Method for Nonlinear Optimization with Discrete Design Variables

Gregory R. Olsen*
Failure Analysis Associates, Palo Alto, California
and
Garret N. Vanderplaats†
VMA Engineering, Santa Barbara, California

A numerical method is presented for the solution of nonlinear discrete optimization problems. The applicability of discrete optimization to engineering design is discussed, and several standard structural optimization problems are solved using discrete design variables. The method uses approximation techniques to create subproblems suitable for linear mixed-integer programming methods. The method employs existing software for continuous optimization and integer programming.

Introduction

MATHEMATICAL optimization techniques have become increasingly important in engineering design. The field of structural optimization has grown steadily in recent years particularly in the aerospace industry, where weight reduction is of paramount importance. In 1960, Schmit proposed the use of mathematical programming techniques to automate the design of nonlinear inequality-constrained structures.[1] The process was termed "structural synthesis," and today numerical optimization techniques are used in a variety of structural design applications. Among optimization codes, general-purpose programs that combine finite-element analysis and mathematical programming techniques are of particular interest.

A common limitation of existing structural optimization approaches is the requirement that the design variables be continuous. Each design variable must be allowed to assume any value within its bounds. This restriction excludes the large category of problems in which the design variables may only assume discrete values from a predetermined set. Discrete variable problems abound in structural engineering design. Often, practical designs must use only those components that are available from a manufacturer's inventory. Therefore, only certain member sizes, pipe diameters, or panel thicknesses produce meaningful designs. Composite structural design often involves design variables that must be integer valued—for instance, the number of plies in a laminate. To date, no completely satisfactory methods exist to solve discrete nonlinear optimization problems.

Schmit and Fleury[2] use dual methods to solve the discrete-continuous variable problem. This procedure has the attraction that it deals with a separable approximation to the original problem and is therefore quite efficient. On the other hand, because the discrete variable problem is nonconvex, dual methods do not guarantee an optimum. However, if the available discrete values are closely spaced, this approach can be expected to reliably find a very near optimum solution.

Various other methods have been proposed, usually based on the classical branch and bound approach. This has the advantage of a solid theoretical basis, at the expense of solving a large number of optimization subproblems. A recent example is contained in the method presented by Mesquita and Kamat,[3] which offers some innovative modifications to Dakin's method[4] to improve efficiency.

A method is presented here to provide a practical approach to the nonlinear discrete optimization problem. The method employs established operations research techniques and is intended to augment existing continuous optimization approaches.[5]

Discrete Design-Variable Problem

The general formulation for a problem having discrete variables is presented here. It should be noted that many problems are actually mixed—they contain both discrete and continuous design variables. The following formulation accommodates the mixed case. The general problem statement is find the set of design variables that will

Minimize

$$F(X) \tag{1}$$

Subject to

$$g_j(X) \le 0 \qquad j = 1,m \tag{2}$$

$$h_k(X) = 0 \qquad k = 1,l \tag{3}$$

where

$$X = (X_1,\ldots,X_p,X_{p+1},\ldots,X_n)^T \tag{4}$$

$$x_i \in (d_{i1},d_{i2},\ldots,d_{iq}) \qquad i = 1,p \tag{5}$$

$$X_k^L \le X_k \le X_k^u \qquad k = p+1,n \tag{6}$$

where p is the number of discrete design variables, n the total number of design variables, d_{ij} the ith discrete value for design variable i, X_k^L the lower bound for design variable k, and X_k^u the upper bound for design variable k.

Initially one might suppose that, since fewer possible solutions exist, the discrete problem may be easier to solve than the continuous problem. However, except for the most trivial cases, the discrete problem presents a great increase in difficulty. In general, the discrete design space is disjoint and nonconvex. Thus, standard mathematical programming techniques and mathematical optimality criteria cannot be applied directly.

Presented as Paper 87-9789 at the AIAA/ASME/AHS/ASEE 28th Structures, Structural Dynamics and Materials Conference, Monterey, CA, April 6–8, 1987; received April 20, 1987; revision received Oct. 5, 1988.

