

SOLUTION OF MIXED-DISCRETE STRUCTURAL OPTIMIZATION PROBLEMS WITH A NEW SEQUENTIAL LINEARIZATION ALGORITHM

M. BREMICKER, P. Y. PAPALAMBROS† and H. T. LOH‡

Department of Mechanical Engineering and Applied Mechanics, The University of Michigan,
 2250 G. G. Brown, Ann Arbor, MI 48109-2125, U.S.A.

‡Department of Mechanical and Production Engineering, National University of Singapore, Singapore

(Received 27 September 1989)

Abstract—In practical structural optimization problems it is often desirable to obtain solutions where all or some of the design variables take their values from a given set of discrete values. As structural optimization problems typically include large models that are expensive to compute, one of the major demands for optimization algorithms is that the number of structural evaluations (i.e. calculations of deformations and stresses) that are needed during the iterative optimization process is as small as possible. In this article an algorithm is developed that meets this requirement, while finding global solutions for the mixed-discrete problem. The method is based on a combination of the well established branch and bound method with a sequential linearization procedure. Branch and bound is applied within a subproblem that is based on a linearization of the original problem. After a brief literature survey the method is described, followed by some comments on its algorithmic implementation. The algorithm is then applied to several structural optimization problems of different type and size to demonstrate its efficiency. All results are compared with solutions obtained by branch and bound.

INTRODUCTION

Structural optimization problems (SOP) can generally be represented as nonlinear mathematical programming problems (NLP) of the following standard form:

$$\begin{aligned}
 &\text{minimize} && f(\mathbf{x}) \\
 &\text{subject to} && h_j(\mathbf{x}) = 0 \quad j = 1 \dots q \\
 & && g_j(\mathbf{x}) \leq 0 \quad j = (q + 1) \dots p \\
 & && l_i < x_i < u_i \quad i = i \dots n,
 \end{aligned} \tag{1}$$

where f , h_j and g_j are scalar objective, equality and inequality constraint functions, respectively, and u_i and l_i are upper and lower bounds for the design variables \mathbf{x} (see Table 1). Most optimization algorithms are applied under the assumption that f , h_j and g_j are C^2 -continuous functions on the n -dimensional Euclidean space \mathcal{R}^n , and that the design variables \mathbf{x} can hold arbitrary real values. This is true for a large variety of structural optimization tasks (e.g. most of the shape and geometry optimization problems). In many situations, however, it is desirable to obtain solutions where all or some of the design variables take their values from a given set of discrete values. Examples are cross-sectional values of standardized profiles, standard thicknesses of sheets, available dimensions of machine parts, number of layers or orientations of fiber angles in a fiber composite structure and material selection problems.

A variable that can take only integer values is called an *integer variable*, while one that can take its value only from a set of discrete ones is called a *discrete variable*. Since integer variables are also discrete the term 'discrete variables' will refer to both integer and discrete variables, and 'integer variables' to integer variables only. When some variables are discrete and some are continuous the problem is a *mixed-discrete* one. The presence of discrete variables makes the solution of NLP problems substantially more difficult.

Equality constraints cannot usually be satisfied when the variables are discrete, or if satisfied they can be eliminated by a variable reduction and elimination procedure, at least theoretically. For example, the equation $x_1^2 x_2 + x_2^3 - 21 = 0$ has no solution when x_1 and x_2 must be integers. Hence, research has focused mostly on problems with only inequality constraints. This restriction is not crucial for SOP as the usually occurring equality constraints are the system equations of the mechanical structure which are generally given implicitly by numerical procedures, such as finite element analysis, and can be solved separately. Thus, the term *Mixed-Discrete Nonlinear Programming* (MDNLP) problem is used here for mathematical models stated as:

$$\begin{aligned}
 &\text{minimize} && f(\mathbf{x}) \\
 &\text{subject to} && g_j(\mathbf{x}) \leq 0 \quad j = 1 \dots p \\
 & && l_i \leq x_i \leq u_i \quad i = 1 \dots n \\
 & && \mathbf{x} \in \mathcal{X} \subset \mathcal{D}^d \times \mathcal{R}^{(n-d)} \\
 & && f: \mathcal{X} \rightarrow \mathcal{R} \quad g_j: \mathcal{X} \rightarrow \mathcal{R}.
 \end{aligned} \tag{2}$$

† To whom correspondence should be addressed.

Table 1. Notation and abbreviations

Notation	
\mathcal{S}	script uppercase denotes a set or vector space
A	bold uppercase denotes a matrix
x	bold lowercase denotes a point or a vector
\mathbf{x}_1	bold lowercase with subscript refers to a particular point (or vector)
x_i	normal lowercase with subscript refers to a component of a vector
Abbreviations	
SOP	structural optimization problem
NLP	nonlinear programming
MDNLP	mixed-discrete nonlinear programming
SLP	sequential linear programming
SQP	sequential quadratic programming
BBM	branch and bound method

In this model, \mathbf{x} is a design point in an n -dimensional design 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. We assume that f and g_i are C^2 -continuous functions and that the MDNLP has a continuous solution when relaxing the discreteness requirement.

LITERATURE SURVEY

One of the main approaches to treat MDNLP problems is the branch and bound method (BBM). Its first use for mixed-discrete linear programming is attributed to Land and Doig [2], whose algorithm was subsequently modified by Dakin [3]. In branch and bound procedures subproblems are created by partitioning the feasible domain to force the integer variables to take integer values. The partitioning is done perpendicular to the axis of the integer chosen for the branching. Reinschmidt [4] used the BBM for the plastic design of frames by posing the problem as an integer linear programming problem. Gupta and Ravindran [5] applied Dakin's method to nonlinear mixed-integer problems, using a generalized reduced gradient method to solve the nonlinear continuous subproblem at each node. Sandgren [6] used basically the same approach as Gupta, but included cases with equality constraints having 0-1 variables. He applied the method to solve various design problems.

The main advantage of branch and bound is that a proper fathoming rule guarantees finding the global minimum. The bounding process assumes that at a discrete minimizer the objective function value is a lower bound on the discrete minimum of any other node in the tree, an assumption valid for objectives that are at least pseudo-convex and constraints that are convex. For non-convex problems branch and bound cannot guarantee finding the global minimizer, since it may end up fathoming nodes that should not have been

fathomed. However, branch and bound usually finds a good solution for these cases. A major drawback is that a large number of nodes are created during the branching process and this number tends to increase exponentially with the number of discrete design variables. As structural optimization problems often include large structural models that are very costly to be solved in terms of cpu time, the BBM is becoming computationally impractical when the number of discrete design variables exceeds three or four.

