COMPUTATIONAL IMPLEMENTATION AND TESTS OF A SEQUENTIAL LINEARIZATION ALGORITHM FOR MIXED-DISCRETE NONLINEAR DESIGN OPTIMIZATION

by

Han Tong Loh

and

Panos Y. Papalambros

Technical Report UM-MEAM-89-09

Abstract

A previous article gave the theoretical background and motivation for a new sequential linearization approach to the solution of mixed-discrete nonlinear optimal design problems. The present sequel article gives the implementation details of program MINSLIP based on this approach. Illustrative examples, modeling issues, and program parameter selection are discussed. A report on extensive computational results with test problems, as well as comparisons with other methods, shows advantages in both robustness and efficiency. Sample design applications are included.

June 1989

DESIGN LABORATORY

THE UNIVERSITY OF MICHIGAN

ANN ARBOR

Introduction

In the present article we address numerical solution of *Mixed-Discrete Nonlinear Programming* (MDNLP) problems expressed in the mathematical model

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{g}_j(\mathbf{x}) \le 0 \qquad j = 1,\ldots, m \\ & && \mathbf{l}_i \le \mathbf{x}_i \le \mathbf{u}_i \qquad i = 1,\ldots, n \\ & && \mathbf{x} \in \mathcal{X} \subset \mathcal{D}^d \times \mathcal{R}^{(n-d)} \\ & && f : \mathcal{X} \longrightarrow \mathcal{R},\ \mathbf{g}_j : \mathcal{X} \longrightarrow \mathcal{R} \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a design point in a n-dimensional space and consists of d discrete variables and (n-d) continuous variables. $\mathcal{D}$ denotes a discrete set for each of the discrete variables and $\mathcal{R}$ is the real continuous space.

In a previous article (Loh & Papalambros, 1989), it was shown that for convex problems, if a sequence of linear approximation problems generates solutions that converge, then the final solution is the global minimizer of the nonlinear problem (Algorithm SLP1). The ideas of decreasing step bounds and ε-feasibility were introduced to overcome some convergence difficulties (Algorithm SLP2). Imposition of decreasing step bounds aims at reaching minimizers that are interior solutions, while reduction of the ε-feasibility aims at forcing the linear approximation algorithm to accept points which are MDNLP-feasible. As the material described in this article is based on the previously cited one and to avoid lengthy repetitions, familiarity with the cited reference (including several definitions) is assumed in the exposition of the following sections.

It is convenient to partition the variables into two sets, discrete and continuous, i.e. $\mathbf{x} = \{\mathbf{y}_D, \mathbf{u}_C\}$. Thus, $\mathbf{x}$ is used to designate the entire vector of variables, $\mathbf{y}_D$ to designate the discrete components and $\mathbf{u}_C$ to designate the continuous components, and the following definition is stated. Given a MDNLP problem of the form defined by Eq.(1), the *Continuous SubProblem* (CSP) at point $\mathbf{x}$ is defined as follows:

$$\begin{aligned} &\min_{\mathbf{u}_C} && f(\mathbf{y}_D, \mathbf{u}_C) && \text{for a fixed } \mathbf{y}_D \\ &\text{subject to} && \mathbf{g}(\mathbf{y}_D, \mathbf{u}_C) \le \mathbf{0} \\ & && \mathbf{u}_C \in \mathcal{R}^{n-d} \end{aligned} \tag{2}$$

The enhanced algorithm SLP2 has been implemented in a program called MINSLIP written in standard FORTRAN 77 for the Apollo Domain operating system. The algorithm can be viewed as a multistage two-phase hybrid of mixed-discrete sequential linear programming (SLP) and a continuous nonlinear programming (NLP) algorithm. The mixed-discrete linear programming (MDLP) method solves a linear approximation of the original problem. The NLP method solves the continuous subproblem, Eq.(2). In the

initial implementation of MINSLIP, the generalized reduced gradient (GRG) method was chosen for the NLP solution, mainly due to the authors' experience in using GRG successfully for continuous, small, dense, nonlinear design problems. With the discrete variables fixed, GRG can reasonably quickly find the optimal values for the continuous variables, but any other NLP algorithm could also be used. For the MDLP problem, a version of the simplex method with branch and bound was used as a variant of Dakin's method (Dakin, 1965), again for reasons of convenience and robustness. If the problem is purely discrete, the algorithm becomes a multistage two-phase SLP algorithm.

The term 'multistage' derives from the fact that the reduction of ε-feasibility to remove the MDLP-feasible points that are MDNLP-infeasible may occur in several stages. In each stage, there are two phases. In the first phase, the algorithm moves the iterate from an ε-infeasible domain to an ε-feasible domain without regard to the objective function, and in the second phase, it strictly improves the objective without losing ε-feasibility.

TABLE OF ABBREVIATIONS

CSP	- continuous subproblem
LP	- linear programming
GRG	- reneralized reduced gradient
NLP	- nonlinear programming
MDLP	- mixed-discrete linear programming
MDNLP	- mixed-discrete nonlinear programming
MIBBGRG	- mixed integer branch and bound using GRG
MIBBSLP	- mixed integer branch and bound using SLP
MIBBSQP	- mixed integer branch and bound using SQP
QP	- quadratic programming
RMDLP	- restricted mixed-discrete linear program
SLP	- sequential linear programming
SQP	- sequential quadratic programming
SUMINF	- sum of infeasibilities

The next section describes the algorithm in MINSLIP step-by-step, followed by a section with exampes of small dimension to illustrate the algorithm, and a section on modeling and parameter selection. A subsequent section gives results on computational performance with many test problems, taken from Gupta (1980), Cha (1987), Sandgren (1988), and Duran & Grossman (1986). Three design applications with discrete variables conclude the article.

Computational Implementation

In this section the major steps of the algorithm in MINSLIP are given and discussed. Table 1 defines the symbols used in the description below.

Table1. Nomenclature of symbols in Algorithm MINSLIP

δ	-	parameter value at which convergence is considered achieved (>0)
ε_0	-	initial allowable ε-feasibility (>0)
ε_f	-	final allowable ε-feasibility (>0)
$\mathbf{dx}$	-	difference between the current iterate and the incumbent
f	-	objective function value
r_ε	-	rate of reducing ε-feasibility (>1)
r_t	-	rate of reducing step bounds (>1)
$\mathbf{t_0}$	-	initial step bounds
$\mathbf{t}$	-	current step bounds (>0)
$\mathbf{u_C}$	-	continuous variables in the continuous subproblem
$\mathbf{x_0}$	-	initial starting vector
$\mathbf{x_b}$	-	incumbent vector
$\mathbf{x_k}$	-	current iterate

Note that, given the MDNLP model, Eq.(1), the restricted mixed-discrete linearized program (RMDLP) is (Loh & Papalambros, op.cit.)

$$\begin{aligned} &\text{minimize} && \nabla f(\mathbf{x_0})^T\mathbf{x} \\ &\text{subject to} && \nabla \mathbf{g}(\mathbf{x_0})^T(\mathbf{x}-\mathbf{x_0}) + \mathbf{g}(\mathbf{x_0}) \le 0 \\ & && -\mathbf{t_0} \le \mathbf{x}-\mathbf{x_0} \le \mathbf{t_0} \\ & && \mathbf{x} \in \mathcal{X} \end{aligned} \tag{3}$$

and the sum of infeasibilities at a point x is given by

$$\text{SUMINF}(\mathbf{x_1}) = \Sigma g_j(x_1),\ g_j > 0. \tag{4}$$

Algorithm MINSLIP

1. Given δ, ε_0, ε_f, $\mathbf{t_0}$, r_t, r_ε and initial $\mathbf{x_0}$, set $\varepsilon = \varepsilon_0$, $\mathbf{t} = \mathbf{t_0}$.
2. If $\mathbf{x_0}$ is discrete, set $\mathbf{x_b} = \mathbf{x_0}$; else, set $\mathbf{x_b} = \text{round}(\mathbf{x_0})$.
3. If $\text{SUMINF}(\mathbf{x_b}) > \varepsilon_0$, set phase = 1; else, set phase = 2.
4. Construct RMDLP($\mathbf{x_b}$).
5. Solve RMDLP($\mathbf{x_b}$) to get $\mathbf{x_k}$.
   (Use the simplex method with branch and bound).
6. If problem is mixed discrete, with discrete values of $\mathbf{y_D}$ fixed by
   Step 5, solve CSP by GRG to get better $\mathbf{u_C}$.
7. Set $\mathbf{dx} = \mathbf{x_k} - \mathbf{x_b}$. If $(\mathbf{dx})_i < \delta$, stop.
8. If phase = 2, go to step 10.

9. If SUMINF($\mathbf{x_k}$) < SUMINF($\mathbf{x_b}$), set $\mathbf{x_b} = \mathbf{x_k}$, go to step 3;
   else, go to step 13.
10. If SUMINF($\mathbf{x_k}$) > ε, go to Step 13.
11. If $f(\mathbf{x_k}) \geq f(\mathbf{x_b})$, go to Step 13; else, set $\mathbf{x_b} = \mathbf{x_k}$, go to Step 12.
12. Set $\varepsilon = \max(\text{SUMINF}(\mathbf{x_k})/r_\varepsilon, \varepsilon_f)$. Set phase = 1, go to Step 3.
13. Set $\mathbf{t} = \mathbf{t}/r_t$. If $|\mathbf{t}| < \delta$, stop with error message; else, go to Step 4.

Step 1. This is the initialization step with parameters as defined in Table 1. Note that δ is usually set to a value smaller than the interval of the smallest discretization for purely discrete problems or to a small value (typically 0.001) for mixed- discrete values. We shall describe selection of some program parameters in the next section. For the starting point, $\mathbf{x_0}$, the minimizer to the problem with the discreteness requirement relaxed provides a good choice after rounding to a discrete point (usually the nearest one). The linearization approximates a nonlinear constraint more closely when it is near the constraint than when it is farther away. Hence, linearization at the rounded minimizer of the relaxed problem will often yield a good starting subproblem.

