AIAA JOURNAL
Vol. 33, No. 9, September 1995

Methods for Finding Feasible Points in Constrained Optimization

Ossama A. Elwakeil* and Jasbir S. Arora†
University of Iowa, Iowa City, Iowa 52242

Methods for finding feasible points are needed in some numerical optimization algorithms to initiate the search for a local minimum. Other methods perform more efficiently if a nearly feasible starting point is used. In addition, a feasible solution may be acceptable as the final design in some practical applications. Therefore, constraint correction methods are quite useful. Such methods based on penalty functions and primal approach are described and discussed. To gain insight into the methods a two-variable problem is used to analyze the methods. Several structural design problems are used to study the numerical behavior of the methods and compare their performance. It is concluded that, in general, the primal methods are more efficient than the penalty function methods. Also, no method has been found that is guaranteed to find a feasible point for general constrained nonlinear problems starting from an arbitrary point. In case a method fails, random points should be used to restart the procedure to search for a feasible point.

I. Introduction

THE problem of finding feasible solutions for general constrained nonlinear optimization problems is addressed in this paper. Some optimization algorithms require a feasible point to initiate the search for a minimizer whereas others perform better when started from a nearly feasible point. Therefore, if the starting point specified by the user has constraint violations, some procedure must first be used to correct them. The constraint correction process can be useful in its own right because in some applications a feasible design is all that is needed. Thus, methods for finding feasible points can be useful in practical applications.

Purpose of this paper is to describe, analyze, and evaluate various methods for correcting constraint violations and finding feasible designs. This is done by studying the available methods and implementing them for their numerical evaluation. The methods are first analyzed using a simple two-variable problem whose analytical solution is available. Then, they are evaluated using larger scale structural design problems.

The problem of finding a feasible point is defined as follows.

Problem P: Find a vector x of n variables such that x is in the constraint set S defined as

$$S = \{x \mid g_i(x) = 0, \quad i = 1, \ldots, p \qquad g_i(x) \le 0, \quad i = p+1, \ldots, m\} \tag{1}$$

where p is the number of equality constraints and m is the total number of constraints. Note that the explicit bounds on variables x_i are assumed to be included in $g_i(x) \le 0$. In numerical calculations, however, they are treated appropriately to take advantage of their special form.

Next, the penalty methods are presented in Sec. II and the primal methods in Sec. III. Analysis of the methods is presented in Sec. IV using a two-variable problem. In Sec. V, larger scale structural design problems are used to evaluate numerical performance of the methods. In Sec. VI, summary and conclusions are presented.

II. Penalty Methods

In these methods, a penalty function is defined using the constraints. This function is required to monotonically decrease at each iteration by means of a minimization technique (sequential quadratic programming method is used in the present work). In this section we present five penalty functions for the constraint correction problem.

A. Interior Penalty Method for Inequality Constraints

This method is for inequality constraints only and is based on the interior penalty approach of sequential unconstrained minimization described in Ref. 1 and other references cited there. In the method, two index sets are defined at each iteration based on the constraint activity. The set N contains indices of all active or violated constraints and the set K contains indices of all of the strictly inactive constraints. The penalty function to be minimized is defined as

$$\Phi_1 = \sum_{i \in N} g_i(x) + r \sum_{j \in K} -1/g_j(x) \tag{2}$$

where $r > 0$ is a continuously decreasing positive penalty parameter. The first part of the function consists of active and violated constraints and the second part inverse of the inactive constraints. The basic strategy of the method is to keep the design in the interior of the inactive constraints by constructing walls near them. The function is minimized using an unconstrained minimization technique where r is continuously decreased for two reasons: to balance the increase in the value of Φ_1 due to an increase in the number of inactive constraints in the second part, and to emphasize minimization of constraint violations in the first part. If, at any iteration, one or more constraints in the set N become inactive, they are transferred to the set K. Once transferred to the set K, they will not return back to the set N. If, at any iteration, the set N is empty (or if $\Phi_1 < 0$), then the feasible point is obtained and the algorithm is terminated. If optimum of Φ_1 is reached and there are still some violations, then a feasible point cannot be found starting from $x^{(0)}$. If a feasible point can be reached as $r \to 0$, then the minimum of Φ_1 approaches zero. On the other hand, if a feasible point cannot be found, then minimum of Φ_1 is positive.

