the benefit of the user (Figure 4) that reflects the monotonicities shown in Eqs (8)-(14). The sign in each column represents the monotonicity of each variable with respect to the objective or constraints associated with each row. Because the direction of the equalities is not known before monotonicity analysis is performed, SYMON initially places question marks in the columns associated with the equalities.

vars	i	t	f	p	s
Obj	+	+			+
1			-		
2		-			
3				+	
4					+
5	?		?	?	
6	?	?		?	?

Figure 4 Monotonicity table.

When monotonicity analysis is performed by SYMON, the output is a list of seven combinations of inactive constraints (Figure 5): one 2 d.o.f. case (CASE PST), three 1 d.o.f. cases (CASES PT, PS, ST), and three 0 d.o.f. cases (CASES P, T, and S). (Here the letters of each case represent the inactive constraints.) The force constraint (9) is set unconditionally active, because no bounded solution can be found if the constraint is assumed inactive. In order to satisfy rule two, inequalities (15) and (18) are also set unconditionally active and the associated doubles (16) and (17) are eliminated from the problem for all cases. A diagram showing the hierarchy of these cases is shown in Figure 6. The 0 d.o.f. cases represent subsets of the 1 d.o.f. cases up to the limits imposed by the maximal activity principle. Monotonicity analysis and the maximal activity principle have reduced the number of possible cases from 64 (or 2^6) to 7.

Because the problem is structurally different if the problem is optimized for each objective independently, monotonicity analysis must also be performed on each single objective in order to find the extreme points (or asymptotic limits) of the Pareto-Optimal curve. When considering the cross-sectional area objective $z_1(i^+, t^+)$, SYMON finds only four cases: one 1 d.o.f. case (CASE PT) and three 0 d.o.f. cases (CASES P, T, and S). SYMON finds the single objective problem unbounded when the stress objective $z_2(s^+)$ is considered and thus there are no potential optimal solution candidates for this case. Let us explore this latter case in more detail to illustrate the reasoning behind SYMON's analysis of the problem. According to the monotonic redundancy principle, constraint (17) is considered inactive because of its positive monotonicity w.r.t. s. In applying rule one, constraint (18) is set unconditionally active (verifying the previous assumption that the double (17) is inactive). Rule two is applied to verify that all variables that are not in the objective but in active constraint (18) are sufficiently constrained so that they are bounded from above and below. If SYMON sets the inequality constraint (15) active and the double (16) inactive in order to bound i and p from below, the force constraint (9) must be set active in order to bound f from below. The thickness t is bounded from below by

ANALYSIS SUMMARY :

Table of combinations of active constraints which yield results:

case	inactive constraints	active constraints	d.o.f
1	[4, 2, 3]	[1, 5, 6]	2

Table of other combinations of active constraints:

cases with 1 or more degrees of freedom

case	inactive constraints	active constraints	d.o.f
2	[2, 3]	[1, 4, 5, 6]	1
3	[3, 4]	[1, 2, 5, 6]	1
4	[2, 4]	[1, 3, 5, 6]	1
5	[3]	[1, 2, 4, 5, 6]	0
6	[2]	[1, 3, 4, 5, 6]	0
7	[4]	[1, 2, 3, 5, 6]	0

Figure 5 SYMON output for multiobjective hydraulic cylinder problem.