This fact motivated many researchers to develop heuristic methods to get approximate discrete or mixed-discrete solutions starting from the continuous solutions. One common practice is to round off the continuous solution to the 'next' feasible discrete point using some heuristic rules (see, e.g. Ringertz [7]). This approach generally leads to non-optimal solutions, and it may even indicate that no feasible point exists in the neighborhood of the continuous solution. As will be seen later, the mixed-discrete solution of an SOP can be relatively 'far away' from the continuous solution and may even result in unexpected alterations of the structural configuration.

Another approach is to treat discrete requirements as explicit constraints and construct an objective function penalizing deviations from discrete values. Davydov and Sigal [8] devised a number of penalty functions for 0-1 problems and convex problems with regularly spaced discrete intervals. Gisvold and Moe [9] offered a similar method for discrete variables with arbitrarily spaced intervals. Shin *et al.* [10] recently applied a penalty method to the optimization of truss structures. There are two main problems which arise in employing this method, both difficult to overcome. As in the continuous penalty method, penalty parameters are difficult to set *a priori*, with different parameters leading to different results. Unless one has some knowledge about the discrete solution, it is difficult to decide whether the result obtained is optimal or the parameters need adjustment to continue iterating. The more serious difficulty is that adding a penalty for moving away from a non-discrete point creates a local optimum for each discrete point. This makes finding global optima with a continuous method even more difficult.

Schmidt and Fleury [11] used a Lagrangian relaxation approach, originally developed for linear problems by Geoffrion [12], to solve nonlinear discrete SOP. The problem is replaced by a sequence of convex and separable approximate subproblems that are solved using Lagrangian relaxation and minimizing the dual function with a subgradient method. This strategy does not guarantee finding global solutions as the subproblems, being convex in the continuous case, are becoming non-convex when discrete variables are introduced. The method was applied by Ringertz [7] to solve six structural

optimization problems, and the results were compared with those using branch and bound. It turns out that the minima obtained for these non-linear structural problems are very close or equal to those given by branch and bound, but the design variable values obtained are different. For a more general survey of mixed-discrete optimization methods, see [1, 13].

DESCRIPTION OF THE ALGORITHM

Branch and bound is a reliable method for finding global mixed-discrete solutions, but it generally needs far too high a number of objective function and constraint evaluations to be used for practical structural problems. On the other hand, it is well understood in continuous nonlinear programming that approximation methods, such as sequential linear programming (SLP), the method of moving asymptotes or sequential quadratic programming (SQP), provide highly efficient tools for solving NLP formulations of structural problems. The new idea here is to apply the concept of sequential approximation to MDNLP by combining an SLP technique with a branch and bound algorithm. John *et al.* [18] have recently proposed a similar coupling, but their algorithm is set up in a different way and has apparently been developed specifically for purely discrete truss sizing problems.

The global iteration scheme of the proposed algorithm can be outlined as follows.

- Step 1: Choose a starting point \mathbf{x}_0 , let the iteration index $k = 0$.
- Step 2: Given an iteration point x_k , compute f , g , ∇f and ∇g .
- Step 3: Generate a linear subproblem L_k .
- Step 4: Solve L_k by a branch and bound method, using a Simplex algorithm as linear solver.
- Step 5: Check convergence criteria: if satisfied, stop; if not, continue with step 6.
- Step 6: Select a new point x_{k+1} , let $k = k + 1$ and go on with step 2.

The main advantage of this approach is that the BBM, expensive in terms of function evaluations, is applied to a linear subproblem, where the function evaluations are computationally 'cheap'. This implies that only one structural analysis and one gradient calculation must be performed in each iteration k . As the BBM solves the linear subproblems exactly the convergence of the algorithm basically depends on the approximation quality of the linear subproblem L_k .

It will now be explained in more detail how the steps 3–6 must be implemented in order to obtain an effective algorithm.

Step 3: The mixed-discrete linear subproblem at a point \mathbf{x}_k is:

$$\begin{aligned} & \text{minimize} && \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) \\ & \text{subject to} && \mathbf{g}(\mathbf{x}_0) + \nabla \mathbf{g}(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) \leq 0 \\ & && \mathbf{x} \in \mathcal{X} \subset \mathcal{D}^d \times \mathcal{R}^{(n-d)}. \end{aligned} \quad (3)$$

As in the case of the continuous SLP method, decreasing step bounds or move limits have to be incorporated to achieve convergence [15], so constraints are added to the model:

$$\mathbf{l}_k \leq \mathbf{x}_k \leq \mathbf{u}_k$$

with

$$\begin{aligned} l_{ik} &= \max \left\{ x_{ik} + \frac{\alpha}{2}(u_i - l_i), l_i \right\}, \quad i = 1 \dots n \\ u_{ik} &= \min \left\{ x_{ik} + \frac{\alpha}{2}(u_i - l_i), u_i \right\}, \quad i = 1 \dots n. \end{aligned}$$

There are many possibilities for choosing α . Computational experience suggests that the following recursive formula be applied whenever α is changed (see step 6):

$$\alpha_{\text{new}} = \frac{\alpha_{\text{old}}}{1 + \alpha_{\text{old}}}. \quad (4)$$

This formula generates the geometric series $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \dots$

Step 4: The branch and bound method of Dakin [3] is applied in a slightly modified form to solve the linear subproblem [3]. The method was extended to handle arbitrarily distributed discrete design variables. Following a recommendation of Gupta [5] the branching was done from the variable with the most fractional part. For selecting the expansion node the node with the lowest objective function value was used.

Step 5: Convergence is considered to be achieved when

$$\|\mathbf{x}_k - \mathbf{x}_{\text{best}}\| < \delta, \quad (5)$$

where \mathbf{x}_{best} denotes the ϵ -feasible point with the lowest objective function value that has been found in previous iterations, the so-called incumbent. A point \mathbf{x}_k is said to be ϵ -feasible for an MDNLP model, if \mathbf{x}_k satisfies the discreteness requirement and if

$$\sum_1^p \Delta g_j \leq \epsilon, \quad \Delta g_j = g_j, \quad g_j > 0. \quad (6)$$

With δ being a small value, eqn (5) implies that two different subproblems yield approximately the same mixed-discrete solution and so no better feasible point can be found. Note that for MDNLP there are

not sufficient optimality conditions, such as the Karush–Kuhn–Tucker equations for convex NLP.

