CHAPTER 16

Structural Optimization

This final chapter on network-type problems deals with finding the best design of a structure to support a specified load at a fixed set of points. The *topology* of the problem is described by a graph where each node represents a *joint* in the structure and each arc represents a potential *member*.[1] We shall formulate this problem as a linear programming problem whose solution determines which of the potential members to include in the structure and how thick each included member must be to handle the load. The optimization criterion is to find a minimal weight structure. As we shall see, the problem bears a striking resemblance to the minimum-cost network flow problem that we studied in Chapter 14.

1. An Example

We begin with an example. Consider the graph shown in Figure 16.1. This graph represents a structure consisting of five joints and eight possible members connecting the joints. The five joints and their coordinates are given as follows:

Joint	Coordinates
1	(0.0 , 0.0)
2	(6.0 , 0.0)
3	(0.0 , 8.0)
4	(6.0 , 8.0)
5	(3.0 , 12.0)

Since joints are analogous to nodes in a network, we shall denote the set of joints by $\mathcal{N}$ and denote by m the number of joints. Also, since members are analogous to arcs in network flows, we shall denote the set of them by $\mathcal{A}$. For the structure shown in Figure 16.1, the set of members is

$$\mathcal{A} = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\},\{3,5\},\{4,5\}\}\,.$$

Note that we enclosed the pairs of endjoints in braces to emphasize that their order is irrelevant. For example, $\{2,3\}$ and $\{3,2\}$ refer to one and the same member spanning between joints 2 and 3. In network flows, the graphs we considered were directed graphs. Here, they are undirected. Also, the graphs here are embedded in a d-dimensional Euclidean space (meaning that every node comes with a set of coordinates indicating its location in d-dimensional space). No such embedding was

[1]Civil engineers refer to beams as members.

R.J. Vanderbei, *Linear Programming*, International Series in Operations Research & Management Science 196, DOI 10.1007/978-1-4614-7630-6_16,

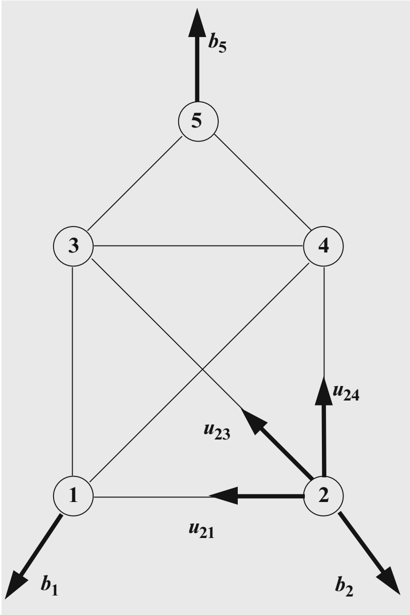

FIGURE 16.1. Sample topology for a two-dimensional structure.

imposed before in our study of network flows, even though real-world network flow problems often possess such an embedding.

Following the standard convention of using braces to denote sets, we ought to let $x_{\{i,j\}}$ denote the force exerted by member $\{i, j\}$ on its endjoints. But the braces are cumbersome. Hence, we shall write this force simply as x_{ij}, with the understanding that x_{ij} and x_{ji} denote one and the same variable.

We shall assume that a positive force represents tension in the member (i.e., the member is pulling "in" on its two endjoints) and that a negative value represents compression (i.e., the member is pushing "out" on its two endjoints).

If the structure is to be in equilibrium (i.e., not accelerating in some direction), then forces must be balanced at each joint. Of course, we assume that there may be a nonzero external load at each joint (this is the analogue of the external supply/demand in the minimum-cost network flow problem). Hence, for each node i, let b_i denote the externally applied load. Note that each b_i is a vector whose dimension equals the dimension of the space in which the structure lies. For our example, this dimension is 2. In general, we shall denote the spatial dimension by d.