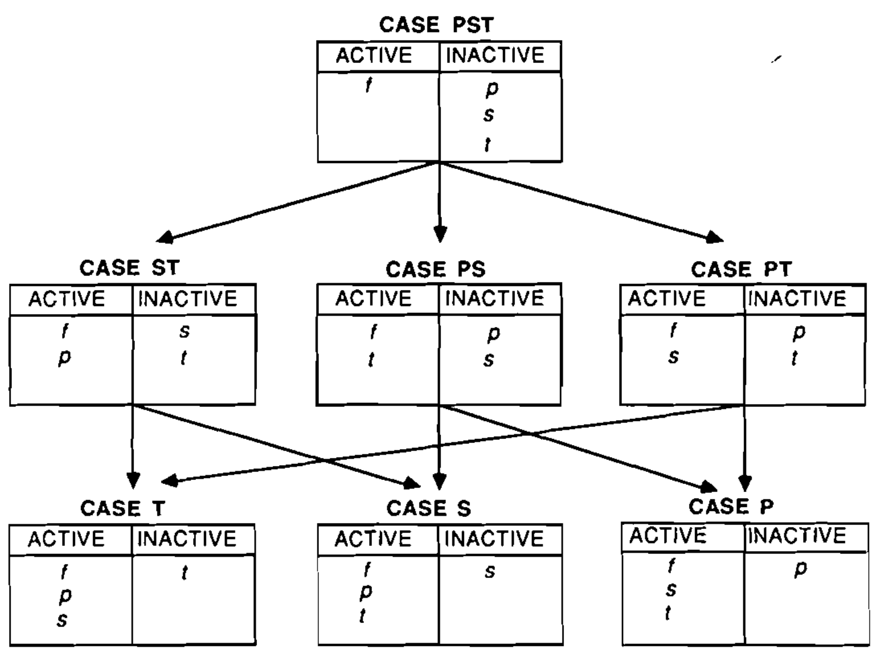

Figure 6 Seven possible cases for the multiobjective hydraulic cylinder problem.

active constraint (18) but no constraint in the system can bound t from above. Thus optimization for the stress single objective case would lead to an infinite value for the thickness t.

3.2 *Symbolic Functional Evaluation*

The formal statement of the multiobjective hydraulic cylinder problem with functional information is given below. A thin-walled cylinder assumption is used in the tangential stress equation in constraint (25). The conflicting objectives are multiplied by a weighting factor and the sum minimized. Convexity guarantees that any feasible point satisfying the Karush-Kuhn-Tucker conditions is also a global minimum and thus the set of all optimal points obtained by parametrically varying the weighting factors on the objectives is guaranteed to be part of the Pareto-optimal set.

$$\underset{f,i,t,p,s}{\text{Minimize}} \quad Z = \alpha_1(it + t^2) + \alpha_2 s \tag{19}$$

$$\text{subject to} \quad g_1 = F_{\min} - f \leq 0 \tag{20}$$

$$g_2 = T_{\min} - t \leq 0 \tag{21}$$

$$g_3 = p - P_{\max} \leq 0 \tag{22}$$

$$g_4 = s - S_{\max} \leq 0 \tag{23}$$

$$h_1 = f - \pi i^2 p/4 = 0 \tag{24}$$

$$h_2 = s - ip/2t = 0 \tag{25}$$

$$f, p, s, i, t > 0$$

where α_1 and α_2 are weighting factors on the two conflicting objectives ($\alpha_1 + \alpha_2 = 1$).

Because SYMON finds g_1 to be unconditionally active for all cases, SYMFUNE sets $f = F_{\min}$ for all possible cases. This equality, along with the two relevant system equations h_1 and h_2, is substituted into all other expressions. The reduced Lagrangean, then excluding the positivity constraints is:

$$L = \alpha_1(it + t^2) + \alpha_2\left\{\frac{2F_{\min}}{\pi it}\right\} + \mu_2(T_{\min} - t) + \mu_3\left\{\frac{4F_{\min}}{\pi i^2} - P_{\max}\right\} + \mu_4\left\{\frac{2F_{\min}}{\pi it} - S_{\max}\right\} \tag{26}$$

The boundedness test (back-substitution into the objective and applying first-order conditions) in SYMFUNE quickly reveals that CASES PST and PT are unbounded, revealing hidden monotonicities. Five cases can now be considered for feasibility and optimality, and the Karush-Kuhn-Tucker test finds that CASES ST and T have multipliers unconditionally negative, so these cases are never optimal. This implies that minimization of cross-sectional area at a functional level specifies that thickness should always be set to its lower limit, a result undetected by SYMON at the qualitative level. SYMFUNE's output for CASE T is given in Figure 7 (note that a1 is α_1 and a2 is α_2). Three final solutions are output from SYMFUNE, each with an accompanying set of inequalities which define the domain of optimality and the values of the Lagrange multipliers in parametric form. See Figure 8 for sample output of SYMFUNE for the optimal candidate CASE PS and Figure 9 for a parametric chart summarizing all of SYMFUNE's solutions. SYMFUNE also detects that there are

CASE T:

$$mu_4 = - \frac{\pi\, a2\, Smax^3 - 2\, a1\, Fmin\, Smax - 2\, a1\, Fmin\, Pmax}{\pi\, Smax^3}$$

$$mu_3 = - \frac{a1\, Fmin}{\pi\, Smax^2}$$

The optimality conditions are:

$$mu_4 = - \frac{\pi\, a2\, Smax^3 - 2\, a1\, Fmin\, Smax - 2\, a1\, Fmin\, Pmax}{\pi\, Smax^3} >= 0$$

$$mu_3 = - \frac{a1\, Fmin}{\pi\, Smax^2} >= 0$$

NOTE: mu_3 is always negative.