Step 6: A point \mathbf{x}_k is accepted as an incumbent, when either

$$f_k \leq f_{\text{best}}, \text{ if } \mathbf{x}_k \text{ is } \epsilon\text{-feasible} \quad (7a)$$

or

$$\left\{ \sum_1^p \Delta g_j \right\}_k \leq \left\{ \sum_1^p \Delta g_j \right\}_{k-1}, \text{ if } \mathbf{x}_{k-1} \text{ is } \epsilon\text{-feasible.} \quad (7b)$$

The parameter ϵ essentially enlarges the feasible domain of an NLP model by an amount ϵ for each constraint. The above given acceptance rule mandates that for a given ϵ a new point is accepted if it improves ϵ -feasibility of an ϵ -infeasible incumbent, or if it strictly improves the objective without sacrificing ϵ -feasibility. Numerical test studies have shown that it is beneficial for the convergence speed of the algorithm to adapt the feasibility acceptance rule (6) during the optimization process, i.e. to introduce an intermediate feasibility parameter ϵ_i starting with a relatively large value and reducing it whenever eqn (7a) is fulfilled, until a final user-defined feasibility parameter ϵ is achieved. If neither (7a) nor (7b) is satisfied \mathbf{x}_k is rejected and a new linear subproblem is constructed at \mathbf{x}_{k-1} with a reduced step bound size according to eqn (4).

The above described algorithm will usually terminate quickly if the problem is purely discrete, as the number of possible solutions is finite. However, this is not necessarily true for the mixed–discrete case

when some of the design variables are continuous. As in the continuous SLP algorithm, cycling can occur between upper and lower step bounds, and the convergence speed will mainly depend on the step bound reduction scheme. This effect is amplified by the fact that the discreteness requirement leads to a smaller number of active inequality constraints compared to the purely continuous problem. The convergence can however be stabilized by modifying step 4 as follows:

Step 4a: Solve L_k by a branch and bound method, using a Simplex algorithm as linear solver.

Step 4b: Solve the following continuous subproblem for \mathbf{x}_c with fixed \mathbf{x}_{dk} :

$$\begin{aligned} & \min_{\mathbf{x}_c} f(\mathbf{x}_{dk}, \mathbf{x}_c) \\ & \text{subject to} \\ & g(\mathbf{x}_{dk}, \mathbf{x}_c) \leq 0 \\ & \mathbf{x}_c \in \mathcal{R}^{(n-d)}, \end{aligned} \quad (8)$$

where \mathbf{x}_{dk} is the discrete design vector part, whereas \mathbf{x}_c contains the continuous design variables. Any continuous NLP-code can be used for solving eqn (8). In order to save cpu time it is advisable to skip step 4b when the sum of the constraint violations resulting from step 4a exceeds a certain degree, since then eqn (8) may have no feasible solution.

Figure 1 shows a global flow chart of the algorithm. A more detailed description of the

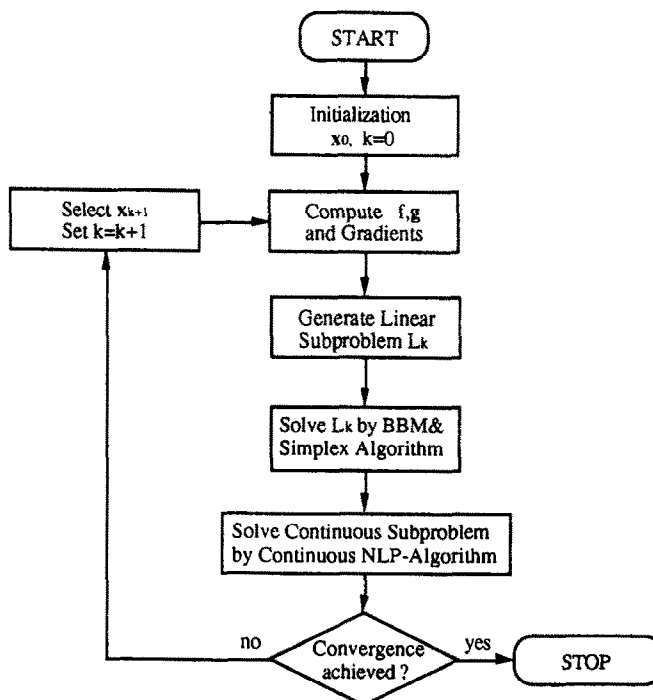


Fig. 1. Global flow chart of the mixed–discrete SLP method.

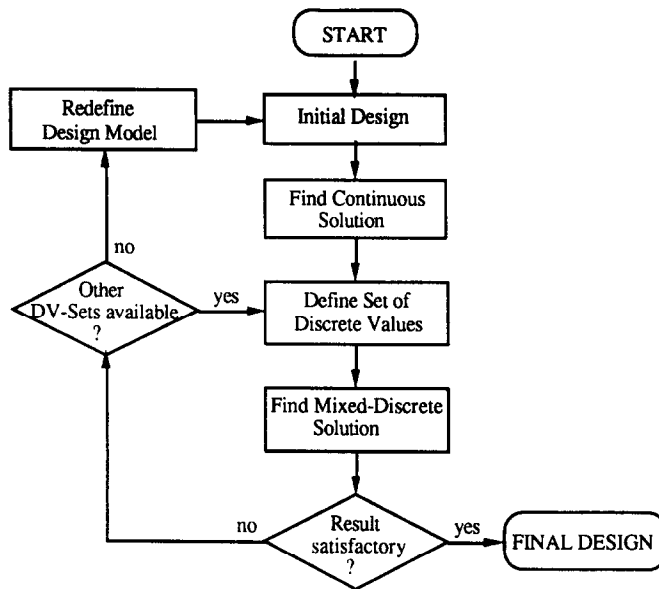


Fig. 2. Interactive mixed-discrete design process.

algorithm including convergence proofs is given in [1, 13]. Elaborate numerical studies are reported in [14].

INTEGRATION INTO AN OPTIMIZATION PROCEDURE

The algorithm has been programmed in Fortran77, and the resulting code, named MINSLIP, has been implemented into a structural optimization package, called SAPOP [15]. The latter provides several structural analysis methods, optimization algorithms and standardized pre- and post-processors (i.e. for sizing and shape optimization). The discrete values can be either generated equidistantly, or taken from a user-provided file. In general, it is of interest for the designer to compute first the continuous solution for the problem in order to obtain a lower bound on the objective function. This solution provides at the same time a good starting point for MINSLIP. Furthermore, the designer will typically want to compare the solutions obtained with different sets of discrete variables, e.g. from different catalogs of manufacturers. These ideas lead to an interactive implementation of the algorithm, as shown in Fig. 2.