*President. Formerly Graduate Student, Mechanical Engineering, University of California, Santa Barbara. Member AIAA.

†Engineer. Associate Fellow AIAA.

In a purely discrete problem, one could conceivably test every possible combination of design variables and select the best optimum. This process, called enumeration, will certainly find the optimum to the discrete problem. Consider however, a problem of 10 design variables where each variable may assume one of 10 values. Ten to the 10th power possible solutions exist. Even with the tremendous computing power available today, this approach is much too costly, especially if a complete finite-element analysis must be executed for each combination.

The most common approach to discrete variable problems in engineering design is rounding. The design variables are treated as continuous and, once the optimum is found, the design variables are rounded off to the nearest available discrete values. Caution must be taken to insure that the rounding does not violate constraints. In the structural design, this task can usually be accomplished by rounding up to increase the member sizes. Obviously, the insight of the engineer can help in making good rounding decisions, and rules of thumb are available in some texts.[6] Rounding off in any form, however, fails to provide any certainty of optimality. As the number of design variables is increased, rounding becomes more difficult. Even the most efficient optimization procedure is of dubious value if the final design must be chosen in such an uncertain manner.

Rigorous and practical mathematical methods for solving nonlinear discrete and mixed-discrete optimization problems are not readily available. Linear discrete optimization methods are well developed and successful. Integer programming is an area of extensive study in operations research. Textbooks on the subject are abundant, and many general algorithms have been developed. Commercial programs are available to solve linear mixed-integer optimization problems, and these programs are used extensively by city and regional planners and resource managers to solve a variety of problems.

Since nearly all engineering problems are nonlinear, it is no mystery why linear integer programming techniques are not commonly used in engineering design. Also, many of the models used in integer programming formulations are ill-suited for engineering problems. However, while integer programming techniques cannot be directly applied to the standard engineering design problem, they provide tools that can be used with nonlinear continuous optimization techniques to provide practical approaches to nonlinear discrete optimization. The typical engineering design problem has several characteristics that make these approaches plausible:

1) Discrete engineering problems can usually be analyzed as continuous. Although a 3.5-ply lamina cannot be constructed, the values of the objective and the constraints can be calculated in the analysis. Most problems that arise from an available size requirement fit into this category.

2) Engineering problems generally have a small number of design variables. "Small" here is in comparison to linear operations research problems, which often have thousands of design variables.

3) Derivative information is available since it is used in almost every continuous optimization procedure. Some analysis routines, including a few finite-element codes, include analytic or semi-analytic gradients.

Sequential Linear Discrete Programming (SLDP)

The SLDP method begins with the creation of a linear mixed-integer approximate problem from the nonlinear discrete problem. Existing integer programming techniques are then used on the approximate problem directly. A series of approximations and optimizations is carried out until convergence occurs. This procedure has two very attractive features:

1) Advanced software capable for efficiently solving linear mixed-integer problems already exists. These programs are well tested and reliable.

2) For each complete mixed-integer optimization, only a single nonlinear analysis must be completed, together with a sensitivity analysis. When an extensive finite-element analysis is required, the cost of the analysis usually dominates other computational costs.

To understand the SLDP method better, first consider the Sequential Linear Programming Method (SLP) for nonlinear continuous optimization.[7] In the SLP approach, a problem is linearized and the approximate linear problem is solved using linear programming or equivalent methods. The approximate solution is then used as the seed point for a new linearization and new linear programming problem. The process is repeated until a satisfactory solution is found. This procedure was proposed in 1960 by Kelley[8] and exists as an option in the ADS program.[9]

The addition of discrete variables does not fundamentally change the sequential linear programming process. Consider again the general nonlinear discrete problem statement given by Eqs. (1-6).

