Semi-infinite programming in multi-hazard structural design

F. A. Potra[1] & E. Simiu[2]
[1]*Mathematical and Computational Sciences Division,*
National Institute of Standards and Technology, Gaithersburg, MD, USA
[2]*Materials and Construction Research Division,*
National Institute of Standards and Technology, Gaithersburg, MD, USA

Abstract

Optimal multi-hazard structural design consists in determining a vector of design variables, subjected to the constraints imposed by all the hazards to which the structure is exposed, such that the cost or weight of the structure is minimized. In particular, the design variables should be determined is such a way that the load-induced stresses and deflections are kept below specified thresholds at all points of the resulting structure. Since there are infinitely many such points, the optimization problem becomes a semi-infinite programming problem. In the present paper we discuss the difficulties involved in the numerical solution of the semi-infinite programming problems arising in multi-hazard structural design. We show that it is possible to construct efficient and robust optimization algorithms, by adaptively choosing a family of finite sets of points on the structure, and by using interior point methods for solving the corresponding optimization problems.

1 Introduction

Optimal structural design consists in determining a vector $d = (d_1, d_2, \ldots, d_n)$ of n design variables such that the structure defined by d satisfies a set of given requirements, and a certain objective function, $f(d)$, related to the cost of the structure is minimized. Multi-hazard design is required where the structure may be subjected to p different hazards. Some of the requirements to be satisfied by the design variables are independent of the different hazards, and can be expressed as inequality constraints of the form $g(d) \leq 0$, where $g : I\!R^n \rightarrow I\!R^m$ is a given vector function. The inequality $g(d) \leq 0$ is interpreted component-wise,

WIT Transactions on The Built Environment, Vol 106, © 2009 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/OP090021

so that the design variables are subject to m constraints that are independent of the hazards. For example, some of these m constraints are simple bound constraints, imposing lower and upper bounds for the values of the design variables. On the other hand, each hazard i, $i = 1, 2, \ldots, p$, imposes constraints of the form $g^i(d, h) \leq 0, \forall h \in \mathcal{H}$, where $g^i : I\!R^n \times \mathcal{H} \to I\!R^{m_i}$. Here $\mathcal{H}$ is an infinite family of points belonging to the structure. The constraint $g^i(d, h) \leq 0, \forall h \in \mathcal{H}$ specifies that the load-induced stresses and deflections caused by hazard i are kept below specified thresholds at all points of $\mathcal{H}$. With this notation, the optimal multi-hazard structural design can be written as

$$\begin{aligned} &\min f(d) \\ \text{such that } &g(d) \leq 0\,, \\ &g^1(d,h) \leq 0, \;\; \forall h \in \mathcal{H}\;, \\ &\ldots \\ &g^p(d,h) \leq 0, \;\; \forall h \in \mathcal{H}\;. \end{aligned} \tag{1}$$

Since $\mathcal{H}$ is an infinite set, the above optimization problem is a semi-infinite optimization problem [1–3]. Such problems are very difficult to solve, because they contain an infinite number of constraints. A simple minded approach is to replace in (1) the infinite set $\mathcal{H}$ by a finite subset $\widetilde{\mathcal{H}} \subset \mathcal{H}$, and to solve the resulting (finitely constrained) programming problem. The design parameter vector $\widetilde{d}$, obtained by solving this problem, may not be useful for the original optimal multi-hazard design problem, because at some points of the resulting structure, belonging to $\mathcal{H} \setminus \widetilde{\mathcal{H}}$, the stresses may be too large.

In previous work [4, 5], we have proposed an iterative method for solving the optimal multi-hazard structural design problem (1), where each iteration requires the numerical solution of a (finitely constrained) optimization problem of the form,

$$\begin{aligned} &\min f(d) \\ \text{such that } &g(d) \leq 0\,, \\ &g^1(d,h) \leq -\varepsilon, \;\; \forall h \in \mathcal{H}_k\;, \\ &\ldots \\ &g^p(d,h) \leq -\varepsilon, \;\; \forall h \in \mathcal{H}_k\;, \end{aligned} \tag{2}$$

as well as the computation of the following quantities:

$$h^i_j = \text{argmax}_{h \in \mathcal{H}}\; g^i_j(\widetilde{d}, h), \quad i = 1, \ldots, p,\; j = 1, \ldots, m_i. \tag{3}$$