Step 2. If $\mathbf{x_0}$ is not discrete, round the necessary components to the nearest discrete values.

Step 3. Check if this initial point is ε-feasible. If it is ε-feasible, phase is set at 2, otherwise phase is set at 1.

Step 4. Construct the restricted mixed-discrete linear problem.

Step 5. Solve the restricted mixed-discrete linear problem using Dakin's branch and bound rules and a LP solver. The specific algorithm for this step is well-known and is omitted here. (See [Loh, 1989] for a description.) It is only noted that code ZX3LP from the IMSL Library (1982) is the LP solver used. ZX3LP uses the revised simplex method and was chosen for its robustness - although some efficiency would be gained by using a dual simplex method.

Step 6. From the previous step, a candidate iterate was obtained. If the problem is mixed-discrete, the discrete variables may be at their optimal values but the continuous ones are probably not. The discrete values are set as parameters for the CSP, which is then solved by a GRG algorithm. The specific implementation here is the robust code GRG2 (Lasdon,

1980). If the problem is purely discrete, this step is skipped. If GRG2 cannot find a feasible solution for the CSP, the values for the continuous variables in Step 5 are returned.

Step 7. If the iterate produced from Steps 5 and 6 is within a distance δ from the incumbent $\mathbf{x_b}$, the algorithm has converged. If ε-feasibility of this point is less than or equal to ε_f, terminate with this value of the incumbent, else terminate with an error message.

Steps 8-11. A new point $\mathbf{x_k}$ is accepted as better than $\mathbf{x_b}$ under one of two conditions:

1) if phase is 1, SUMINF($\mathbf{x_k}$) < SUMINF($\mathbf{x_b}$);

or 2) if phase is 2, $\mathbf{x_k}$ is ε-feasible and $f(\mathbf{x_k}) < f(\mathbf{x_b})$.

In phase 1, we are interested in moving the iterate from an ε-infeasible domain towards an ε-feasible domain without regard to the objective function; in phase 2, we accept an iterate only if it strictly improves the objective without sacrificing ε-feasibility.

Step 12. If the point generated improves a phase 2 incumbent, then ε-feasibility can be reduced unless it is already at ε_f. We want to reduce ε-feasibility to a value such that this point will be excluded from further consideration. The phase is therefore reset to 1.

Step 13. If the iterate generated is not accepted, the step bounds are reduced. Eventually one of two things must happen: either a particular linearization will produce an optimizer that is the same point at which the linearization is made, or the algorithm cannot converge as we decrease the step bounds to a value lower than the parameter for convergence, δ, and so it terminates with a error message. In this case, some of the parameters can be reset and the algorithm restarted.

Some comments are now in order.

In some SLP strategies the successive linearizations are added to the preceding ones. In convex problems, accumulating each successive linearization generates an envelope confining the nonlinear feasible domain. Such additions are used, for example, in generalized Bender's decomposition (Geoffrion, 1972) and in outer-approximation algorithms (Duran & Grossmann, 1986). MINSLIP does not accumulate linear cuts for each iteration. In nonconvex problems, the linearization cuts off part of the nonlinear feasible domain. Accumulating all preceding linearizations cuts off an increasingly larger part of the feasible region, and the solution may be inadvertedly excluded. The algorithm will often terminate quickly with suboptimal answers for nonconvex problems. In MINSLIP, instead of accumulating cuts a new restricted mixed-discrete linear program is

generated at each iteration, so parts of the feasible domain that may have been cut off by a previous linearization can be reconsidered. This provides an opportunity for finding a better minimum in nonconvex problems. The acceptance criteria above maintain the algorithm's good performance on convex problems.

Another comment is that, when a new restricted mixed discrete linear program is created, the step bounds are restored to their initial values. This strategy works always, but leads to more iterations. Empirically, it has been found that it is only necessary to restore the step bounds until they have decreased below four discrete units. Restoring the step bounds to four discrete units at that time is sufficient, and the number of iterations is substantially reduced.

Illustrative Examples

In this section, three two-dimensional examples demonstrate the algorithm graphically. In the first two the algorithm converges to the global discrete minimizer for a convex and nonconvex problem respectively, and in the third it fails for a nonconvex problem.

Example 1. Consider the following convex problem (Figure 1a)

$$\begin{aligned} &\text{minimize} && f = x_1^2 - 8x_2 && x_1 \text{ integer} \\ &\text{subject to} && g_1: -8.63x_1 + x_2^3 \le 0 && (5) \\ & && g_2: x_1 - 5 \le 0 \\ & && g_3: x_2 - 5 \le 0 \\ & && g_4: 0.5 - x_1 \le 0 \end{aligned}$$

Linearize initially at $(4, 5)^T$, where f = -7 and SUMINF = 20.85 (infeasible). Set $t_{0i} = 4$, $\varepsilon_0 = 1$, $\varepsilon_f = 0.01$, and $r_t = 2$. The linearization gives the *restricted mixed-integer linear program* (RMILP)

$$\begin{aligned} &\text{minimize} && 10x_1 - 8x_2 && x_1 \text{ integer} \\ &\text{subject to} && g_1: -8.63x_1 + 48x_2 \le 128 \\ & && g_2: x_1 - 5 \le 0 \\ & && g_3: x_2 - 5 \le 0 \\ & && g_4: 0.5 - x_1 \le 0 \\ & && -4 \le x_1 - 5 \le 4 && (6) \\ & && -4 \le x_2 - 4 \le 4 \end{aligned}$$

with solution $(1, 2.846)^T$. GRG2 is called with x_1 fixed at 1, obtaining the solution $(1, 2.051)^T$ since the continuous subproblem has one less dimension and 2.846 is near 2.051. At $(1, 2.051)^T$ the nonlinear objective value is f = -15.409, and since the point is feasible, it becomes the incumbent. Linearizing again yields the new RMILP

$$
\begin{aligned}
\text{minimize} \quad & 2x_1 - 8x_2 && x_1 \text{ integer} \\
\text{subject to} \quad & g_1: -8.63x_1 + 12.622x_2 \le 17.26 \\
& g_2: x_1 - 5 \le 0 \\
& g_3: x_2 - 5 \le 0 \\
& g_4: 0.5 - x_1 \le 0 \\
& -4 \le x_1 - 1 \le 4 \\
& -4 \le x_2 - 2.051 \le 4
\end{aligned} \tag{7}
$$

with solution $(5, 4.786)^T$. GRG2 with x_1 fixed at 5, obtains the point $(5, 3.507)^T$ with f(5, 3.507) = -3.06, worse than the incumbent. The point is rejected and the step bounds are reduced. Eventually the bounds are reduced until $(2, 2.735)^T$ is obtained for the RMILP in Eq.(7). GRG2 moves this iterate to the point $(2, 2.584)^T$, with f(2, 2.584) = -16.68 and so the incumbent is replaced. Step bounds are restored to 4. A new linearization gives

$$
\begin{aligned}
\text{minimize} \quad & 4x_1 - 8x_2 && x_1 \text{ integer} \\
\text{subject to} \quad & g_1: -8.63x_1 + 20.036x_2 \le 34.52 \\
& g_2: x_1 - 5 \le 0 \\
& g_3: x_2 - 5 \le 0 \\
& g_4: 0.5 - x_1 \le 0 \\
& -4 \le x_1 - 2 \le 4 \\
& -4 \le x_2 - 2.584 \le 4
\end{aligned} \tag{8}
$$

This problem is solved, points with objective values worse than the incumbent are rejected, and step bounds are reduced twice. Convergence occurs at $(2, 2.584)^T$. The process is illustrated in Figure 1. □

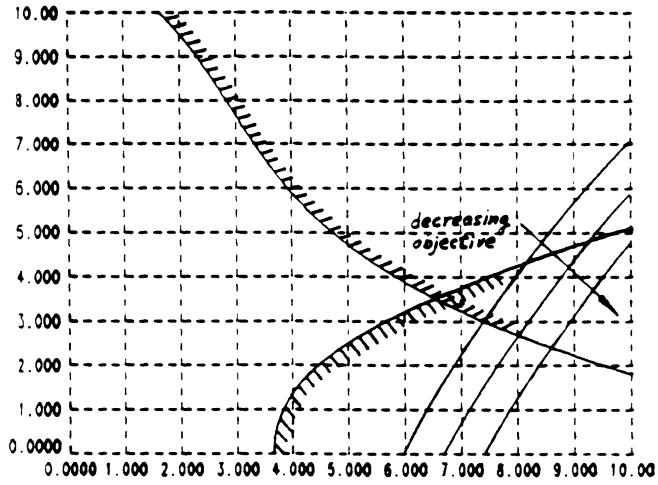
Example 2. Consider the following non-convex problem (see Figure 2a)

$$
\begin{aligned}
\text{minimize} \quad & f = -9x_1^2 + 10x_1x_2 - 50x_1 + 8x_2 + 460 && x_1, x_2 \text{ integer} \\
\text{subject to} \quad & g_1: x_1 - (0.2768x_2^2 - 0.235x_2 + 3.718) \le 0 \\
& g_2: x_1 - (-0.019x_2^3 + 0.446x_2^2 - 3.98x_2 + 15.854) \le 0
\end{aligned} \tag{9}
$$