Now linearize this problem about a point X^0 using a first-order Taylor series expansion. The resulting approximate problem is:

Minimize

$$F(X) \cong F(X^0) + \nabla F(X^0) \cdot \delta X \tag{7}$$

Subject to

$$g_j(X) \cong g_j(X^0) + \nabla g_j(X^0) \cdot \delta X \leq 0 \qquad j = 1,m \tag{8}$$

$$h_k(X) \cong h_k(X^0) + \nabla h_k(X^0) \cdot \delta X = 0 \qquad k = 1,l \tag{9}$$

where

$$\delta X = X - X^0 \tag{10}$$

$$X = (x_1, \ldots X_p, X_{p+1}, \ldots, X_n)^T \tag{11}$$

$$X_i^0 + \delta X_i \in (d_{i1}, d_{i2}, \ldots, d_{iq}) \qquad i = 1,p \tag{12}$$

$$X_k \leq X_k^0 + \delta X_k \leq X_k^u \qquad k = p+1,n \tag{13}$$

The approximate problem remains intractable since some of the design variables are discrete and noninteger. A transformation provides the remedy.[10] Since each X_i can take on one of several values $(d_{i1}, d_{i2}, \ldots, d_{iq})$, the following construct can be made:

$$X_i = z_{i1} d_{i1} + z_{i2} d_{i2} + \ldots + z_{iq} d_{iq} \tag{14}$$

with

$$z_{i1} + z_{i2} + \ldots + z_{iq} = 1 \tag{15}$$

and

$$z_{ij} = 0 \text{ or } 1 \qquad j = 1,q \tag{16}$$

This expression for X_i can be inserted into the linear approximate problem, yielding

Minimize

$$F(X) \cong F(X^0) + \sum_{i=1}^{p} \frac{\partial F}{\partial X_i} \left[\left(\sum_{j=1}^{q} z_{ij} d_{ij} \right) - X_i^0 \right] + \sum_{k=p+1}^{n} \frac{\partial F}{\partial X_k} (X_k - X_k^0) \tag{17}$$

Subject to

$$g_j(X) \cong g_j(X^0) = \sum_{i=1}^{p} \frac{\partial g_j}{\partial X_i} \left[\left(\sum_{m=1}^{q} z_{im} d_{im} \right) - X_i^0 \right] + \sum_{k=p+1}^{n} \frac{\partial g_j}{\partial X_k} (X_k - X_k^0) \leq 0 \tag{18}$$

[Note: $h_k(X)$ will have the same form as $g_j(X)$.]

$$\sum_{l=1}^{q} z_{il} = 1 \qquad i = 1,p \qquad (19)$$

$$z_{ij} = 0 \text{ or } 1 \qquad \text{all } i \text{ and } j \qquad (20)$$

$$X_k^L < X_k^0 + \Sigma X_k < X_k^u \qquad k = p+1,n \qquad (21)$$

Both the z_{ij} and X_k are design variables.

The problem is now in a form suitable for mixed-integer programming techniques. Several decisions can now be made.

The linearization point X^0 needs to be chosen. An assumption is made that the discrete optimum is in the vicinity of the continuous optimum. Therefore, a logical first step is to optimize the true problem with all design variables treated as continuous using a nonlinear optimization program. It is not essential to find the precise optimum to the continuous problem, as will be seen from the examples. The purpose of the continuous optimization is simply to reach the vicinity of the discrete solution and to provide a good starting point. The continuous optimum is thus used for the initial value of X^0. This choice, however, fails to guarantee a feasible discrete solution.

Instead, in the present method, the X_i from the continuous optimum are rounded (usually in a direction away from constraint violation) to provide an initially feasible discrete solution for X^0. For all future linearizations, the result of the preceding approximate optimization is used for X^0.

In the general approximate formulation, the entire set of available discrete values for each variable is included. An intelligent option is the use of a truncated set of discrete possibilities. A truncated set has several justifications:

1) It is assumed the present solution X^0 is in the vicinity of the discrete optimum. Discrete values distant from the values in X^0 are unlikely members of the optimum set.

2) The Taylor series approximation is most valid in the vicinity of the present X^0. Move limits of some sort should be imposed to prevent extreme constraint violation.

3) Inclusion of more possible discrete values in the approximate problem does not necessarily improve the final solution to the true problem.

