SIAM J. OPTIM.
Vol. 7, No. 4, pp. 991–1016, November 1997

ROBUST TRUSS TOPOLOGY DESIGN VIA SEMIDEFINITE PROGRAMMING*

A. BEN-TAL† AND A. NEMIROVSKI†

Abstract. We present and motivate a new model of the truss topology design problem, where the rigidity of the resulting truss with respect both to given loading scenarios and small "occasional" loads is optimized. It is shown that the resulting optimization problem is a semidefinite program. We derive and analyze several equivalent reformulations of the problem and present illustrative numerical examples.

Key words. structural optimization, truss topology design, robustness, semidefinite programming, interior point methods

AMS subject classifications. 19C25, 19C30, 19C50, 73K40

PII. S1052623495291951

1. Introduction.

Truss topology design (TTD) deals with the selection of optimal configuration for structural systems (mechanical, civil engineering, aerospace) and constitutes one of the newest and most rapidly growing fields of structural design (see the excellent survey paper by Rozvany, Bendsøe, and Kirsch [12]). The TTD problem was studied extensively, both mathematically and algorithmically, in [1, 2, 3, 4, 5].

In this paper we bring forth the issue of the *robustness* of the truss; here we say that a truss is *robust* if it is reasonably rigid with respect both to the given set of loading scenarios and to all small uncertain (in size and direction) loads, which may act at any of the *active* nodes of the truss, i.e., those which are linked at least by one bar. In the engineering literature, rigidity is modeled by considering different *loading scenarios* on the structure (the multiload TTD problem) or by imposing upper and lower bounds on nodal displacements. The first approach depends on the engineer's ability to "guess right" the relevant scenarios, while the second approach leads to a mathematical problem which is not tractable computationally. Here we suggest a new modeling approach, which circumvents both of the above mentioned difficulties.

The paper is organized as follows. Section 2 describes the modeling approach in question. The preliminary section 2.1 presents the basic notions related to the TTD problem and the traditional formulations of the problem. We demonstrate by simple example (section 2.2) that robustness restrictions (which are basically ignored in the traditional formulations) are critical to obtain reasonable designs; this observation motivates our modeling approach presented in section 2.3. Its computational tractability is demonstrated in section 2.4, where we show that the TTD problem in our new formulation can be equivalently cast as a *semidefinite program.* This brings the problem into the realm of convex programming for which efficient (polynomial time) interior point algorithms can be employed. Sections 3–5 are devoted to mathematical processing of the semidefinite program of section 2.4; the goal is to get a program better

---

*Received by the editors September 15, 1995; accepted for publication (in revised form) June 28, 1996.

http://www.siam.org/journals/siopt/7-4/29195.html

†Faculty of Industrial Engineering and Management at Technion Israel Institute of Technology, Technion City, Haifa 32000, Israel (morbt@ie.technion.ac.il, nemirovs@ie.technion.ac.il). This research was partly supported by the Israeli Academy of Sciences and by The Fund for Promotion of Research at the Technion.

suited for interior point algorithms. Possibilities for robust truss topology design by these algorithms are discussed in section 6. We end up (section 7) with illustrating usefulness of our approach by considering several examples of optimal trusses with and without robustness considerations. We show that at least for these examples *robustness* can be gained without sacrificing much in the optimality of the resulting trusses. Concluding section 8 contains remarks on the possibility to extend the idea of "robust reformulation" of an optimization program from the particular case of the TTD problem to other problems of mathematical programming.

2. Truss topology design with robustness constraints.

2.1. Trusses, loads, compliances. Informally, a *truss* is a 2D or 3D construction composed of thin elastic *bars* linked with each other at *nodes*—points from finite *nodal set* $\mathcal{V}$ given in advance in 2D plane (respectively, 3D) space. When subjected to a given *load*—distribution of external forces applied at the nodes—the construction deformates until the reaction forces caused by deformations of the bars compensate the external load. The deformated truss capacitates certain potential energy, and this energy, the *compliance*, measures stiffness of the truss, its ability to withstand the load; the less is compliance, the more rigid is the truss with respect to the load.

In the usual TTD problem we are given the nodal set and one (*single-load* TTD) or several (*multi-load* TTD) loads along with total volume of the bars. The displacements of some of the nodes are completely or partially fixed, so that the space R_v of virtual displacements of node v is certain linear subspace and the problem is to distribute the given volume of the truss between the bars in order to get the most rigid construction, i.e., the one which minimizes the maximal compliance over the given set of loads. Some of the bars can get zero volume, i.e., be eliminated from the resulting construction, so that in fact the topology of the construction is optimized as well (this is the origin of the term "topology design").

The mathematical formulation of the problem, in its simplest form, is as follows.

Given are

(i) graph $(\mathcal{V}, \mathcal{B})$ (ground structure) with the nodal set $\mathcal{V} \subset R^D$ $(D = 2, 3)$ composed of $\widehat{n}$ nodes and with arc set $\mathcal{B}$ of m tentative bars;

(ii) collection of linear subspaces $R_v \subset R^D$, $v \in \mathcal{V}$—the spaces of virtual displacements of the nodes.

We refer to the quantity $n = \sum_{v \in \mathcal{V}} \dim R_v$ as the number of degrees of freedom of the nodal set and call the space $R^n = \prod_{v \in \mathcal{V}} R_v$ the *space of nodal displacements.* A vector $x \in R^n$ can be naturally interpreted as collection of virtual displacements of the nodes. Similarly, a load—collection of external forces applied at the nodes – can be interpreted as a vector from R^n. (One can ignore the components of the forces orthogonal to the subspaces of virtual nodal displacements, since these components are compensated by supports restricting virtual displacements of nodes; the remaining components of the forces can be naturally assembled in a vector from R^n.)

(iii) When designing the truss, we are given a finite set $F \subset R^n$ of *loading scenarios*; the truss should be able to carry the load for each of the scenarios.

(iv) The design variables in the problem are *bar volumes* t_i, $i = 1, ..., m$; along with the nodal set $\mathcal{V}$, they completely determine the truss. We allow ourselves, for the sake of brevity, *truss* t. We are given the total volume $V > 0$ of the bars, so that the set of all admissible vectors of bar volumes is the simplex

$$T = \left\{ t \in R^m |\, t \geq 0,\ \sum_{i=1}^{m} t_i = V \right\}.$$

With the elastic model of the bars, deformation of truss accompanied by displacement $x \in R^n$ of the nodes results in the vector of reaction forces $A(t)x$, where t is the vector of bar volumes and

$$A(t) = \sum_{i=1}^{m} t_i A_i$$

is the $n \times n$ *bar-stiffness matrix* of the truss. The *bar-stiffness matrix* A_i of the ith bar is readily given by the geometry of the nodal set and involves the Young modulus of the material. What is crucial for us is that for all i,

$$A_i = b_i b_i^T \tag{2.1}$$

is a rank 1 positive semidefinite symmetric matrix (for explanations and details, see, e.g., [1, 2, 3]).

Given $t \in T$ and a load $f \in F$, one can associate with this pair the equilibrium equation

$$A(t)x = f. \tag{2.2}$$

(As was explained, x is the vector of nodal displacements caused by the load f, provided that the vector of bar volumes is t.) Solvability of this equation means that the truss is capable of carrying the load f, and if this is the case, then the *compliance*[1]

$$c_f(t) \equiv f^T x = \sup_{u \in R^n} \left[2f^T u - u^T A(t) u \right] \tag{2.3}$$

is regarded as a measure of internal work done by the truss with respect to the load f; the smaller is the compliance, the larger is the stiffness of the truss. If the equilibrium equation (2.2) for a given t is unsolvable, then it is convenient to define the compliance $c_f(t)$ as $+\infty$, which is compatible with the second equality in (2.3).

The problem of optimal minmax TTD is to find the vector of bar volumes which results in the smallest possible worst-case compliance:

($\mathrm{TD}_{\mathrm{minmax}}$) : *find* $t \in T$ *which minimizes the worst-case compliance* $c^F(t) = \sup_{f \in F} c_f(t)$.

From now on we assume that the problem is *well posed*, i.e., that

A. The matrix $\sum_{i=1}^{m} A_i$ is positive definite.

(This actually means that the supports prevent rigid body motion of the truss.)

2.2. Robustness constraint: Motivation. The "standard" case of problem ($\mathrm{TD}_{\mathrm{minmax}}$) is the one when F is a singleton (*single-load TTD problem*) or a finite set composed of small number (3-5) of loads (*multiload TTD problem*). An evident shortcoming of both these settings is that they do not take "full" care of the robustness of the resulting truss. The associated optimal design ensures reasonable (in fact the best possible) behavior of the truss under the loads from the list of scenarios F; it may happen, however, that a load not from this set, even a "small" one, will cause an inappropriately large deformation of the truss. Consider, e.g., the following toy example. Figure 2.1 represents a six-element nodal set with two fixed nodes ($R_v = \{0\}$) and four free nodes ($R_v = R^2$), the "ground structure"—the set of all tentative bars and the load f which is the unique element of F.

[1]The "true" compliance, as defined in mechanics, is one half of the quantity given by (2.3); we rescale the compliance in order to avoid multiple fractions $\frac{1}{2}$.

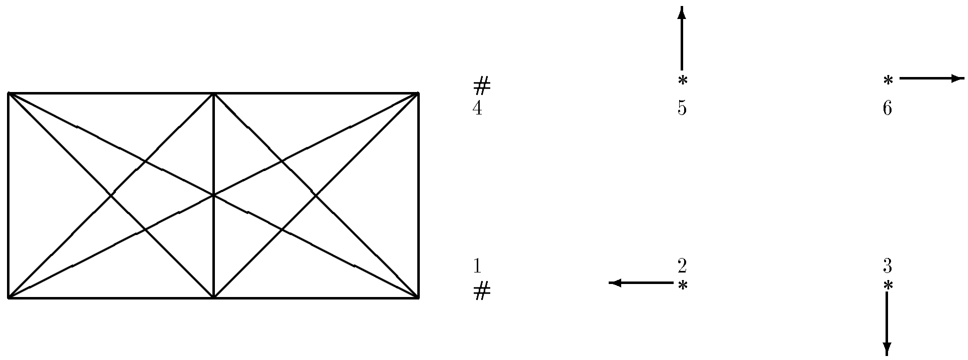

FIG. 2.1. *Ground structure and loading scenario* * *– free nodes;* # *– fixed nodes; arrows – forces.*

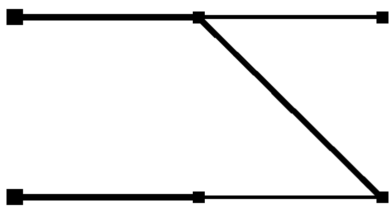

FIG. 2.2. *Optimal single-load design.*

Figure 2.2 shows the results of the usual single-load design, which results in the optimal compliance 16.000. Note that the resulting truss is completely unstable; e.g., the bar linking nodes 5 and 6 can rotate around node 5, so that arbitrarily small nonhorizontal force applied at node 6 will cause infinite compliance.

It seems that a "good" design should ensure reasonable compliances under *all* tentative loads of reasonable magnitude acting at the nodes of the resulting truss, not only "the best possible" compliance under the small list of loads in F of primary interest.

The indicated requirement can be modeled as follows. When formulating the problem, the engineer embeds a small finite set of loads $F = \{f_1, ..., f_q\}$ he is especially interested in ("primary" loads) into a "more massive" set M containing F, but also "occasional loads" of perhaps much smaller magnitude ("secondary" loads), and looks for the truss $t \in T$ which minimizes the worst-case compliance $c^M(t)$ taken with respect to this extended set M of loading scenarios.