A drawback of the method is that it can handle only inequality constraints. Also a numerical difficulty with the method occurs when a constraint is inactive with values close to zero, which causes singularity in the second part of Φ_1. To overcome this difficulty, an ε-satisfied constraint strategy is followed. That is, a constraint is moved to the set K only if it is satisfied by a predetermined scalar $\varepsilon > 0$ (machine dependent); e.g., when $g_i + \varepsilon \le 0$.

B. Quadratic Penalty Method

This method is based on the exterior penalty approach of sequential unconstrained minimization techniques which can handle both equality and inequality constraints. The basic idea is to minimize a positive penalty function consisting of two parts, defined as follows:

$$\Phi_2 = \frac{1}{2}\sum_{i=1}^{p}[g_i(x)]^2 + \frac{1}{2}\sum_{i=p+1}^{m}[g_i(x) + |g_i(x)|]^2 \tag{3}$$

Received March 23, 1994; revision received Dec. 8, 1994; accepted for publication Dec. 8, 1994.

*Graduate Research Assistant, Optimal Design Laboratory, College of Engineering.

†Professor, Civil & Environmental Engineering, and Mechanical Engineering, Optimal Design Laboratory. Member AIAA.

The function Φ_2 is slightly different form of the quadratic loss function given in Ref. 2. Since there is no objective function involved here, no penalty parameter is needed. Note also that, unlike Φ_1, this function includes only violated constraints. If all constraints are satisfied, then Φ_2 has a zero value, otherwise there are some violations. The iterative minimization process is stopped once a feasible point is found; otherwise it is continued until the minimum of Φ_2 is reached. If the design is still infeasible, a different starting point has to be chosen.

C. Exponential Function Methods

A method that uses an exponential penalty function to find a feasible point of a set of nonlinear inequalities is given in Ref. 3 as

$$\Phi_3 = \frac{1}{r}\sum_{i=p+1}^{m}[\exp(rg_i(\boldsymbol{x})) - 1] \tag{4}$$

The function penalizes violated constraints and rewards inactive constraints slightly. The solution $\boldsymbol{x}^*$ with a function value Φ_3^* (note that a superscript $*$ implies optimum value) will fall into one of the three cases[3]: 1) $\boldsymbol{x}^*$ is feasible, 2) $\Phi_3^* > 0$, the minimum point of Φ_3^* is infeasible, or 3) $\Phi_3^* \le 0$ and $\boldsymbol{x}^*$ is infeasible which can occur even if a different value of r leads to a feasible point.

In the following algorithm,[3] if condition 3 is reached, r is increased until either of case 1 or case 2 is encountered. 1) Choose a value for the penalty parameter $r > 0$ and an initial point $\boldsymbol{x}^{(0)}$. 2) Minimize Φ_3 starting from $\boldsymbol{x}^{(0)}$ until either the minimum is reached or a feasible point is found. 3) At the minimum of Φ_3, if $\Phi_3^* > 0$, then stop; a feasible point cannot be found starting from $\boldsymbol{x}^{(0)}$. Otherwise if $\Phi_3^* \le 0$ then increase r and restart from the same initial point (unless r has reached a specified upper limit $r_{\max}$).