The approach from [4, 5] can be formalized as the following generic algorithm:

```
Algorithm 1
Select a (small) finite subset $\mathcal{H}_0 \subset \mathcal{H}$ and a tolerance $\varepsilon > 0$;
    Set $k \leftarrow 0$;
    repeat
        Find the solution $\widetilde{d}$ of the optimization problem (2);
        If $\widetilde{d}$ satisfies the constraints of (1),
            then return $\widetilde{d}$ as the solution of the problem and stop;
        Update $\mathcal{H}_k$ to obtain a larger subset $\mathcal{H}_{k+1} \subset \mathcal{H}$;
        Set $k \leftarrow k + 1$.
    continue
```

The updating of $\mathcal{H}_k$ in the above algorithm was done by adding to $\mathcal{H}_k$ the points at which the maximum violations of the constraints in (1) are attained and, possibly, some randomly generated points in $\mathcal{H}$. The addition of these randomly generated points had a stabilization effect on the algorithm. However, because of the randomness, the behavior of the algorithm varied slightly, both in the number of iterations before termination and the CPU time, each time it was used.

In the present paper we propose a purely deterministic variant of Algorithm 1, that turns out to be very stable, and outperforms most of time the random algorithms from [4,5]. For the sake of simplicity, we present the new algorithm only on the simple example considered in [4, 5]. This example will be presented in Section 2. The description of our adaptive iterative strategy is contained in Section 3. We give some numerical results in Section 4, showing the robustness and efficiency of our approach.

2 A simple example of multi-hazard structural design

In [4, 5] we have examined a structure consisting of equally spaced steel columns supporting a long, straight horizontal metal pipe filled with water, and have considered the problem of designing a typical supporting column. The steel column has hollow elliptical cross sections with constant thickness w. Geometrically, the column is defined as the region between two frustums of cones with elliptical cross sections, and given hight H. The cross sections of the bottom and the top of the outer frustum are the ellipses that can be represented in the xy plane by the equations

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1\,, \qquad \frac{x^2}{A^2} + \frac{y^2}{B^2} = 1.$$

We assume that the two ellipses are similar, in the sense that $a/b = A/B$, which is equivalent to stating that they have the same eccentricity. Therefore A can be expressed as

$$A = \frac{aB}{b}\,. \tag{4}$$

Therefore, we have only four design variables a, b, B and w that have to satisfy the following bound constraints

$$w_{min} \leq w \leq a_{min} \leq a \leq b \leq B \leq B_{max}\,, \tag{5}$$

where the quantities $w_{min} < a_{min} < B_{max}$ are specified.

A vertical force denoted by V, equal to the weight of the pipe and the water it contains, is applied concentrically at the top of the column. Horizontal forces are applied at the center line of the pipe. They can be due to seismic effects or to wind effects. The seismic forces are denoted by F^e, can act in any direction, and are the same for all directions. The wind forces also act from any direction and are denoted by $F^w \cos\alpha$, where α is the angle between the wind direction and the normal to the longitudinal axis of the pipe. They are largest when normal to the pipe and are assumed to be negligible when parallel to the pipe.

It is easily shown that at coordinate h from the top of the column the compression stress due to gravity loading (defined as the sum of the vertical concentric load V and the self-weight of the column above the coordinate h, divided by the area of the cross section at coordinate h) is given by

$$s_g(a,b,B,w,h) = \frac{2bHV + 2bH\pi(a+b-w)w\gamma h + (a+b)(B-b)\pi w\gamma h^2}{2\pi(H(a+b-w)b + (a+b)(B-b)h)w}\,, \tag{6}$$

where γ is the specific weight of the column material.

It is also shown (see [5] for technical details) that the bending stress at coordinate h, induced by a force F that makes an angle α with the major axis of the ellipse and is applied at a point situated at distance c above the column, can be expressed by the formula

$$s_b(a,b,B,w,F,\alpha,h) = \frac{F(c+h)\,q_0(a(h),b(h),\alpha)}{\mu(a(h),b(h),w,\alpha)}\,, \tag{7}$$

where

$$a(h) = \left(1 - \frac{h}{H}\right)a + \frac{h}{H}\frac{aB}{b}\,, \quad b(h) = \left(1 - \frac{h}{H}\right)b + \frac{h}{H}B\,,$$

$$q_0(a,b,\alpha) = \sqrt{.5(a^2 + b^2 + (b^2 - a^2)\cos(2\alpha))},$$

$$\mu(a,b,w,\alpha) = \frac{\pi}{8}ab\left(a^2 + b^2 + \left(b^2 - a^2\right)\cos(2\alpha)\right)$$

$$-\frac{\pi}{8}(a-w)(b-w)\left((a-w)^2 + (b-w)^2 + \left((b-w)^2 - (a-w)^2\right)\cos(2\alpha)\right).$$