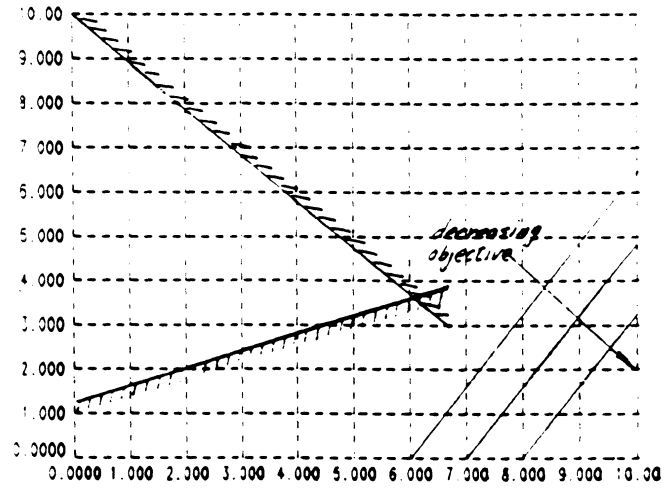
Linearize initially at $(7, 5)^T$, with $t_{0i} = 4$, SUMINF = 2.271, and f = 59. Let $\varepsilon_0 = 1$, $\varepsilon_f = 0.01$. The linearization gives

$$
\begin{aligned}
\text{minimize} \quad & -126x_1 + 78x_2 && x_1, x_2 \text{ integer} \\
\text{subject to} \quad & g_1: x_1 + 0.945x_2 \le 9.454 \\
& g_2: x_1 - 2.533x_2 \le -3.2 \\
& -4 \le x_1 - 7 \le 4 \\
& -4 \le x_2 - 5 \le 4
\end{aligned} \tag{10}
$$

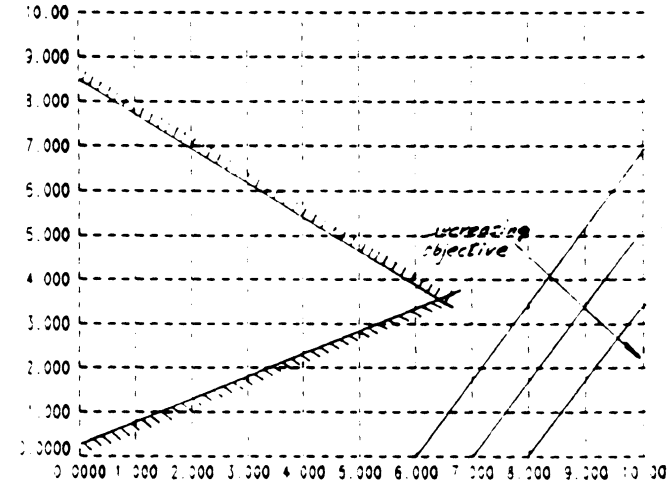
with solution $(5, 4)^T$. The value f(5, 4) = 217 is worse than the incumbent, but the point is feasible, so it is accepted and ε is set to 0.01. The new linearization gives



(a)



(b)



(c)

Figure 2. Convergence for a nonconvex problem (Example 2). (a) The mixed-discrete nonlinear problem; (b) linearization at $(7, 5)^T$ giving $(5, 4)^T$ as the next iterate; (c) linearization at $(5,4)^T$ giving $(5, 3)^T$, the discrete minimizer.

$$\begin{aligned} \text{minimize} \quad & -100x_1 + 58\,x_2 && x_1, x_2 \text{ integer} \\ \text{subject to} \quad & g_1: \; x_1 + 1.324x_2 \le 11.15 && (11) \\ & g_2: \; x_1 - 1.979x_2 \le -0.71 \\ & -4 \le x_1 - 5 \le 4 \\ & -4 \le x_2 - 4 \le 4 \end{aligned}$$

The solution $(5, 3)^T$ is feasible, and f(5,3) = 159. Being better than the incumbent, it becomes the incumbent. Linearizing at $(5, 3)^T$ gives

$$\begin{aligned} \text{minimize} \quad & -110x_1 + 58\,x_2 && x_1, x_2 \text{ integer} \\ \text{subject to} \quad & g_1: \; x_1 + 1.817x_2 \le 12.866 && (12) \\ & g_2: \; x_1 - 1.425x_2 \le 1.227 \\ & -4 \le x_1 - 5 \le 4 \\ & -4 \le x_2 - 3 \le 4 \end{aligned}$$

which produces the same point $(5, 3)^T$ and the process terminates (see Figures 2b and 2c). As seen from Figure 2a, $(5, 3)^T$ is the global discrete minimizer. Notice that if we had accumulated all linearizations in RMILP Eq.(10) and RMILP Eq.(11), this point would have been permanently cut off by RMILP Eq.(10). Accumulation of linearizations will have led to the erroneous conclusion that point $(5, 4)^T$ is the discrete minimizer. □

Example 3. Consider now another nonconvex problem (see Figure 3a)

$$\begin{aligned} \text{minimize} \quad & f = -1.5x_1 - 1.2x_2 && x_1, x_2 \text{ integer} \\ \text{subject to} \quad & g_1: \; 4.64 - (x_1 - 6)^2 - (x_2 - 2.8)^2 \le 0 && (13) \\ & g_2: \; x_2 - (0.0643x_1^2 - 0.7564x_1 + 6.7857) \le 0 \end{aligned}$$

Linearize initially at $(5, 4)^T$ with $t_{0i} = 4$. Going through the same process as for the ptrevious examples produces iterates $(3, 4)^T$ and $(3, 5)^T$, with convergence at $(3, 5)^T$ where f = -10.5. However, a better point exists. Point $(4, 4)^T$ has f = -10.8 and is the global discrete minimizer. Figure 3b illustrates why the sequential linearization did not find $(4, 4)^T$. The linearization at $(5, 3)^T$ is

$$\begin{aligned} \text{minimize} \quad & -1.5x_1 + 1.2x_2 && x_1, x_2 \text{ integer} \\ \text{subject to} \quad & g_1: \; 6x_1 - x_2 \le 19.56 && (14) \\ & g_2: \; 0.3706x_1 + x_2 \le 6.207 \end{aligned}$$

cuting off $(4, 4)^T$ as a feasible solution, and the algorithm fails to find this point. □

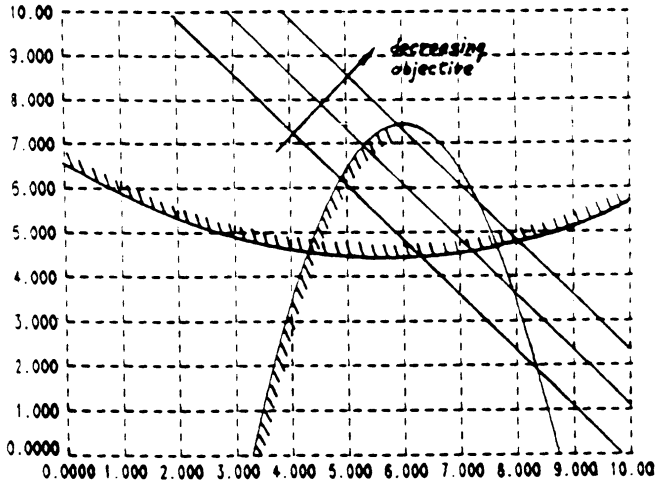
Modeling Considerations and Parameter Selection

In most engineering problems, the model is non-convex and convergence to the global optimizer cannot be guaranteed in general. Proper modeling may assist MINSLIP attain

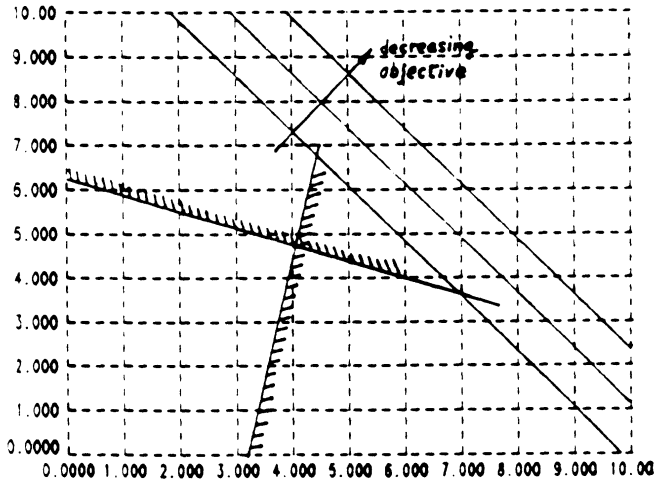
better feasible solutions. Program parameter selection is also often critical for success, good values usually discovered with experience.

Scaling of Constraints

Scaling of constraints can affect convergence of optimization algorithms in general. In MINSLIP, constraint scaling relative to each other is important, because convergence depends on finding successive points within the ε-feasible nonlinear domain with progressive tightening of this ε-feasibility. Since ε-feasibility is a measure of the domain's



(a)



(b)

Figure 3. Algorithm failure for a nonconvex problem (Example 3). (a) The mixed-discrete nonlinear problem; (b) linearization at $(5,3)^T$ cuts off $(4, 4)^T$, the discrete minimizer.

enlargement, the size of the enlargement for each constraint depends on the scale of the constraints. To assist convergence, the constraints should be scaled so that a unit change of any variable produces a nearly uniform change in the binding constraints.

Scaling of constraints is often needed for structural problems with stress and deflection constraints of the form $\sigma(\mathbf{x}) - \sigma_{allowable} \leq 0$ and $\delta(\mathbf{x}) - \delta_{allowable} \leq 0$, respectively, with $\sigma_{allowable}$ of order $\mathbf{O}(10^6$ to $10^9)$ and $\delta_{allowable}$ of order $\mathbf{O}(10^{-3}$ to $1)$. Usually one of these constraints is binding, and scaling is required.

Constraint Transformation

Consider two inequality constraints with the same feasible domain, if $x_2 = 0$ is excluded.

$$g_{3a} : x_2 - x_1 \leq 0$$
$$g_{3b} : 1 - \frac{x_1}{x_2} \leq 0 \qquad (15)$$

Their linearizations are very different. For g_{3a}, the linearization of g_{3a} is g_{3a} itself. For g_{3b}, its linearization at $(2, 5)^T$ is $2x_2 - 5x_1 + 15 \leq 0$ and at $(4, 2)^T$ is $2x_2 - x_1 - 2 \leq 0$ (see Figure 4a). Notice that g_{3b} is non-convex and the linearization cuts off part of the feasible region. The form of constraint g_{3a} is much preferred to that of g_{3b}.