Table 2. Criteria to compare numerical results

f_{opt}	optimal objective function value
x_{iopt}	optimal value for design variable i
n_f	number of function evaluations needed to achieve the optimal solution
n_g	number of gradient evaluations needed to achieve the optimal solution
n_{sub}	number of equivalent evaluations to solve the continuous subproblem, eqn (8), when using finite differences to compute the required gradients
n_{tot}	total number of equivalent evaluations when using finite differences to compute the required gradients, $n_{tot} = n_f + n \cdot n_g + n_{sub}$
t_{cpu}	cpu time needed on an Apollo 4000 computer

EXAMPLES

Four structural optimization problems serve to demonstrate the convergence behavior of MINSLIP. The first two examples, well known in the literature, are used to show principally the convergence characteristics of the algorithm. The last two examples exhibit the efficiency of the SLP approach in handling larger problems, when the number of design variables is high and/or the structural analyses are computationally expensive. Table 2 summarizes the criteria utilized in this section in order to compare the numerical results.

In addition to computations with MINSLIP, the problems are solved with the program BBSQP, a branch and bound method where a SQP method (program VMCWD by Powell [16]) is applied to solve the NLP problems created by the branching procedure. The same SQP algorithm is used to determine the solutions of the continuous subproblem, (8), with MINSLIP. The BBSQP program is straightforward; more details on it can be found in [1].

Example 1: Three-bar truss

The weight of the three-bar truss structure shown in Fig. 3 needs to be minimized, subject to stress

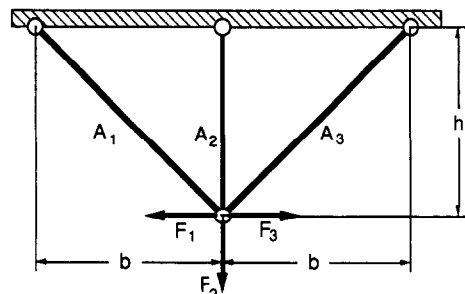


Fig. 3. Three-bar truss.

Table 3. Solutions and numerical comparison for Example 1, purely discrete case

	Continuous (SQP)	Discretization D ₁		Discretization D ₂	
		BBSQP†	MINSLIP†	BBSQP†	MINSLIP†
f_{opt} (kg)	14.648	14.697	14.697	14.703	14.703
$x_{1\text{opt}}$ (mm ²)	557.7	570.0	570.0	582.0	582.0
$x_{2\text{opt}}$ (mm ²)	288.5	260.0	260.0	227.0	227.0
$x_{3\text{opt}}$ (mm ²)	557.7	570.0	570.0	582.0	582.0
n_f	16	109 (124)	8 (13)	60 (75)	5 (6)
n_g	15	108 (123)	7 (12)	59 (74)	4 (5)
n_{tot}	61	433 (493)	29 (49)	237 (297)	17 (21)

† Numbers in parentheses are achieved when starting at \mathbf{x}_0 instead of \mathbf{x}_{con} .

constraints in the three members. The structure is loaded by three forces under two loading conditions:

$$\text{Load case } L_1: F_1 = F_2 = 10^5 \text{N}, \quad F_3 = 0;$$

$$\text{Load case } L_2: F_1 = 0, \quad F_2 = F_3 = 10^5 \text{N}.$$

The design variables are the cross section A_i of the bars (discrete variables) and the distance b (continuous variable). The design model can thus be posed as follows:

$$\min f(\mathbf{x}) = \sum_1^3 \rho_i l_i(x_4) x_i,$$

subject to

$$g_j(\mathbf{x}) = \frac{\sigma_j(\mathbf{x})}{\sigma_{\text{fea}}} - 1 \leq 0, \quad j = 1, 2, 3(L_1), 4, 5, 6(L_2) \quad (9)$$

$$1 \text{ mm}^2 \leq x_i \leq 1000 \text{ mm}^2, \quad i = 1, 2, 3$$

$$400 \text{ mm} \leq x_4 \leq 2000 \text{ mm}$$

$$x \in \mathcal{X} \subset \mathcal{D}^3 \times \mathcal{R}^1,$$

where l_i denotes the length of the bar i , σ_i the normal stress in bar i and σ_{fea} the maximum allowable stress. The fixed design parameters are:

$$E = 2.1 \times 10^6 \text{ N/mm}^2$$

$$\rho = 7.85 \times 10^{-6} \text{ kg/mm}^3$$

$$\sigma_{\text{fea}} = 200 \text{ N/mm}^2$$

$$h = 1000 \text{ mm}.$$

Two different discretizations are defined:

D₁: Small, regular discrete steps 1, 10, 20, 30, 40 . . . 1000 (mm²).

D₂: Large, irregular discrete steps, values taken from Deutsche Industrie Norm, DIN1028, single profiles (see Table A1, Appendix).

Let us first consider the purely discrete case, i.e. the length b is fixed to $b = 1000$ mm and the design variables are the cross sections only. The initial design is $\mathbf{x}_0 = (1000 \text{ mm}^2, 1000 \text{ mm}^2, 1000 \text{ mm}^2)^T$, $f = 29.9$ kg. Table 3 summarizes the results obtained with MINSLIP and BBSQP, both starting from the continuous solution \mathbf{x}_{con} . The numbers in parentheses denote the numerical effort when starting from \mathbf{x}_0 instead of \mathbf{x}_{con} . In a second, mixed-discrete case, the length b is additionally defined as design variable x_4 , with the starting value $x_4 = 1000$ mm. The results for this model are given in Table 4.

The following conclusions can be drawn from the above reported results.

(a) MINSLIP always finds the same solution as BBSQP.

(b) In every case MINSLIP needs a considerably smaller number of function and gradient evaluations than BBSQP to achieve the optimum. The differences are more obvious in the purely discrete case than in the mixed-discrete case, as in the latter the solution of the continuous subproblem requires additional function calls (see numbers n_{sub} in Table 4).

Table 4. Solutions and numerical comparison for Example 1, mixed-discrete case

	Continuous (SQP)	Discretization D ₁		Discretization D ₂	
		BBSQP†	MINSLIP†	BBSQP†	MINSLIP†
f_{opt} (kg)	14.173	14.176	14.176	14.338	14.338
$x_{1\text{opt}}$ (mm ²)	753.74	750.0	750.0	691.0	691.0
$x_{2\text{opt}}$ (mm ²)	1.0	1.0	1.0	112.0	112.0
$x_{3\text{opt}}$ (mm ²)	753.74	750.0	750.0	691.0	691.0
$x_{4\text{opt}}$ (mm)	657.99	669.14	669.14	734.25	734.25
n_f	30	16 (45)	7 (16)	37 (66)	7 (10)
n_g	29	15 (44)	6 (15)	35 (64)	6 (9)
n_{sub}	—	—	36 (89)	—	37 (44)
n_{tot}	146	71 (221)	67 (165)	177 (321)	68 (90)

† Numbers in parentheses are achieved when starting at \mathbf{x}_0 instead of \mathbf{x}_{con} .