In order to get a computationally tractable problem, in what follows we restrict ourselves to the case where M is an ellipsoid centered at the origin.[2]

$$M = QW_q \equiv \{Qe |\, e \in R^q,\, e^T e \le 1\}.$$

Here Q is a given $n \times q$ "scale" matrix and W_q is the unit Euclidean ball in R^q. Note that we allow the case $q < n$ as well, where M is a "flat" q-dimensional ellipsoid.

The corresponding modification of $(\mathrm{TD}_{\mathrm{minmax}})$ is as follows:

[2]The only other case when the indicated problem is computationally tractable seems to be that one of a polytope M given by the list of its vertices. This case hardly deserves a special consideration, since it leads to the standard multiload TTD problem.

$(\mathrm{TD}_{\mathrm{robust}})$: *find* $t \in T$ *which minimizes the compliance*

$$c^M(t) = \max_{e^T e \le 1} \max_{x \in R^n} \left[2(Qe)^T x - x^T A(t) x \right].$$

2.3. Selection of scale matrix Q. Problem $(\mathrm{TD}_{\mathrm{robust}})$ takes care of all loads $f \in M$, M being the image of the unit q-dimensional Euclidean ball under the mapping $e \mapsto Qe$. It follows that if a load $f \in M$ has a nonzero force acting at certain node l, then this node will for sure be present in the resulting construction. This observation means that we should be very careful when forming Q; otherwise we enforce incorporating into the final construction the nodes which in fact are redundant. There are two ways to meet the latter requirement.

A. We could use the indicated approach as a postoptimality analysis; after we have found the solution to the usual multiload TTD problem, given the resulting nodal structure, we can improve the robustness of the solution by solving $(\mathrm{TD}_{\mathrm{robust}})$ associated with this nodal structure.

B. We know in advance some nodes which for sure will be present in the solution (certainly the nodes where the forces from the given loading scenarios are applied) and it seems to be natural to require rigidity with respect to all properly scaled forces acting at these "active" nodes.

Let us discuss in more detail the latter possibility. Let $F = \{f_1, ..., f_k\}$ be the given set of loading scenarios. We say that a node $v \in \mathcal{V}$ is *active* with respect to F if the projection of certain load f_j on the space R_v of virtual displacements of the node is nonzero. Let $\mathcal{V}^*$ be the set of all active nodes. Our goal is to embed F into a "reasonably chosen" ellipsoid M in the space $R^q = \prod_{v \in \mathcal{V}^*} R_v$ (which for sure will be the part of the displacement space in the final construction). According to our motivation, M should contain

(i) the set F of given loads;

(ii) the ball $B = \{f \in R^q | \, f^T f \le r^2\}$ of all "occasional" loads of prescribed magnitude r.

The setup $M = F \cup B$ most adequate to our motivation is inappropriate; as it was explained, we need M to be an ellipsoid in order to get a computationally tractable problem, so that we should look for "the smallest possible" ellipsoid M containing $F \cup B$. The simplest interpretation of "the smallest possible" here is in terms of q-dimensional volume. Thus, it is natural to choose as M the ellipsoid in R^q centered at the origin and containing $F \cup B$ of the minimum q-dimensional volume. To form the indicated *ellipsoidal envelope* M of F and B is a convex problem; since normally q is not large, there is no difficulty to solve the problem numerically. Note, however, that there exists an "easy case" where M can be pointed out explicitly. Namely, let $L(F) \subset R^k$ be the linear span of F. Assume that

1. the loads $f_1, ..., f_k$ are linearly independent;
2. the convex hull $\hat{F}$ of the set $F \cup (-F)$ contains the k-dimensional ball $B' = B \cap L(F)$.

Note that in actual design both these assumptions normally are satisfied.

LEMMA 2.1. *Under the indicated assumptions the ellipsoidal envelope of F and B is*

$$M = QW_q, \quad Q = [f_1; ...; f_k; re_1; ...; re_{q-k}], \tag{2.4}$$

where $e_1, ..., e_{q-k}$ is an orthonormal basis in the orthogonal complement to $L(F)$ in R^q.

Proof. We can choose an orthonormal basis in R^q in such a way that the first k vectors of the basis span $L(F)$ and the rest $q-k$ vectors span the orthogonal complement $L^\perp(F)$ to $L(F)$ in R^q. Let $x=(u,v)$ be the coordinates of a vector in this basis (u are the first k and v are the rest $q-k$ coordinates). A centered at the origin ellipsoid E in R^q can be parameterized by a positive definite symmetric $q\times q$ matrix A:

$$E=\{x|\, x^TAx\le 1\};$$

the squared volume of E is inversely proportional to $\det A$. The matrix A_* corresponding to the minimum volume centered at the origin ellipsoid containing F and B is therefore an optimal solution to the following convex program:

$$\ln\det A\to\max\,|\,A=A^T>0,\ x^TAx\le 1\ \forall x\in B\cup\hat{F}. \tag{2.5}$$

The problem clearly is solvable, and since its objective is strictly concave on the cone of positive definite symmetric $q\times q$ matrices, the solution is unique. On the other hand, let J be the matrix of the mapping $(u,v)\mapsto(u,-v)$; then the mapping $A\mapsto J^TAJ$ clearly is a symmetry of (2.5). This mapping preserves feasibility and does not vary the value of the objective. We conclude that the optimal solution is invariant with respect to the indicated mapping: $A_*=JA_*J$, whence A_* is block diagonal with $k\times k$ diagonal block U_* and $(q-k)\times(q-k)$ diagonal block V_*. Since the ellipsoid $\{x|\,x^TA_*x\le 1\}$ contains $B\cup\hat{F}$, the k-dimensional ellipsoid $M'=\{u|\,u^TU_*u\le 1\}$ in $L(F)$ contains $\hat{F}$, while the $(q-k)$-dimensional ellipsoid $M''=\{v|\,v^TV_*v\le 1\}$ in $L^\perp(F)$ contains the ball B'' centered at the origin of the radius r in $L^\perp(F)$.

Now let $U=U^T>0$ and $V=V^T>0$ be $k\times k$ and $(q-k)\times(q-k)$ matrices such that the ellipsoids $E'=\{u|\,u^TUu\le 1\}$ and $E''=\{v|\,v^TVv\le 1\}$ contain $\hat{F}$ and B'', respectively. We claim that then the ellipsoid $\{x|\,x^TAx\le 1\}$, $A=\mathrm{Diag}(U,V)$, contains $B\cup\hat{F}$. Indeed, the ellipsoid clearly contains $\hat{F}$, and all we need is to verify that if $x=(u,v)\in B$, i.e., $u^Tu+v^Tv\le r^2$, then $u^TUu+v^TVv\le 1$. This is immediate: since $E'\supset\hat{F}\supset B'$, we have $u^TUu\le 1$ whenever $u^Tu\le r^2$, or, which is the same, $u^TUu\le r^{-2}u^Tu$ for all u. Similarly, $E''\supset B''$ implies that $v^TVv\le r^{-2}v^Tv$, so that $u^Tu+v^Tv\le r^2$ indeed implies $u^TUu+v^TVv\le 1$.

The above observations combined with the identity $\ln\det A=\ln\det U+\ln\det V$ for positive definite symmetric $A=\mathrm{Diag}(U,V)$ demonstrate that the block U_* of the optimal solution to (2.5) corresponds to the minimum volume ellipsoid in $L(F)$ containing $\hat{F}$, and similarly for V_*, $L^\perp(F)$ and B''. In other words, M is the "ellipsoidal product" of the ellipsoid M' of the minimum volume in $L(F)$ containing $F\cup(-F)$ and the ball B'' in $L^\perp(F)$. If $M'=Q'W_k$, then

$$M=[Q';re_1;...;re_{q-k}]\,W_q.$$

To conclude the proof, it suffices to verify that one can choose, as Q', the matrix $[f_1;...;f_k]$, which is immediate. Indeed, let $s_1,...,s_k$ be an orthonormal basis in $L(F)$, and let D be the linear transformation of $L(F)$ which maps s_i onto f_i, $i=1,...,k$. Since the ratio of k-dimensional volumes of solids in $L(F)$ remains invariant under the transformation D, $M'=DN'$, where N' is the minimum volume ellipsoid centered at the origin in $L(F)$ containing $s_1,...,s_k$. The latter ellipsoid is clearly $[s_1;...;s_k]\,W_k$, whence

$$M'=DN'=\left\{D\left(\sum_{i=1}^k\lambda_is_i\right)\,|\,\lambda\in W_k\right\}=\left\{\sum_{i=1}^k\lambda_if_i|\,\lambda\in W_k\right\}=[f_1;...;f_k]\,W_k. \qquad\square$$

Remark 2.1. Evident modification of the proof of Lemma 2.1 demonstrates that the minimum volume ellipsoid in R^q centered at the origin and containing $F \cup B$ always is the "ellipsoidal product" of the minimum volume ellipsoid M' in $L(F)$ containing $F \cup (-F) \cup B'$ and the ball B'' in $L^{\perp}(F)$. If $M' = Q' W_{\widehat{k}}$, $\widehat{k} = \dim L(F)$, then $M = [Q'; re_1, ..., re_{q-\widehat{k}}] W_q$, $e_1, ..., e_{q-\widehat{k}}$ being an orthonormal basis in $L^{\perp}(F)$. Thus, to find M is, basically, the same as to find M', and this latter convex problem is normally of quite a small dimension, since $\widehat{k} \le k$ and typically $k \le 5$.

The outlined way of modeling the robustness constraint is, perhaps, more reasonable than the usual multiload setting of the TTD problem. Indeed, the new model enforces certain level of rigidity of the resulting construction with respect not only to the primary loads, but also to loads associated with "active" nodes. At the same time, it turns out, as we are about to demonstrate, that the resulting problem ($\mathrm{TD}_{\mathrm{robust}}$) is basically not more computationally demanding than the usual multiload TTD problem of the same size (i.e., with the same ground structure and the number of scenario loads equal to the dimension of the loading ellipsoid used in ($\mathrm{TD}_{\mathrm{robust}}$)).

2.4. Semidefinite reformulation of ($\mathbf{TD}_{\mathbf{robust}}$). Our goal now is to rewrite ($\mathrm{TD}_{\mathrm{robust}}$) equivalently as a so-called *semidefinite program.* To this end we start with the following simple result.

LEMMA 2.2. *Let A be a positive semidefinite $n \times n$ matrix, and let*

$$c = \max_{x \in R^n; e \in R^q : e^T e \le 1} \left[2(Qe)^T x - x^T A x \right]. \tag{2.6}$$

Then the inequality $c \le \tau$ is equivalent to positive semidefiniteness of the matrix

$$\mathcal{A} = \begin{pmatrix} \tau I_q & Q^T \\ Q & A \end{pmatrix},$$

I_q being the unit $q \times q$ matrix.