4) A truncated set reduces the size of the approximate problem and hence improves efficiency.

In this study, each discrete design variable is allowed three possibilities: its present value, the adjacent higher value, and the adjacent lower value. A two-variable example is presented to demonstrate the approach:

Minimize

$$F(X) = X_1^2 + X_2^2 \qquad (22)$$

Subject to

$$g(X) = \frac{1}{X_1} + \frac{1}{X_2} - 2 \le 0 \qquad (23)$$

$$X_1 \in (0.3,\ 0.7,\ 0.9,\ 1.2,\ 1.5,\ 1.8) \qquad (24)$$

$$X_2 \in (0.4,\ 0.8,\ 1.1,\ 1.4,\ 1.6) \qquad (25)$$

$$X^0 = (1.2,\ 1.1) \qquad (26)$$

$$F(X^0) = 2.65 \qquad (27)$$

$$\nabla F(X^0) = (2.4,\ 2.2)^T \qquad (28)$$

$$g(X^0) = -0.26 \qquad (29)$$

$$\nabla g(X^0) = (-0.69\ -0.83)^T \qquad (30)$$

Now

$$X_1 = Z_{11}(0.8) + Z_{12}(1.2) + Z_{13}(1.5) \qquad (31)$$

$$X_2 = Z_{21}(0.8) + Z_{22}(1.1) + Z_{23}(1.4) \qquad (32)$$

$$\delta X_1 = Z_{11}(0.8-1.2) + Z_{12}(1.2-1.2) + Z_{13}(1.5-1.2) \qquad (33)$$

$$\delta X_2 = Z_{21}(0.8-1.1) + Z_{22}(1.1-1.1) + Z_{23}(1.4-1.1) \qquad (34)$$

$$F \cong 2.65 + \begin{Bmatrix} 2.4 \\ 2.2 \end{Bmatrix}^T \begin{Bmatrix} -0.4Z_{11} + 0.3Z_{13} \\ -0.3Z_{21} + 0.3Z_{23} \end{Bmatrix} \qquad (35)$$

$$g \cong -0.26 + \begin{Bmatrix} -0.69 \\ -0.83 \end{Bmatrix}^T \begin{Bmatrix} -0.4Z_{11} + 0.3Z_{13} \\ -0.3Z_{21} + 0.3Z_{23} \end{Bmatrix} \qquad (36)$$

The approximate problem becomes:

Minimize

$$2.65 - 0.96Z_{11} + 0.72Z_{13} - 0.66Z_{21} + 0.66Z_{23} \qquad (37)$$

Subject to

$$-0.26 + 0.28Z_{11} + 0.21Z_{13} + 0.25Z_{21} - 0.25Z_{23} \le 0 \qquad (38)$$

$$Z_{11} + Z_{12} + Z_{13} = 1 \qquad (39)$$

$$Z_{21} + Z_{22} + Z_{23} = 1 \qquad (40)$$

$$Z_{ij} = 0 \text{ or } 1 \qquad (41)$$

Since there are only nine possible solutions, they can be enumerated. The minimum feasible solution is:

$$\begin{matrix} Z_{11} = 0 & Z_{12} = 1 & Z_{13} = 0 \\ Z_{21} = 1 & Z_{22} = 0 & Z_{23} = 0 \end{matrix} \qquad (42)$$

Translated into variables of the true problem:

$$X_1 = 1.2 \qquad (43)$$

$$X_2 = 0.8 \qquad (44)$$

$$F(X) = 2.08 \qquad (45)$$

$$g(X) = 0.08 \qquad (46)$$

The process is then repeated with $(1.2,\ 0.8)^T$ as the new X^0.

The chosen truncated set makes for an interesting comparison to rounding. Consider the basic problem of rounding once a continuous optimum is found and assume that gradients of the objective and constraints with respect to the design variables are available. With the gradients it is possible to discern which design variables have the most effect on the objective function and constraints. This derivative information should then provide a basis for an intelligent decision as to which design variables should be rounded up and which should be rounded down. A valid description of a single iteration of the SLDP procedure is that it finds the best rounding combination based upon first-order derivative information. In order to provide a series of viable discrete designs, the SLDP procedure does not actually round but updates an already rounded solution. The analogy, however, is conceptually valid, and the search for an intelligent rounding scheme provided the motivation for the SLDP procedure.