(c) MINSLIP terminates earlier when the discrete steps are larger (compare discretization D_1 with D_2). Figure 4 clarifies this fact. In the case of small spaces between the discrete values (as in discretization D_1) many possible solutions are lying very closely together and the algorithm cycles for some iterations between different points with small differences in the objective function values. This effect vanishes with an increasing size of spaces.

(d) One further conclusion can be made concerning the characteristics of the discrete or mixed-discrete solutions. The optimal design can be located relatively far away from the continuous solution. In Table 4, for instance, we see that due to large 'jumps' in the available cross-sectional areas of discretization D_2 , the optimal value of bar 2 is not 1 mm^2 (i.e. 'removal' of the bar), as it is for the continuous solution and discretization D_1 . This shows that adding the discreteness requirement to a continuous SOP can even result in a different topology of the optimum design.

Example 2: Ten-bar truss

Consider now the classical 10-bar truss problem shown in Fig. 5. We seek the minimum weight of this structure, subject to stress and displacement constraints. The design variables are the cross-sectional areas of the 10 members. The design model is thus:

$$\min f(\mathbf{x}) = \sum_1^{10} \rho_i l_i x_i$$

subject to

$$g_j(\mathbf{x}) = \frac{\sigma_j(\mathbf{x})}{\sigma_{\text{fea}}} - 1 \leq 0, \quad j = 1 \dots 10$$

$$g_{11}(\mathbf{x}) = \frac{\delta_p}{\delta_{p\text{fea}}} - 1 \leq 0$$

$$0.1 \text{ in}^2 \leq x_i \leq 40 \text{ in}^2, \quad i = 1 \dots 10$$

$$\mathbf{x} \in \mathcal{X} \subset \mathcal{D}^{10}, \quad (10)$$

where l_i denotes the length of bar i , σ_i the normal stress in bar i , σ_{fea} the maximum allowable stress, δ_p the vertical deflection of nodal point P , and $\delta_{p\text{fea}}$ the

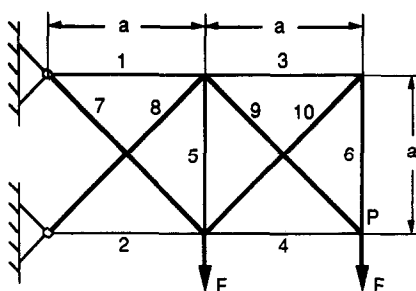


Fig. 5. Ten-bar truss.

associated maximum allowable deflection. The design parameters are:

- $E = 10,000 \text{ ksi}$
- $F = 100,000 \text{ lbf}$
- $\rho = 0.1 \text{ lb kg/in}^3$
- $\sigma_{\text{fea}} = 25 \text{ psi}$
- $\delta_{p\text{fea}} = 2 \text{ in.}$
- $a = 360 \text{ in.}$

Let us define two cases concerning the constraint definition:

- Case I: Only the stress constraints $g_1 - g_{10}$ are considered.
- Case II: Constraint g_{11} is defined additionally.

As in the first example, two different discretizations are considered:

- D_1 : Uniform discretization, $0.1, 1, 2 \dots 40 \text{ (in}^2\text{)}$
- D_2 : Discretization according to DIN1028, double angle profiles (see Table A2, Appendix).

The results are given in Tables 5 and 6. In three of the four runs identical solutions could be achieved by MINSLIP and BBSQP, the latter algorithm, however, requiring a very large number of function evaluations, as up to 700 branching nodes are generated. MINSLIP shows a better convergence in Case I than in Case II. Case I/ D_2 reveals the same effect that we observed in Example 1: some bars are not 'removed' from the structure, in contrast to the continuous solution.

In case II/ D_2 MINSLIP fails to find the global solution located by BBSQP. The reason for this is that the discretization steps are extremely large in the upper region of the discrete value table (see Table A2, Appendix) and no further progress can be achieved in the linearized problem due to large approximation errors. However, MINSLIP locates the global solution when doing an inverse variable transformation

$$z_i = \frac{1}{x_i}, \quad i = 1 \dots n \quad (11)$$

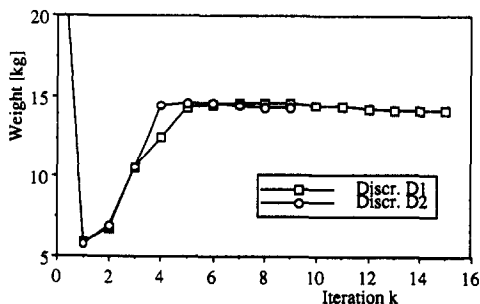


Fig. 4. Convergence history of MINSLIP.

Table 5. Solutions and numerical comparison for Example 2, Case I

	Discretization D ₁			Discretization D ₂	
	Continuous (SQP)	BBSQP	MINSLIP	BBSQP	MINSLIP
f_{opt} (lb)	1593.18	1688.30	1688.30	1706.4	1706.4
x_{1opt} (in ²)	7.9379	8.0	8.0	8.525	8.525
x_{2opt} (in ²)	0.1	0.1	0.1	0.347	0.347
x_{3opt} (in ²)	8.0621	9.0	9.0	8.525	8.525
x_{3opt} (in ²)	3.9379	4.0	4.0	3.813	3.813
x_{5opt} (in ²)	0.1	0.1	0.1	0.1	0.1
x_{6opt} (in ²)	0.1	0.1	0.1	0.347	0.347
x_{7opt} (in ²)	5.7447	6.0	6.0	5.592	5.592
x_{8opt} (in ²)	5.5690	6.0	6.0	5.592	5.592
x_{9opt} (in ²)	5.5690	6.0	6.0	5.592	5.592
x_{10opt} (in ²)	0.1	0.1	0.1	0.347	0.347
n_f	24	239	4	244	5
n_g	17	213	3	238	4
n_{tot}	194	2549	34	2624	45