We assume that the wind force, of magnitude F^w, acts along the y-axis, i.e., along the major axis of the ellipse. The corresponding bending stress is obtained by taking $F = F^w$ and $\alpha = 0$ in (7),

$$s_b^w(a,b,B,w,h) = s_b(a,b,B,w,F^w,0,h)\,. \tag{8}$$

Therefore the total stress in this case is

$$s^w(a,b,B,w,h) = s_g(a,b,B,w,h) + s_b^w(a,b,B,w,h)\,. \tag{9}$$

Since we would like this stress to be below the maximum admissible stress σ we obtain a constraint of the form

$$s^w(a,b,B,w,h) \le \sigma, \quad \forall j \in \mathcal{H}\,. \tag{10}$$

As the direction of the force F^e of the earthquake is not known, the corresponding constraint becomes

$$\begin{aligned} s^{\,e}(a,b,B,w,h) = s_g(a,b,B,w,h) + \max_{\alpha\in[0,\pi]} s_b(a,b,B,w,F^e,\alpha,h) \ \le \sigma, \\ 0 \le h \le H\,. \end{aligned} \tag{11}$$

The optimal design problem we are considering is to find design variables a, b, B, w satisfying the constraints (5), (10), and (11), such that the mass of the corresponding column is minimized. That mass is our objective function. It can be expressed in terms of the design variables as

$$\varphi(a,b,B,w) = \frac{H\pi\rho(a(b+B)+b(b+B-2w))w}{2b}, \tag{12}$$

where ρ is the specific mass of the column material. We note that in the International System of Units (SI), the specific mass and the specific weight are related by the equation

$$\rho = \frac{\gamma}{9.80665}\,. \tag{13}$$

With the notation introduced above, our optimal multi-hazard structural design problem can be written as

$$\begin{aligned} &\min_{a,b,B,w} \ \varphi(a,b,B,w) && (14)\\ \text{such that } & s^w(a,b,B,w) \le \sigma\,, \\ & s^e(a,b,B,w) \le \sigma\,, \\ & w_{min} \le w \le a_{min} \le a \le b \le B \le B_{max}\,. \end{aligned}$$

This is a particular case of the general formulation (1), described in the introduction.

3 An iterative method

In this section we present our deterministic variant of Algorithm 1 for solving the semi-infinite programming problem (14). In this case we have $\mathcal{H} = [0, H]$, and we choose the initial finite subset of $\mathcal{H}$ to be

$$\mathcal{H}_0 = \left\{ 0 \, , \, \frac{H}{3} \, , \, \frac{2H}{3} \, , \, H \right\} . \tag{15}$$

At a typical iteration we compute the solution $(\widetilde{a}, \widetilde{b}, \widetilde{B}, \widetilde{w})$ of the nonlinear optimization problem

$$\min_{a,b,B,w} \; \varphi(a,b,B,w) \tag{16}$$

$$\begin{aligned} \text{such that } s^w(a,b,B,w,h) &\le \sigma - \frac{\varepsilon}{2\pi w} , \quad h \in \mathcal{H}_k , \\ s^e(a,b,B,w,h) &\le \sigma - \frac{\varepsilon}{2\pi w} , \quad h \in \mathcal{H}_k , \\ w_{min} \le w \le a_{min} &\le a \le b \le B \le B_{max} . \end{aligned}$$

Then we compute the maximum stresses and denote

$$hW = \text{argmax}_{h \in [O,H]} s^w(\widetilde{a}, \widetilde{b}, \widetilde{B}, \widetilde{w}, h), \quad msW = s^w(\widetilde{a}, \widetilde{b}, \widetilde{B}, \widetilde{w}, hW). \tag{17}$$

$$hQ = \text{argmax}_{h \in [O,H]} \overline{s}^e(\widetilde{a}, \widetilde{b}, \widetilde{B}, \widetilde{w}, h), \quad msQ = s^e(\widetilde{a}, \widetilde{b}, \widetilde{B}, \widetilde{w}, hQ), \tag{18}$$

In our implementation the points hW and hQ are found within a tolerance of $.01\varepsilon$, by using a golden section method procedure [6, pp. 726].