Force balance imposes a number of constraints on the member forces. For example, the force balance equations for joint 2 can be written as follows:

$$x_{12} \begin{bmatrix} -1 \\ 0 \end{bmatrix} + x_{23} \begin{bmatrix} -0.6 \\ 0.8 \end{bmatrix} + x_{24} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = - \begin{bmatrix} b_2^1 \\ b_2^2 \end{bmatrix},$$

where b_2^1 and b_2^2 denote the components of b_2. Note that the three vectors appearing on the left are unit vectors pointing out from joint 2 along each of the corresponding members.

2. Incidence Matrices

If, for each joint i, we let p_i denote its position vector, then the unit vectors pointing along the arcs can be written as follows:

$$u_{ij} = \frac{p_j - p_i}{\|p_j - p_i\|}, \qquad \{i,j\} \in \mathcal{A}.$$

It is important to note that $u_{ji} = -u_{ij}$, since the first vector points from j towards i, whereas the second points from i towards j. In terms of these notations, the force balance equations can be expressed succinctly as

$$\sum_{\substack{j: \\ \{i,j\} \in \mathcal{A}}} u_{ij} x_{ij} = -b_i \qquad i = 1, 2, \ldots, m. \tag{16.1}$$

These equations can be written in matrix form as

$$Ax = -b \tag{16.2}$$

where x denotes the vector consisting of the member forces, b denotes the vector whose elements are the applied load vectors, and A is a matrix containing the unit vectors pointing along the appropriate arcs. For our example, we have

$$x^T = \begin{bmatrix} x_{12} & x_{13} & x_{14} & x_{23} & x_{24} & x_{34} & x_{35} & x_{45} \end{bmatrix}$$

$$A = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{bmatrix} .6 \\ .8 \end{bmatrix} & & & & & \\ \begin{bmatrix} -1 \\ 0 \end{bmatrix} & & & \begin{bmatrix} -.6 \\ .8 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & & & \\ & \begin{bmatrix} 0 \\ -1 \end{bmatrix} & & \begin{bmatrix} .6 \\ -.8 \end{bmatrix} & & \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} .6 \\ .8 \end{bmatrix} & \\ & & \begin{bmatrix} -.6 \\ -.8 \end{bmatrix} & & \begin{bmatrix} 0 \\ -1 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \end{bmatrix} & & \begin{bmatrix} -.6 \\ .8 \end{bmatrix} \\ & & & & & & \begin{bmatrix} -.6 \\ -.8 \end{bmatrix} & \begin{bmatrix} .6 \\ -.8 \end{bmatrix} \end{bmatrix}, \quad b = \begin{bmatrix} b_1^1 \\ b_1^2 \\ b_2^1 \\ b_2^2 \\ b_3^1 \\ b_3^2 \\ b_4^1 \\ b_4^2 \\ b_5^1 \\ b_5^2 \end{bmatrix}.$$

Note that we have written A as a matrix of 2-vectors by putting "inner" brackets around appropriate pairs of entries. These inner brackets could of course be dropped—they are included simply to show how the constraints match up with (16.1).

In network flows, an incidence matrix is characterized by the property that every column of the matrix has exactly two nonzero entries, one $+1$ and one -1. Here, the matrix A is characterized by the property that, when viewed as a matrix of d-vectors, every column has two nonzero entries that are unit vectors pointing in opposite directions from each other. Generically, matrix A can be written as follows:

$$A = \begin{array}{c} \\ i \rightarrow \\ \\ j \rightarrow \end{array} \begin{array}{c} \{i,j\} \\ \downarrow \\ \left[\begin{array}{ccc} & u_{ij} & \\ & & \\ & u_{ji} & \end{array} \right] \end{array}$$

where, for each $\{i, j\} \in \mathcal{A}$,

$$u_{ij}, u_{ji} \in \mathbb{R}^d,$$

$$u_{ij} = -u_{ji},$$

and

$$\|u_{ij}\| = \|u_{ji}\| = 1.$$

By analogy with network flows, the matrix A is called an *incidence matrix*. This definition is a strict generalization of the definition we had before, since, for $d = 1$, the current notion reduces to the network flows notion. Incidence matrices for network flows enjoy many useful properties. In the following sections, we shall investigate the extent to which these properties carry over to our generalized notion.