Similarly, the following constraints g_{4a} and g_{4b} have the same feasible domain, if $x_2 = 0$ is excluded,

$$g_{4a} : x_1 - x_2^2 \leq 0$$
$$g_{4b} : \frac{x_1}{x_2^2} - 1 \leq 0 \qquad (16)$$

and both forms are non-convex. One form may still be preferred to the other. For example, linearizing at $(1, 2)^T$ gives $x_1 - 4x_2 + 4 \leq 0$ for g_{4a}, but $x_1 - x_2 - 2 \leq 0$ for g_{4b} (see Figure 4b). Linearizing at $(2, 1)^T$ gives $x_1 - 2x_2 + 1 \leq 0$ for g_{4a} and $x_1 - 4x_2 + 3 \leq 0$ for g_{4b} (see Figure 4c). The linearization cuts off different discrete points for different constraint forms. Depending on where the optimizer is expected to be, one form may be preferred to the other. Empirically, it has been found that form g_{4a}, where the variables are not in the denominator, usually works better.

Selection of Parameters

As mentioned earlier, to choose a starting point it is highly recommended to relax the discreteness requirement, solve the continuous problem, and use the continuous solution to start MINSLIP. Of the six program parameters, ε_0 and $\mathbf{t_0}$ are found to have the greatest influence on robustness and number of iterations. The following guidelines for setting parameter values are based on experience.

Values for ε_0, ε_f, r_ε. Parameter ε_f should be strictly greater than 0, the value 0.001 being a good choice. One way to set ε_0 is to take the starting point and vary the variables by one unit to get an idea of how the ε-feasibility is changing. When the algorithm finds a better phase 2 iterate, the epsilon value has to be reduced. Parameter r_ε should be strictly between 1 and 2.

Values for δ. If $(\mathbf{x_k} - \mathbf{x_b})_i < \delta$ for every i, convergence is considered achieved. For purely discrete problems, setting δ smaller than the interval of discretization will work. For mixed discrete problems, δ has to be small. Usually $\delta = 0.001$ to 0.005 is found to work well.

Values for the initial step bounds t_0. If $\mathbf{t_0}$ is too small, the nonlinear feasible region will not be covered, while if it is too large more iterations than necessary will occur. The choice $t_i = (x_u)_i - (x_l)_i$ has been found generally successful.

Values for r_t. Choosing a small r_t will lead to more iterations. Its value should be strictly between 1 and 2. The choice $r_t = 2$ has worked well.

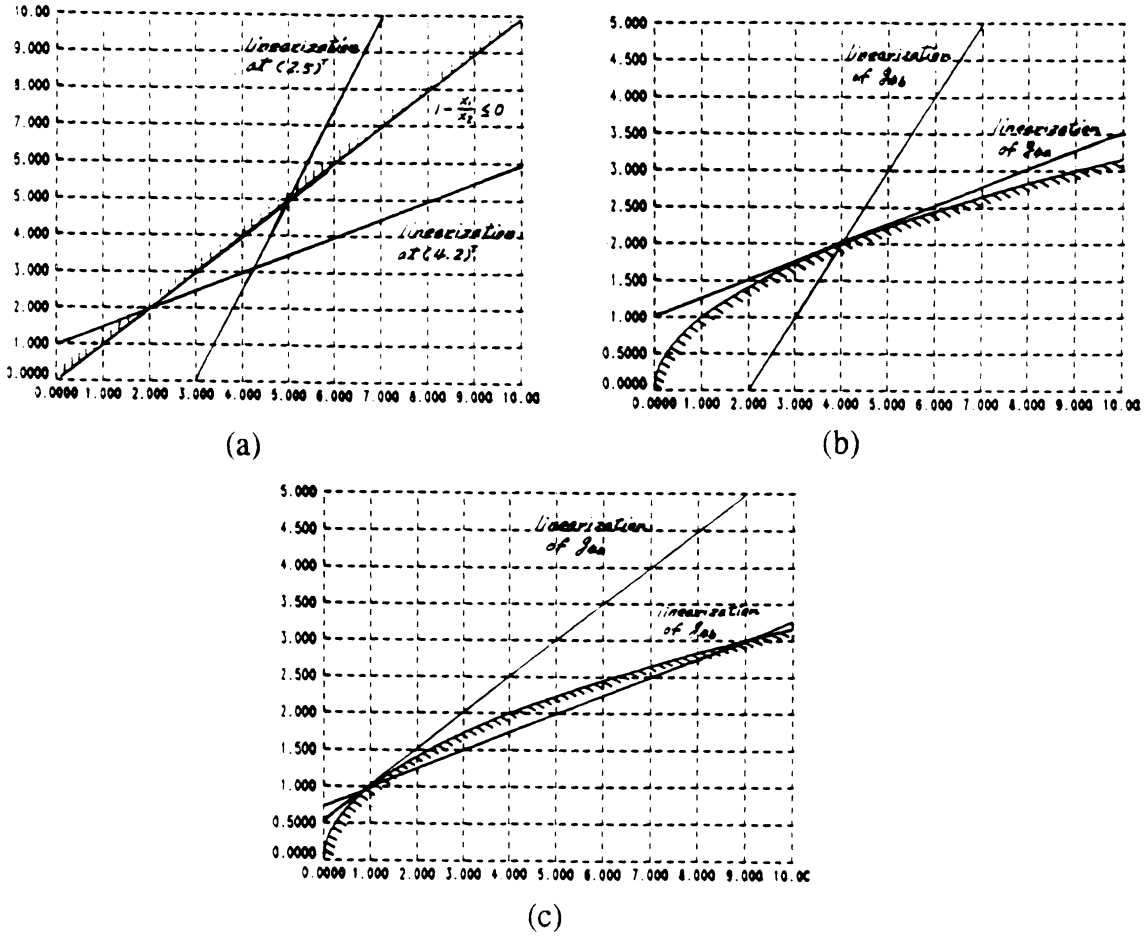


Figure 4. Constraint transformations. (a)Linearization of $1 - (x_1/x_2) \leq 0$ at two different points; (b) linearization at $(1, 2)^T$ for constraints g_{4a} and g_{4b}; (c) linearization at $(2, 1)^T$ for constraints g_{4a} and g_{4b}.

Computational Tests

In the computational tests, the primary emphasis was on robustness (whether the algorithm solves the problem) and efficiency (measured roughly by the number of function evaluations). In MDNLP applications to design, the most comprehensive tests were reported by Gupta (1980) and Cha (1987) on a variety of problems documented in other sources (e.g., Himmelblau, 1972; Eason & Fenton, 1974). MINSLIP was tested on these problems as well as some others.

MINSLIP usually obtained the same mixed-discrete minimizers as those by branch and bound (Gupta, 1980; Sandgren, 1988), by discrete sequential quadratic programming approach (Cha, 1987), and by outer-approximation (Duran & Grossmann, 1986). The number of objective and constraint function evaluations required by MINSLIP compares very favorably against branch and bound, and is in the same order of magnitude as discrete sequential quadratic programming. Performance on some other problems from the literature was also satisfactory.

Performance on Gupta's and Cha's Test Problems

Table 2 summarizes problems documented by Gupta and Cha used for testing MINSLIP. In all problems, it is assumed that an underlying continuous problem describes the mixed discrete model, for which only discrete variable values are acceptable solutions. Not all problems are listed, because (1) there is overlap in the two sets, (2) unconstrained problems in Gupta's set were discarded, (3) problems not C^1 continuous in Cha's set were discarded, and (4) problems in Cha's set where the continuous minimizers are also the mixed-discrete minimizers were discarded. Gradients of objective and constraints for the test problems were computed with forward differencing (when the discreteness requirement is relaxed).

Most problems are nonconvex with no guarantee that any solutions obtained are global minima. Thus the results reported, including those from MINSLIP, are the best to date. Also, while these are engineering problems, Gupta and Cha arbitrarily discretized the variables to demonstrate their approaches, so solutions obtained may not have any particular physical meaning.

As Gupta did not report number of function evaluations, program MIBBGRG (Mixed Integer Branch and Bound using the Generalized Reduced Gradient algorithm) was written to confirm the solutions reported for these test problems and to give an indication of the number of function evaluations needed by branch and bound. MIGGBRG uses Gupta's basic branch and bound with GRG2 (Lasdon *et al* , 1980) instead of OPT (Grabriele & Ragsdell, 1975) as the nonlinear optimizer. The branching rule in MIBBGRG is the

Table 2. Summary of Test problems from Gupta(1980) and Cha(1987)

Problem	Number of Variables NV	Number of Discrete Variables ND	Number of Constraints NG
Gupta1	2	2	2
Gupta2	5	5	6
Gupta3	2	2	2
Gupta5	4	2	5
Gupta7	3	3	2
Gupta8	3	2	3
Gupta10	2	2	2
Gupta11	3	3	3
Gupta12	4	4	4
Gupta13	5	5	5
Gupta14	5	5	6
Gupta15	3	3	1
Gupta17	7	7	7
Gupta18	6	6	6
Gupta19	8	8	8
Gupta21	3	2	2
Gupta22	4	4	5
Cha1	2	2	2
Cha2	2	2	1
Cha3	5	5	6
Cha4	2	2	3
Cha8	5	5	10
Cha11	3	3	14
Cha12	5	4	7
Cha14	4	4	5
Cha15	6	6	4
Cha16	2	2	1
Cha17	2	2	1
Cha18	2	2	4
Cha20	2	2	3
Cha25	4	4	2

best-node-first criterion for node selection and the most-fractional-part criterion for variable selection. This was reported by Gupta as the best overall strategy. Clearly, the comparison based on MIBBGRG does not necessarily reflect exactly what results may be obtained by the program written by Gupta. However, as that was a major early study it was felt that even an indirect comparison would be useful.