Table 6. Solutions and numerical comparison for Example 2, Case II

	Discretization D ₁			Discretization D ₂	
	Continuous (SQP)	BBSQP	MINSLIP	BBSQP	MINSLIP
f_{opt} (lb)	5022.9	5051.65	5051.65	5100.32	5153.90
x_{1opt} (in ²)	30.126	30.0	30.0	28.080	33.703
x_{2opt} (in ²)	0.1	0.1	0.1	0.1	0.1
x_{3opt} (in ²)	22.931	22.0	22.0	28.080	19.180
x_{4opt} (in ²)	15.394	16.0	16.0	14.290	17.170
x_{5opt} (in ²)	0.1	0.1	0.1	0.1	0.1
x_{6opt} (in ²)	0.1	0.1	0.1	0.1	0.1
x_{7opt} (in ²)	7.242	7.0	7.0	7.192	8.525
x_{8opt} (in ²)	20.751	19.0	19.0	19.180	23.680
x_{9opt} (in ²)	21.771	22.0	22.0	23.680	19.180
x_{10opt} (in ²)	0.1	0.1	0.1	0.1	0.1
n_f	59	2556	6	630	5
n_g	33	2505	7	606	4
n_{tot}	389	27,606	76	6690	45

and using z_i as new design variables. It can be shown easily that this transformation leads to ‘less nonlinear’ stress and deflection constraints for sizing problems and hence to more precise linearizations [19].

Example 3: Three-dimensional tower

We now consider a more complex mixed-discrete problem taken from [17], where it has been treated as a continuous problem. We want to determine the minimum weight design of the 39-bar tower shown in Fig. 6, subject to stress constraints in the bars and a single deflection constraint under three loads applied at the nodes 13, 14 and 15 in the positive y -direction. The mathematical formulation of the design model is similar to eqn (10). The three bottom nodes (1, 2 and 3) have coordinates (0, 4, 0), $(-2\sqrt{3}, -2, 0)$ and $(2\sqrt{3}, -2, 0)$, while the three top nodes (13, 14 and 15) have coordinates (0, 1.12, 16), $(-0.56\sqrt{3}, -0.56, 16)$ and $(0.56\sqrt{3}, -0.56, 16)$; all coordinates in m (meters). The other coordinates are defined as design variables in the following way (see Fig. 6): x_1, x_2 and x_3 define the z -coordinates of three levels as shown in Fig. 6; $x_4 =$ the distances AS, BS, CS in the x - y -plane at the level of the nodes 4, 5 and 6; $x_5 =$ the distances AS, BS, CS in the x - y -plane at the level of the nodes 7, 8 and 9; $x_6 =$ the distances AS, BS, CS in the x - y -plane at the level of the nodes 10, 11 and 12. Moreover, we have five sizing

variables: $x_7 =$ the cross-sectional area of the elements defined by the node-pairs (1, 4), (2, 5) and (3, 6); $x_8 =$ the cross-sectional area of the elements defined by the node-pairs (4, 7), (5, 8) and (6, 9); $x_9 =$ the cross-sectional area of the elements defined by the node-pairs (7, 10), (8, 11) and (9, 12); $x_{10} =$ the cross-sectional area of the elements defined by the node-pairs (10, 13), (11, 14) and (12, 15); $x_{11} =$ the cross-sectional area of the rest of the elements.

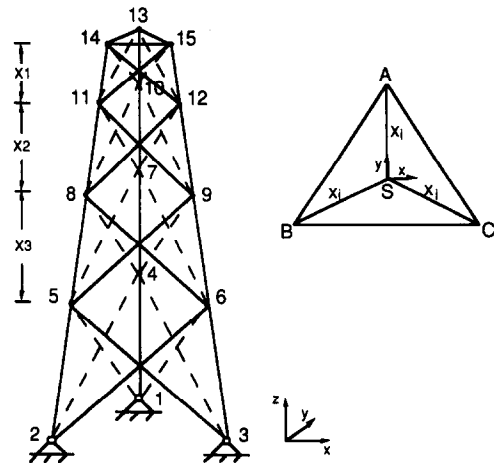


Fig. 6. Three-dimensional tower.

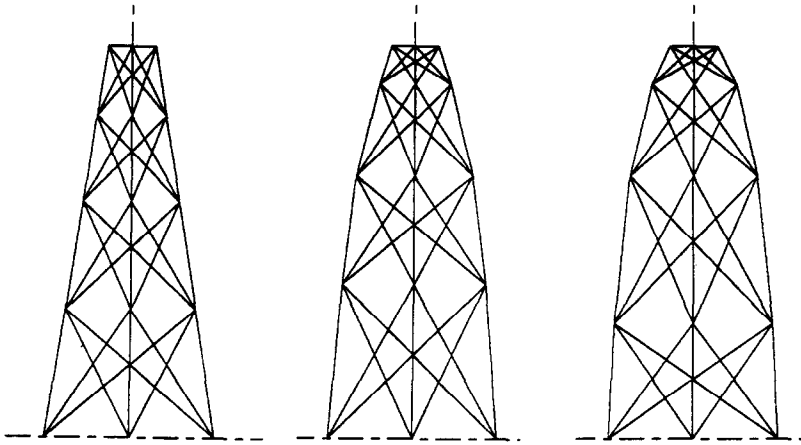


Fig. 7. Truss geometry for initial design (left), continuous solution (middle) and final mixed-discrete design (right).

Variables x_1 – x_6 are treated as continuous variables, variables x_7 – x_{11} as discrete variables, the values of the latter again taken from DIN 1028 and listed in Table A1 (Appendix). Bounds are defined for the geometry variables as listed: $500 \text{ mm} \leq x_1 \leq 4000 \text{ mm}$; $1000 \text{ mm} \leq x_2 \leq 5000 \text{ mm}$; $2000 \text{ mm} \leq x_3 \leq 6000 \text{ mm}$; $1000 \text{ mm} \leq x_i \leq 4000 \text{ mm}$, $i = 4, 5, 6$.

The design parameters are: $E = 2.1 \cdot 10^6 \text{ N/mm}^2$; $\rho = 7.85 \cdot 10^{-6} \text{ kg/mm}^3$; $\sigma_{\text{tea}} = 150 \text{ N/mm}^2$.

The displacement constraint limits the deflection of the top nodes 14 and 15 in the positive y -direction to a maximum of 3 mm. We solve this design problem in two different ways, both starting with the continuous solution:

Case I: Fix the continuous variables describing the geometry of the structure and solve for the discrete variables only.

Case II: Solve the complete mixed-discrete problem.

As a matter of fact, Case I only gives an approximate solution of the problem, but the numerical effort

should be less than that of Case II. The differences between the two solutions should be an indicator of how strongly the geometry and sizing variables are coupled. The inverse transformation scheme (10) is applied to the sizing variables x_6 – x_{11} for both BBSQP and MINSLLIP, in order to improve convergence. Figure 7 shows the geometry of the initial design, the continuous solution and the mixed-discrete solution. Table 7 summarizes the results of the computations, revealing two interesting points, as follows.