3. Stability

Recall that for network flows, the sum of the rows of the incidence matrix vanishes and that if the network is connected, this is the only redundancy. For $d > 1$, the situation is similar. Clearly, the sum of the rows vanishes. But is this the only redundancy? To answer this question, we need to look for nonzero row vectors y^T for which $y^T A = 0$. The set of all such row vectors is a subspace of the set of all row vectors. Our aim is to find a basis for this subspace and, in particular, to identify its dimension. To this end, first write y in component form as $y^T = [\ y_1^T \cdots y_m^T\ ]$ where each of the entries y_i, $i = 1, 2, \ldots, m$, are d-vectors (transposed to make them into row vectors). Multiplying this row vector against each column of A, we see that $y^T A = 0$ if and only if

(16.3) $$y_i^T u_{ij} + y_j^T u_{ji} = 0, \qquad \text{for all } \{i, j\} \in \mathcal{A}.$$

There are many choices of y that yield a zero row combination. For example, we can take any vector $v \in \mathbb{R}^d$ and put

$$y_i = v, \qquad \text{for every } i \in \mathcal{N}.$$

Substituting this choice of y_i's into the left-hand side of (16.3), we get

$$y_i^T u_{ij} + y_j^T u_{ji} = v^T u_{ij} + v^T u_{ji} = v^T u_{ij} - v^T u_{ij} = 0.$$

This set of choices shows that the subspace is at least d-dimensional.

But there are more! They are defined in terms of skew symmetric matrices. A matrix R is called *skew symmetric* if $R^T = -R$. A simple but important property of skew symmetric matrices is that, for every vector ξ,

(16.4) $$\xi^T R \xi = 0$$

(see Exercise 16.1). We shall make use of this property shortly. Now, to give more choices of y, let R be a $d \times d$ skew symmetric matrix and put

$$y_i = Rp_i, \qquad \text{for every } i \in \mathcal{N}$$

(recall that p_i denotes the position vector for joint i). We need to check (16.3). Substituting this definition of the y's into the left-hand side in (16.3), we see that

$$\begin{aligned} y_i^T u_{ij} + y_j^T u_{ji} &= p_i^T R^T u_{ij} + p_j^T R^T u_{ji} \\ &= -p_i^T R u_{ij} - p_j^T R u_{ji} \\ &= (p_j - p_i)^T R u_{ij}. \end{aligned}$$

Now substituting in the definition of u_{ij}, we get

$$(p_j - p_i)^T R u_{ij} = \frac{(p_j - p_i)^T R (p_j - p_i)}{\|p_j - p_i\|}.$$

Finally, by putting $\xi = p_j - p_i$ and using property (16.4) of skew symmetric matrices, we see that the numerator on the right vanishes. Hence, (16.3) holds.

How many redundancies have we found? For $d = 2$, there are two independent v-type redundancies and one more R-type. The following two vectors and a matrix can be taken as a basis for these redundancies

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

For $d = 3$, there are three independent v-type redundancies and three R-type. Here are three vectors and three matrices that can be taken as a basis for the space of redundancies:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{16.5}$$

In general, there are $d + d(d-1)/2 = d(d+1)/2$ independent redundancies. There could be more. But just as for network flows, where we showed that there is one redundancy if and only if the network is connected, further redundancies represent a defect in the underlying graph structure. In fact, we say that the graph is *stable* if the rank of the incidence matrix A is exactly $md - d(d+1)/2$, that is, if and only if the above redundancies account for the entire rank deficiency of A.

4. Conservation Laws

Recall that for network flows, not all choices of supplies/demands yield feasible flows. For connected networks, it is necessary and sufficient that the total supply equals the total demand. The situation is similar here. The analogous question is: which external loads give rise to solutions to (16.2)? We have already identified several row vectors y^T for which $y^T A = 0$. Clearly, in order to have a solution to (16.2), it is necessary that $y^T b = 0$ for all these row vectors. In particular for every $v \in \mathbb{R}^d$, we see that b must satisfy the following condition:

$$\sum_i v^T b_i = 0.$$

Bringing the sum inside of the product, we get

$$v^T \left(\sum_i b_i \right) = 0.$$

Since this must hold for every d-vector v, it follows that

$$\sum_i b_i = 0.$$

This condition has a simple physical interpretation: the loads, taken in total, must balance.

