Linear Programming: Chapter 15
Structural Optimization

Robert J. Vanderbei

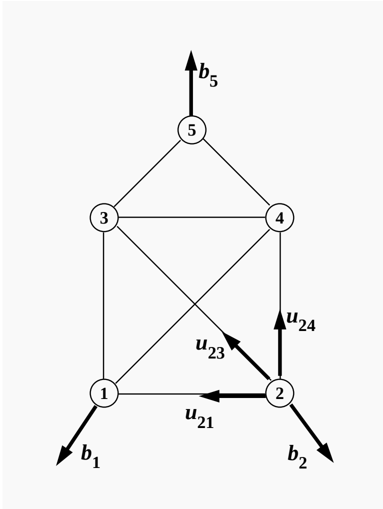
November 4, 2007

Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544
http://www.princeton.edu/∼rvdb

Structural Optimization

Forces: x_{ij} = tension in *member* $\{i, j\}$.

- $x_{ij} = x_{ji}$.
- Compression = -Tension.



Force Balance:

Look at joint 2:

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

Notations:

$$p_i = \text{position vector for joint } i$$

$$u_{ij} = \frac{p_j - p_i}{\|p_j - p_i\|} \quad (\text{ Note } u_{ji} = -u_{ij})$$

Constraints:

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

Matrix Form

$$Ax = -b$$

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

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

Notes:

- $\|u_{ij}\| = \|u_{ji}\| = 1$.
- $u_{ij} = -u_{ji}$.
- Each column contains a u_{ij}, a u_{ji}, and rest are zero.
- In one dimension, exactly a node-arc incidence matrix.

Minimum Weight Structural Design

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

Not quite an LP.
Use a common trick:

$$\begin{aligned} x_{ij} &= x_{ij}^+ - x_{ij}^-, \qquad x_{ij}^+, x_{ij}^- \geq 0 \\ |x_{ij}| &= x_{ij}^+ + x_{ij}^- \end{aligned}$$

Reformulated as an LP:

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

Redundant Equations

Recall network flows:

number of redundant equations $=$ number of connected components.

Row combinations:

$$y_i^T u_{ij} + y_j^T u_{ji}$$

Sum of "x"-component rows:

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} u_{ij}^{(1)} \\ u_{ij}^{(2)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} u_{ji}^{(1)} \\ u_{ji}^{(2)} \end{bmatrix} = 0$$

Sum of "y"-component rows, "z"-component rows, etc. is similar.

Are There Others?

Yes. Put

$$y_i = Rp_i, \qquad R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \qquad R^T = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = -R.$$

Compute:

$$\begin{aligned} y_i^T u_{ij} + y_j^T u_{ji} &= p_i^T R^T u_{ij} + p_j^T R^T u_{ji} \\ &= (p_i - p_j)^T R^T u_{ij} \\ &= -\frac{(p_j - p_i)^T R^T (p_j - p_i)}{\|p_j - p_i\|} \\ &= 0 \end{aligned}$$

Last equality follows from:

$$\begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \xi_1\xi_2 - \xi_1\xi_2 = 0 \qquad \text{for all } \xi_1, \xi_2$$

Skew Symmetric Matrices

Definition.

$$R^T = -R$$

For $d = 1$: no nonzero ones.

For $d = 2$:

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

For $d = 3$:

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

Structure is *stable* if the redundancies just identified represent the *only* redundancies.

Conservation Laws

Suppose a combination of rows of A vanishes.
Then the same combination of elements of b must vanish.

Force Balance:

$$\sum_i b_i^{(1)} = 0 \quad \text{and} \quad \sum_i b_i^{(2)} = 0$$

What is meaning of the other redundancies?

$$\sum_i (Rp_i)^T b_i = 0$$

Answer...

Torque Balance

Consider *two-dimensional* case:

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

Physically, this matrix rotates vectors 90° counterclockwise.

Let $v_i = p_i / \|p_i\|$ be a *unit vector* pointing in the direction of p_i:

$$p_i = \|p_i\| v_i.$$

Then,

$$\begin{aligned} (Rp_i)^T b_i &= \|p_i\| (Rv_i)^T b_i \\ &= (\text{length of moment arm})(\text{proj of force perp to moment arm}) \end{aligned}$$

In *three dimensions*, three independent torques: *roll, pitch, yaw*.

They correspond to the three basis matrices given before.

Note: torque balance is invariant under parallel tranlation of axis.

Trusses

Definition.

- Stable
- Has $md - d(d+1)/2$ members (d is dimension).

Anchors

No force balance equation at *anchored* joints.

Earth provides counterbalancing force.

If enough $(d(d+1)/2)$ independent constraints are dropped (due to anchoring), then no force balance or torque balance limitations remain.