(a) The results of Cases I and II differ considerably, both in the objective function and the constraints. This indicates a relatively strong interaction between sizing and geometry variables.

(b) Compared with Case I, MINSLLIP requires a relatively high number of function calls in Case II, most of which are needed for repeatedly solving the continuous subproblem [eqn (8)] by the SQP method. Attempts to achieve the solution without executing step 4b of the algorithm fail; the method terminates with an infeasible design.

Table 7. Solutions and numerical effort for Example 3

	Case I			Case II	
	Continuous (SQP)	BBSQP	MINSLLIP	BBSQP	MINSLLIP
f_{opt} (kg)	7020.32	7571.14	7571.14	7238.79	7238.78
$x_{1\text{opt}}$ (mm)	1558.6	1558.6	1558.6	1563.3	1567.0
$x_{2\text{opt}}$ (mm)	3563.5	3563.5	3563.5	3760.9	3756.1
$x_{3\text{opt}}$ (mm)	4532.6	4532.6	4532.6	6000.0	6000.0
$x_{4\text{opt}}$ (mm)	3386.4	3386.4	3386.4	3740.9	3738.8
$x_{5\text{opt}}$ (mm)	2726.5	2726.5	2726.5	3004.8	3004.3
$x_{6\text{opt}}$ (mm)	1720.3	1720.3	1720.3	1990.6	1994.6
$x_{7\text{opt}}$ (mm ²)	1719.9	1920.0	1920.0	1920.0	1920.0
$x_{8\text{opt}}$ (mm ²)	1215.3	1230.0	1230.0	1220.0	1220.0
$x_{9\text{opt}}$ (mm ²)	736.98	870.0	870.0	691.0	691.0
$x_{10\text{opt}}$ (mm ²)	239.34	267.0	267.0	227.0	227.0
$x_{11\text{opt}}$ (mm ²)	215.67	227.0	227.0	227.0	227.0
n_f	129	125	3	1589	31
n_g	75	343	2	658	30
n_{sub}	—	—	—	—	1373
n_{tot}	954	968	20	8827	1734

Example 4: Rod panel

An example with a larger mechanical model serves to demonstrate the efficiency of the proposed SLP method. We want to minimize the structural weight of a shear panel, reinforced in both directions with welded rods that have axial stiffness only (Fig. 8), subject to stress constraints with the maximum allowable stress of $\sigma_{fea} = 200 \text{ N/mm}^2$ in sheet and rods. The structure is loaded by forces $F_1 = 49.05 \text{ kN}$ and $F_2 = 49.05 \text{ kN}$ defining two load cases L_1 and L_2 respectively. The finite element model to perform the structural analysis consists of 144 four-node plain stress elements and 109 truss elements.

The design variables are defined as follows: $x_1 =$ cross-sectional area of rods R_1 and R_5 ; $x_2 =$ cross-sectional area of rods R_2 and R_4 ; $x_3 =$ cross-sectional area of rod R_3 ; $x_4 =$ cross-sectional area of rod R_6 ; $x_5 =$ cross-sectional area of rod R_7 ; $x_6 =$ cross-sectional area of rod R_8 ; $x_7 =$ cross-sectional area of rod R_9 ; $x_8 =$ thickness of sheet areas P_1 to P_4 ; $x_9 =$ thickness of sheet areas P_5 to P_8 ; and $x_{10} =$ thickness of sheet areas P_9 to P_{12} .

The bounds for the design variables are: $10 \text{ mm}^2 < x_i < 1000 \text{ mm}^2, i = 1 \dots 7$ and $0.01 \text{ mm} < x_i < 3 \text{ mm}, i = 8 \dots 10$.

Ten stress constraints are defined for the different panel parts according to the above given variable definition. The Huber-von Mises stress criterion is applied to compute equivalent stresses for the sheet. The variable linking and constraint linking capabilities of SAPOP [15] are used to set up the model. The size of the problem is designed in such a way that we are still able to compute the solution with BBSQP for comparison within an acceptable CPU time.

The discrete values of the rods are again taken from DIN1028 (see Table A1, Appendix). The sheet parts may have the thicknesses (in mm) 0.8, 1.0, 1.25, 1.5, 1.75, 2.0, 2.5 and 3.0, according to DIN1016. The design parameters for this problem are: $E = 2.1 \cdot 10^6 \text{ N/mm}^2$; $\rho = 7.85 \cdot 10^{-6} \text{ kg/mm}^3$; $b = 1200 \text{ mm}$; $h = 600 \text{ mm}$.

Table 8 compares the results achieved by BBSQP and MINSLIP, including the required cpu time for solving the problem on an Apollo 4000 workstation. BBSQP and MINSLIP located different solutions

Table 8. Results and numerical comparison for Example 4

	Continuous (SQP)	BBSQP	MINSLIP
f_{opt} (kg)	7.452	7.582	7.582
x_{1opt} (mm ²)	295.61	308.0	308.00
x_{2opt} (mm ²)	10.00	10.00	10.00
x_{3opt} (mm ²)	622.21	691.0	691.0
x_{4opt} (mm ²)	451.49	379.0	430.0
x_{5opt} (mm ²)	306.58	308.0	308.0
x_{6opt} (mm ²)	311.08	328.0	328.0
x_{7opt} (mm ²)	314.85	379.0	328.0
x_{8opt} (mm)	1.537	1.25	1.25
x_{9opt} (mm)	0.952	1.00	1.00
x_{10opt} (mm)	0.519	0.80	0.80
n_f	24	2337	5
n_g	23	1271	4
n_{tot}	254	15047	45
t_{cpu} (hr)	0.586	41.66	0.167

with exactly the same objective function value, a fact that shows the non-convexity of the discrete problem. However, the BBM requires 200 times the computational time needed by MINSLIP, for it creates 141 branching nodes, which means that 141 NLP problems have to be solved. This result indeed shows that the SLP approach considerably enlarges the application range of the BBM method.

One remaining question is: What is the maximum number of design variables that can be practically handled by MINSLIP? To come closer to that limit the rod panel problem is formulated with 21 design variables, i.e. the thickness of each sheet section and the cross-sectional area of each rod represent one design variable. MINSLIP needs 12 iterations to solve this problem, a number that is comparable to the 10-variable problem. However, the number of nodes created by BBM within the linearized problem rises up to about 4000, compared with 200 in the 10-variable case. Already about 60% of the total computational time is needed to solve the linearized subproblem (3) by BBM, compared with ca 10% in the 10-variable problem. This result indicates that the application limit of the method is determined by the problem size dependent performance of the BBM, and should lie somewhere in the region of 20 to 30 discrete design variables.