What about the choices of y^T arising from skew symmetric matrices? We shall show that these choices impose the conditions necessary to prevent the structure from spinning around some axis of rotation. To show that this is so, let us first consider the two-dimensional case. For every 2×2 skew symmetric matrix R, the load vectors b_i, $i \in \mathcal{N}$, must satisfy

$$\sum_i (Rp_i)^T b_i = 0. \tag{16.6}$$

This expression is a sum of terms of the form $(Rp)^T b$, where p is the position vector of a point and b is a force applied at this point. We claim that $(Rp)^T b$ is precisely the torque about the origin created by applying force b at location p. To see this connection between the algebraic expression and its physical interpretation, first decompose p into the product of its length r times a unit vector v pointing in the same direction and rewrite the algebraic expression as

$$(Rp)^T b = r(Rv)^T b.$$

Now, without loss of generality, we may assume that R is the "basis" matrix for the space of skew symmetric matrices,

$$R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

This matrix has the additional property that its two columns are unit vectors that are orthogonal to each other. That is, $R^T R = I$. Hence,

$$\|Rv\|^2 = \|v\|^2 = 1.$$

Furthermore, property (16.4) tells us that Rv is orthogonal to v. Therefore, the product $(Rv)^T b$ is the length of the projection of b in the direction of Rv, and so $r(Rv)^T b$ is the distance from the origin (of the coordinate system) to p, which is called the *moment arm*, times the component of the force that is orthogonal to the moment arm in the direction of Rv (see Figure 16.2). This interpretation for each summand in (16.6) shows that it is exactly the torque around the rotation axis passing through the origin of the coordinate system caused by the force b_i applied to

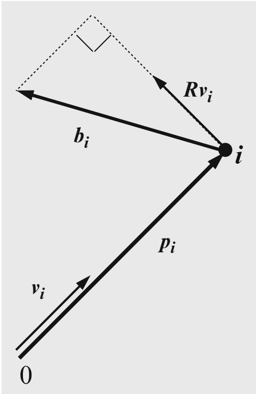

FIGURE 16.2. The ith summand in (16.6) is the length of p_i times the length of the projection of b_i onto the direction given by Rv_i. This is precisely the torque around an axis at 0 caused by the force b_i applied at joint i.

joint i. In $d = 2$, there is only one rotation around the origin. This fact corresponds to the fact that the dimension of the space of skew symmetric matrices in two dimensions is 1. Also, stipulating that the total torque about the origin vanishes implies that the total torque around any point other point also vanishes—see Exercise 16.4.

The situation for $d > 2$, in particular for $d = 3$, is slightly more complicated. Algebraically, the complications arise because the basic skew symmetric matrices no longer satisfy $R^T R = I$. Physically, the complications stem from the fact that in two dimensions rotation takes place around a point, whereas in three dimensions it takes place around an axis. We shall explain how to resolve the complications for $d = 3$. The extension to higher dimensions is straightforward (and perhaps not so important). The basic conclusion that we wish to derive is the same, namely that for basic skew symmetric matrices, the expression $(Rp)^T b$ represents the torque generated by applying a force b at point p. Recall that there are just three basic skew symmetric matrices, and they are given by (16.5). To be specific, let us just study the first one:

$$R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

This matrix can be decomposed into the product of two matrices:

$$R = UP$$

where

$$U = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

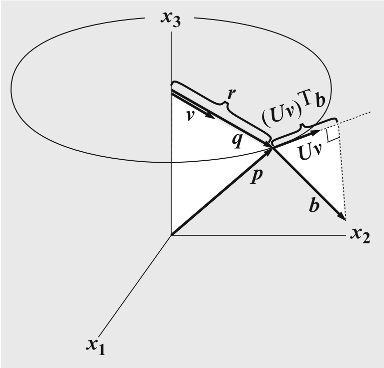

FIGURE 16.3. The decomposition of $(Rp)^T b$ into the product of a moment arm r times the component of b in the direction Uv shows that it is precisely the torque around the third axis.