SLDP—The Program

Following a continuous optimization by ADS, the linear mixed-integer approximation is carried out. The coefficients of the variables in the approximate problem are used to create a Mathematical Programming System (MPS) (IBM TM) format file. This format is the standard used by most commercial linear optimization programs.

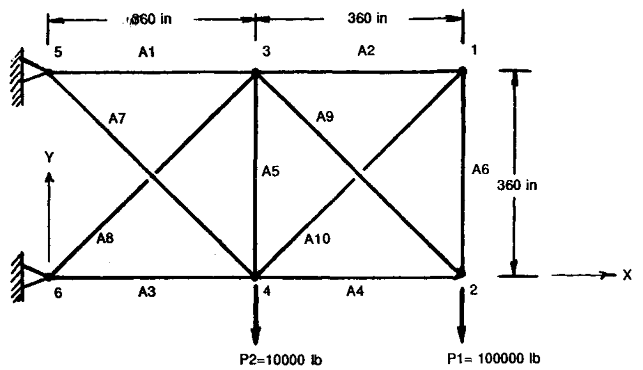

Fig. 1 10-bar truss: $P1 = 10,000$ lb; $P2 = 10,000$ lb.

Table 1 10 bar truss data

Material	Aluminum
Young's modulus	1.0×10^7
Specific weight	0.1 lbm/in.^{3}
Allowable stress	$\pm 25,000$ psi
Minimum area	0.1 in.^{2}
Allowed displacement	
All nodes (Y direction)	± 2.0 in.

Table 2 Allowable discrete values for test problems

		Cross-sectional area, in.^{2}
10-bar truss		0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 39.5, 40.0
25-bar truss		0.1, 0.4, 0.7, 1.1, 1.5, 2.0, 2.5, 3.0 3.5, 4.0, 4.5
	Number of plies	Fiber orientation
Composite :		
:	All integers	Continuous
:		

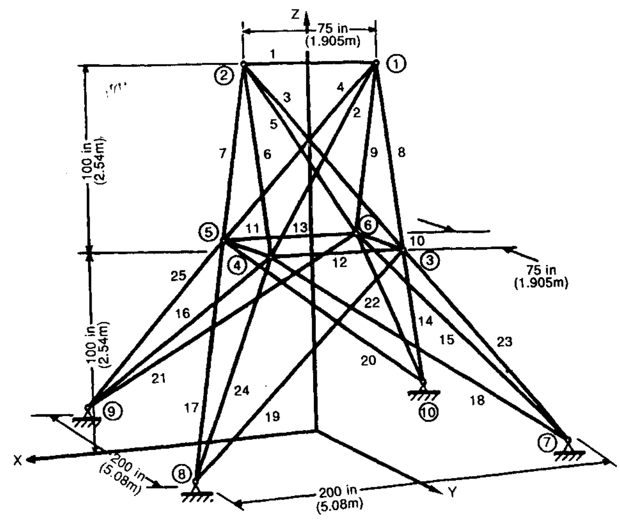

Fig. 2 25-bar truss.

Table 3 Results for 10-bar truss (SLDP method)

Design variable no.	Continuous optimization solution	Rounded upward solution	Discrete optimization solution
1	29.94	30.00	31.50
2	0.10	0.10	0.10
3	24.55	25.00	23.00
4	14.43	14.50	15.50
5	0.10	0.10	0.10
6	0.10	0.50	0.50
7	8.47	8.50	7.50
8	21.18	21.50	20.50
9	21.05	21.50	21.00
10	0.12	0.50	0.10
Mass (lbm)	5079.0	5174.6	5045.8
Maximum constraint component			0.004
Number of function evaluations			7

The linear optimization program LINDO[11] is used to solve the approximate linear integer programming problem. LINDO uses a branch and bound procedure with revised simplex. The values returned by LINDO are converted into the values they represent in the true problem. The new X vector is used for a new linearization. An MPS file is created and a new linear integer optimization is performed. The sequence is repeated until no improved designs are created.