Proof. We have

$$\begin{aligned}
& c \le \tau \Leftrightarrow \forall (x \in R^n, e \in R^q, e^T e \le 1): \quad \tau - 2(Qe)^T x + x^T A x \ge 0 \Leftrightarrow \\
& \qquad \text{[by homogeneity reasons]} \\
& \forall (\lambda > 0, x \in R^n, e \in R^q, e^T e \le 1): \quad \tau\lambda^2 - 2(Q\lambda e)^T(\lambda x) + (\lambda x)^T A (\lambda x) \ge 0 \Leftrightarrow \\
& \qquad \text{[set } \lambda e = f, \lambda x = y] \\
& \forall (\lambda > 0, y \in R^n, f \in R^q, f^T f \le \lambda^2): \quad \tau\lambda^2 - 2(Qf)^T y + y^T A y \ge 0 \Rightarrow \\
& \forall \left(\begin{pmatrix} f \\ y \end{pmatrix} \in R^{q+n} \right): \quad \begin{pmatrix} f \\ y \end{pmatrix}^T \begin{pmatrix} \tau I_q & Q^T \\ Q & A \end{pmatrix} \begin{pmatrix} f \\ y \end{pmatrix} \equiv \tau f^T f - 2(Qf)^T y + y^T A y \ge 0.
\end{aligned}$$

Thus, $\tau \ge c \Rightarrow \mathcal{A} \ge 0$. Vice versa, if $\mathcal{A} \ge 0$, then clearly $\tau \ge 0$, and, therefore, the implication $\Rightarrow$ in the above chain can be inverted. ∎

Remark 2.2. It is well known that a symmetric matrix $\begin{pmatrix} U & Q^T \\ Q & A \end{pmatrix}$ with positive definite U is positive semidefinite if and only if $A \ge QU^{-1}Q^T$. Applying this observation to the case of $U = \tau I_q$, we can reformulate the result of Lemma 2.2 as follows:

The compliance c of a truss t, with respect to the ellipsoid of loads $M = QW_q$ is $\le \tau$ if and only if $A(t) \ge \tau^{-1} QQ^T$.

In the particular case when QQ^T is the orthoprojector P onto the linear span L of the columns of Q, the above observation can be reformulated as follows:

$c \le \tau$ if and only if the minimum eigenvalue of the restriction of $A(t)$ onto L is $\ge \tau^{-1}$.

(In the general case, the interpretation is similar, but instead of the usual minimum eigenvalue of the restriction we should speak about minimum eigenvalue of the matrix pencil $(A|_L, QQ^T|_L)$ on L.)

In view of Lemma 2.2, problem $(\mathrm{TD}_{\mathrm{robust}})$ can be rewritten equivalently as the following *semidefinite program*:

$$(\mathrm{TD}_{\mathrm{sd}}) \qquad \min_{t\in R^m,\tau\in R} \tau$$

$$\text{subject to} \quad \begin{array}{rcl} \begin{pmatrix} \tau I_q & Q^T \\ Q & A(t) \end{pmatrix} & \geq & 0, \\ t & \geq & 0, \\ \sum_{i=1}^m t_i & = & V. \end{array}$$

(Here and in what follows the inequality $A \geq B$ between symmetric matrices means that the matrix $A - B$ is positive semidefinite.)

3. Deriving a dual problem to $(\mathbf{TD_{sd}})$. Here we derive the Fenchel–Rockafellar [11] dual to the problem $(\mathrm{TD}_{\mathrm{sd}})$. The latter problem is of the form

$$\min\{\tau : \ \mathcal{A}(\tau, t) + B \in \mathbf{S}_+, \, t \in T\},$$

where

$$\mathcal{A}(\tau, t) = \begin{pmatrix} \tau I_q & 0 \\ 0 & A(t) \end{pmatrix}$$

is a linear mapping from $R \times R^n$ to the space $\mathbf{S}$ of symmetric $(n+q)\times(n+q)$ matrices equipped with the standard Frobenius Euclidean structure $\langle X, Y\rangle = \mathrm{Tr}(XY)$, $\mathbf{S}_+$ is the cone of positive semidefinite matrices from $\mathbf{S}$ and

$$B = \begin{pmatrix} 0 & Q^T \\ Q & 0 \end{pmatrix} \in \mathbf{S}.$$

We write the problem in the Fenchel–Rockafellar primal scheme:
(P) $\quad \min\{f(\tau, t) - g(\mathcal{A}(\tau, t))\}$,
where

$$f(\tau, t) = \tau + \delta(t|T), \quad g(X) = -\delta(X + B|\mathbf{S}_+)$$

and $\delta(x|W)$ is the indicator function of a set W. To derive the dual to (P), we need to compute the conjugates f^* and g_* of the convex function f and the concave function g, which is quite straightforward:

$$\begin{array}{rcl} f^*(\sigma, s) & = & \sup_{\tau,t}\{\sigma\tau + s^T t - \tau |\, t \in T\} = \begin{cases} V \max_{1\le i\le n} s_i, & \sigma = 1, \\ +\infty & \text{otherwise}, \end{cases} \\ g_*(R) & = & \inf_S\{\mathrm{Tr}(SR) | S + B \in \mathbf{S}_+\} = \inf\{\mathrm{Tr}((Z - B)R) | Z \in \mathbf{S}_+\} \\ & = & \begin{cases} -\mathrm{Tr}(BR), & R \in \mathbf{S}_+, \\ -\infty & \text{otherwise}. \end{cases} \end{array}$$

(We have used the well-known fact that the cone of positive semidefinite matrices is self-conjugate with respect to the Frobenius Euclidean structure.)

The Fenchel–Rockafellar dual to (P) is

(D) $\sup_{R\in\mathbf{S}}\{g_*(R) - f^*(\mathcal{A}^*R)\}$,

where $\mathcal{A}^* : \mathbf{S} \to R \times R^n$ is the adjoint to $\mathcal{A}$.

Representing $R \in \mathbf{S}$ in the block form

$$R = \begin{pmatrix} \Lambda & X^T \\ X & Y \end{pmatrix}$$

(Λ is $q \times q$, Y is $n \times n$), we get

$$\mathcal{A}^*R = \begin{pmatrix} \tau = \operatorname{Tr}\Lambda \\ t_1 = \operatorname{Tr}(A_1Y) \\ \dots \\ t_n = \operatorname{Tr}(A_nY) \end{pmatrix}.$$

Substituting the resulting expressions for f^*, g_*, and A^*, we come to the following explicit formulation of the dual problem (D):

(D) $\max\left[-2\operatorname{Tr}(QX^T) - V\max_{i=1,\dots,m}[\operatorname{Tr}(A_iY)]\right]$,

s.t.

$$\begin{pmatrix} \Lambda & X^T \\ X & Y \end{pmatrix} \geq 0,\ \operatorname{Tr}\Lambda = 1,$$

the design variables being symmetric $q \times q$ and $n \times n$ matrices Λ, Y, respectively, and $n \times q$ matrix X.

Note that the functions f and g in (P) are clearly closed convex and concave, respectively. Moreover, from the well-posedness assumption **A**, it immediately follows that (P) is strictly feasible (i.e., the relative interiors of the domains of $f(\tau, t)$ and $\phi(\tau, t) = g(\mathcal{A}(\tau, t))$ have nonempty intersection, and the image of the mapping $\mathcal{A}$ intersects the interior of the domain of g); to see this, choose arbitrary positive $t \in T$ and enforce τ to be large enough. Of course (P) is bounded below (the compliance always is nonnegative); thus, all requirements of the Fenchel–Rockafellar duality theorem are satisfied, and we come to the following.

PROPOSITION 3.1. (D) *is solvable, and the optimal values in* (P) *and* (D) *are equal to each other.*

Remark 3.1. Until now, we dealt with the TTD problem with *simple constraints* on the bar volumes:

$$t \in T = \left\{ t \in R^n |\, t \geq 0, \sum_{i=1}^n t_i = V \right\}.$$

In the case when there are also lower and upper bounds on the bar volumes so that the constraints on t are

$$t \in T^+ = \{t \in T |\, L \leq t \leq U\}$$

($U > L \geq 0$ are given n-dimensional vectors), the above derivation results in a dual problem as follows:

($\mathrm{D_b}$) $\max\left[-2\operatorname{Tr}(QX^T) - \lambda V - \sum_{i=1}^n \max\left[(\operatorname{Tr}(YA_i) - \lambda)L_i; (\operatorname{Tr}(YA_i) - \lambda)U_i\right]\right]$

s.t.

$$\begin{pmatrix} \Lambda & X^T \\ X & Y \end{pmatrix} \geq 0,\ \operatorname{Tr}\Lambda = 1,$$

the design variables being real λ, symmetric $q \times q$ matrix Λ, symmetric $n \times n$ matrix Y, and $n \times q$ matrix X.

4. A simplification of the dual problem (D). Our next goal is to simplify problem (D), derived in the previous section, by eliminating the matrix variable Y. To this end it suffices to note that (D) can be rewritten as

$$
\begin{array}{llcl}
(\mathrm{TD}_{\mathrm{dl}}) & & & \\
& \min\limits_{X\in R^{n\times q},\Lambda=\Lambda^T\in R^{q\times q},Y=Y^T\in R^{n\times n},\rho\in R} & & [2\,\mathrm{Tr}(QX^T)+V\rho] \\
\text{s.t.} & & & \\
(\alpha) & \mathrm{Tr}(YA_i) & \le & \rho,\ i=1,\dots,m, \\
(\beta) & \begin{pmatrix} \Lambda & X^T \\ X & Y \end{pmatrix} & \ge & 0, \\
(\gamma) & \mathrm{Tr}(\Lambda) & = & 1.
\end{array}
$$

(We have replaced the maximization problem (D) by an equivalent minimization one.) Note that $(\mathrm{TD}_{\mathrm{dl}})$ is strictly feasible—there exists a feasible solution where all scalar inequality constraints and the matrix inequality one are strict (take $\Lambda = q^{-1}I_q$, $Y = I_n$, and enforce ρ to be large enough).

The matrix inequality (β) clearly implies that Λ is positive semidefinite. Thus, we do not vary $(\mathrm{TD}_{\mathrm{dl}})$ when adding (in fact, redundant) inequality $\Lambda \ge 0$. Now let us strengthen, for a moment, the latter inequality to

$$\Lambda > 0, \tag{4.1}$$

i.e., to positive definiteness of Λ; it is immediately seen from strict feasibility of $(\mathrm{TD}_{\mathrm{dl}})$ that the transformation does not violate the optimal value of the problem, although it may cut off the optimal solution (anyhow, from the computational viewpoint the exact solution is nothing but a fiction). Thus, we may focus on the problem $(\mathrm{TD}'_{\mathrm{dl}})$ obtained from $(\mathrm{TD}_{\mathrm{dl}})$ by adding to the list of constraints inequality (4.1).