The matrix U has the property that R had before, namely,

$$U^T U = I.$$

Such matrices are called *unitary*. The matrix P is a projection matrix. If we let

$$q = Pp,$$

$$v = \frac{q}{\|q\|},$$

and

$$r = \|q\|,$$

then we can rewrite $(Rp)^T b$ as

$$(Rp)^T b = r(Uv)^T b.$$

Since v is a unit vector and U is unitary, it follows that Uv is a unit vector. Hence, $(Uv)^T b$ represents the scalar projection of b onto the direction determined by Uv. Also, it is easy to check that Uv is orthogonal to v. At this point, we can consult Figure 16.3 to see that r is the moment arm for the torque around the third coordinate axis and $(Uv)^T b$ is the component of force in the direction of rotation around this axis. Therefore, the product is precisely the torque around this axis. As we know, for $d = 3$, there are three independent axes of rotation, namely, *pitch*, *roll*, and *yaw*. These axes correspond to the three basis matrices for the space of 3×3 skew symmetric matrices (the one we have just studied corresponds to the yaw axis).

Finally, we note that (16.6) simply states that the total torque around each axis of rotation must vanish. This means that the forces cannot be chosen to make the system spin.

5. Minimum-Weight Structural Design

For a structure with m nodes, the system of force balance equations (16.2) has md equations. But, as we now know, if the structure is stable, there are exactly $d(d+1)/2$ redundant equations. That is, the rank of A is $md - d(d+1)/2$. Clearly, the structure must contain at least this many members. We say that the structure is a *truss* if it is stable and has exactly $md - d(d+1)/2$ members. In this case, the force balance equations have a unique solution (assuming, of course, that the total applied force and the total applied torque around each axis vanish). From an optimization point of view, trusses are not interesting because they leave nothing to optimize—one only needs to calculate.

To obtain an interesting optimization problem, we assume that the proposed structure has more members than the minimum required to form a truss. In this setting, we introduce an optimization criterion to pick that solution (whether a truss or otherwise) that minimizes the criterion. For us, we shall attempt to minimize total weight. To keep things simple, we assume that the weight of a member is directly proportional to its volume and that the constant of proportionality (the *density* of the material) is the same for each member. (These assumptions are purely for notational convenience—a real engineer would certainly include these constants and let them vary from one member to the next). Hence, it suffices to minimize the total volume. The volume of one member, say, $\{i, j\}$, is its length $l_{ij} = \|p_j - p_i\|$ times its cross-sectional area. Again, to keep things as simple as possible, we assume that the cross-sectional area must be proportional to the tension/compression carried by the member (members carrying big loads must be "fat"—otherwise they might break). Let's set the constant of proportionality arbitrarily to one. Then the function that we should minimize is just the sum over all members of $l_{ij}|x_{ij}|$. Hence, our optimization problem can be written as follows:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{\{i,j\}\in\mathcal{A}} l_{ij}|x_{ij}| \\ \text{subject to} & \displaystyle\sum_{\substack{j:\\ \{i,j\}\in\mathcal{A}}} u_{ij}x_{ij} = -b_i \qquad i = 1, 2, \ldots, m. \end{array}$$

This problem is not a linear programming problem: the constraints are linear, but the objective function involves the absolute value of each variable. We can, however, convert this problem to a linear programming problem with the following trick. For each $\{i, j\} \in \mathcal{A}$, write x_{ij} as the difference between two nonnegative variables:

$$x_{ij} = x_{ij}^+ - x_{ij}^-, \qquad x_{ij}^+,\ x_{ij}^- \geq 0.$$

Think of x_{ij}^+ as the tension part of x_{ij} and x_{ij}^- as the compression part. The absolute value can then be modeled as the sum of these components

$$|x_{ij}| = x_{ij}^+ + x_{ij}^-.$$

We allow both components to be positive at the same time, but no minimum-weight solution will have any member with both components positive, since if there were such a member, the tension component and the compression component could be decreased simultaneously at the same rate without changing the force balance equations but reducing the weight. This reduction contradicts the minimum-weight assumption.