This case is NEVER OPTIMAL.

Figure 7 SYMFUNE output for Case T.

only two optimal solution candidates for the cross-sectional area single objective problem. Recall that SYMON found no bounded solution for the stress single objective problem, and thus SYMFUNE does not consider this case.

The SYMON-SYMFUNE procedure has solved a nonlinear optimal design problem, initially with three degrees of freedom and sixty-four potential cases (2^6 combinations of active constraints). This entirely automated procedure has reduced the solution to a set of three parametric cases presented symbolically in functional form. Lagrange multipliers are also supplied in nonlinear functional form for further use in sensitivity analysis. The corresponding solution, derived numerically, would have only been valid for one specific set of parameters. The symbolic solution, on the other hand, allows one to solve a whole class of problems in terms of unspecific parameters. The parametric form has proven extremely useful in reducing the numerical complexity in generating Pareto-optimal plots for an extended version of the multiobjective hydraulic cylinder design problem in Michelena and Agogino[31].

4 INTEGRATION WITH NUMERICAL PROGRAMS

Because each numerical optimization code or numerical analysis program has its own input format requirements, it is difficult to develop a general preprocessor, like the SYMON-SYMFUNE system, with output in the specified format. One advantage of writing in an AI language like FranzLISP is the ease with which it operates on text

CASE PS :

$mu_2 = 2\ a1\ Tmin$

The optimality conditions are:

Setting $mu_2 >= 0$ introduces no new constraints.

The feasibility conditions are:

$$Pmax > \frac{2\ a1\ Tmin^2}{a2}$$

$$Smax > \frac{sqrt(2)\ sqrt(a1)\ sqrt(Fmin)}{sqrt(\pi)\ sqrt(a2)}$$

with solutions:

$$s = \frac{sqrt(2)\ sqrt(a1)\ sqrt(Fmin)}{sqrt(\pi)\ sqrt(a2)}$$

$$p = \frac{2\ a1\ Tmin^2}{a2}$$

$$i = \frac{2\ sqrt(a2)\ sqrt(Fmin)}{sqrt(2)\ sqrt(\pi)\ sqrt(a1)\ Tmin}$$

f = Fmin
t = Tmin

This case is BOUNDED.

Figure 8 SYMFUNE output for Case PS.

data, making such a direct translation possible. For example, the FORTRANIZER program in VAXIMA can rewrite the mathematical formulas produced in VAXIMA (and thus SYMFUNE) into standard FORTRAN code. Under the Unix™ operating system, it would then be possible to write shell programs utilizing the pipe features of Unix™ to integrate a SYMON-SYMFUNE type of preprocessor to a numerical program written in FORTRAN such that the integration details are transparent to the user. Of course, if the output from SYMON-SYMFUNE is entirely numerical (e.g., a list of the constraints that are active at optimality for cases that are bounded and feasible), then such a translator is not necessary and a shared data file is sufficient to perform the integration.