Table 3 summarizes the number of function evaluations needed for MIBBGRG to find the discrete solution starting from the continuous one. Even for a small number of discrete variables, the number of function evaluations is relatively large, and increasing exponentially with the number of discrete variables, to be expected for branch and bound. Even when solving a subsequent node using information from its predecessor, the number of function evaluations is large. For example, a typical four variable problem will need 80 to 200 evaluations to arrive at the continuous minimizer, if finite differencing is used for the

Table 3. Number of function evaluations using Branch and Bound with GRG2

Problem	Number of Objective & Constraint Evaluations to reach discrete minimizer	Total Number of Objective & Constraint Evaluations to fathom all nodes
Gupta1	64	139
Gupta2	138	158
Gupta3	37	45
Gupta5	549	807
Gupta7	5	53
Gupta8	167	180
Gupta10	52	56
Gupta11	190	229
Gupta12	296	339
Gupta13	963	963
Gupta14	158	158
Gupta15	119	149
Gupta17	4552	4851
Gupta18	2714	2714
Gupta19	4655	5763
Gupta21	30	88
Gupta22	769	823
Cha1	62	90
Cha2	43	55
Cha3	112	112
Cha4	36	36
Cha8	273	273
Cha11	55	55
Cha12	404	438
Cha14	622	663
Cha15	616	659
Cha16	54	101
Cha17	99	111
Cha18	77	100
Cha20	32	52
Cha25	216	216

gradient. Branch and bound typically would require 300 to 800 more evaluations to obtain the mixed-discrete optimizer and fathom all nodes.

Table 4 compares results for MIBBGRG and MINSLIP for Gupta's test set. MINSLIP obtains the same minimizers as MIBBGRG in most problems but needs far fewer function evaluations. For Problems 17, 18 and 19, MINSLIP could not find the same minimizer as branch and bound. These are not real-world engineering problems, but randomly generated (Rosen & Suzuki, 1964). The objectives and constraints are highly nonlinear and nonconvex with only one constraint binding for the seven, six, and eight variables, respectively. These problems essentially behave as highly nonconvex unconstrained problems with six, five, and seven dimensions, respectively, so the sequential linearization approach is less effective. For each problem, however, MINSLIP

was able to find a minimum that is close to that reported, see Table 5. The number of function evaluations required in each case is still substantially reduced (Table 4).

Table 4. Results and Comparison with Branch and Bound for Gupta's test set [Gupta 1980]

Prob No.	MIBBGRGl NF/N	MINSLIP NF	NC	NGRG
1	139	16	12	
2	158	9	7	
3	45	11	8	
5	807	46	38	285
7	53	13	10	
8	180	35	28	72
10	56	51	40	
11	229	55	45	
12	339	50	42	
13	963	65	56	
14	158	9	7	
15	149	31	25	
17*	4851	71	63	
18*	2714	82	72	
19*	5763	56	50	
21	88	13	10	20
22	823	50	42	

NC - Number of constraint function evaluations
NF - Number of objective function evaluations
NGRG - Number of objective and constraint function evaluations used by GRG2 in CSP
* MINSLIP did not get the optimizer obtained by MIBBGRG but the optimum is close to that obtained by branch and bound (see Table 5.4)

Table 5. Comparison of objective function values for Problems 17, 18 and 19 (Gupta, 1980)

Problem No.	MIBBGRG	MINSLIP
17	-1100.4	-1067.0
18	-778.4	-767.8
19	-1098.4	-1082.8

Table 6 shows results obtained by MIBBGRG, MINSLIP, and DRQP (Cha, op.cit.). For the reasons mentioned earlier, some of Cha's problems were not appropriate for testing MINSLIP. Cha also reported results on several problems (Nos. 4, 7, 8, 11, and 14) with discretization interval 0.001 that is in the same order of magnitude as the accuracy obtained from a typical continuous optimizer, making a discrete solution rather meaningless. Three problems (Nos. 16, 17, and 18) had irregular interval of discretization, but MINSLIP testing was performed with interval of discretization changed to 1.

For problems with the same intervals of discretization, MINSLIP obtained the same minimizers except for Problem 12. For this problem, DRQP reported a minimizer at (12, 0.5625, 55, 0.375, 0.713)^T with f = 47005.3; MIBBGRG obtained a minimizer at

Table 6. Results and comparison with Cha's test set [Cha 1987]

Prob No.	MIBBGRG NF/NC	DRQP NF	DRQP NC	MINSLIP NF	MINSLIP NC	MINSLIP NGRG
1	90	18	25	11	18	
2	55	27	34	21	16	
3	112	58	58	17	14	
4#	36	28	30	31	24	
8*#	273	30	74	81	70	
11#	55	112	123	49	40	
12@	438	74	107	46	39	41
14#	663	869	1276	57	48	
15	659	124	165	19	16	
16#	101	16	21	26	20	
17#	111	17	20	26	20	
18#	100	11	15	21	16	
20	52	21	30	41	32	
25	216	32	41	29	24	

* MINSLIP did not get the optimizer obtained by MIBBGRG
# The interval of discretization is changed to 1 for MIBBGRG and for MINSLIP
@MINSLIP obtained an answer better than both MIBBGRG and DRQP
Note: The starting point for the program DRQP is different from those of MIBBGRG and MINSLIP. The comparison here only gives a rough estimate of the number of function evaluations needed

$(10, 0.4375, 55, 0.4375, 0.733)^T$ with f = 45543.9; and MINSLIP obtained a better minimizer at $(10, 0.5, 55, 0.375, 0.733)^T$ with f = 42866.7. This problem is non-convex and the local search in DRQP appears unable to leave the neighborhood of the continuous minimizer. For the same reason of nonconvexity, the bounding rule in branch and bound fathoms a node it should not. For problems with different discretization intervals, we can compare only solutions by MINSLIP and MIBBGRG. MINSLIP obtained the same minimizers as MIBBGRG except for Problem 8. In this problem, MINSLIP obtained the minimizer $(3, 2, 4, 4, 5)^T$ with f = -30.0; MIBBGRG reported a better minimizer $(3, 3, 4, 4, 3)^T$ with f = -32.0. MINSLIP fails to find $(3, 3, 4, 4, 3)^T$ because the constraints are non-convex and the last linearization at $(3, 2, 4, 4, 5)^T$ cuts off the true minimizer.

In general, MINSLIP outperforms MIBBGRG in terms of function evaluations. From Cha's reported results, in DRQP the number of function evaluations is not very sensitive to the interval of discretization. This being the case, Table 6 indicates that MINSLIP is only slightly more efficient than DRQP for most problems. A newer version of MINSLIP handles irregular intervals, but new tests on these problems have not yet been conducted.

Other Test Problems

Some other demonstration problems were reported by Duran (1984) and Sandgren (1988), but without data on number of objective and constraint function evaluations.

Duran & Grossmann (1986) reported an outer-approximation algorithm for a special class of MDLP models. The MDNLP model class for which MINSLIP was designed is a more general one, so one would expect that MINSLIP should solve problems in that special class but less efficiently. Indeed MINSLIP produced the same minimizers as those given in Duran's thesis (1984), but since function evaluation data were not given by Duran, further comparison is not possible. See Table 7.

Table 7. MINSLIP results for Duran's problems (Duran, 1984)

Duran Problem 1	
Minimizer	$(0, 1, 0, 1.301, 0, 1)^T$
Objective Function	6.010
NF	28
NC	24
NGRG	27
Duran Problem 2	
Minimizer	$(0, 1, 1, 1, 0, 0, 2, 0.652, 0.326, 1.078, 1.078)^T$
Objective Function	73.035
NF	57
NC	52
NGRG	200

Sandgren (1988) used a branch and bound approach similar to Gupta but with an SQP method for the continuous optimizer. Results on five problems were reported, but there were sufficient numerical data for replicating only two of them. MINSLIP produced better results than those reported (Table 8). MIBBGRG was then used for confirmation and the same solutions as MINSLIP were produced. Table 9 shows the number of objective and constraint function evaluations required for MIBBGRG and MINSLIP. Again, this information is not available in the cited reference.

Table 8. Comparison with Sandgren's results (Sandgren, 1988)

	Minimizer	Objective
Gear Train Problem		
MINSLIP	$(19, 16, 42, 50)^T$	0.233E-6
Sandgren's B&B	$(18, 22, 45, 60)^T$	5.700E-6
Pressure Vessel Problem		
MINSLIP	$(1.125, 0.625, 58.290, 43.693)^T$	7179.7
Sandgren's B&B	$(1.125, 0.625, 47.7008, 117.701)^T$	8129.8

Table 9. Comparison with Branch and Bound for Sandgren's problems

Problem	MIBBGRG NF/NC	MINSLIP NF	MINSLIP NC	MINSLIP NGRG
Gear Train	207	64	—	—
Pressure Vessel	147	15	12	60

Branch and Bound with Different Continuous Optimizers

One may argue that the inferior performance on function evaluations observed for MIBBGRG is due in part to the inefficiency of GRG for the types of problems solved. To investigate this possibility empirically, two other branch and bound programs were developed and tested on a sample of Gupta's and Cha's examples. They were based on MIBBGRG, with GRG2 replaced by another nonlinear continuous optimizer. Program MIBBSQP used the SQP algorithm NLPQLD (Schittkowski, 1985), and program MIBBSLP used the SLP algorithm SEQLI2 by Post & Bremicker (Eschenauer et al., 1988).