We can now state the linear programming formulation of the minimum weight structural design problem as follows:

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{\{i,j\}\in\mathcal{A}} (l_{ij}x_{ij}^+ + l_{ij}x_{ij}^-) & \\
\text{subject to} & \displaystyle\sum_{\substack{j:\\ \{i,j\}\in\mathcal{A}}} (u_{ij}x_{ij}^+ - u_{ij}x_{ij}^-) = -b_i & i = 1, 2, \ldots, m \\
& \qquad\qquad x_{ij}^+,\ x_{ij}^- \geq 0 & \{i,j\} \in \mathcal{A}.
\end{array}$$

In terms of the incidence matrix, each column must now be written down twice, once as before and once as the negative of before.

6. Anchors Away

So far we have considered structures that are free floating in the sense that even though loads are applied at various joints, we have not assumed that any of the joints are anchored to a large object such as the Earth. This setup is fine for structures intended for a rocket or a space station, but for Earth-bound applications it is generally desired to anchor some joints. It is trivial to modify the formulation we have already given to cover the situation where some of the joints are anchored. Indeed, the d force balance equations associated with an anchored joint are simply dropped as constraints, since the Earth supplies whatever counterbalancing force is needed. Of course, one can consider dropping only some of the d force balance equations associated with a particular joint. In this case, the physical interpretation is quite simple. For example, in two dimensions it simply means that the joint is allowed to roll on a track that is aligned with one of the coordinate directions but is not allowed to move off the track.

If enough "independent" constraints are dropped (at least three in two dimensions and at least six in three dimensions), then there are no longer any limitations on the applied loads—the structure will be sufficiently well anchored so that the Earth will apply whatever forces are needed to prevent the structure from moving. This is the most typical scenario under which these problems are solved. It makes setting up the problem much easier, since one no longer needs to worry about supplying loads that can't be balanced.

We end this chapter with one realistic example. Suppose the need exists to design a bracket to support a hanging load at a fixed distance from a wall. This bracket will be molded out of plastic, which means that the problem of finding an optimal design belongs to the realm of continuum mechanics. However, we can get an idea of the optimal shape by modeling the problem discretely (don't tell anyone). That is, we define a lattice of joints as shown in Figure 16.4 and introduce a set of members from which the bracket can be constructed. Each joint has members connecting it to several nearby joints. Figure 16.5 shows the members connected to one specific joint. Each joint in the structure has this connection "topology" with,

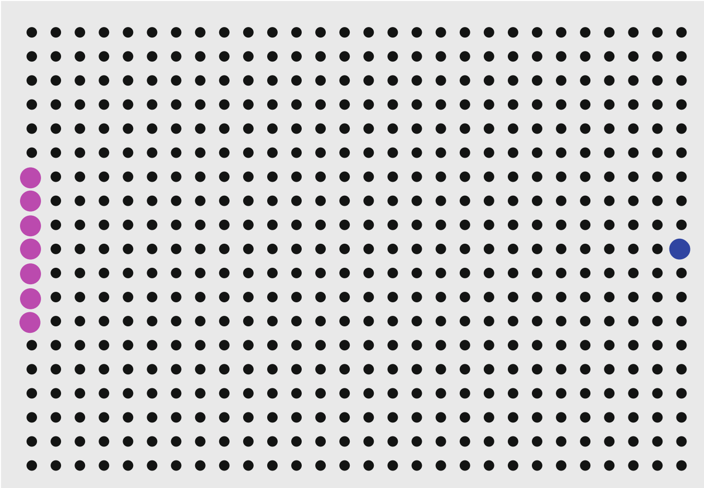

FIGURE 16.4. The set of joints used for the discrete approximation to the bracket design problem. The highlighted joints on the left are anchored to the wall, and the highlighted joint on the right must support the hanging load.