The pair of matrix inequalities (β), (4.1), which are present among the constraints of $(\mathrm{TD}'_{\mathrm{dl}})$, is equivalent to the pair of matrix inequalities

$$\Lambda > 0; \quad Y \ge Y^*(\Lambda, X) = X\Lambda^{-1}X^T.$$

Now let (Λ, X, Y, ρ) be a feasible solution to $(\mathrm{TD}'_{\mathrm{dl}})$; then, as we have just mentioned, $Y \ge Y^*(\Lambda, X)$ and the collection $(\Lambda, X, Y^* = Y^*(\Lambda, X), \rho)$ satisfies (β), (γ) and (4.1). Moreover, since A_i are symmetric positive semidefinite and $Y \ge Y^*$, we have $\mathrm{Tr}(YA_i) \ge \mathrm{Tr}(Y^*A_i)$ so that the updated collection satisfies (α) as well, and (Λ, X, Y^*, ρ) is feasible for $(\mathrm{TD}'_{\mathrm{dl}})$. Note that the transformation $(\Lambda, X, Y, \rho) \mapsto (\Lambda, X, Y^*(\Lambda, X), \rho)$ does not affect the objective function of the problem. We conclude that $(\mathrm{TD}'_{\mathrm{dl}})$ can be equivalently rewritten as

$$
\begin{array}{ll}
 & \min_{X\in R^{n\times q},\Lambda=\Lambda^T\in R^{q\times q},\rho\in R}\ 2\,\mathrm{Tr}(QX^T)+V\rho \\
\text{s.t.} & \\
 & \Lambda > 0,\ \mathrm{Tr}(\Lambda)=1,\ \rho \ge \mathrm{Tr}(X\Lambda^{-1}X^TA_i),\ i=1,\dots,m.
\end{array}
$$

Substituting $A_i = b_ib_i^T$ (see (2.1)), we can rewrite the constraints

$$\rho \ge \mathrm{Tr}(X\Lambda^{-1}X^TA_i)$$

as

$$\rho \ge (X^Tb_i)^T\Lambda^{-1}(X^Tb_i),$$

which is the same (since $\Lambda = \Lambda^T > 0$) as

$$\begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix} \geq 0.$$

With this substitution, the problem $(\mathrm{TD}'_{\mathrm{dl}})$ becomes

$$\begin{array}{ll} & \min_{X \in R^{n\times q}, \Lambda=\Lambda^T \in R^{q\times q}, \rho\in R} \; 2\,\mathrm{Tr}(QX^T) + V\rho \\ \text{s.t.} & \\ & \Lambda > 0,\ \mathrm{Tr}(\Lambda) = 1,\ \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix} \geq 0,\ i = 1, ..., m. \end{array}$$

When replacing the strict inequality $\Lambda > 0$ in the latter problem with the nonstrict one $\Lambda \geq 0$, we clearly do not vary the optimal value of the problem; in the modified problem, the inequality $\Lambda \geq 0$ is in fact redundant (it follows from positive semidefiniteness of any of the matrices $\begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix}$). With these modifications, we come to the final formulation of the problem dual to $(\mathrm{TD}_{\mathrm{robust}})$:

$$\begin{array}{lrcl} (\mathrm{TD}_{\mathrm{fn}}) & & & \\ & \min\limits_{\Lambda=\Lambda^T\in R^{q\times q}, X\in R^{n\times q}, \rho\in R} & & 2\,\mathrm{Tr}(QX^T) + V\rho \\ \text{s.t.} & & & \\ & \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix} & \geq & 0,\ \ i = 1, ..., m, \\ & \mathrm{Tr}(\Lambda) & = & 1. \end{array}$$

Note that $(\mathrm{TD}_{\mathrm{fn}})$ is very similar to the standard multiload TTD problem in dual setting [5]; the only difference is that in the latter problem Λ is further restricted to be diagonal.

5. Recovering the bar volumes. Until now, the only relation between the initial primal problem $(\mathrm{TD}_{\mathrm{robust}})$ and the dual one $(\mathrm{TD}_{\mathrm{fn}})$ is that their optimal values are negations of each other (note that when coming to $(\mathrm{TD}_{\mathrm{fn}})$ from the maximization problem $(\mathrm{TD}_{\mathrm{dl}})$, which has the same optimal value as $(\mathrm{TD}_{\mathrm{sd}})$, we have changed the sign of the objective and have replaced maximization with minimization). Thus, the problem arises: how to restore good approximate solutions to $(\mathrm{TD}_{\mathrm{robust}})$ via good approximate solutions to $(\mathrm{TD}_{\mathrm{fn}})$. To resolve this problem, we first derive the Fenchel–Rockafellar dual $(\mathrm{TD}^*_{\mathrm{fn}})$ to $(\mathrm{TD}_{\mathrm{fn}})$ and recognize in it the initial problem $(\mathrm{TD}_{\mathrm{robust}})$, and then use the well-known relation in interior point theory between "central path" approximate solutions to $(\mathrm{TD}_{\mathrm{fn}})$ and approximate solutions to $(\mathrm{TD}^*_{\mathrm{fn}})$.

5.1. A dual problem to $(\mathbf{TD_{fn}})$. Similar to the above, we represent problem $(\mathrm{TD}_{\mathrm{fn}})$ in the Fenchel–Rockafellar scheme:

(PI) $\quad \min \{f(\Lambda, X, \rho) - g(\mathcal{A}(\Lambda, X, \rho))\}$,

where

$$f(\Lambda, X, \rho) = 2\,\mathrm{Tr}(QX^T) + V\rho + \delta(\mathrm{Tr}(\Lambda)|\{1\}),$$

$$\mathcal{A}(\Lambda, X, \rho) = \mathrm{Diag}\left\{ \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix}, i = 1, ..., m \right\}$$

is the linear mapping from the space of design variables of $(\mathrm{TD_{fn}})$ to the space $\mathbf{S}$ of block-diagonal symmetric matrices with m diagonal blocks of the sizes $(q+1)\times(q+1)$ each, and

$$g(W) = -\delta(W|\mathbf{S}_+),$$

$\mathbf{S}_+$ being the cone of positive semidefinite matrices from $\mathbf{S}$.

The dual to (P) is

$$\text{(DI)} \qquad \max_{R\in\mathbf{S}} \{g_*(R) - f^*(\mathcal{A}^*R)\},$$

where $\mathcal{A}^*$ is the operator adjoint to $\mathcal{A}$. Here

$$\begin{array}{rcl} f^*(L,\Xi,r) & = & \sup_{\Lambda,X,\rho}\left[\mathrm{Tr}(\Lambda L) + \mathrm{Tr}(\Xi X^T) + r\rho - f(\Lambda,X,\rho)\right] \\ & = & \sup_\Lambda\left[\mathrm{Tr}(\Lambda L) - \delta(\mathrm{Tr}(\Lambda)|\{1\})\right] + \sup_X\left[\mathrm{Tr}(\Xi X^T) - 2\,\mathrm{Tr}(QX^T)\right] \\ & & +\sup_\rho\left[r\rho - V\rho\right] \\ & = & \dfrac{1}{q}\,\mathrm{Tr}(L) + \delta((L,\Xi,r)|\{(L=\lambda I_q, 2Q, V)|\,\lambda\in R\}) \\ & = & \begin{cases} \lambda & \text{if } L=\lambda I_q \text{ for some } \lambda\in R \text{ and } \Xi = 2Q,\ r=V, \\ \infty & \text{otherwise} \end{cases} \end{array}$$

and

$$g_*(R) = \inf_S\left[\mathrm{Tr}(SR) + \delta(S|\mathbf{S}_+)\right] = -\delta(R|\mathbf{S}_+).$$

(Here we again used the fact that the cone $\mathbf{S}_+$ is self-dual with respect to the Frobenius Euclidean structure of $\mathbf{S}$.)

Denoting a generic element of $\mathbf{S}$ as

$$R = \mathrm{Diag}\left\{\begin{pmatrix} L_i & d_i \\ d_i^T & t_i \end{pmatrix},\ i=1,...,m\right\}$$

(L_i are symmetric $q\times q$ matrices, d_i are q-dimensional vectors, t_i are reals) it can be seen that

$$\mathcal{A}^*R = \left(L = \sum_{i=1}^m L_i, \Xi = 2\sum_{i=1}^m b_i d_i^T, r = \sum_{i=1}^m t_i\right).$$

With these relations, the dual (DI) to (PI) becomes

$$\begin{array}{llcl} (\mathrm{TD}^*_{\mathrm{fn}}) & \min\limits_{\lambda\in R, L_i=L_i^T\in R^{q\times q}, d_i\in R^q, t_i\in R} \lambda & & \\ \text{s.t.} & & & \\ (\alpha) & \sum\limits_{i=1}^m L_i & = & \lambda I_q, \\ (\beta) & \sum\limits_{i=1}^m b_i d_i^T & = & Q, \\ (\gamma) & \sum\limits_{i=1}^m t_i & = & V, \\ (\delta) & \begin{pmatrix} L_i & d_i \\ d_i^T & t_i \end{pmatrix} & \geq & 0,\ i=1,...,m. \end{array}$$

(We again have replaced a maximization problem with the equivalent minimization one.)

Problem $(\mathrm{TD}_{\mathrm{fn}})$ clearly satisfies the assumption of the Fenchel–Rockafellar duality theorem, and this together with Proposition 3.1 proves the following.

PROPOSITION 5.1. *Problem* $(\mathrm{TD}^*_{\mathrm{fn}})$ *is solvable, and its optimal value* λ^* *is equal to the optimal value* c^* *of the initial problem* $(\mathrm{TD}_{\mathrm{robust}})$.

It is not difficult to guess that the variables t_i involved into $(\mathrm{TD}^*_{\mathrm{fn}})$ can be interpreted as our initial bar volumes t_i. The exact statement is given by the following theorem.

THEOREM 5.2. *Let* $R = \{\lambda; L_i, d_i, t_i,\ i = 1, ..., m\}$ *be a feasible solution to* $(\mathrm{TD}^*_{\mathrm{fn}})$. *Then the vector* $t = (t_1, ..., t_m)$ *is a feasible solution to* $(\mathrm{TD}_{\mathrm{robust}})$, *and the value of the objective of the latter problem at* t *is less than or equal to* λ. *In particular, if* R *is an* ϵ*-solution to* $(\mathrm{TD}^*_{\mathrm{fn}})$ *(i.e.,* $\lambda - \lambda^* \le \epsilon$*), then* t *is an* ϵ*-solution to* $(\mathrm{TD}_{\mathrm{robust}})$ *(i.e.,* $c^M(t) - c^* \le \epsilon$*).*

Proof. The "in particular" part of the statement follows from its first part due to Proposition 5.1, and all we need is to prove the first part. From the positive semidefiniteness constraints (δ) in $(\mathrm{TD}^*_{\mathrm{fn}})$ it follows that $t \ge 0$, which combined with (γ) implies the inclusion $t \in T$. To complete the proof, we should verify that $c^M(t) \le \lambda$.

Let $e \in R^q$, $e^T e \le 1$. From (β) we have

$$Qe = \sum_{i=1}^{m} (d_i^T e) b_i.$$

Let $x \in R^n$. Due to $A_i = b_i b_i^T$, we have

$$\begin{aligned}
\phi_e(x) &\equiv 2(Qe)^T x - x^T A(t) x \\
&= \sum_{i=1}^{m} 2(d_i^T e)(b_i^T x) - t_i \sum_{i=1}^{m} (b_i^T x)^2 \\
&= \sum_{i=1}^{m} \left[2(d_i^T e)(b_i^T x) - t_i (b_i^T x)^2 \right] \\
&\qquad [\text{denoting } s_i = -b_i^T x] \\
&= -\sum_{i=1}^{m} \left[e^T L_i e + 2(d_i^T e) s_i + t_i s_i^2 \right] + \sum_{i=1}^{m} e^T L_i e \\
&= -\sum_{i=1}^{m} \begin{pmatrix} e \\ s_i \end{pmatrix}^T \begin{pmatrix} L_i & d_i \\ d_i^T & t_i \end{pmatrix} \begin{pmatrix} e \\ s_i \end{pmatrix} + \sum_{i=1}^{m} e^T L_i e \\
&\qquad [\text{by } (\delta)] \\
&\le \sum_{i=1}^{m} e^T L_i e \\
&\qquad [\text{by } (\alpha)] \\
&= \lambda.
\end{aligned}$$

Thus, $\phi_e(x) \le \lambda$ for all x. By definition, $c^M(t)$ is the upper bound of $\phi_e(x)$ over x, and the inequality $c^M(t) \le \lambda$ then follows. ◻

Remark 5.1. Note that $(\mathrm{TD}^*_{\mathrm{fn}})$ is a natural modification of the "bar-forces" formulation of the usual multiload TTD problem; see [5].

6. Solving ($\mathrm{TD_{fn}}$) and ($\mathrm{TD^*_{fn}}$) via interior point methods. Among numerical methods available for solving semidefinite programs like ($\mathrm{TD_{fn}}$) and ($\mathrm{TD^*_{fn}}$), the most attractive (and, in fact, the only meaningful in the large scale case) are the recent interior point algorithms (for relevant general theory, see [10]). Here we discuss the corresponding possibilities. In what follows we restrict ourselves to outlining the main elements of the construction, since our goal now is not to present detailed description of the algorithms, but to demonstrate the following.