The SLDP method cannot guarantee convergence to a discrete optimum. Because of the limits of the linear approximation, the discrete optimum may not be accessible by a series of moves from discrete solutions. The error introduced by the linearization depends on the spacing between the discrete values (which determined the size of δX and the nonlinearity of the problem at the points chosen).

Typically, the SLDP process ends with the repetition of a linear mixed-integer solution. It is also possible for the series of discrete solutions to oscillate between two or several discrete values. The number of iterations is usually small enough to identify such a pattern and terminate execution. Choosing the best solution from the set is then a simple procedure.

Numerical Examples

The results of several test problems are presented to demonstrate the discrete variable optimization capability of the method discussed. The test cases represent classical optimization examples. The examples are not intended to represent the scope of application of the method.

Some reference solutions exist for the 10- and 25-bar truss examples,[12,13] A comparison is difficult because of the importance of internal parameters used in numerical approaches. In numerical optimization methods, feasibility is determined by a constraint tolerance. The constraint tolerance is the maximum value allowed for a constraint to be considered unviolated. This tolerance may vary from program to program and can affect results. In this paper, the ADS default tolerance of 0.005 was used. The maximum constraint value is listed in the result tables for each case.

10-Bar Truss

The first example is the classical 10-bar truss (Fig. 1). The truss material information is given in Table 1. The design variable are the cross-sectional area of the 10 bars. Each member is allowed a maximum stress of $\pm 25,000$ psi and a minimum are of 0.1 in.^{2} The vertical displacement of all joints is limited to ± 2.0 in. The available discrete values of the variables are given in Table 2.

The SADT[14] finite-element analysis program is used for analysis. This includes analytic gradient information for the constraints and the objective function.

Table 3 shows the results for continuous optimization, rounded-up solution, and discrete optimization solution for the SLDP method. It is apparent that the continuous optimization result is inferior to reference solutions. The nonlinear numerical optimization procedure used is capable of achieving better results than those produced. A crude convergence criterion was used here to provide only a near-optimum solution to the continuous problem. This was used to test the discrete optimization method's abilities given an inaccurate continuous optimization starting position.

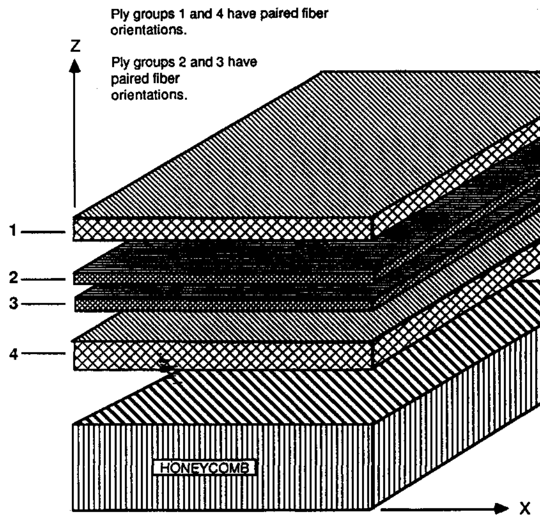

Fig. 3 Symmetry about X plane.

Table 4 Iteration history data (SLDP method)

Analysis no.	10-bar truss weight	25-bar truss weight	Composite panel weight
Round up	5081.5	570.7	2.54
1	5052.0	539.5	2.49
2	5063.0	544.0[a]	2.59[b]
3	5045.8[a]	544.0	
4	5059.9		
5	5059.9		
6	5059.9		
7	5059.9		

[a]Optimum. [b]No convergence.