A variation of Φ_3 that gives different weights for constraints[4] is defined as

$$\Phi_4 = \frac{1}{r}\sum_{i=p+1}^{m} v_i[\exp(rg_i(\boldsymbol{x})) - 1] \tag{5}$$

where v_i are the weights or multipliers updated at the beginning of each iteration as

$$v_i^{(k+1)} = v_i^{(k)}\exp(rg_i(\boldsymbol{x})); \qquad i = p+1, m \tag{6}$$

An algorithm suitable for this function has the following six steps. 1) Choose initial values of v_i, the parameter $r > 0$ and $\boldsymbol{x}^{(0)}$, and set the iteration counter $k = 1$. 2) Perform one iteration of unconstrained minimization. 3) If current design $\boldsymbol{x}^{(k)}$ is feasible, then stop. Else, if $\boldsymbol{x}^{(k)}$ is a minimum of Φ_4, then go to step 5, otherwise continue. 4) Set $k = k+1$, update $v_i, i = p+1, m$ according to Eq. (6) and go to step 2. 5) If $\Phi_4^* > 0$, then stop (a feasible point cannot be found starting from $\boldsymbol{x}^{(0)}$), otherwise continue. 6) Increase r, and go to step 1.

The update rule given in Eq. (6) is derived by comparing the necessary conditions for optimality of Φ_4 with those of the augmented Lagrangian method using the same function. A numerical difficulty with the method occurs due to the use of exponential function with the increasing parameter r. If a constraint $g_i(\boldsymbol{x})$ has large violation at any iteration, the function could blow up. Thus, it is important to use some safeguards for this situation. An advantage of the penalty functions using exponentials is that they have better differentiability properties than the logarithmic ones. In addition they are defined on feasible as well as infeasible domains.[5]

D. Logarithmic-Exponential Function Method

Several forms of the logarithmic-exponential function have been used in the context of solving inequality constrained problems. In this paper we use one such popular function to find feasible points. The function, given in Refs. 6 and 7, consists of the logarithm of the summation of exponential of the inequalities as

$$\Phi_5 = \frac{1}{r}\ln\left[\sum_{i=p+1}^{m}\exp(rg_i(\boldsymbol{x}))\right] \tag{7}$$

The algorithm given for Φ_3 is suitable for Φ_5 but with different stopping criterion. Here Φ_5 does not become negative. Instead, the algorithm is stopped if Φ_5^* is larger than $\ln(m - p)$.

Another form of this functional (called the KS functional), in the context of control design, was given in Ref. 8. Both the KS and Φ_5 functionals have the property

$$\max(g_i) \le \Phi_5 \le \max(g_i) + (1/r)\ln(m - p) \tag{8}$$

where $(m - p)$ is the number of inequality constraints. Therefore, Φ_5 follows and envelopes the maximum constraint. When r is very large, the gradient of Φ_5 will also follow gradient of the maximum constraint. This results in a discontinuous gradient because the critical constraint can shift suddenly. A smaller r reduces such a problem but the gradient will be away from that of the maximum constraint. Therefore, a compromise value of r has to be chosen.

III. Primal Methods

In contrast to the penalty methods, the primal methods treat the constraints directly. Three such methods are summarized here.

A. Constraint Correction Subproblem

This procedure is a minor modification of the constrained steepest descent method.[2] A quadratic programming (QP) subproblem is defined that seeks the shortest path to the feasible domain, as follows:

$$\text{minimize } 0.5\,\boldsymbol{d}^T\boldsymbol{d} \text{ subject to } \boldsymbol{N}^T\boldsymbol{d} = \boldsymbol{e} \text{ and } \boldsymbol{A}^T\boldsymbol{d} \le \boldsymbol{b} \tag{9}$$

where $\boldsymbol{d}$ is the search direction, $\boldsymbol{N}$ and $\boldsymbol{A}$ are gradient matrices of equality and inequality constraints, respectively, at the current design, and $\boldsymbol{e}$ and $\boldsymbol{b}$ are evaluated using the equality and inequality constraints as $e_i = -g_i$, and $b_i = -g_i$. Note that a potential set strategy can be used here, so all of the constraints need not be considered at each iteration. Any QP solver can be used to solve the problem. In an iterative process, this problem is redefined after each iteration. Since only linear information is used, a few iterations are usually needed to achieve feasibility.