(i) From the above semidefinite programs related to truss topology design with robustness constraints, the most convenient for numerical processing by interior point methods is the problem ($\mathrm{TD_{fn}}$).

(ii) Solving ($\mathrm{TD_{fn}}$) by *interior point path-following methods*, one has the possibility of generating, as a byproduct, good approximate solutions to the problem of interest ($\mathrm{TD^*_{fn}}$), i.e., of recovering the primal design variables (bar volumes).

When solving a generic semidefinite program

(SP) $\qquad \sigma^T\xi \to \min \mid \mathcal{A}(\xi) \in \mathbf{S}_+,$

$\xi \in R^N$ being the design vector, $\mathcal{A}(\xi)$ being an affine mapping from R^N to the space $\mathbf{S}$ of symmetric matrices of certain fixed block-diagonal structure, and $\mathbf{S}_+$ being the cone of positive semidefinite matrices from $\mathbf{S}$, by a path-following interior point method, one defines the family of barrier-type functions

$$F_s(\xi) = s\sigma^T\xi + \Phi(\mathcal{A}(\xi)), \quad \Phi(\Xi) = -\ln \mathrm{Det}\, \Xi,$$

and traces the *central path*—the path of minimizers

$$\xi^*(s) = \operatorname*{argmin}_{\xi \in \mathrm{Dom}\, F_s} F_s(\xi).$$

If (SP) is strictly feasible (i.e., $\mathcal{A}(\xi)$ is positive definite for certain ξ) and the level sets

$$\{\xi \in R^N \mid \mathcal{A}(\xi) \in \mathbf{S}_+, \sigma^T\xi \le a\},$$

$a \in R$, are bounded, then the path ξ^* is well defined and converges, as $s \to \infty$, to the optimal set of the problem. In the path-following scheme, one generates close (in certain exact sense) approximations ξ_i to the points $\xi^*(s_i)$ along certain sequence $\{s_i\}$ of penalty parameters "diverging to ∞ fast enough," thus generating a sequence of strictly feasible approximate solutions converging to the optimal set. Updating $(s_i, \xi_i) \mapsto (s_{i+1}, \xi_{i+1})$ is as follows: first, we increase, according to certain rule, the current value s_i to a larger value s_{i+1}. Second, we restore closeness to the path of the new point $\xi^*(s_{i+1})$ by running the *damped Newton method*—the recurrence

$$\begin{array}{rcl} y & \mapsto & y^+ = y - (1 + \lambda(F_s, y))^{-1}[\nabla_y^2 F_s(y)]^{-1}\nabla_y F_s(y), \\ \lambda(F_s, y) & = & \sqrt{\nabla_y^T F_s(y)[\nabla_y^2 F_s(y)]^{-1}\nabla_y F_s(y)}, \end{array} \tag{6.1}$$

with s set to s_{i+1}. The recurrence is started at $y = \xi_i$ and is terminated when, for the first time, it turns out that $\lambda(F_{s_{i+1}}, y) \le \kappa$, $\kappa \in (0,1)$ being a once forever fixed threshold. (Thus, the exact meaning of "closeness of a point ξ to the point $\xi^*(s)$" is given by the inequality $\lambda(F_s, \xi) \le \kappa$. In what follows, for the sake of definiteness, it is assumed that $\kappa = 0.1$.) The resulting y is chosen as ξ_{i+1}, and the process is iterated.

The following is known:

(i) it is possible to trace the path "quickly": with reasonable policy of updating the values of the penalty parameter, it takes, for any $T > 2$, no more than

$$M = M(T) = O(1)\sqrt{\mu}\ln T$$

Newton steps (6.1) to come from a point ξ_0 close to $\xi^*(s_0)$ to a point ξ_M close to $\xi^*(s_M)$ with $s_M \geq Ts_0$; here μ is the total row size of the matrices from $\mathbf{S}$ and $O(1)$ is an absolute constant;

(ii) if ξ is close to $\xi(s)$, then the quality of ξ as an approximate solution to (SP) can be expressed via the value of s alone:

$$\sigma^T\xi - \sigma^* \leq \frac{2\mu}{s}, \tag{6.2}$$

σ^* being the optimal value in (SP);

(iii) being close to the path, it is easy to come "very close" to it; if $\lambda \equiv \lambda(F_s, y) \leq 0.1$, then (6.1) results in

$$\lambda^+ \equiv \lambda(F_s, y^+) \leq 2.5\lambda^2. \tag{6.3}$$

Although the indicated remarks deal with the path-following scheme only, the conclusions related to the number of "elementary steps" required to solve a semidefinite program to a given accuracy and to the complexity of a step (dominated by the computational cost of the Newton direction; see (6.1)) are valid for other interior point methods for semidefinite programming. The "integrated" complexity characteristic of an interior point method for (SP) is the quantity

$$\mathcal{C} = \sqrt{\mu}\mathcal{C}_{\rm Nwt},$$

where $\mathcal{C}_{\rm Nwt}$ is the arithmetic cost of computing the Newton direction. Indeed, according to the above remarks, it takes $O(1)\sqrt{\mu}$ Newton steps to increase the value of the penalty by an absolute constant factor, or, which is the same, to reduce by the same factor the (natural upper bound for) inaccuracy of the current approximate solution.

Now let us look at the complexity characteristic $\mathcal{C}$ for the semidefinite programs related to $(\mathrm{TD}_{\rm robust})$. In the table below we write down the principal terms of the corresponding quantities (omitting absolute constant factors); it is assumed (as it is normally the case for TTD) that

$$m = O(n^2); \quad q << n.$$

The expression for $\mathcal{C}_{\rm Nwt}$ corresponds to the "explicit" policy when we first assemble, in the natural manner, the Hessian matrix $\nabla^2_\xi F_s(\cdot)$ and then solve the resulting Newton system by traditional direct linear algebra routines like Choleski decomposition. It turns out that the specific structure of matrix inequalities in our problems[3] allows us to assemble the Hessians at a relatively low cost, so that the cost of a single Newton step is dominated by the complexity of Choleski factorization of the Hessian, i.e., by cube of the design dimension of the corresponding problem. With this remark, we come to the results as follows:

[3]In particular, the fact that in TTD design each of the vectors b_i has $O(1)$ nonzero entries—at most four in the case of 2D and at most six in the case of 3D trusses.

Model	μ	$\mathcal{C}_{\mathrm{Nwt}}$	$\mathcal{C}$
$(\mathrm{TD}_{\mathrm{sd}})$	m	m^3	$m^{3.5} \approx n^7$
$(\mathrm{TD}_{\mathrm{dl}})$	m	m^3	$m^{3.5} \approx n^7$
$(\mathrm{TD}_{\mathrm{fn}})$	qm	$q^3 n^3$	$q^{3.5} n^4$
$(\mathrm{TD}^*_{\mathrm{fn}})$	qm	$q^6 m^3$	$q^{6.5} m^{3.5} \approx q^{6.5} n^7$

The reader should be aware that there are "implicit" schemes of computing the Newton direction in $(\mathrm{TD}^*_{\mathrm{fn}})$ with arithmetic cost $O(q^3n^3)$ (the same as in $(\mathrm{TD}_{\mathrm{fn}})$). Thus, in fact, the primal and dual problems in primal-dual pairs $((\mathrm{TD}_{\mathrm{sd}}), (\mathrm{TD}_{\mathrm{dl}}))$, $((\mathrm{TD}_{\mathrm{fn}}), (\mathrm{TD}^*_{\mathrm{fn}}))$ are theoretically equivalent in complexity; moreover, there are "symmetric" primal-dual methods which solve simultaneously the primal-dual pair of the problems at the complexity, respectively, $O(n^7)$ and $O(q^{3.5}n^4)$. Nevertheless, we believe that at the moment practical considerations still are in favor of "purely primal" methods as applied to $(\mathrm{TD}_{\mathrm{sd}})$ in the first primal-dual pair and to $(\mathrm{TD}_{\mathrm{fn}})$ in the second pair. The reason is that the feasible planes $\mathcal{L}$ in the "unfavorable" problems of the above pairs are given by linear equalities, while in the "favorable" components of the pairs they are parameterized (from the very beginning they are represented as images of affine mappings). Now, the theoretically efficient way to compute the Newton direction for an "unfavorable" problem represents the direction as the difference of a certain "exactly known" vector and its projection on the orthogonal complement to $\mathcal{L}$. Such a computation is relatively unstable—rounding errors make the actually computed Newton directions nonparallel to $\mathcal{L}$, and the iterates eventually become far from the feasible plane. In order to overcome this instability, in the existing software for semidefinite problems, "expensive" linear algebra routines, like QR factorization, are used, at least at the final phase of computations. In contrast to this, in the "favorable" problems the Newton direction is computed in the space of parameters identifying a point on the feasible plane, so that there is no danger of being kicked off this plane.

With the above remarks, it is clear that among the semidefinite programs we introduced, the most convenient for numerical processing by interior point methods is $(\mathrm{TD}_{\mathrm{fn}})$, as it was claimed in **I**. There is, however, an a priori drawback of this approach; what we need are the bar volumes, and they "are not seen" at all in $(\mathrm{TD}_{\mathrm{fn}})$. We are about to demonstrate that in order to overcome this difficulty it suffices to solve $(\mathrm{TD}_{\mathrm{fn}})$ not by an arbitrary interior point method, but with a path-following one.

Assume that we are applying a path-following method to $(\mathrm{TD}_{\mathrm{fn}})$ and have computed a point $\xi = (\Lambda, X, \rho)$ close (in the aforementioned sense) to the point $\xi^*(s)$. From (6.3) it follows that a small number of steps of the recurrence (6.1) started at ξ allows to come "very close" to $\xi^*(s)$ (six steps of the recurrence restore $\xi^*(s)$ within machine accuracy). We may, therefore, assume for the sake of simplicity that we can "stand at the path," i.e., operate with $\xi^*(s)$ itself rather than with a tight approximation of the point.[4] It turns out that given $\xi^*(s)$, one can explicitly generate a feasible solution to $(\mathrm{TD}^*_{\mathrm{fn}})$ of inaccuracy $\leq O(1/s)$. The exact statement is as follows.

PROPOSITION 6.1. *Let $s > 0$, and let $\xi^*(s) = (\Lambda_s, X_s, \rho_s)$ be the minimizer of the function*

$$F_s(\Lambda, X, \rho) = s\left[2\,\mathrm{Tr}(QX^T) + V\rho\right] + \Phi(\mathcal{A}(\Lambda, X, \rho)) \tag{6.4}$$

[4]This is an idealization, of course, but it is as well motivated as the standard model of precise real arithmetic. We could replace in the forthcoming considerations $\xi^*(s)$ by its tight approximation, with minor modification of the construction, but we do not think it makes sense.

over the set of strictly feasible solutions to $(\mathrm{TD_{fn}})$. *Here*

$$\Phi(S) = -\ln \mathrm{Det}\, S : \mathrm{int}\, \mathbf{S}_+ \to R. \tag{6.5}$$

$\mathbf{S}$ *is the space of block-diagonal symmetric matrices with* m *diagonal blocks of the size* $(q+1)\times(q+1)$ *each, and*

$$\mathcal{A}(\Lambda, X, \rho) = \mathrm{Diag}\left\{ \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix},\ i = 1, ..., m \right\}. \tag{6.6}$$

Then the matrix

$$\begin{array}{rcl} R(s) & \equiv & \mathrm{Diag}\left\{ \begin{pmatrix} L_i & d_i \\ d_i^T & t_i \end{pmatrix} i = 1, ..., m \right\} \\ & = & s^{-1}\mathcal{A}^{-1}(\Lambda_s, X_s, \rho_s) \quad \left[= -s^{-1}\nabla_S|_{S=\mathcal{A}(\Lambda_s,X_s,\rho_s)}\Phi(S)\right] \end{array} \tag{6.7}$$

is such that $\sum_{i=1}^m L_i = \lambda_s I_q$ *for some real* λ_s, *and*

$(R(s), \lambda_s)$ *is a feasible solution to* $(\mathrm{TD}^*_{\mathrm{fn}})$ *with the value of the objective*

$$\lambda_s \le c^* + \frac{\mu}{s}, \tag{6.8}$$

where c^* *is the optimal value in* $(\mathrm{TD}^*_{\mathrm{fn}})$ *and* $\mu = m(q+1)$ *is the total row size of the matrices from* $\mathbf{S}$.