Table 5 25-bar truss

Material	Aluminum
Young's modulus	1.0×10^7 psi
Specific mass	0.1 lbm/in.^{3}
Minimum area	0.1 in.^{2}

Table 6 Allowable stresses 25-bar truss

Members	Stress limited (psi) Tension	Compression
1, 10–13	40,000	−35,092
2–5	40,000	−11,590
6–9	40,000	−17,305
14–17	40,000	− 6,759
18–21	40,000	− 6,959
22–25	40,000	−11,082

Table 7 Allowable displacements 25-bar truss

Nodes 1 and 2 (X, Y, Z, directions)	±0.35 in.

The SLDP method succeeded in producing a lower mass solution than the continuous solution and the result listed in Ref. 8 for the discrete solution. The method found the solution listed in Table 3 in four iterations after the continuous solution had been reached. Since the constraint tolerances are much smaller in the linear subproblem, four more iterations took place before repetition occurred. Interestingly, all four solutions had the same objective value, clearly displaying the potential for nonunique solutions to a discrete problem. The iteration histories are shown in Table 4.

Table 8 Loading information 25-bar truss

Load case	Node	Component (lbf) X	Y	Z
1	1	1,000	10,000	−5,000
	2	0	10,000	−5,000
	3	500	0	0
	6	500	0	0
2	5	0	20,000	−5,000
	6	0	−20,000	−5,000

Table 9 Results for 25-bar truss (SLDP method)

Design variable no.	Continuous optimization solution	Rounded upward solution	Discrete optimization solution
1	0.03	0.10	0.10
2	0.09	0.10	0.10
3	3.29	3.50	3.50
4	1.37	1.50	1.50
5	1.76	1.80	1.50
6	2.00	2.00	2.00
7	0.02	0.10	0.10
8	3.67	4.00	3.50
Mass (lbm)	534.7	570.7	539.5
Maximum constraint component			0.005
Number of iterations			4

Table 10 Composite panel data

Fiber properties	
Modulus in fiber direction	26.25 Mpsi
Inplane modulus transverse to fiber direction	1490 ksi
Poisson's ratio for load in fiber direction	0.28
Inplane shear modulus	1040 ksi
Material density	0.056 lb/in.^{3}
Tensile strength in fiber direction	217.5 ksi
Compressive strength in fiber direction	217.5 ksi
Tensile strength in transverse direction	5.8 ksi
Compressive strength in transverse direction	
Shear strength	
Maximum allowable strain	
Thickness of one ply in group	

Loading information	1	2	3
Membrane load X direction	5700	11,400	11,400
Membrane load Y direction	5700	5700	−5700
Membrane load X-Y direction	5700	0	0
Bending load X plane	5000	2000	0
Bending load Y plane	5000	4000	0
Bending coupling load	0	0	1000

25-Bar Truss

The second example is another classical optimization test case—the 25-bar truss (Fig. 2). Due to symmetry, only eight different member sizes are allowed, and hence eight design variables exist. Truss data is given in Table 5, and stress and displacement constraints are shown in Table 6 and 7. Two separate loading conditions are imposed, as shown in Table 8.

For this example, unevenly and widely spaced discrete values were chosen for the possibility set, as listed in Table 2. The iteration history is given in Table 4, and the optimum member sizes are given in Table 9. The continuous optimization solution agreed with reference solutions.

The method found the discrete solution after one iteration. This solution had a 0.005 constraint violation, which was the limit allowed by the program. A second solution was produced

Table 11 Results for composite panel (SLDP method)

Design variable no.	Continuous optimization solution	Rounded upward solution	Discrete optimization solution
1	12.09	13.0	12.0
2	32.83	33.0	33.0
3	14.22	15.0	14.0
4	11.27	12.0	12.0
5	9.28	9.28	9.28
6	55.05	55.05	56.67
Mass (lbm)	2.47	2.54	2.49

with no constraint violations; then repetition occurred. The solution is significantly better than the rounded-up solution.

Composite Panel

The third example is a symmetric composite honeycomb panel (Fig. 3). The outer skins are composed of four ply groups. Each ply group has a variable number of lamina. From the four ply groups, two fiber orientations are allowed to vary. The other two are determined by symmetry. Various planar and bending loads are applied. Minimization of weight is the objective, subject to the Tsai-Wu strain failure criteria. Panel data is shown in Table 10.