Table 10 shows comparisons among MIBBGRG, MIBBSQP, MIBBSLP and MINSLIP. Using a method other than GRG did reduce the number of function evaluations for several examples, but this was not always true (Gupta7, Gupta11, Cha2, Cha3, and Cha8). Also, MIBBSQP and MIBBSLP found solutions worse than those by MIBBGRG for some of the problems (Gupta3, Gupta17, Gupta19, and Cha3). NLPQLD and SEQLI2 are not as reliable as GRG2 in finding a feasible solution for each of the continuous

Table 10. Comparisons using different continuous optimizers for branch and bound

Prob No.	MIBBGRG NF/NC	MIBBSQP NF/NC	MIBBSLP NF/NC	MINSLIP NF/NC
Gupta1	139	108	66	16/12
Gupta2	158	155	69	9/7
Gupta3	45	45*	39	11/8
Gupta7	53	30	176	13/10
Gupta11	229	129	250	55/45
Gupta17	4851	3513	2106*	71/63*
Gupta19	5763	3437*	5569*	56/50*
Cha2	55	65	56	21/16
Cha3	112	497*	251	17/14
Cha8	273	126	1406	81/70*

* The program found a worse solution than that obtained by MIBBGRG

subproblems. So for these problems, nodes are fathomed when a feasible solution cannot be found by NLPQLD or SEQLI2, when in fact the discrete minimizer is in one of these nodes. MINSLIP needed fewer function evaluations than the other programs (Table 10), and one may conclude that the reduction in number of function evaluations achieved throughout the reported tests comes mainly from the linearization strategy in MINSLIP.

Three Design Applications

In this chapter, we use MINSLIP to solve three design problems with discrete variables. More design applications of MINSLIP can be found in Loh (1989), and in Bremicker, Loh, and Papalambros (1989) where structural problems are examined in some detail.

Presssure Vessel Design

A horizontal pressure vessel for storage of liquid butene and mounted on two equidistant saddle supports (Nguyen & Mistree, 1986) is to be sized minimizing cost of manufacture and using the ASME code stipulations as constraints. The vessel has a hollow cylindrical body (shell) of 50 feet length and 5 feet diameter, capped by two ellipsoidal ends (heads), see Figure 5. Two saddles support the vessel, 6 feet from each end. Saddle and vacuum rings strengthen the shell and provide stiffness at partial vacuum conditions. Wear plates are placed on the saddles to reduce local stress concentration.

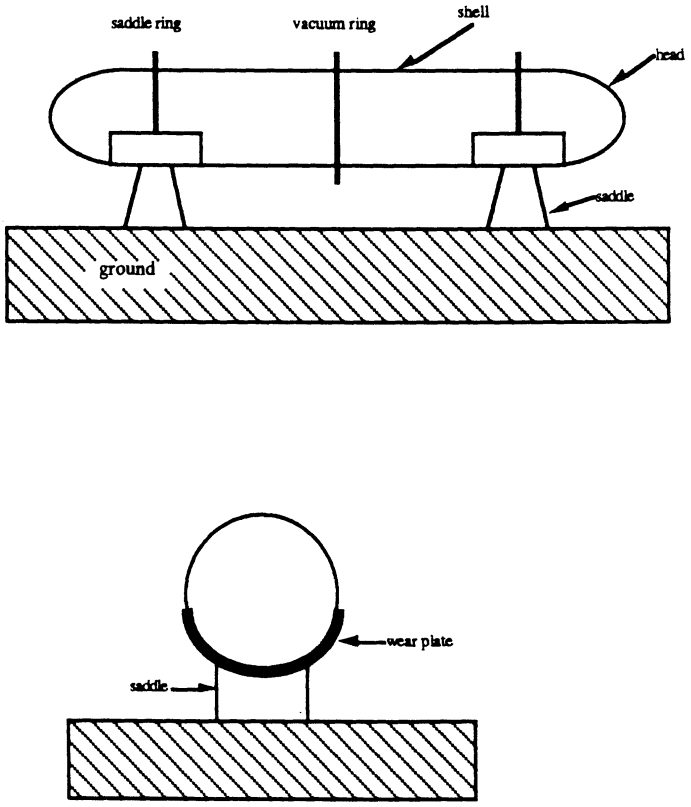


Figure 5. Horizontal pressure vessel on two saddle supports

Vessel thickness is assumed small relative to other dimensions and thin-wall theory applies. Design variables are thicknesses for head (T_H), shell (T_S), saddle ring (T_{SR}),

vacuum ring (T_{VR}) and wear plate (T_{WP}). They are formed from plates available only in thickness increments of 1/16 inch. Material chosen for shell, head and wear plate is SA-516-70 steel and for ring and saddles is SA-36. The cost of manufacture is estimated to vary linearly with T_S, T_H and T_{WP} and quadratically with respect to T_{SR} and T_{VR}.

The mathematical model is given below.

$$\text{minimize } f = 19380 + 32154T_S + 13189T_H + 2376T_{WP} + 5329(T_{VR}^2 + 2T_{SR}^2)$$

subject to

$$g_1: \frac{P_i R}{2T_S} + \frac{QL_A}{\pi R^2 T_S}\left[1 - \frac{1 - \frac{L_A}{12L} + \frac{R^2 - H^2}{24L_A L}}{1 + \frac{H}{9L}}\right] - \sigma_A \le 0$$

$$g_2: \frac{P_i R}{2T_S} + \frac{3QL}{\pi R^2\left(T_S + \frac{A_{VR}}{L_{RR}}\right)}\left[\frac{1 + 2\frac{R^2 - H^2}{144L^2}}{1 + \frac{H}{9L}} - \frac{L_A}{3L}\right] - E_F\sigma_A \le 0$$

$$g_3: \frac{P_i R}{2T_S} - \frac{QL_A}{0.603R^2 T_S}\left[1 - \frac{1 - \frac{L_A}{12L} + \frac{R^2 - H^2}{24L_A L}}{1 + \frac{H}{9L}}\right] - \frac{E_Y T_S}{29R}\left(2 - \frac{200T_S}{3R}\right) \le 0$$

$$g_4: \frac{P_i R}{2T_S} - \frac{3QL}{\pi R^2\left(T_S + \frac{A_{VR}}{L_{RR}}\right)}\left[\frac{1 + 2\frac{R^2 - H^2}{(12L)^2}}{1 + \frac{H}{9L}} - \frac{L_A}{3L}\right] - \frac{E_Y T_S}{29R}\left(2 - \frac{200T_S}{3R}\right) \le 0$$

$$g_5: \frac{P_i R}{2T_S} - \frac{QL_A}{0.603R^2 T_S}\left[1 - \frac{1 - \frac{L_A}{12L} + \frac{R^2 - H^2}{24L_A L}}{1 + \frac{H}{9L}}\right] - 0.5\sigma_y \le 0 \qquad (17)$$

$$g_6: \frac{P_i R}{2T_S} - \frac{3QL}{\pi R^2\left(T_S + \frac{A_{VR}}{L_{RR}}\right)}\left[\frac{1 + 2\frac{R^2 - H^2}{(12L)^2}}{1 + \frac{H}{9L}} - \frac{L_A}{3L}\right] - 0.5\sigma_y \le 0$$

$$g_7: \frac{K_7 Q}{(T_S + T_{WP})\left(B + 1.56\sqrt{RT_S}\right)} - 0.5\sigma_Y \le 0$$

$$g_8: \frac{K_3 Q}{R(T_S + T_{WP})}\left[\frac{12L - 2L_A}{12L + \frac{4}{3}H}\right] - 0.8\sigma_A \le 0$$

$$g_9: \frac{1}{T_H} - 0.8\sigma_A \le 0$$

$$g_{10}: \frac{K_{10}QRL_C}{I_{SR}} - \frac{K_9 Q}{A_{SR}} - 0.5\sigma_{yR} \le 0$$

$$g_{11}: \frac{K_9 Q}{A_{SR}} + \frac{K_{10}QRL_D}{I_{SR}} - 0.5\sigma_{yR} \le 0$$

$$g_{12}: \frac{3K_{11}Q}{RT_{WEB}} - \frac{2}{3}\sigma_{AS} \le 0$$

$$g_{13}: \delta_{ms} - 0.024L \le 0$$

$$g_{14}: 4 + \frac{B}{2} + 0.78\sqrt{RT_S} - L_A \le 0$$

$$g_{15}: T_S - T_H - 0.125 \le 0$$

$$g_{16}: T_S - T_{WP} \le 0$$

$$g_{17}: T_S - \frac{1}{5}R \le 0$$

$$g_{18}: T_H - \frac{1}{5}R \le 0 \qquad \text{(17, con'd)}$$

$$g_{19}: I_{VRM} - I_{VR} \le 0$$

$$g_{20}: I_{VRM} - I_{SR} \le 0$$

$$g_{21}: P_E - \frac{E_Y}{16}\left(\frac{T_H}{0.9DOH}\right)^2 \le 0$$

$$g_{22}: P_E - \frac{0.93E_Y L_{RR}}{DOS}\left(\frac{T_S}{DOS}\right)^{2.5} \le 0$$

where the intermediate variables DOS, DOH, L_{RR}, Q, δ_{ms}, A_{SR}, L_C, L_D, I_{SR}, A_{VR}, I_{VR}, I_{VRM} are defined by the formulae below:

$$DOS = D+2T_S$$

$$DOS = D+2T_H$$

$$L_{RR} = 0.5(12L)-L_A$$

$$Wves = 1.1\rho_s(12\pi\, DL(T_S+T_{CA})+2.16D^2\,(T_H+T_{CA}))$$

$$Wliq = \rho_b(3\pi D^2 L+0.262D^3)$$

$$Q = (Wves + Wliq)/2$$

$$Xie = \pi T_S(D+2T_S)^3/8 \qquad (18)$$

$$Wu = 2Q/(12L+4H/3)$$

$$\delta_{ms} = Wu(12L-2L_A)^2(5\,(12L-2L_A)^2 - 24(2H/3+L_A)^2)/(384E_Y Xie)$$

$$A_1 = (T_{SR}+1.56\sqrt{RT_S})T_S$$

$$A_2 = 8T_{SR}^2$$

$$A_{SR} = A_1+A_2$$

$$L_C = A_1T_S/2 + A_2(T_S+4T_{SR})/A_{SR}$$

$$L_D = T_S+8T_{SR}-L_C$$

$$H_1 = L_C-T_S/2$$

$$H_2 = T_S + 4T_{SR} - L_C$$
$$I_{SR} = A_1 H_1^2 + A_2 H_2^2 + A_1 T_S^2/12 + T_{SR}(8T_{SR})^3/12$$
$$A_{1V} = 1.56 T_S \sqrt{R T_S}$$
$$A_{2V} = 8T_{VR}^2$$
$$A_{VR} = A_{1V} + A_{2V}$$
$$L_{CV} = A_{1V} T_S/2 + A_{2V}(T_S + 4T_{VR})/A_{VR} \qquad \text{(18, Cont'd)}$$
$$L_{DV} = T_S + 8T_{VR} - L_{CV}$$
$$H_{1V} = L_{CV} - T_S/2$$
$$H_{2V} = T_S + 4T_{VR} - L_{CV}$$
$$I_{VR} = A_{1V} H_{1V}^2 + A_{2V} H_{2V}^2 + A_{1V} T_S^2/12 + T_{VR}(8T_{VR})^3/12$$
$$I_{VRM} = 0.12798 D^3 (T_S + A_{VR}/L_{RR})(T_S/DOS)^{1.5}$$