The proposition is an immediate consequence of general results of [10]; to make the paper self-contained, below we present a direct proof.

Let us set $Y = \mathcal{A}(\Lambda_s, X_s, \rho_s)$, $Z = Y^{-1}$, so that

$$R(s) = s^{-1}Z; \quad \nabla\Phi(Y) = -Z.$$

The set G of strictly feasible solutions to $(\mathrm{TD_{fn}})$ is comprised of all triples $\xi = (\Lambda, X, \rho)$, which correspond to positive definite $\mathcal{A}(\xi)$ and are such that $\mathrm{Tr}\,\Lambda = 1$; this is an open convex subset in the hyperplane given by the equation $\mathrm{Tr}\,\Lambda = 1$. Since $\xi^*(s) = (\Lambda_s, X_s, \rho_s)$ is the minimizer of F_s over G, we have, for certain real p,

$$\nabla_\Lambda F_s(\xi^*(s)) = pI_q; \quad \nabla_X F_s(\xi^*(s)) = 0; \quad \nabla_\rho F_s(\xi^*(s)) = 0.$$

Substituting the expression for F_s and $\mathcal{A}$, we obtain

$$\begin{array}{rcccccl} \sum_{i=1}^m L_i & \equiv & [\mathcal{A}^*R(s)]_\Lambda & \equiv & -s^{-1}\left[\mathcal{A}^*\nabla\Phi(Y)\right]_\Lambda & = & -s^{-1}pI_q, \\ 2\sum_{i=1}^m b_i d_i^T & \equiv & [\mathcal{A}^*R(s)]_X & \equiv & -s^{-1}\left[\mathcal{A}^*\nabla\Phi(Y)\right]_X & = & 2Q, \\ \sum_{i=1}^m t_i & \equiv & [\mathcal{A}^*R(s)]_\rho & \equiv & -s^{-1}\left[\mathcal{A}^*\nabla\Phi(Y)\right]_\rho & = & V. \end{array}$$

(Here $[\cdot]_\Lambda$, $[\cdot]_X$ and $[\cdot]_\rho$ denote, respectively, the Λ-, the X-, and the ρ-component of the design vector of $(\mathrm{TD_{fn}})$.) Note also that Y (and therefore Z) is positive definite. We see that $(R(s), \lambda \equiv -s^{-1}p)$ indeed is a feasible solution of $(\mathrm{TD}^*_{\mathrm{fn}})$.

Now, if (Λ, X, ρ) is a feasible solution to $(\mathrm{TD_{fn}})$, and

$$\left(R \equiv \mathrm{Diag}\left\{ \begin{pmatrix} M_i & c_i \\ c_i^T & r_i \end{pmatrix},\ i = 1, ..., m \right\}, \lambda \right)$$

is a feasible solution to $(\mathrm{TD}^*_{\mathrm{fn}})$, then

$$\begin{aligned}2\,\mathrm{Tr}(QX^T) + V\rho &= \big[\mathrm{Tr}([\mathcal{A}^*R]_X X^T) + [\mathcal{A}^*R]_\rho \rho + \mathrm{Tr}([\mathcal{A}^*R]_\Lambda \Lambda)\big] - \lambda \\ &\qquad [\text{since } [\mathcal{A}^*R]_\Lambda = \lambda I_q,\ [\mathcal{A}^*R]_X = 2Q,\ [\mathcal{A}^*R]_\rho = V \text{ by the} \\ &\qquad\ \text{constraints of } (\mathrm{TD}^*_{\mathrm{fn}}) \text{ and } \mathrm{Tr}\,\Lambda = 1 \\ &\qquad\ \text{by the constraints of } (\mathrm{TD}_{\mathrm{fn}})\,] \\ &= \mathrm{Tr}(R\mathcal{A}(\Lambda, X, \rho)) - \lambda,\end{aligned}$$

whence

$$[2\,\mathrm{Tr}(QX^T) + V\rho] + \lambda = \mathrm{Tr}(R\mathcal{A}(\Lambda, X, \rho)).$$

Since the optimal values in $(\mathrm{TD}_{\mathrm{fn}})$ and $(\mathrm{TD}^*_{\mathrm{fn}})$, by the Fenchel–Rockafellar duality theorem, are negations of each other, we come to

$$\epsilon[\Lambda, X, \rho] + \epsilon^*[R, \lambda] = \mathrm{Tr}(R\mathcal{A}(\Lambda, X, \rho)); \tag{6.9}$$

here $\epsilon[\Lambda, X, \rho]$ is the accuracy of the feasible solution (Λ, X, ρ) of $(\mathrm{TD}_{\mathrm{fn}})$ (i.e., the value of the objective of $(\mathrm{TD}_{\mathrm{fn}})$ at (Λ, X, ρ) minus the optimal value of the problem), and $\epsilon^*[\cdot]$ is similar accuracy in $(\mathrm{TD}^*_{\mathrm{fn}})$.

Specifying (Λ, X, ρ) as (Λ_s, X_s, ρ_s) and (R, λ) as $(R(s), \lambda_s)$, we make the right-hand side of (6.9) equal to

$$\mathrm{Tr}(R(s)Y) = s^{-1}\,\mathrm{Tr}(ZY) = s^{-1}\,\mathrm{Tr}(Y^{-1}Y) = s^{-1}\mu,$$

and with this equality (6.9) implies (6.8). □

7. Numerical examples. Let us illustrate the developed approach by a few examples.

Example 1. Our first example deals with the toy problem presented in Fig. 2.1; as was explained in section 2.2, here the single-load optimal design results in an unstable truss capable of carrying only very specific loads; the compliance of the truss with respect to the given load is 16.000. Now let us apply approach **B** from section 2.3, where the robustness constraint is imposed before solving the problem and corresponds to "active" nodes—those where the given load is applied. When imposing robustness requirement, we choose Q as explained in section 2.3. Namely, in our case we have 2 fixed and 4 free nodes, so that the dimension n of the space of virtual nodal displacements is $2 \times 4 = 8$. Since all free nodes are active, the ellipsoid of loads in robust setting is full-dimensional ($q = n = 8$); this ellipsoid is chosen as explained in section 2.3—one of the half-axes is the given load, and the remaining 7 half-axes are 10 times smaller. The corresponding matrix (rounded to 3 decimal places after the dot) is

$$Q = \begin{pmatrix} 2.000 & 0.014 & -0.026 & 0.117 & -0.063 & 0.170 & -0.264 & -0.054 \\ 0 & 0.235 & 0.216 & 0.125 & -0.032 & -0.161 & -0.070 & 0.104 \\ 0 & -0.040 & -0.107 & 0.099 & 0.311 & -0.158 & -0.117 & -0.035 \\ 2.000 & 0.045 & 0.137 & -0.263 & 0.162 & 0.039 & 0.002 & 0.043 \\ 0 & -0.202 & 0.148 & -0.081 & -0.111 & -0.190 & -0.124 & -0.164 \\ -2.000 & 0.149 & -0.108 & -0.203 & -0.030 & 0.006 & -0.210 & -0.009 \\ -2.000 & -0.089 & 0.219 & 0.057 & 0.129 & 0.203 & -0.052 & -0.003 \\ 0 & 0.173 & 0.028 & 0.020 & 0.042 & 0.035 & 0.098 & -0.341 \end{pmatrix}.$$

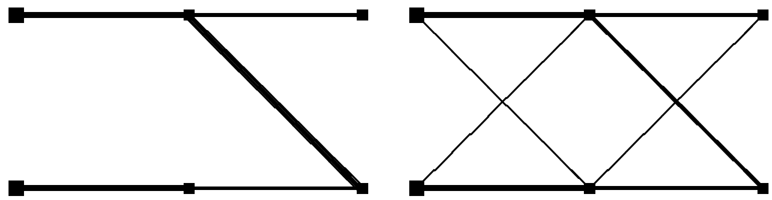

FIG. 7.1. *Optimal design without* (*left*) *and with* (*right*) *robustness constraints.*

TABLE 7.1
Optimal designs for Example 1.

Problem setting	Compliance	Bars, node : node	Bar volumes, %
without robustness constraints	16.000	1 : 2	25.00
		4 : 5	25.00
		3 : 5	25.00
		5 : 6	12.50
		2 : 3	12.50
with robustness constraints	17.400	4 : 5	24.48
		1 : 2	24.48
		3 : 5	23.68
		2 : 3	11.95
		5 : 6	11.95
		2 : 4	1.27
		1 : 5	1.27
		2 : 6	0.92

(To relate Q to the nodal structure presented on Fig. 2.1, note that the coordinates of virtual displacements are ordered as 2X,2Y,3X,3Y,5X,5Y,6X,6Y, where, say, 3X corresponds to the displacement of node #3 along the X-axis.)

The result of "robust" design is presented in Fig. 7.1 and Table 7.1.

Now the maximum over the 8-dimensional loading ellipsoid compliance becomes 17.400 (8.75% growth). But the compliance of the truss with respect to the load f is 16.148; i.e., it is only larger by 0.9% than for the truss given by single-load setting.

Example 2 (Console). The second example deals with approach **A** from section 2.3, where the robustness constraint is used for postoptimality analysis. The left part of Fig. 7.2 represents optimal single-load design for a 9×9 nodal grid on a 2D plane; nodes from the very left column are fixed, the remaining nodes are free, and the load is the unit force acting down and applied at the midnode of the very right column (long arrow). The compliance of the resulting truss with respect to f^*, in appropriate scale, is 1.00. Now note that the compliance of t with respect to very small (of magnitude $0.005\|f^*\|$) "occasional" load (short arrow) applied at properly chosen node is > 8.4 ! Thus, in fact, t is highly unstable.

The right part of Fig. 7.2 represents the truss obtained via postoptimality design with robustness constraint. We marked the nodes incident to the bars of t (there were only 12 of them) and formed a new design problem with the nodal set composed of these marked nodes, and the tentative bars given by all 66 possible pair connections in this nodal set (in the original problem, there were 2040 tentative bars). The truss represented in the right part corresponds to optimal design with robustness constraint imposed at all 10 free nodes of this ground structure in the same way as in the previous example (i.e., the first column in the 20×20 matrix Q is the given load f^*, and the remaining 19 columns formed orthogonal basis in the orthogonal complement to f^* in of 20-dimensional space of virtual displacements of the construction; the Euclidean lengths of these additional columns were set to 0.1 (10% of the magnitude of f^*).

The maximal compliance, over the resulting ellipsoid of loads, of the "robust" truss is now 1.03, and its compliance with respect to f is 1.0024—i.e., it is only larger by 0.24% than the optimal compliance c^* given by the single-load design; at the same time, the compliance of the new truss with respect to all "occasional" loads of magnitude 0.1 is at most by 3% greater than c^*.