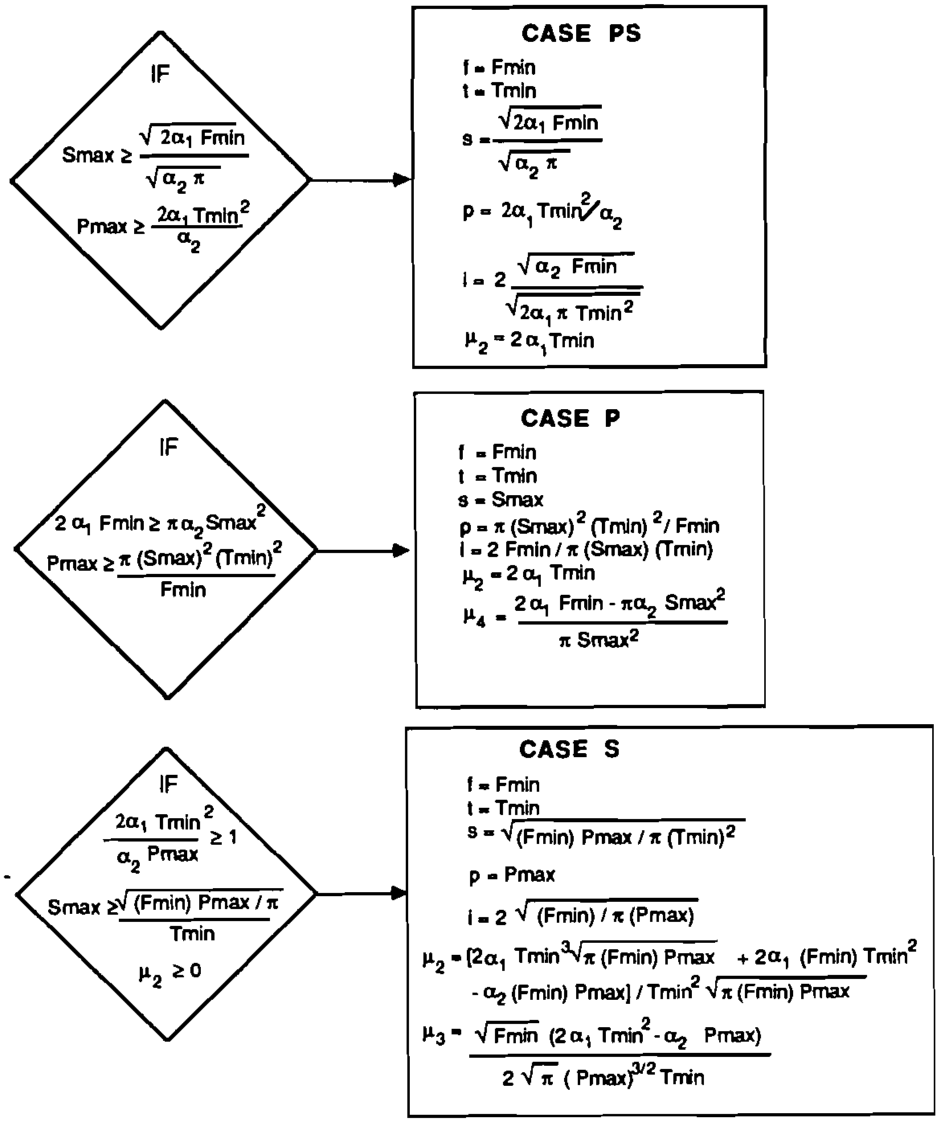

Figure 9 Parametric solution to multiobjective hydraulic cylinder problem.

5 INTEGRATION WITH KNOWLEDGE-BASED SYSTEMS

Although SYMON-SYMFUNE is a domain-independent system, there are a number of ways it could be integrated with a domain-specific knowledge-based system. The rules of monotonicity analysis provide a rigorous calculus for logically analyzing constraint activity relative to design goals. Yet the SYMON-SYMFUNE implementation has been shown to parallel the reasoning processes used by human designers[25]. For example, the single objective hydraulic cylinder design problem has been tested on numerous experts in both industry and academia with the following specializations: (1) engineering design with expertise in pressure vessel design and (2) mathematics and optimization theory. All tests were performed in periods ranging from 20 to 120 minutes (with an average of around 30 minutes). The goal was to design a hydraulic cylinder with minimum cross-sectional area subject to constraints on thickness, force, pressure, and stress. Although the experts were unfamiliar with the theory of monotonicity analysis, they approached the problem in much the same way as SYMON. Most were able to determine that the force constraint was unconditionally active (design for the minimum required force). However, not all of the experts were able to identify all of the feasible and bounded parametric solutions and none were able to determine (as did SYMFUNE) that the optimal cylinder should be designed for the minimum allowable level of thickness. A domain-specific application of SYMON-SYMFUNE might be useful in strengthening the knowledge base of an expert system by adding a "deeper-level" of understanding of the physical problem.