B. Constraint Correction with Limit on Cost Increase

In the previous method, no consideration is given to the cost function in the constraint correction process. The process may give a feasible point but the cost function may become large. Therefore, it is useful to impose a limit on the cost function increase. This can be done quite easily by adding the following constraint on the foregoing QP subproblem[2]:

$$\boldsymbol{c}^T\boldsymbol{d} \le \Delta \tag{10}$$

where $\boldsymbol{c}$ is gradient of the cost function and $\Delta \ge 0$ is the limit on increase in the cost. If $\Delta = 0$ is specified, we seek constraint correction at no increase in the linearized cost function.[2] On the other hand, if $\Delta = \infty$ is specified, it reduces to the method A. Note that this modified QP subproblem can be infeasible which does not imply that the original problem is infeasible.

C. Min-Max Approach

In this approach, the constraint correction problem is posed as a min-max problem:

$$\min\max\{|g_i|, i = 1, p; \quad g_i, i = p+1, m\} \tag{11}$$

This min-max problem can be transformed to the standard nonlinear programming problem by introducing an artificial variable U as follows:

$$\begin{aligned} &\text{minimize } U \text{ subject to } |g_i(\boldsymbol{x})| \le U, \quad i = 1, \ldots, p \\ &\qquad g_i(\boldsymbol{x}) \le U, \quad i = p+1, \ldots, m \end{aligned} \tag{12}$$

Note that U actually represents the maximum constraint violation. If a feasible point exists, then $U = 0$ is a solution to the problem; otherwise U cannot be driven to zero.

It is interesting to note that the min-max problem defined in Eq. (11) can also be transformed to the logarithmic penalty function Φ_5 defined in Eq. (7).[5]

IV. Analysis of the Methods

The penalty functions Φ_1–Φ_5, and primal methods A, B, and C are implemented to test their performance and gain insight into their numerical behavior. The numerical tests are performed on an HP workstation series 425e. The following two-variable problem is considered to analyze the methods and gain insights.

Find x to satisfy $x_1, x_2 \geq 0$, and

$$g_1 \equiv \tfrac{1}{6}x_1 + \tfrac{1}{2}x_2 - 1.0 \leq 0 \tag{13}$$

$$g_2 \equiv \tfrac{1}{2}x_1 + \tfrac{1}{5}x_2 - 1.0 \leq 0 \tag{14}$$

$$g_3 \equiv \tfrac{4}{7}x_1 + \tfrac{6}{7}x_2 - \tfrac{2}{7}x_1^3 - \tfrac{4}{7}x_2^2 + 1.0 \leq 0 \tag{15}$$

For the primal methods, the cost function expression is also needed which is given as $2x_1 + 3x_2 - x_1^3 - 2x_2^2$. The example problem was attempted from three different starting points, namely (2, 2) with violation of 40%, (5, 4) with violation of 23%, and (4, 6) with violation of 266%. The problem can be solved easily using the graphical procedure, see Fig. 1. The only feasible region for the problem is a small triangular neighborhood of (2, 0). In the implementation, any resulting point close to that region is considered an acceptable solution to the problem.

The penalty methods were tested with parameter values ranging between 2 and 100. None of the methods found a feasible point starting from all three points. However, most of the methods were able to find a feasible point when started from one of the points. The methods differ in their sensitivity to starting point and the penalty parameter. In the following paragraphs, starting and ending points for the methods are given (refer to Fig. 1) and some observations are summarized.

1) Interior penalty function Φ_1: for $r = 5$: $(2, 2) \to F, (5, 4) \to F, (4, 6) \to S$; for $r = 20$: $(2, 2) \to R, (5, 4) \to F, (4, 6) \to S$. The method is unstable with respect to the initial value of the penalty parameter. It behaved better with a lower value of penalty parameter when started from (2, 2) and with a higher value from (4, 6). The method always converged from (5, 4). When started from (4, 6), the performance was very sensitive to the starting penalty parameter value. It found a feasible point for r_0 close to 70.