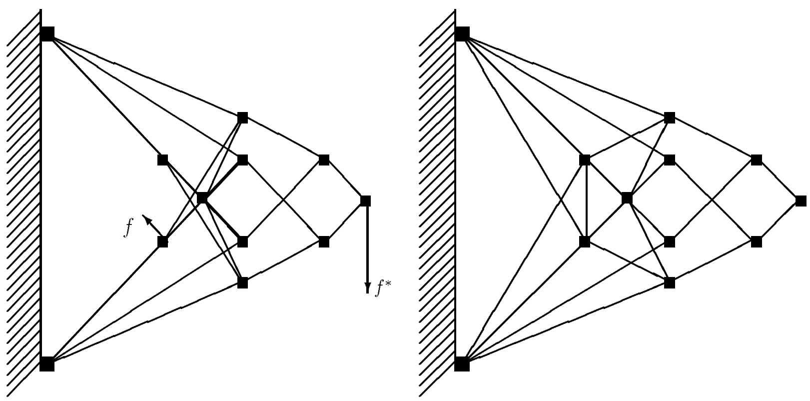

FIG. 7.2. *Single-load optimal design* (*left*) *and its postoptimal "robust correction"* (*right*).

Example 3 ($N \times 2$-truncated pyramids). The examples below deal with simple 3D trusses. The nodal set is composed of $2N$ points. N "ground" nodes are the vertices of equilateral N-polygon in the plane $z = 0$:

$$x_i = \cos(2\pi i/N),\ y_i = \sin(2\pi i/N),\ z_i = 0,\ \ i = 1, \dots, N,$$

and N "top" nodes are the vertices of twice smaller concentric polygon in the plane $z = 2$:

$$x_i = \frac{1}{2}\cos(2\pi i/N),\ y_i = \frac{1}{2}\sin(2\pi i/N),\ z_i = 2,\ \ i = N+1, \dots, 2N.$$

The ground nodes are fixed, and the top ones are free. The ground structure is composed of all pair connections of the nodes, except connections between the ground-fixed ones.

We dealt with two kinds of loading scenarios, referred to, respectively, as "$N \times 2$s"- and "$N \times 2$m"-design data. $N \times 2$s-data corresponds to a singleton scenario set, where the load is composed of N nearly horizontal forces acting at the top nodes and "rotating" the construction. The force acting at the ith node, $i = N+1, ..., 2N$, is

$$f_i = \alpha(\sin(2\pi i/N), -\cos(2\pi i/N), -\rho),\ \ i = N+1, \dots, 2N, \tag{7.1}$$

where ρ is a small parameter and α is a normalizing coefficient which makes the Euclidean length of the load equal to 1 (i.e., $\alpha = 1/\sqrt{N(1+\rho^2)}$). $N \times 2$m-data correspond to N-scenario design where the forces (7.1) act nonsimultaneously (and are renormalized to be of unit length, i.e., $\alpha = 1/\sqrt{1+\rho^2}$).

Along with the traditional "scenario design" (single load in the case of s-data and multiload in the case of m-data), we carried out "robust design" where we minimized

TABLE 7.2
Compliances in Example 3.

Design data	Scenario design			Robust design	
	Compl(Scen)	Compl(0.1)	Compl(0.3)	Compl(Scen)	Compl(0.3)
`3x2s`	1.0000	7.5355	67.820	1.0029	1.0029
`4x2s`	1.0000	12.209	109.88	1.0028	1.0028
`5x2s`	1.0000	2.7311	24.580	1.0022	1.0022
`3x2m`	1.0000	1.2679	1.2679	1.0942	1.0943
`4x2m`	1.0000	4.1914	37.722	1.2903	1.2903
`5x2m`	1.0000	1.5603	1.6882	1.5604	1.5604

the maximum compliance with respect to a full-dimensional ellipsoid of loads M_θ—the "ellipsoidal envelope" of the unit ball in the linear span $L(F)$ of the scenario loads and the ball of radius θ in the orthogonal complement of $L(F)$ in the $3N$-dimensional space of virtual displacements of the nodal set. In other words, dim $L(F)$ of the principal half-axes of M_θ are of unit length and span $L(F)$, and the remaining principal half-axes are of length θ. In our experiments with robust design, we used $\theta = 0.3$ and measured the worst-case compliance of the resulting trusses, same as those given by the usual scenario design, with respect to three sets of loads:

(i) the original set of scenarios,
(ii) the ellipsoid of loads $M_{0.1}$,
(iii) the ellipsoid of loads $M_{0.3}$.

The resulting structures are shown in Fig. 7.3 (data $N \times 2$s) and Fig. 7.4 (data $N \times 2$m), and the corresponding compliances are seen in Table 7.2. In Table 7.2, Compl(Scen) means the maximum compliance of the designed structure with respect to the set of loading scenarios given by the corresponding data, while Compl(θ), $\theta = 0.1, 0.3$ is the maximum compliance with respect to the ellipsoid M_θ. In order to make the comparison more clear, we normalize the data in each row to make the compliance of the truss given by scenario design with respect to the underlying set of scenarios equal to 1.

The summary of the numerical results in question is as follows.

1. $N \times 2$s *design data.* The trusses given by the scenario and the robust designs have the same topology and differ only in bar volumes; the difference basically is in the thickness of the "top" – horizontal – bars (see Fig. 7.3): for the "robust" truss they are approximately 80 times larger in volume than for the "scenario" one (0.1% of the total bar volume instead of 0.0012% for $N = 3$). Although this difference in sizing seems small, it is in fact quite significant. The scenario design results in highly unstable constructions: appropriately chosen "occasional" loads with magnitude only 10% of the scenario load, result in 2.6–13.0 times larger compliance than the "scenario" one. When the occasional load is allowed to be 30% of the scenario one, the ratio in question may become 15–100. Note that bad robustness of the trusses given by the scenario design has very simple origin: in the limiting case of $\rho = 0$ (purely horizontal rotating load—the torque) the top bars disappear at all, and the optimal truss given by the usual single-load design becomes completely unstable.

The robust design associated with the ellipsoid $M_{0.3}$ ("occasional" loads may be as large as 30% of the scenario one) results in trusses nearly optimal with respect to the scenario load ("nonoptimality" is at most 0.3%). Surprisingly enough, for the trusses given by the robust design the maximum compliance with respect to the ellipsoid of loads is the same as their compliance with respect to the scenario load. Thus, in the case in question, the robustness is "almost costless."

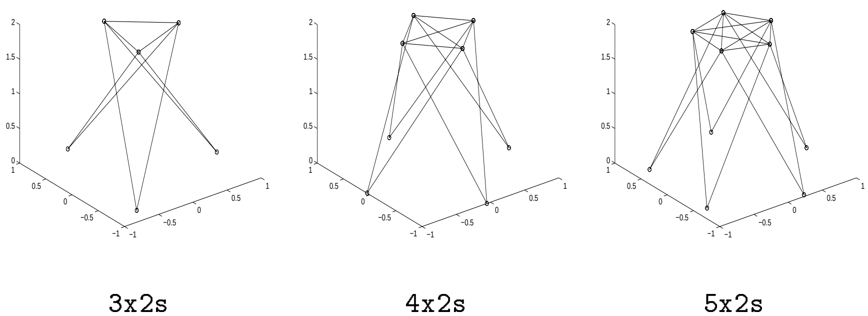

FIG. 7.3. *Scenario and robust design, single "rotating" load* ($\rho = 0.001$ *for* 3×2s *and* 4×2s, $\rho = 0.01$ *for* 5×2s).

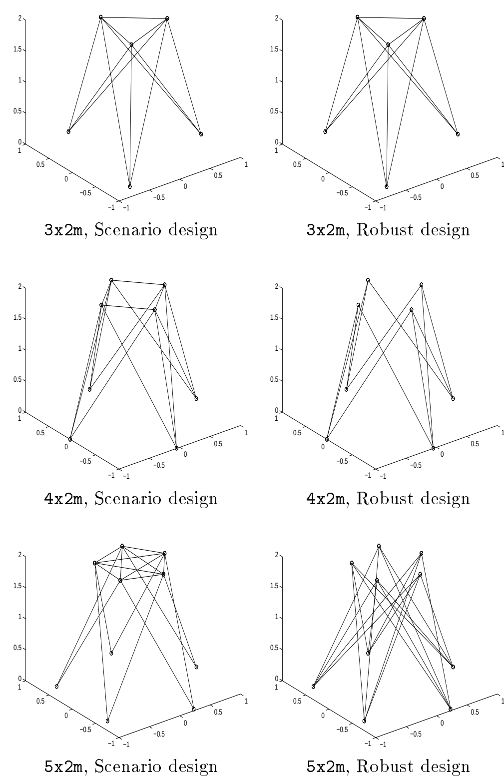

FIG. 7.4. *Scenario design vs. robust design, multiple "rotating" loads* ($\rho = 0.001$ *for* 3×2m *and* 4×2m, $\rho = 0.01$ *for* 5×2m).

2. $N \times 2$m *design data.* Here the trusses given by the scenario design are of course much more stable than in the case of $N \times 2$s data, and both kinds of design possess their own advantages and drawbacks. On one hand, the maximum compliance, over the ellipsoid $M_{0.3}$ of loads, of the truss given by the scenario design is considerably larger than the optimal value of this quantity (by 27% for $N = 3$, by 3670 % for $N = 4$,[5] and by 69% for $N = 5$). On the other hand, the maximum compliance, over the scenario set, of the truss given by robust design is also considerably larger than the optimal value of this quantity (by 9% for $N = 3$, by 29% for $N = 4$, and by 56% for $N = 5$). Thus, it is difficult to say which design—the scenario or the robust one—results in better construction.

The results in question suggest a seemingly better approach to ensuring robustness than those mentioned in section 2.3, namely, as follows. Given a scenario set F, we embed it into an ellipsoid M (see section 2.3) and solve the resulting problem $(\mathrm{TD}_{\mathrm{robust}})$; let c^*_{robust} be the corresponding optimal value. After this value is found, we increase it in certain fixed proportion $1 + \chi$, say, by 10%, and solve the problem

$$\begin{array}{c} \text{find } t \in T \text{ which minimizes the compliance } c^F(t) = \max_{f\in F} c_f(t) \\ \text{s.t.}\quad c^M(t) \equiv \max_{f\in M} c_f(t) \le (1+\chi)c^*_{\mathrm{robust}}. \end{array}$$

Note that the latter problem can be posed as a semidefinite program, which only slightly differs from $(\mathrm{TD}_{\mathrm{sd}})$:

$$\begin{array}{l} \min\limits_{t\in R^m,\tau\in R} \tau \\ \text{s.t.} \\ \qquad \begin{pmatrix} \tau & f^T \\ f & \sum_{i=1}^m t_i A_i \end{pmatrix} \ \ge\ 0,\ \forall f \in F \\ \qquad \begin{pmatrix} a & Q^T \\ Q & \sum_{i=1}^m t_i A_i \end{pmatrix} \ \ge\ 0, \end{array}$$

where

$$a = (1+\chi)c^*_{\mathrm{robust}}.$$

The dual to the latter problem is the computationally more convenient program

$$\begin{array}{l} \min \left\{ a\,\mathrm{Tr}(\Lambda) + 2\,\mathrm{Tr}(QX^T) + 2\sum\limits_{f\in F} f^T x_f + V\rho \right\} \\ \text{s.t.} \\ \qquad \begin{array}{rcl} \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \sigma_i \end{pmatrix} & \ge & 0,\ i = 1,\dots,m, \\ \sigma_i + \sum\limits_{f\in F} \dfrac{(b_i^T x_f)^2}{\lambda_f} & \le & \rho,\ i = 1,\dots,m, \\ \lambda_f & \ge & 0,\ f \in F, \\ \sum\limits_{f\in F}^{k} \lambda_f & = & 1, \end{array} \end{array}$$

[5]This huge difference mainly comes not from the difference in the topology of trusses but from different sizing of the bars linking bottom nodes with "the same" top ones; for the robust design these bars are approximately 30 times thicker than for the scenario design (1.5% of the total bar volume vs. 0.05%, resp.).