If both msW and msQ are less than $\sigma - .1\varepsilon$ then we return $(\widetilde{a}, \widetilde{b}, \widetilde{B}, \widetilde{w})$ as the solution of the optimal multi-hazard structural design problem (14). Otherwise we update $\mathcal{H}_k$ and start a new iteration. It turns out that there are some advantages if at the end of the first iteration ($k = 0$)we construct $\mathcal{H}_1$ by enlarging the set

$$\left\{ 0 \, , \, \frac{H}{2} \, , \, H \right\} , \tag{19}$$

rather than the set (15). For $k >$ we always construct $\mathcal{H}_{k+1}$ by enlarging $\mathcal{H}_k$. The precise updating method is described in the following algorithm:

```
Algorithm 2
Choose a tolerance $\varepsilon > 0$;
Consider the set $\mathcal{H}_0 \subset \mathcal{H}$ defined by (15);
    Set $k \leftarrow 0$;
    repeat
        Compute the solution $\widetilde{d} = (\widetilde{a}, \widetilde{b}, \widetilde{B}, \widetilde{w})$ of (16);
        Compute (17) and (18) within a tolerance of $.01\varepsilon$;
        If $\max\{msW, msQ\} < \sigma - .1\varepsilon$,
            then return $\widetilde{d}$ as the solution of the problem and stop;
        If $k = 0$, then replace $\mathcal{H}_k$ with the set (19);
        If $msW > \sigma - .1\varepsilon$, then set $\mathcal{H}_{k+1} = \mathcal{H}_k \bigcup \{hW\}$;
        If $msQ > \sigma - .1\varepsilon$, then set $\mathcal{H}_{k+1} = \mathcal{H}_k \bigcup \{hQ\}$;
        Set $k \leftarrow k + 1$.
    continue
```

4 Numerical results

In the numerical results to be presented in this section we have taken the following values for the constants defining our model

$$H = 3.6\,m,\ c = 0.6\,m,\ V = 5000\,N,\ \gamma = 76820\,N/m^3,$$
$$F^e = 6000\,N,\ F^w = 8000\,N,\ \sigma = 165\,MPa,$$
$$w_{min} = .003\,m,\ a_{min} = .01\,m,\ B_{max} = .24\,m.$$

Previous numerical experiments reported in [5] showed that interior point methods (see [7] and the literature cited therein) are very efficient in solving nonlinear programming problems in structural optimization.

The numerical results from this section were obtained by using the *Mathematica* function `FindMinimum` with the option `Method -> "InteriorPoint"`, and the following trivial starting point

$$a = 0.4B_{max}, \quad b = 0.5B_{max}, \quad B = 0.9B_{max}, \quad w = 1.5w_{min}\ .$$

By using Algorithm 2 we obtained the following values for the design parameters:

$$a = 0.0532\,m,\ b = 0.0915\,m,\ B = 0.2031\,m,\ w = 0.003\,m.$$

The (optimal) weight of the corresponding column is 61.1 kg. Algorithm 2 needed 6 iterations to find this solution and the total CPU time (on a Dell Optiplex machine with 1 GB of RAM and a 3.20 GHz Pentium 4 processor) was 2.8 seconds. The same values for the design variables were obtained with the method described in [4,5]. As we mentioned before that method has a random component so that the CPU time varies each time the program is run. We run the program 10 times for the problem considered in this section. While the program gave the same solution in all of the runs the execution time varied. The average CPU time was 5.4 seconds with a standard deviation of one second. Also the average number of iterations was 8.5 with a standard deviation of almost .97 iterations.

5 Conclusions

In this paper we have described an iterative method that can be easily coupled with interior point methods in order to solve efficiently semi-infinite programming problems arising in multi-hazard structural design. Our method is robust and strictly deterministic. The new method appears to be more efficient, both in terms of number of iterations and computational time, than our previous algorithms which were not completely deterministic.

Disclaimer Certain commercial software products are identified in this paper in order to adequately specify the computational procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology nor does it imply that the software products identified are necessarily the best available for the purpose.

References

[1] Goberna, M.Á. & López, M.A., (eds.) *Semi-infinite programming*, volume 57 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers: Dordrecht, pp. xviii+386, 2001. Recent advances, Papers from the conference (SIP'99) held in Alicante, September 1999.

[2] Stein, O., *Bi-level strategies in semi-infinite programming*, volume 71 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers: Boston, MA, pp. xxviii+202, 2003.

[3] Weber, G.W., *Generalized semi-infinite optimization and related topics*, volume 29 of *Research and Exposition in Mathematics*. Heldermann Verlag: Lemgo, pp. xxii+361, 2003.

[4] Potra, F.A. & Simiu, E., Optimization and structural multi-hazard design. *Journal of Engineering Mechanics*, (submitted for publication), 2008.

[5] Potra, F.A. & Simiu, E., Multi-hazard design: A structural optimization approach. *Journal of Optimization Theory and Applications*, (submitted for publication), 2009.

[6] Bertsekas, D.P., *Nonlinear programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

[7] Wright, M.H., The interior-point revolution in optimization: history, recent developments, and lasting consequences. *Bull Amer Math Soc (NS)*, **42(1)**, pp. 39–56 (electronic), 2005.