2) Quadratic penalty function Φ_2: for $r = 5$: $(2, 2) \to T, (5, 4) \to S, (4, 6) \to S$; for $r = 20$: $(2, 2) \to T, (5, 4) \to S, (4, 6) \to S$. Numerical behavior with this function is quite stable and the method converges to a minimum of the Φ_2, which may or may not be feasible.

3) Exponential function Φ_3: for $r = 5$: $(2, 2) \to F, (5, 4) \to S, (4, 6) \to F$; for $r = 20$: $(2, 2) \to F, (5, 4) \to S, (4, 6) \to F$. This method converges to a solution point in most cases. Better results were obtained when the penalty parameter between 3 and 20 was used. Note that the penalty parameter is kept fixed during the minimization process. The method was trapped in an infeasible area only when started from (5, 4) for all values of r.

4) The function Φ_4: for $r = 5$: $(2, 2) \to R, (5, 4) \to R, (4, 6) \to F$; for $r = 20$: $(2, 2) \to R, (5, 4) \to R, (4, 6) \to F$. This function is fairly stable but not robust. A numerical overflow occurs with $r \geq 50$ for most cases. It found a feasible point starting only from (4, 6) with $3 \leq r \leq 50$. The other two points led to the infeasible domain.

5) The logarithmic function Φ_5: for $r = 5$: $(2, 2) \to F, (5, 4) \to F, (4, 6) \to F$; for $r = 20$: $(2, 2) \to F, (5, 4) \to S, (4, 6) \to S$. This function behaves in an unstable but interesting manner. It was possible to find a feasible point starting from any of the three points for some range of the penalty parameter. Furthermore, for penalty $r \geq 4$, the point (4, 6) always led to a feasible point. But the points (2, 2) and (5, 4) needed a smaller value to reach feasible points.

6) Primal method A: $(2, 2) \to T, (5, 4) \to F, (4, 6) \to S$; primal method B: $(2, 2) \to T, (5, 4) \to F, (4, 6) \to S$; and primal method C: $(2, 2) \to T, (5, 4) \to S, (4, 6) \to F$. Primal methods A and B behave similarly whereas the min-max method behaves differently; it ends at different point than that of methods A and B when started from points (5, 4) and (4, 6). Primal methods are more stable than penalty methods and in most cases require less function evaluations and CPU time.

Behavior of different penalty functions can be better understood by observing their three-dimensional graphs; two typical graphs of the penalty functions are shown in Figs. 2 and 3. It can be seen from the graphs that the contours are formed according to the shape of the constraints. Some areas have very high peaks (depending on constraint values and gradients) and no deep valleys exist. If a

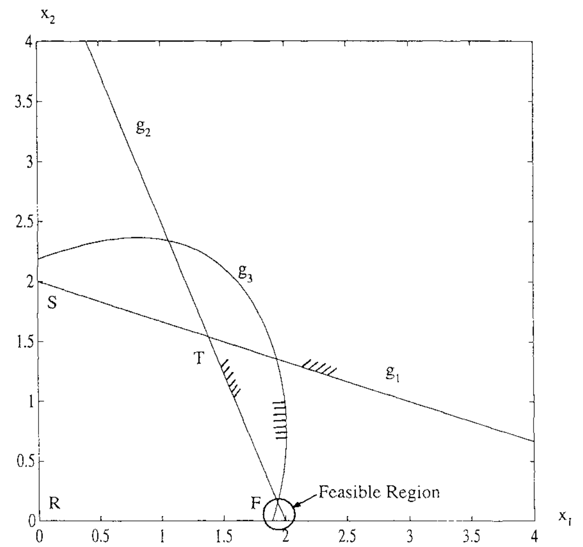

Fig. 1 Graphical representation for two-variable problem; hatched region is infeasible.