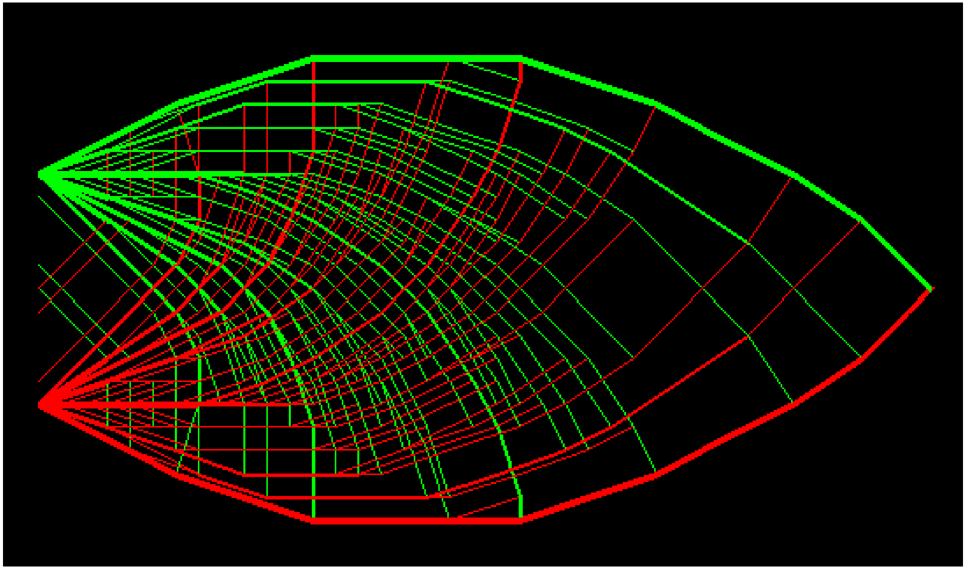
AMPL Model

```
param m default 26;         # must be even
param n default 39;

set X := {0..n};
set Y := {0..m};

set NODES := X cross Y;     # A lattice of Nodes

set ANCHORS within NODES
    := { x in X, y in Y :
         x == 0 && y >= floor(m/3) && y <= m-floor(m/3) };

param xload {(x,y) in NODES: (x,y) not in ANCHORS} default 0;
param yload {(x,y) in NODES: (x,y) not in ANCHORS} default 0;

param gcd {x in -n..n, y in -n..n} :=
    (if x <  0 then gcd[-x,y] else
    (if x == 0 then y else
    (if y < x  then gcd[y,x] else
    (gcd[y mod x, x])
    )));
```

```
set ARCS := { (xi,yi) in NODES, (xj,yj) in NODES:
    abs( xj-xi ) <= 3            &&
    abs(yj-yi) <=3               &&
    abs(gcd[ xj-xi, yj-yi ]) == 1 &&
    ( xi > xj || (xi == xj && yi > yj) )
    };

param length {(xi,yi,xj,yj) in ARCS} := sqrt( (xj-xi)^2 + (yj-yi)^2 );

var comp {ARCS} >= 0;
var tens {ARCS} >= 0;

minimize volume:
    sum {(xi,yi,xj,yj) in ARCS}
        length[xi,yi,xj,yj] * (comp[xi,yi,xj,yj] + tens[xi,yi,xj,yj]);

subject to Xbalance {(xi,yi) in NODES: (xi,yi) not in ANCHORS}:
    sum { (xi,yi,xj,yj) in ARCS }
        ((xj-xi)/length[xi,yi,xj,yj]) * (comp[xi,yi,xj,yj]-tens[xi,yi,xj,yj])
    +
    sum { (xk,yk,xi,yi) in ARCS }
        ((xi-xk)/length[xk,yk,xi,yi]) * (tens[xk,yk,xi,yi]-comp[xk,yk,xi,yi])
    =
    xload[xi,yi];
    ;
```

```
subject to Ybalance {(xi,yi) in NODES: (xi,yi) not in ANCHORS}:
    sum { (xi,yi,xj,yj) in ARCS }
        ((yj-yi)/length[xi,yi,xj,yj]) * (comp[xi,yi,xj,yj]-tens[xi,yi,xj,yj])
    +
    sum { (xk,yk,xi,yi) in ARCS }
        ((yi-yk)/length[xk,yk,xi,yi]) * (tens[xk,yk,xi,yi]-comp[xk,yk,xi,yi])
    =
    yload[xi,yi];
    ;

let yload[n,m/2] := -1;

solve;

printf: "%d \n",
    card({(xi,yi,xj,yj) in ARCS: comp[xi,yi,xj,yj]+tens[xi,yi,xj,yj] > 1.0e-4})
    > structure.out;

printf {(xi,yi,xj,yj) in ARCS: comp[xi,yi,xj,yj] + tens[xi,yi,xj,yj] > 1.0e-4}:
    "%3d %3d %3d %3d %10.4f \n",
    xi, yi, xj, yj, tens[xi,yi,xj,yj] - comp[xi,yi,xj,yj]
    > structure.out;
```

The Michel Bracket



Constraints:	2,138
Variables:	31,034
Time:	193 secs

Click here for parametric self-dual simplex method animation tool.
Click here for affine-scaling method animation tool.