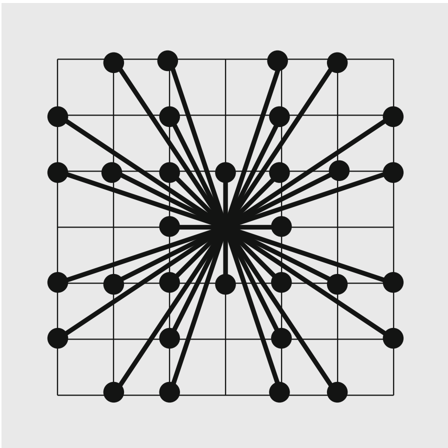

FIGURE 16.5. The members connected to a single interior joint.

of course, the understanding that joints close to the boundary do not have any member for which the intended connecting joint does not exist. The highlighted joints on the left side in Figure 16.4 are the anchored joints, and the highlighted joint on the right side is the joint to which the hanging load is applied (by "hanging," we mean that the applied load points downward). The optimal solution is shown in Figure 16.6. The thickness of each member is drawn in proportion to the square root of

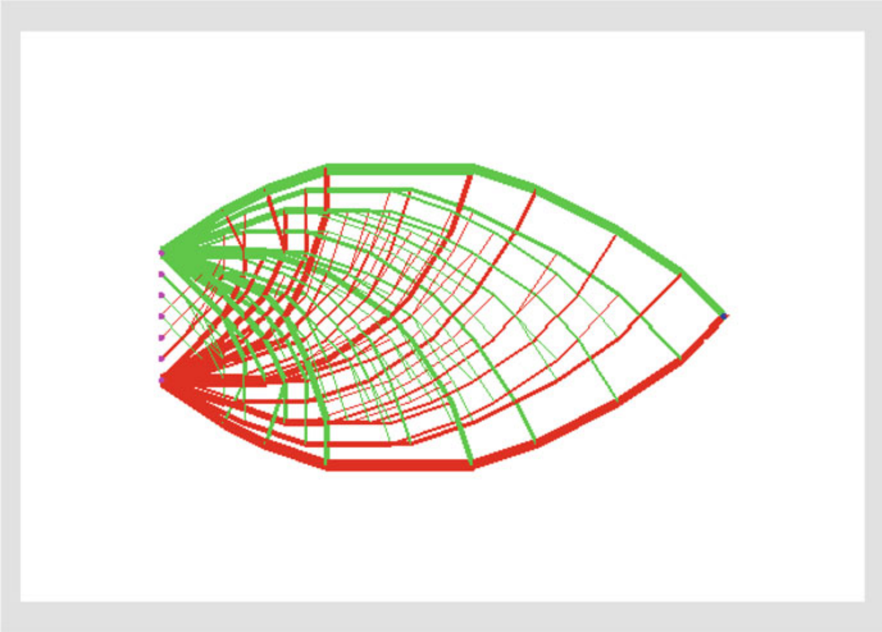

FIGURE 16.6. The minimum weight bracket.

the tension/compression in the member (since if the structure actually exists in three dimensions, the diameter of a member would be proportional to the square root of the cross-sectional area). Also, those members under compression are drawn in dark gray, whereas those under tension are drawn in light gray. Note that the compression members appear to cross the tension members at right angles. These curves are called *principle stresses*. It is a fundamental result in continuum mechanics that the principle tension stresses cross the principle compression stresses at right angles. We have discovered this result using optimization.

Most nonexperts find the solution to this problem to be quite surprising, since it covers such a large area. Yet it is indeed optimal. Also, one can see that the continuum solution should be roughly in the shape of a leaf.

Exercises

16.1 Show that a matrix R is skew symmetric if and only if

$$\xi^T R \xi = 0, \qquad \text{for every vector } \xi.$$

16.2 Which of the structures shown in Figure 16.7 is stable? (Note: each structure is shown embedded in a convenient coordinate system.)