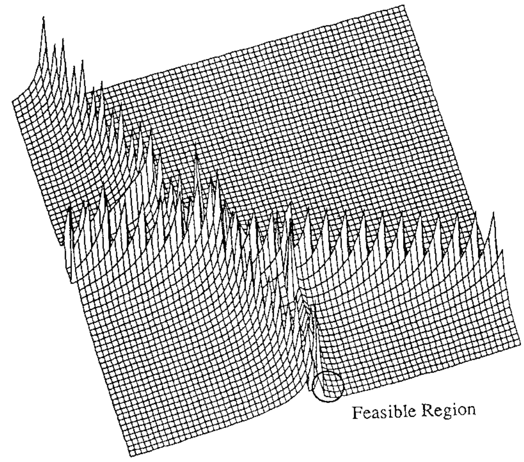

Fig. 2 Three-dimensional graph of Φ_1 for two-variable problem with $r = 5$.

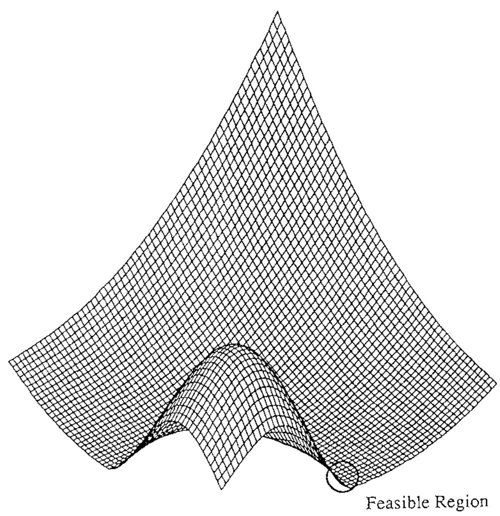

Fig. 3 Typical three-dimensional graph of Φ_2–Φ_5 for two-variable problem with $r = 5$.

method is started from a top-of-a-hill then it is difficult to predict the search trajectory. Further, if a method is started from a flat region and the search trajectory faces a hill, the next step tries to go around the hill, which in most cases leads to a trap. It is noted here that the steepness of the contours depends on the penalty parameter. It is possible based on results from different starting points to determine a domain or domains from which a particular method will find a feasible point.

V. Numerical Evaluation of the Methods for Structural Problems

To evaluate the methods on larger scale problems, nine structural design problems are considered. These consist of 10-bar, 47-bar, 200-bar trusses, one-bay two-story frame, and two-bay six-story frame subjected to multiple loading conditions. There are constraints on stress, displacement, member buckling, and fundamental natural frequency. The problem descriptions are summarized in Table 1. These problems are solved starting from a variety of infeasible points and a 5–100 range for the penalty parameter r. They have a feasible domain and are well behaved, so most methods discussed can find a feasible point. To compare performance of the methods, the following data have been collected: initial and final violations, number of function evaluations, cost function at the feasible or the best solution, and CPU time (seconds). The best results with the penalty and the primal methods are summarized in Tables 2 and 3, respectively. The following observations are made based on the results.

1) The function Φ_1 is unstable with different values of initial penalty value r_0 for all of the nine structural design problems. However, each problem can be solved using a particular range of r_0 values.

2) All of the other functions are able to find a feasible or near feasible point for all the nine problems except for a few cases: the functions Φ_3 and Φ_4 failed to solve problem 9 due to overflow. However, the function Φ_3 was able to find a near feasible point when a penalty parameter value of 5 was used.

3) In general, Φ_3 and Φ_4 are likely to find feasible points more often than other functions (except for problem 9). They are also more stable.

4) The primal methods behaved in a similar manner to that for the mathematical problem tested in Sec. IV. They are more stable and require less computational effort than the penalty methods. The results are summarized in Table 3.