The parameter values for this problem are

$\sigma_A = 17500$	$D = 120$	$K_9 = 0.340$	$R = 60$
$\sigma_{AR} = 12700$	$E_F = 0.85$	$K_{10} = 0.053$	$T_{CA} = 0.125$
$\sigma_{AS} = 12700$	$E_Y = 29*10^6$	$K_{11} = 0.204$	$T_{WEB} = 0.5$
$\sigma_y = 38000$	$H = 30$	$L = 50$	
$\sigma_{yR} = 36000$	$K_3 = 0.319$	$L_A = 50$	
$\rho_S = 0.28$	$K_4 = 0.880$	$P_e = 5$	
$\rho_b = 0.0216$	$K_7 = 0.760$	$P_i = 50$	(19)

Constraints 1 to 12 limit tensile bending stresses, compressive and shear stresses at various locations. Constraint 13 gives allowable deflection at midspan as 2.4% of the length. Constraints 14 and 15 are for ease of fabrication. Constraint 16 requires the wear plate thickness to be larger than the shell thickness. Constraints 17 and 18 require that for the stress analysis to be valid, shell and head thicknesses must be less than one-fifth the radii. Constraints 19 and 20 set the minimum allowable moment of inertia for vacuum and saddle rings. Constraints 21 and 22 guard against yielding under partial vacuum.

Using branch and bound, the minimizer $(5/16, 3/16, 9/16, 18/16)^T$ is found, with f = 47,818. With the continuous optimizer as a starting point, MINSLIP converges to the same answer obtained by branch and bound. However, MINSLIP needs 17 objective function and 14 constraint function evaluations, compared with 395 objective and constraint function evaluations by branch and bound. Note that starting at $(8/16, 8/16, 18/16, 29/16, 11/16)^T$, the algorithm converges to $(5/16, 4/16, 9/16, 18/16, 5/16)^T$, f = 48,642. The global minimizer is missed, because the linearization at $(5/16, 4/16, 9/16, 18/16, 5/16)^T$ cuts it off. The algorithm still produces a series of feasible designs (see Table

11). This feature of MINSLIP, producing a sequence of feasible discrete intermediate designs, is clearly a very desirable one for design applications.

Table 11: Intermediate feasible solutions for the pressure vessel design

$(T_H, T_S, T_{SR}, T_{VR}, T_{WP})^T$	Objective
$(\frac{8}{16}, \frac{8}{16}, \frac{18}{16}, \frac{29}{16}, \frac{11}{16})^T$	90,687
$(\frac{6}{16}, \frac{5}{16}, \frac{18}{16}, \frac{22}{16}, \frac{6}{16})^T$	63,344
$(\frac{5}{16}, \frac{4}{16}, \frac{14}{16}, \frac{18}{16}, \frac{5}{16})^T$	51,037
$(\frac{5}{16}, \frac{4}{16}, \frac{11}{16}, \frac{18}{16}, \frac{5}{16})^T$	49,475
$(\frac{5}{16}, \frac{4}{16}, \frac{10}{16}, \frac{18}{16}, \frac{5}{16})^T$	49,038

Selection of Bolts in Standard Sizes

This problem concerns selection of standard size bolts to fasten flange joints of a pressure vessel undergoing rapid pressure fluctuation. An *even* number of bolts are placed uniformly at diameter distance of 350mm around a cylindrical pressure vessel of 250mm internal diameter. Internal gage pressure fluctuates rapidly between 0 and 2.5 MPA. The pressure vessel is made of cast iron (E = 100GPa), and the cover plate is made of aluminum (E = 70GPa). The bolts are made of Class 88 steel with a fatigue-limiting value of alternating norminal stress of 69MPa. The governing design equations can be found in Juvinall (1983). We are interested in selecting the number of standard-sized bolts from Table 12 to minimize total cost. This cost is the sum of two components: price of bolts (CostB), and labor cost for drilling holes and installing the bolts (CostL).

Selection of a bolt from the above set is done by considering two series, one series for bolts with diameter from 3mm to 8mm, and the other from 10mm to 24mm. Three discrete variables are used to describe bolt selection and one discrete variable to represent the number of bolts: x_1 is a binary variable denoting whether the bolt comes from the first or second series; x_2, x_3 are the diameters of bolts chosen from the first, second series, respectively; x_4 denotes an even number of bolts.

The nominal diameter d is modeled by

$$d = x_1x_2 + (1 - x_1)x_3 \tag{20}$$

The effective stress area is approximated (Figure 6) by a quadratic polynomial:

$$A = 2.857 - 1.0692d + 0.6562d^2 \tag{21}$$

Table 12. ISO metric coarse screw threads

Bolt	Nominal Diameter (d)	Stress Area (A)	Cost per Bolt (Costpb)
M3x0.5	3	5.03	15
M4x0.7	4	8.78	16
M5x0.8	5	14.2	17
M6x1	6	20.1	18
M7x1	7	28.9	19
M8x1.25	8	36.6	20
M10x1.5	10	58.0	22
M12x1.75	12	84.3	24
M14x2	14	115	26
M16x2	16	157	28
M18x2.5	18	192	30
M20x2.5	20	245	32
M22x2.5	22	303	34
M24x3	24	353	36

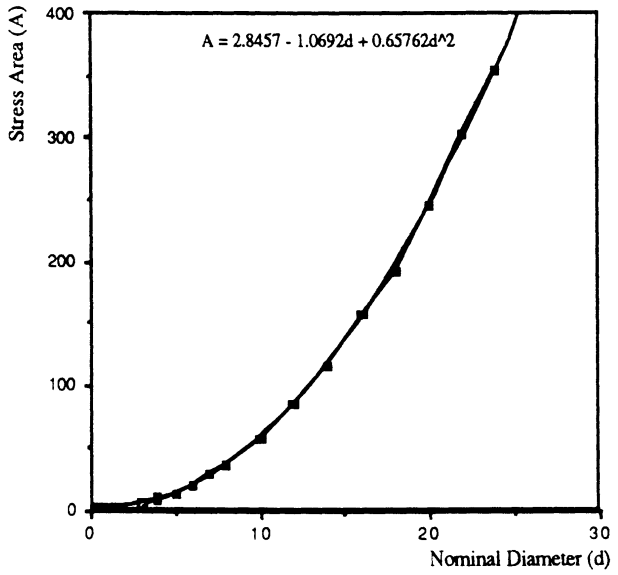


Figure 6. Stress area versus diameter.

Curve fitting causes slight, but acceptable, errors in determinating stress areas (Table 13). The model is thus

$$\begin{aligned} &\text{minimize} && f = \text{CostB} + \text{CostL} = 12\,d\,n + 19\,n \\ &\text{subject to} && g_1:\ x_1 \le 1 \\ & && g_2:\ (F\,K/2\,n\,A) - \sigma \le 0 \\ & && g_3:\ (C/n\,d) - 10 \le 0 \\ & && g_4:\ 5 - (C/n\,d) \le 0 \end{aligned} \tag{22}$$

where $n = 2x_4$, $d = x_1x_2 + (1 - x_1)x_3$, $A = 2.857 - 1.0692d + 0.6562d^2$, $K = 0.3333$, $F = 245400$, $\sigma = 69$, $C = 350\pi$.

Table 13. Fitted stress area for ISO coarse screw threads

Diameter	Fitted Stress Area (A)
3	5.55
4	9.08
5	13.9
6	20.6
7	27.5
8	36.3
10	57.8
12	84.5
14	116
16	153
18	196
20	244
22	297
24	355

The feasible design of selecting twelve M12x1.75 bolts (Juvinall, 1983) is chosen as a starting point $(1, 5, 6, 6)^T$. MINSLIP obtained the minimizer $(1, 3, 10, 3)^T$, i.e., selecting six M20x2.5 bolts, with cost reduction from 516 to 306, after 22 objective and 18 constraint function evaluations. Branch and bound found the same answer after 232 sets of objective and constraint function evaluations.

This problem also demonstrates the importance of choosing a particular form of the nonlinear constraints. Using the model

$$g_1: x_1 \le 1 \qquad g_3: 1 - \frac{10\,n\,d}{C} - 10 \le 0$$

$$g_2: \left(\frac{F\,K}{2\,\sigma} - n\,A\right) 0.001 \le 0 \qquad g_4: \frac{5\,n\,d}{C} - 1 \le 0$$

(23)

the solution at $(1, 3, 10, 3)^T$ with $f = 306$ was obtained, while using the model

$$g_1: x_1 \le 1 \qquad g_3: \frac{C}{n\,d} - 10 \le 0$$

$$g_2: \frac{F\,K}{2\,n\,A} - \sigma \le 0 \qquad g_4: 5 - \frac{C}{n\,d} \le 0 \qquad (24)$$

MINSLIP converged to the feasible point $(1, 3, 7, 4)^T$ with $f = 360$ instead. In the model of Eq.(24) when the linearization at point $(1, 3, 7, 4)^T$ is performed, the minimizer $(1, 3, 10, 3)^T$ is cut off.