6 DISCUSSION

SYMON and SYMFUNE have both been written as domain independent programs to perform qualitative reasoning and symbolic computation on a wide class of optimal design problems. They have been tested successfully on numerous problems published in the literature, including those described in Refs. [30–41].

In its present form the main limitation of SYMON is that it works best when the objective function and constraints are monotonic either globally or over a predefined regional domain. If the monotonicities are ambiguous, SYMON will continue the analysis to the maximum extent allowed by the theory. Current research in qualitative reasoning with nonmonotonic functions is taking two directions; both of which are based on dividing the feasible domain into monotonic subregions: (1) using variable transformations[41] and (2) using higher order derivatives. It is expected that the use of higher order derivatives will provide a continuum between qualitative and functional levels of knowledge representation.

SYMFUNE is further limited by the inability of VAXIMA to solve large systems of nonlinear equations (a limitation it shares with the best of human mathematicians). Further generality in SYMFUNE relies on improvements in the field of symbolic computation.

7 CONCLUSIONS

The results of the qualitative and functional levels of computation in the SYMON-SYMFUNE programs give strong insights into the mathematical structure of optimization problems and can eliminate the need for further numerical analysis. The symbolic results provide useful assistance at the numerical level in generating parametric curves for applications in multiobjective optimization, sensitivity analysis, and parametric, probabilistic or fuzzy design. These powerful techniques for integrating qualitative reasoning and symbolic computation in SYMON and SYMFUNE typically outperform human reasoning in reducing the complexity of optimal design problems.

ACKNOWLEDGEMENTS

This research was funded in part by the National Science PYI Program Grant DMC-8451622 with matching grants from the IBM Corporation and the General Electric Company. The authors wish to thank Manuel Bronstein, Johnston K. Choy, and Nestor Michelena for their valuable contributions to the research program described herein.

REFERENCES

1. McDermott, J. (1980) R1: An expert in the computer systems domain. *AAAI*, 269-271.
2. Dixon, J. R. and Simmons, M. K. (1983) Computers that design: expert systems for mechanical engineers, *Computers in Mechanical Engineering*, ASME, New York, 10-18.
3. Maher, M. L., Sriram, D. and Fenves, S. J. (1984) Tools and techniques for knowledge based expert systems for engineering design. *Adv. Eng. Software*, **6**(4), 178-188.
4. Mittal, S. M., Dym, C. L. and Morjaria, M. (1985) PRIDE: An expert system for the design of paper handling systems. *Computer*, IEEE Computer Society, 102-114.
5. Gero, J. S. and Balachandran, M. (1986) A comparison of procedural and declarative programming languages for the computation of pareto optimal solutions. *Engineering Optimization*, **9**(2), 131-142.
6. Stefik, M. and Bobrow, D. G. Object-oriented programming: themes and variations. *AI Magazine*, **6** (4), 40-62.
7. Forbus, K. D. (1984) Qualitative process theory, *Artificial Intelligence*, **24** (Also in Bobrow, 85-168, (1985).)
8. Bobrow, D. G. (1985) *Qualitative Reasoning about Physical Systems*, Bradford Press, Cambridge, MA.
9. Li, H. L. and Papalambros, P. (1985) A production system for use of global optimization knowledge. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **107**, 277-284.
10. Arora, J. S. and Baenziger, G. (1986) Uses of artificial intelligence in design optimization. *Computer Methods in Applied Mechanics and Engineering*, **54**, 303-323.
11. Dubowsky, S. and Grant, J. L. (1975) Application of symbolic manipulation to time domain analysis on nonlinear dynamic systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 60-68.
12. Rand, R. H. (1984) *Computer Algebra in Applied Mathematics: An Introduction to MACSYMA*, Pitman Publishing Inc., Marshfield, MA.
13. Month, L. A. and Rand, R. H. (1985) Stability of a rigid body with an oscillating particle: an application of MACSYMA. *ASME Journal of Applied Mechanics.*
14. Leu M. C. and Hemati, N. (1986) Automated symbolic derivation of dynamic equations of motion for robotic manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control*, **108**, 172-179.
15. Murtagh, B. A. and Saunders, M. A. (1980) *MINOS-Augmented User's Manual.* Report SOL 80-14, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University.
16. Lasdon, L. S., Warren, A. D., Jain, A., and Ratner, M. (1978) Design and testing of a GRG code for nonlinear optimization. *ACM Trans Math. Software*, **4**, 34-50.
17. Swanson Analysis Systems, Inc., *ANSYS*, Houston, PA, 15342.
18. MacNeal-Schwendler Corp. (1985) *MSC/NASTRAN User's Manual.* Los Angeles, CA.
19. Auslander, D. (1982) Parasol II: A laboratory simulation and control tool, *Computers in Mechanical Engineering*, 25-32.
20. Radford, A. D. and Gero, J. S. (1986) Multicriteria optimization in architectural design. In *Design Optimization*, J. S. Gero, ed., Academic Press, Inc., New York, NY.
21. Papalambros P. and Wilde, D. J. (1979) Global non-iterative design optimization using monotonicity analysis. *ASME Journal of Mechanical Design*, **101**(3), 645-649.
22. Wilde, D. J. (1986) A maximal activity principle for eliminating overconstrained optimization cases. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **108**, 312-314.
23. Johnson R. C. (1978) *Mechanical Design Synthesis.* 2nd Ed., Van Nostrand, New York, NY.
24. Kuhn, H. W. (1976) Nonlinear programming: a historical view. In R. Cottle and C. Lemke (eds.), *SIAM-AMS Proceedings*, 1-26.
25. Choy J. K. and Agogino, A. M. (1986) SYMON: automated SYmbolic MONotonicity analysis system for qualitative design optimization in *ASME Proceedings of the 1986 International Computers in Engineering Conference*, Chicago, IL, **1**, 207-212.
26. Fateman, R. J. (1982) *MACSYMA: Primer for VAX/UNIX 6.3.1.* Academic Computer Services Library, University of California, Berkeley, CA.
27. Wilensky, R. (1984) *LISPcraft*, W. W. Norton & Company, New York, NY.