5) The methods A and B ($\Delta = 10\%$ of current cost) behaved similarly; they found a feasible point for all of the nine problems in a few iterations except for problem 1 which had a much larger initial violation. When $\Delta = 0$ was imposed in method B, it took

Table 1 Description of structural design problems

Problem no.	Description
1	10-bar truss subjected to two vertical loads of 100 kips each[1]
2	10-bar truss subjected to same loading but different starting design[1]
3	10-bar truss subjected to same loading but design variables are flange widths and web heights; also a different Young's modulus is used[1]
4	10-bar truss subjected to same conditions as problem 3 with an additional constraint on the natural frequency[1]
5	1-bay 2-story frame subjected to vertical uniform load and concentrated horizontal wind loads[1,9]
6	1-bay 2-story frame subjected to same loading as problem 5 with an additional frequency constraint[1,9]
7	47-bar truss subjected to two vertical loads and one wind load[10,11]
8	200-bar truss subjected to nodal forces[1,10]
9	2-bay 6-story frame subjected to distributed loads and wind loads[1]

Table 2 Summary of results with penalty functions for structural design problems

Problem	Maximum violation, % Initial	Final	Method	Penalty value for best result	Cost function	Function evaluations	CPU time, s
1	884.89	0.0	Φ_3, Φ_4	$5 \leq r \leq 20$	6.03, 6.18	42, 42	0.8, 1.50
2	96.98	0.0	Φ_3, Φ_4	$20 \leq r \leq 100$	8.11, 8.11	42, 42	0.81, 0.8
3	101.24	0.0	Φ_3, Φ_4	$10 \leq r \leq 100$	4.01, 4.01	82, 82	1.62, 1.6
4	101.24	0.0	Φ_3, Φ_4	$10 \leq r \leq 100$	4.01, 4.01	82, 82	4.3, 4.3
5	26.32	0.0	Φ_3, Φ_4	$50 \leq r \leq 100$	6.13, 6.13	67, 67	2.6, 2.6
6	26.32	0.0	Φ_4	$20 \leq r \leq 50$	4.34	198	21.3
		0.1	Φ_3	$70 \leq r \leq 100$	0.0	300	32.3
7	36.16	0.0	Φ_4	$20 \leq r \leq 70$	4.68	395	33.3
8	38.18	0.1	Φ_5	$50 \leq r \leq 100$	2.3E–2	1480	913.2
9	107.21	0.1	Φ_1, Φ_2	$r = 100$	0.001, 0.0	6073, 1853	733, 227

Table 3 Summary of results with primal methods for structural design problems

Prob. no.	Initial Vio,[a] %	Initial Cost	Method A NI[b]	Method A CPU, s	Method A Cost	Method B NI[b]	Method B CPU, s	Method B Cost	Method B, $\Delta = 0$ NI[b]	Method B, $\Delta = 0$ CPU, s	Method B, $\Delta = 0$ Cost	Min-max NI[b]	Min-max CPU, s
1	884.8	0.8	10	1.0	6.4	38	3.6	5.2	47	3.9	5.1	22	1.31
2	96.9	4.2	6	0.6	7.0	9	0.8	6.0	44	2.9	5.1	17	1.15
3	101.2	3.8	5	0.7	6.2	7	0.9	5.1	17	2.1	3.8[c]	20	1.85
4	101.2	3.8	6	1.0	5.8	8	1.3	5.1	7	1.6	3.8[c]	19	2.62
5	26.3	38.1	3	0.5	41.0	3	0.5	41.0	4	0.5	38.0	9	1.20
6	26.3	38.1	4	0.8	40.0	4	0.8	40.0	4	0.8	38.0	10	1.91
7	36.2	4.3	4	2.4	4.4	4	2.6	4.4	4	2.4	4.3	25	18.80
8	38.2	39.7	5	13.7	43.0	5	13.7	43.0	9	40.0	37.0	16	9.14
9	107.2	48.8	5	3.7	47.0	5	3.8	47.0	11	7.5	40.0	23	52.30

[a]Vio: initial violation. [b]NI: number of iterations (also number of calls for function and gradient evaluations). [c]Method did not converge to a feasible point.

more computational effort and in two cases (problems 3 and 4) did not find a feasible point. The min-max method found feasible points for all of the problems and required more computational effort than the first two primal methods.