CONCLUSIONS

In this paper a method for solving mixed-discrete structural optimization problems based on a sequential linearization technique was presented. It applies the well-established branch and bound method within the linearized subproblems and can therefore save function evaluations that can be extremely computationally intensive. The method was used to solve several structural optimization problems, where the available discrete values for the design variables are either generated equidistantly or selected from standard tables, the latter case being the more usual one in engineering practice.

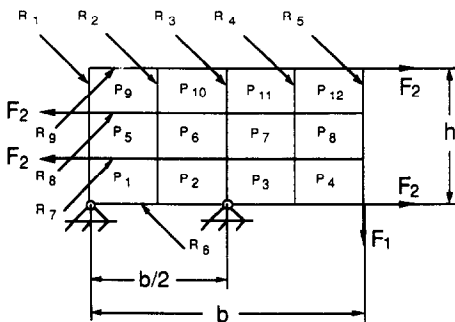


Fig. 8. Rod panel.

The new method generally locates solutions that are equal or very close to those computed by the pure branch and bound method, requiring, however, a significantly smaller number of function evaluations. The latter fact makes the method applicable to larger structural optimization problems than would be practical for branch and bound. The application range of the method is for medium size design models containing not more than, say, 20 to 30 discrete design variables. The method shows the best convergence behavior for purely discrete problems. In the mixed-discrete case it tends to require a relatively high number of function evaluations for solving the continuous subproblem, the total number of evaluations, however, being still much smaller than the number required by branch and bound.

Acknowledgements—This research was supported in part by NSF Grants DMC 85-14721 and DMC 86-11916 at the University of Michigan. The third author was supported also by a Fellowship from the National University of Singapore. This support is gratefully acknowledged.

REFERENCES

1. H. T. Loh, 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 (1988).
2. A. M. Land and A. G. Doig, An automatic method of solving discrete programming problems. *Econometrica* **28**, 497-520 (1960).
3. R. J. Dakin, A tree-search algorithm for mixed integer programming problems. *Comput. Jnl* **8**, 250-255 (1965).
4. K. Reinschmidt, Discrete structural optimization. *ASCE, J. struct. Div.* **94**, (ST1), 133-156 (1971).
5. O. K. Gupta and A. Ravindran, Nonlinear integer programming and discrete optimization. *ASME J. Mechs, Transmiss. Automn Des.* **105**, 160-164 (1983).
6. E. Sandgren, Nonlinear integer and discrete programming in mechanical design. In *Proc. 1988 ASME Design Technology Conference*, Kissimmee, Florida, Sept 25th, pp. 95-105 (1988).
7. U. T. Ringertz, On methods for discrete structural optimization. *Engng Optimizn* **13**, 47-64 (1988).
8. E. G. Davydov and I. Sigal, Application of penalty function method in integer programming problems. *Engng Cybernetics* **10**, 21-24 (1972).
9. K. M. Gisvold and J. Moe, A method for nonlinear mixed integer programming and its application to design problems. *ASME J. Engng Ind.* **94**, 353-364 (1972).
10. D. K. Shin, Z. Gurdal and O. H. Griffin Jr, A penalty approach for nonlinear optimization with discrete design variables. In *Discretization Methods and Structural Optimization, Proc. GAMM-Seminar*, Siegen, F.R.G., Oct. 5-7 1988, pp. 326-334 (1989).
11. L. A. Schmidt and C. Fleury, Discrete-continuous variable structural synthesis using dual methods. *AIAA Jnl* **18**, 1515-1524 (1980).
12. A. M. Geoffrion, Lagrangian relaxation for integer programming. *Mathematical Programming Study* **2**, pp. 82-114. North-Holland, Amsterdam (1974).

13. H. T. Loh and P. Y. Papalambros, A sequential linearization approach for solving mixed-discrete nonlinear design optimization problems. *Trans. ASME, J. Mech. Des.* (to appear).
14. H. T. Loh and P. Y. Papalambros, Computational implementation and tests of a sequential linearization algorithm for mixed-discrete nonlinear design optimization. *Trans. ASME, J. Mech. Des.* (to appear).
15. H. Eschenauer, P. U. Post and M. Bremicker, Einsatz der Optimierungsprozedur SAPOP zur Auslegung von Bauteilkomponenten. *Bauingenieur* **63**, 515-526 (1988).
16. M. J. D. Powell, VMCWD: a FORTRAN subroutine for constrained optimization. Rep. DANTP 1982/NA4. University of Cambridge (1982).
17. A. J. Morris (Editor), *Foundations of Structural Optimization: A Unified Approach*. John Wiley, Chichester (1982).
18. K. V. John, C. V. Ramakrishnan and K. G. Sharma, Optimum design of trusses from available selections—use of sequential linear programming with branch and bound algorithm. *Engng Optimizn* **13**, 119-145 (1988).
19. L. A. Schmidt and H. Miura, Approximation concepts for efficient structural synthesis. NASA CR-2552 (1976).

APPENDIX: DISCRETE VALUE TABLES

Table A1. Available cross sections for Examples 1, 3 and 4, single angle profiles, values taken from Deutsche Industrienorm (DIN) 1028

i	d_i (mm ²)	i	d_i (mm ²)	i	d_i (mm ²)
1	112.0	13	569.0	25	1230.0
2	142.0	14	582.0	26	1510.0
3	174.0	15	656.0	27	1550.0
4	185.0	16	691.0	28	1920.0
5	227.0	17	870.0	29	2120.0
6	267.0	18	903.0	30	2270.0
7	308.0	19	935.0	31	2320.0
8	328.0	20	940.0		
9	349.0	21	1010.0		
10	379.0	22	1150.0		
11	430.0	23	1190.0		
12	480.0	24	1220.0		

Table A2. Available cross sections for Example 2, double angle profiles, values based on DIN1028

i	d_i (in ²)	i	d_i (in ²)	i	d_i (in ²)
1	0.347	13	3.131	25	17.170
2	0.440	14	3.565	26	19.180
3	0.539	15	3.813	27	23.680
4	0.954	16	4.805	28	28.080
5	1.081	17	5.952	29	33.700
6	1.174	18	6.572		
7	1.333	19	7.192		
8	1.488	20	8.525		
9	1.764	21	9.300		
10	2.142	22	10.850		
11	2.697	23	13.330		
12	2.800	24	14.290		