16.3 Which of the structures shown in Figure 16.7 is a truss?

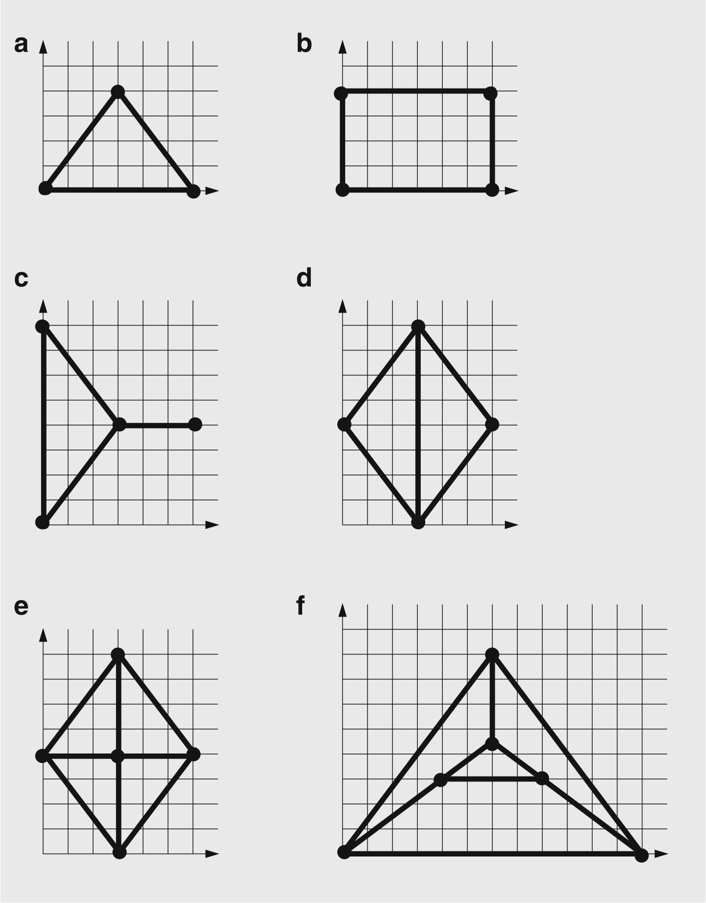

FIGURE 16.7. Structures for Exercises 16.2 and 16.3.

16.4 Assuming that the total applied force vanishes, show that total torque is translation invariant. That is, for any vector $\xi \in \mathbb{R}^d$,

$$\sum_i (R(p_i - \xi))^T b_i = \sum_i (Rp_i)^T b_i.$$

16.5 In 3-dimensions there are 5 regular (Platonic) solids. They are shown in Figure 16.8 and have the following number of vertices and edges:

	vertices	edges
tetrahedron	4	6
cube	8	12
octahedron	6	12
dodecahedron	20	30
icosahedron	12	30

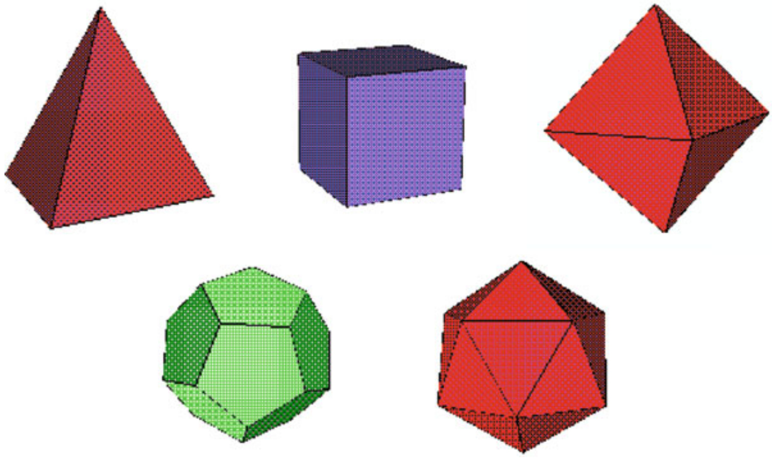

FIGURE 16.8. The five regular solids.

If one were to construct pin-jointed wire-frame models of these solids, which ones would be stable?

Notes

Structural optimization has its roots in Michell (1904). The first paper in which truss design was formulated as a linear programming problem is Dorn et al. (1964). A few general references on the subject include Hemp (1973), Rozvany (1989), Bendsøe et al. (1994), and Recski (1989).