TABLE 7.3
Computational performance.

Problem	Scenario design			Robust design		
	$(N_{\rm dsg}, N_{\rm LMI}, N_{\rm img})$	Nwt	CPU	$(N_{\rm dsg}, N_{\rm LMI}, N_{\rm img})$	Nwt	CPU
Example 2	(146,2041,6121)	75	$3'58''$	(611,67,15247)	95	$24'42''$
`3x2s`	(11,13,37)	14	$0.2''$	(127,13,661)	62	$14.5''$
`4x2s`	(14,23,67)	16	$0.4''$	(223,23,2003)	77	$1'18''$
`5x2s`	(17,36,106)	17	$0.6''$	(346,36,4761)	59	$3'13''$
`3x2m`	(31,13,121)	16	$0.4''$	(127,13,661)	101	$24''$
`4x2m`	(53,23,331)	23	$1.5''$	(223,23,2003)	65	$1'6''$
`5x2m`	(81,36,736)	23	$3''$	(346,36,4761)	65	$3'32''$

In the table:

$N_{\rm dsg}$ – number of design variables in $(\mathrm{TD_{fn}})$,
$N_{\rm LMI}$ – number of linear matrix inequalities in $(\mathrm{TD_{fn}})$,
$N_{\rm ing}$ – total image dimension of $(\mathrm{TD_{fn}})$, i.e., the dimension of the corresponding semidefinite cone,
Nwt – number of Newton steps performed by the interior point solver when solving $(\mathrm{TD_{fn}})$,
CPU – solution time (workstation RS 6000).

the design variables being $\Lambda \in \mathbf{S}^k$, $X \in R^{n\times q}$, $\sigma \in R^n$, $\{(\lambda_f, x_f) \in R \times R^n\}_{f\in F}$, and $\rho \in R$.

The reported numerical experiments were carried out with the LMI Control Toolbox [7], the only software for semidefinite programming available to us at the moment. The projective interior point method [10, Chapter 5], implemented in the Toolbox is of the potential reduction rather than of the path-following type, and we were forced to add to the Toolbox solver a "centering" interior point routine which transforms a good approximate solution to $(\mathrm{TD_{fn}})$ into another solution of the same quality belonging to the central path, which enabled us to recover the optimal truss, as is explained in section 6. The time of solving $(\mathrm{TD_{fn}})$ by the Toolbox solver was moderate, as it is seen in Table 7.3.

8. Concluding remarks. Uncertainty of the data is a generic property associated with optimization problems of real world origin. Accordingly, "robust reformulation" of an optimization model as a way to improve applicability of the resulting solution is a very traditional idea in mathematical programming, and different approaches to implement this idea were proposed. One of the best-known approaches is *stochastic programming*, where uncertainty is assumed to be of stochastic nature. Another approach is *robust optimization* (see [9] and references therein); here, roughly speaking, the "robust solution" should not necessarily be feasible for all "allowed" data, and the "optimal robust solution" minimizes the sum of the original objective and a penalty for infeasibilities, the infeasibilities being taken over a finite set of scenarios. The approach used in our paper is somewhat different: a solution to the "stabilized" problem should be feasible for all allowed data. This approach is exactly the one used in robust control. The goal of this concluding section is to demonstrate that the approach developed in the paper can be naturally extended to other mathematical programming problems. To this end let us look at what in fact was done in section 2.

(ii) We start with an optimization program in the "conic" form

$$\text{(P)} \qquad c^T u \to \min \mid \; Au \in K, \; u \in E,$$

where u is the design vector, A is $M \times N$ matrix, K is closed convex cone in R^M, and

E is an affine plane in R^N.

This is exactly the form of a single-load TTD problem $\min\{\sigma \mid \sigma \geq c_f(t), t \in T\}$ (see section 2.1): to cast TTD as (P) it suffices to specify (P) as follows:

- $u = (t, \tau, \sigma) \in R^m \times R \times R$;
- $E = \{(t, \tau, \sigma) \mid \tau = 1, \sum_{i=1}^m t_i = V\}$;
- K is the direct product of the cone of positive semidefinite symmetric $(n+1) \times (n+1)$ matrices ("matrix part") and R^m_+ ("vector part");
- the "vector" part of the linear mapping $(t, \tau, \sigma) \mapsto A(t, \tau, \sigma)$ is t, and the "matrix" part is $\begin{pmatrix} \sigma & \tau f^T \\ \tau f & A(t) \end{pmatrix}$, f being the load in question.

(ii) We say that the data in (P)(entries in the data matrix A) are inexact (in TTD, these are entries associated with the load vector f). We model the corresponding uncertainty by the assumption that $A \in \mathcal{U}$, where $\mathcal{U}$ is certain ellipsoid in the space of $M \times N$ matrices.[6] Accordingly, we impose on the decision u the requirement to be *robust feasible*, i.e., to satisfy the inclusions $u \in E$ and $Au \in K$ for *all* possible data matrices $A \in \mathcal{U}$. This leads to our *robust reformulation* of (P):

$$(\mathrm{P_{st}}) \qquad c^T u \to \min \mid u \in E, \ Au \in K \ \forall A \in \mathcal{U}.$$

Note that this is a general form of the approach we have used in section 2; and the goal of the remaining sections was to realize, for the case when (P) is the single load TTD problem, what is $(\mathrm{P_{st}})$ as a mathematical programming problem and how to solve it efficiently.

Problem (P) is a quite general form of a convex programming problem; the advantage of this conic form is that it allows to separate the "structure" of the problem (c, K, E) and the "data" (A).[7] The data now become a quite tractable entity—simply a matrix. Whenever a program in question can be naturally posed in the conic form, we can apply the above approach to get a "robust reformulation" of (P). Let us look at some concrete examples.

Robust linear programming. Let K in (P) be the nonnegative orthant; this is exactly the case when (P) is a linear programming problem in the canonical form.[8] It is shown in [6] that $(\mathrm{P_{st}})$ is a conic quadratic program (i.e., a conic program with K being a direct product of the second order cones).

Robust quadratic programming. Let K be a direct product of the second order cones, so that (P) is a conic quadratic program (a natural extension of the usual quadratically constrained convex quadratic program). It can be verified (see [6]) that in this case, under mild restrictions on the structure of the uncertainty ellipsoid $\mathcal{U}$, the problem $(\mathrm{P_{st}})$ can be equivalently rewritten as a semidefinite program (a conic program with K being the cone of positive semidefinite symmetric matrices).

Note that in these examples $(\mathrm{P_{st}})$ is quite tractable computationally, in particular, it can be efficiently solved by interior point methods.

---

[6]Here, as in the main body of the paper, a k-dimensional ellipsoid in R^M is, by definition, the image of the unit Euclidean ball in R^k under an affine embedding of R^k into R^M.

[7]In some applications, the objective c should be treated as a part of the data rather than the structure. One can easily reduce this case to the one in question by evident equivalent reformulation of (P).

[8]Up to the fact that the mapping $u \mapsto Au$ is assumed to be linear rather than affine. This assumption does not restrict generality, since we incorporate into the model the affine constraint $u \in E$; at the same time, the homogeneous form $Au \in K$ of the nonnegativity constraints allows us to handle both uncertainties in the matrix of the linear inequality constraints and those in the right-hand side vector.

A somewhat "arbitrary" element in the outlined general approach is that we model uncertainty as an *ellipsoid*. Note, anyhow, that *in principle* the above scheme can be applied to any other uncertainty set $\mathcal{U}$, and the actual "bottleneck" is our ability to solve efficiently the resulting problem (P_{st}). Note that the robust problem (P_{st}) always is convex, so that there is a sufficient condition for its "efficient solvability." The condition, roughly speaking (for the details, see [8]), is that we should be able to equip the feasible domain

$$G = \{u \mid u \in E, Au \in K \ \forall A \in \mathcal{U}\}$$

of (P_{st}) with a *separation oracle*—a "computationally efficient" routine which, given on input u, reports on output whether $u \in G$, and if it is not the case, returns a linear form which separates G and u. Whether this sufficient condition is satisfied or not depends on the geometry of $\mathcal{U}$ and K, and the "more complicated" $\mathcal{U}$ is, the "simpler" K should be. When $\mathcal{U}$ is very simple (a polytope given as a convex hull of a finite set), K could be an arbitrary "tractable" cone (one which can be equipped with a separation oracle); when $\mathcal{U}$ is an ellipsoid, K could be for sure the nonnegative orthant or a direct product of the second order cones. On the other hand, if K is simple (the nonnegative orthant, as in the linear programming case), $\mathcal{U}$ could be more complicated than an ellipsoid—e.g., it could be an intersection of finitely many ellipsoids. Under mild regularity assumptions, in the latter case (P_{st}) turns out to be a conic quadratic program [6]. In other words, there is a "tradeoff" between the *flexibility* and the *tractability*, i.e., between the ability to express uncertainties, on one hand and the ability to produce computationally tractable problems (P_{st}) on the other hand.

We strongly believe that the approach advocated here is promising and is worthy of investigation, and we intend to devote a separate paper to it.

REFERENCES

[1] W. Achtziger, M. P. Bendsøe, A. Ben-Tal, and J. Zowe, *Equivalent displacement-based formulations for maximum strength truss topology design*, Impact Comput. Sci. Eng., 4 (1992), pp. 315–345.

[2] M. P. Bendsøe, A. Ben-Tal, and J. Zowe, *Optimization methods for truss geometry and topology design*, Structural Optimization, 7 (1994), pp. 141–159.

[3] A. Ben-Tal and M. P. Bendsøe, *A new method for optimal truss topology design*, SIAM J. Optim., 3 (1993), pp. 322–358.

[4] A. Ben-Tal, M. Kočvara, and J. Zowe, *Two nonsmooth approaches to simultaneous geometry and topology design of trusses*, Topology Design of Structures, Proc. NATO-ARW, M. P. Bendsøe, ed., Sesimbra, Portugal, 1992.

[5] A. Ben-Tal and A. Nemirovski, *Potential reduction polynomial time method for Truss Topology Design*, SIAM J. Optim., 4 (1994), pp. 596–612.

[6] A. Ben-Tal and A. Nemirovski, *Robust Convex Programming*, manuscript, Optimization Laboratory, Faculty of Industrial Engineering and Management at Technion, Haifa, Israel, 1995.

[7] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox*, The Mathworks Partner Series, The MathWorks Inc., Natick, MA, 1995.

[8] M. Grötschel, L. Lovasz, and A. Schrijver, *The Ellipsoid Method and Combinatorial Optimization*, Springer-Verlag, Heidelberg, 1988.

[9] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios, *Robust optimization of large-scale systems*, Oper. Res., 43 (1995), pp. 264–281.

[10] Yu. Nesterov and A. Nemirovski, *Interior Point Polynomial Methods in Convex Programming*, SIAM Series in Applied Mathematics, Philadelphia, PA, 1994.

[11] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[12] G. Rozvany, M. P. Bendsøe, and U. Kirsch, *Layout optimization of structures*, Appl. Mech. Rev., 48 (1955), pp. 41–119.