The composite panel is used to test the mixed-discrete optimization capability of the method. The lamina are considered integer variables while the fiber orientations are allowed to vary continuously.

The SLDP method experienced difficulty with the continuous design variables. The difficulty resulted in part from the lack of a rigorous move limit scheme. More robust move limit techniques should remedy the problem. However, it should be noted that this is a fiber orientation problem known to be inherently ill-conditioned. The iterations were terminated due to lack of convergence. The results for the composite panel are shown in Tables 4 and 11.

The composite panel problem proved to have a very "flat" design space. The sensitivities of the design variables in the vicinity of the optimum were very small, which made the search for a unique optimum difficult.

Summary and Conclusions

The results of this study show that integer programming methods and software provide a viable means to solving nonlinear discrete optimization problems. The SLDP method succeeded in converging to discrete soutions that could not be anticipated with conventional methods such as rounding. The present method found these solutions at a small computational cost using existing software for continuous and linear mixed-integer optimization.

The scope of the test cases used in this paper was not sufficient to truly examine the general nonlinear discrete problem. Additional study is needed to better understand the effects of discrete value spacing, number of design variables, and problem specifics of the nonlinear discrete optimization process. In addition, an improved framework for the creation and testing of discrete optimization methods needs to be developed. In the creation of a nonlinear discrete optimization framework, the integration of nonlinear continuous optimization and integer programming techniques provides a broad, practical, and coherent foundation.

Acknowledgments

This research reported here was funded by NASA Grant NAG-1567.

References

[1]Schmit, L. A., "Structural Design by Systematic Synthesis," *Proceedings of the 2nd Conference on Electronic Computation,* American Society of Civil Engineers, New York, 1960, pp. 105-122.

[2]Schmit, L. A. and Fleury, C., "Discrete-Continuous Variable Structural Synthesis Using Dual Methods," *AIAA Journal,* Vol. 18, 1980, pp. 1515-1524.

[3]Mesquita, L. and Kamat, M. P., "Optimization of Stiffened Laminated Composite Plates with Frequency Constraints," *Engineering Optimization,* Vol. 11, 1987, pp. 77-88.

[4]Dakin, R. J., "A Tree Search Algorithm for Mixed Integer Programming Problems," *Computer Journal,* Vol. 8, 1965, pp. 250-255.

[5]Olsen, G. R., "Nonlinear Optimization with Discrete Design Variables," M.S. Thesis, Department of Mechanical and Environmental Engineering, University of California, Santa Barbara, CA, 1986.

[6]Siddall, J. N., *Optimal Engineering Design,* Marcel Decker, New York, 1982, pp. 223-266.

[7]Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design—with Applications,* McGraw-Hill, 1984, pp. 153-201.

[8]Kelley, J. E., "The Cutting Plane Method for Solving Convex Programs," *Journal of SIAM,* Vol. 8, 1960, pp. 702-712.

[9]Vanderplaats, G. N. and Sugimoto, H., "A General-Purpose Optimization Program for Engineering Design," *International Journal of Computers and Structures,* Vol. 24, No. 1, 1986, pp. 13-21.

[10]Salkin, H. M., *Integer Programming,* Addison-Wesley, Philippines, 1975, pp. 3-5.

[11]Schrage, L., "LINDO—Linear Interactive Discrete Optimizer," University of Chicago, Chicago, IL, 1983.

[12]Schmit, L. A. and Fleury, C., "An Improved Analysis/Synthesis Capability Based on Dual Methods—ACCESS 3," *Proceedings of the AIAA/ASME/ASCE/AHS 20th Structures, Structural Dynamics and Materials Conference,* AIAA, New York, April 1979, pp. 23-50.

[13]Haftka, R. T. and Kamat, M. P., *Elements of Structural Optimization,* Martinus Nijhoff, Dordrecht, the Netherlands, 1985, pp. 236-239.

[14]Vanderplaats, G. N., *SADT, User's Manual,* University of California, Santa Barbara, CA, 1984.