Bracket Design

The bracket shown in Figure 7a is modeled as in Figure 7b.This structural optimization problem requires finite element analysis to provide stress and deflection information, so MINSLIP was interfaced to the finite element program FEM2 (Kikuchi, 1986). There are three discrete variables: x_1 is the dimension of the section which fits into a prefabricated

Table 14. Material properties of Aluminum, Brass and Steel

	Aluminum	Brass	Steel
Modulus of Elasticity E (GPa)	71	106	200
Yield Strength (MPa)	186	416	1020

metal strip with slots at regular intervals 5 mm apart; x_2 is the dimension of the mid-section in multiples of 1mm due to manufacturing limitations; x_3 is a variable for selecting one of three materials: aluminum, brass, or steel. Material properties are given in Table 14.

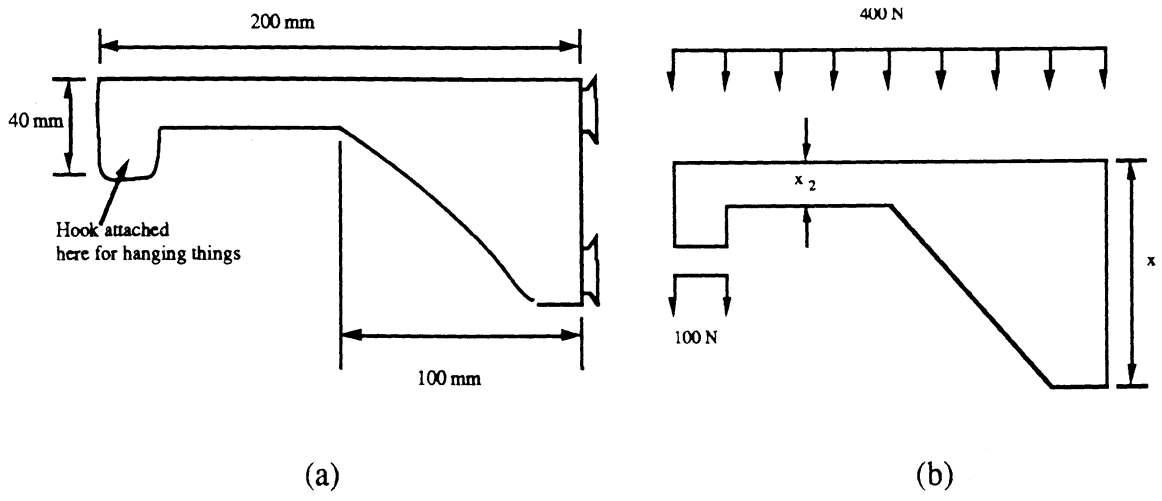


Figure 7. Bracket design. (a) The bracket; (b) the analysis model.

The objective is to minimize material cost subject to stress and deflection constraints. Allowable deflection is 2 mm. With a safety factor of 1.5, the material must not yield or shear under the design load. The maximum shear stress theory (Shigley, 1985) is used for a conservative design. Two cost functions are considered (see Table 15). Since

Table 15 . Two sets of material costs

	Aluminum	Brass	Steel
Cost Function C1($/m^2)	12	14	17
Cost Function C2($/m^2)	12	14	18

MINSLIP requires gradient information with respect to all variables, gradient information for x_3 is obtained from curve fitting the material properties by quadratic functions:

$$E = (95 - 53.5x_3 + 29.5x_3^2)10^3 \text{ MPa}$$
$$\text{Yield strength} = 330 - 331x_3 + 187x_3^2 \text{ MPa} \quad (25)$$
$$C1 = 11 + 0.5x_3 + 0.5x_3^2$$
$$C2 = 12 - x_3 + x_3^2$$

The finite element program solves for stresses and displacements using 4-node plane stress elements.

With a starting point of $(10, 50, 1)^T$, the results obtained are given in Table 16. MINSLIP indicates selecting a different material for each of the two cost functions. MIBBGRG obtained worse results for both cost functions.

Table 16. Results for the bracket design

	MINSLIP		MIBBGRG	
	Cost C1	Cost C2	Cost C1	Cost C2
x_1 (mm)	7	15	15	15
x_2 (mm)	40	40	45	45
x_3 (material)	3	2	2	2
f ($)	0.6868	0.7000	0.7420	0.7420
NF	43	37	334	305
NC	35	30	334	305

Solve this problem separately for each of the three different materials, gives the same results as MINSLIP (Table 17), but is less efficient. Therefore it may be worthwhile to incorporate a material selection variable into an overall model instead of solving for each material separately.

Table 17. Solving for the three materials separately

	Aluminum	Brass	Steel
x_1 (mm)	20	15	7
x_2 (mm)	55	40	40
X-sectional Area (mm^2)	6500	5000	4040
Total Cost at C1	0.7800	0.7000	0.6868
Total Cost at C2	0.7800	0.7000	0.7272
NF (using MINSLIP)	16	16	26
NC (using MINSLIP)	12	12	20

Conclusion

MINSLIP is an attractive algorithm in both aspects of performance, i.e., robustness and efficiency. The computational results from both test problems and design applications confirm the expectations of good performance, as well as the limitations in solving

nonconvex problems. Results from a variety of structural design problems, reported elsewhere as cited earlier, are particularly encouraging for solving models with extensive monotonicities and nearly constraint-bound solutions. As experience accumulates, some further empirical enhancements are possible and are currently under implementation. Complete listing of the program and data files for all problems reported here can be found in Loh (1989).

Acknowledgements

This research was partially supported by NSF Grants DMC 85-14721 and DMC 86-11916 at the University of Michigan. The first author was also supported by a Fellowship from the National University of Singapore. This support is gratefully acknowledged. The authors also express their thanks to John Birge for his insights during the course of this research.

References

Bremicker, M., Loh, H.T., & Papalambros, P.Y., 1989, " Solution of Mixed-Discrete Structural Optimization Problems with a New Sequential Linearization Algorithm", *Computers and Structures* (submitted). Also, Report UM-MEAM-89-10, The University of Michigan, Ann Arbor.

Cha, J.Z. 1987, *Mixed Discrete Constrained Nonlinear Programming via Recursive Quadratic Programming*, Phd Thesis, SUNY at Buffalo.

Dakin, R.J. 1965, "A Tree-Search Algorithm for Mixed Integer Programming Problems", *Computer Journal*, Vol 8, no. 3, pp 250-255.

Duran, M.A. & Grossmann, I.E. 1986, "An Outer-Approximation Algorithm for a Class of Mixed-integer Nonlinear Programs", *Mathematical Programming*, Vol 36, pp307-339.

Duran, M.A. 1984, *A Mixed-integer Nonlinear Programming Approach for the Systematic Synthesis of Engineering Systems*, PhD Thesis, Carnegie-Mellon University.

Eason, E. D. & Fenton, R.G., 1974, "A Comparison of Numerical Optimization Methods for Engineering Design", *Journal of Engineering for Industry*, Vol 96, pp196-200.

Eschenauer, H., Post, P.U. & Bremicker, M. 1988, "Einsatz der Optimierungsprozedur SAPOP zur Auslegung von Bauteilkomponenten", *Bauingenieur*, Vol 63, pp515-526.

Gabriele, G. & Ragsdell, K.M., "The Generalized Reduced Gradient Method: A reliable Tool for Optimal Design", ASME Publication, 75-DET-103.

Geoffrion, A.M. 1972, "Generalized Bender's Decomposition", *Journal of Optimization Theory and Applications*, Vol 10, no. 4, pp237-260.

Gupta, O.K.1980, *Branch and Bound Experiments in Nonlinear Integer Programming*, PhD Thesis, Purdue University.

Himmelblau, D.M. 1972, *Applied Nonlinear Programming*, McGraw-Hill.

IMSL Library, 1982, CC Memo June 1982, University of Michigan Computing Center, Ann Arbor.

Juvinall, R.C. 1983, *Fundamentals of Machine Component Design*, John Wiley & Sons.

Kikuchi, N. 1986, *Finite Element Methods in Mechanics*, Cambridge University Press.

Lasdon, L.S., Waren, A.D. and Ratner, M.W. 1980, *GRG2 User's Guide*, Case-Western Reserve University.

Loh, H.T., 1989, *A Sequential Linearization Approach for Mixed-Discrete Nonlinear Design Optimization*, Doctoral Dissertation, Dept. of Mechanical Engineering and Applied Mechanics, The University of Michigan, Ann Arbor.

Loh, H.T., & Papalambros, P.Y., 1989, " A Sequential Linearization Approach for Solving Mixed-Discrete Nonlinear Design Optimization Problems", *Trans. ASME, J. of Mechanical Design* (in review). Also, Report UM-MEAM-89-08, The University of Michigan, Ann Arbor.

Nguyen, N. & Mistree, F. 1986, "Design of Horizontal Pressure Vessels Using Decision Support Problem Technique", Journal of Mechanism, Transmissions and Automation in Design, Vol 108, pp203-210.

Rosen, J.E. & Suzuki, S. 1964, "Construction of Nonlinear Programming Test Problems", *Communications of the ACM*, Vol 8-2, pp113.

Sandgren, E. 1988, "Nonlinear Integer and Discrete Programming in Mechanical Design", *Proceedings of 1988 ASME Design Technology Conference*, Kissimmee, Florida, Sept 25th 1988, pp 95-105.

Schittkowski, K. 1985, "NLPQL - A FORTRAN Subroutine Solving Constrained Nonlinear Programming Problems", *Annals of Operations Research*, Vol 5, pp485-500.

Shigley, J.E. 1985, *Mechanical Engineering Design*, McGraw-Hill.