28. Karush, W. (1939) Minima of functions of several variables wth inequalities as side conditions. *Master's Thesis*, Dept. of Mathematics, University of Chicago, IL. (referenced by Kuhn, 1976).
29. Kuhn, H. W. and Tucker, A. W. (1950) Nonlinear programming, In J. Neyman (ed.), *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, CA, 481-492.
30. Wilde, D. J. (1975) Monotonicity and dominance in optimal hydraulic cylinder design, *ASME Journal of Engineering for Industry*, **97**, 1310-1314.
31. Michelena N. and Agogino A. M. (1986) *Multiobjective hydraulic cylinder design. ASME Journal of Mechanisms, Transmissions, and Automation in Design*, in press,
32. Siddall, J. N. (1972) *Analytical Decision-Making in Engineering Design*. Prentice-Hall, New York, NY, 107-113.
33. Wilde, D. J. (1978) *Globally Optimal Design*, Wiley-Interscience, New York, NY.
34. Haug, E. J. and Arora, J. S. (1979) *Applied Optimal Design*, Wiley-Interscience, New York, NY.
35. Reklaitis, G. V., Ravindran, A. and Ragsdell, K. M. (1983) *Engineering Optimization*, Wiley, New York, NY.
36. Papalambros, P. and Li, H. L. (1983) Notes on the operational utility of monotonicity in optimization. *ASME Journal of Mechanisms, Transmissions and Automation in Design*, **105**, (2), 174-181.
37. Himmelblau, D. M. (1972) A uniform evaluation of unconstrained optimization techniques. *Proceedings of the Conference on Numerical Methods for Nonlinear Optimization at the University of Dundee, Scotland, 1971*, Academic Press, London, England, 91-113.
38. Beightler, C. S., Phillips, D. T. and Wilde, D. J. (1979) *Foundations of Optimization*, 2nd ed., Prentice-Hall, New York.
39. Gill, P. E., Murray, W. and Wright, M. H. (1981) *Practical Optimization*. Academic Press, San Francisco, CA.
40. Ragsdell K. M. and Phillips D. T. (1983) Optimal design of a class of welded structures using geometric programming. *ASME Journal of Mechanisms, Transmissions and Automation in Design*, **105** (2), 174-181.
41. Agogino, A. M. (1985) *Monotonicity analysis in unconstrained optimization* Working Paper 85-0701, Expert Systems Laboratory, Dept. of Mechanical Engineering, University of California, Berkeley, CA.