VI. Discussion and Conclusions

Methods to correct constraint violations are presented and analyzed. Five methods based on penalty approach and three methods based on the primal approach are included in the study. Another penalty function consisting of quadratic loss term for equalities and logarithmic term for inequalities given in Ref. 5 was also implemented and evaluated. The results for this function were quite similar to those for the functions Φ_3 and Φ_4. Based on the numerical evaluation, the following points are noted.

1) The primal methods require less computational effort than the penalty methods.

2) None of the methods (penalty or primal) was able to find a feasible point for all of the problems solved starting from an arbitrary point.

3) The penalty methods' performance is sensitive to the penalty parameter used and the update strategy. This is a problem dependent parameter; therefore, it is difficult to give general guidelines for its initial selection.

4) It is suggested that if a method fails to find a feasible point (which in many cases does not mean that the problem is infeasible), it should be restarted from a different point. For this purpose, either a random point generator should be used or the designer should select a new starting point. Some global optimization technique may also be used to solve the problem of feasible point determination.

A basic conclusion from the study is that most of the methods are based on the philosophy of requiring decrease in the constraint violations from one iteration to the next. This is interpreted as an improvement in design and progress toward the feasible domain for the problem. In some cases, this movement is away from the feasible domain. In such situations, most of the methods eventually get trapped at points from where no decrease in the constraint violations is possible. A similar situation can arise if the given problem has no feasible solution points and the penalty function approach is used. The reason is that the penalty function has many local minimum points. Some of these local minima are not feasible for the original problem. Therefore, finding a feasible point of the constraint set is equivalent to solving a global optimization problem. To overcome these difficulties, use of random points is suggested to restart the methods. Several such starting points need to be tried before declaring the problem to be infeasible.

References

[1]Haug, E. J., and Arora, J. S., *Applied Optimal Design*, Wiley-Interscience, New York, 1979, pp. 100–105.

[2]Arora, J. S., *Introduction to Optimal Design*, McGraw–Hill, New York, 1989, pp. 338–439.

[3]Schnabel, R. B., "Determining Feasibility of a Set of Nonlinear Inequality Constraints," *Mathematical Programming Study*, North-Holland, Amsterdam, 1980, Vol. 16, pp. 137–148.

[4]Kort, B. W., and Bertsekas, D. P., "A New Penalty Function Algorithm for Constrained Minimization," *Proceedings 1972 IEEE Conference on Decision and Control* (New Orleans, LA), Inst. of Electrical and Electronics Engineers, New York, 1972, pp. 162–166.

[5]Arora, J. S., Chahande, A. I., and Paeng, J. K., "Multiplier Methods for Engineering Optimization," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 7, 1991, pp. 1485–1525.

[6]Bertsekas, D. P., "Approximation Procedures Based on the Methods of Multipliers," *Journal of Optimization Theory and Applications*, Vol. 23, No. 4, 1977, pp. 487–510.

[7]Bertsekas, D. P., *Constrained Optimization and Lagrange Multiplier Methods*, Academic, New York, 1982.

[8]Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," IFAC Symposium on Computer Aided Design of Control Systems (Zurich, Switzerland), International Federation of Automatic Control, 1979.

[9]Nakamura, Y., "Optimum Design of Frame Structures Using Linear Programming," M.S. Thesis, Dept. of Civil Engineering, Massachusetts Inst. of Technology, Cambridge, MA, Feb. 1966.

[10]Arora, J. S., "On Improving Efficiency of an Algorithm for Structural Optimization and a User's Manual for Program Trussopt 3," Tech. Rept. 12, Materials Engineering Div., Univ. of Iowa, Iowa City, IA, 1974.

[11]Johnson, D., and Brotton, D. M., "Optimum Elastic Design of Redundant Trusses," *Journal of the Structural Division, ASCE*, Vol. 95, No. ST12, 1969, pp. 2589–2610.