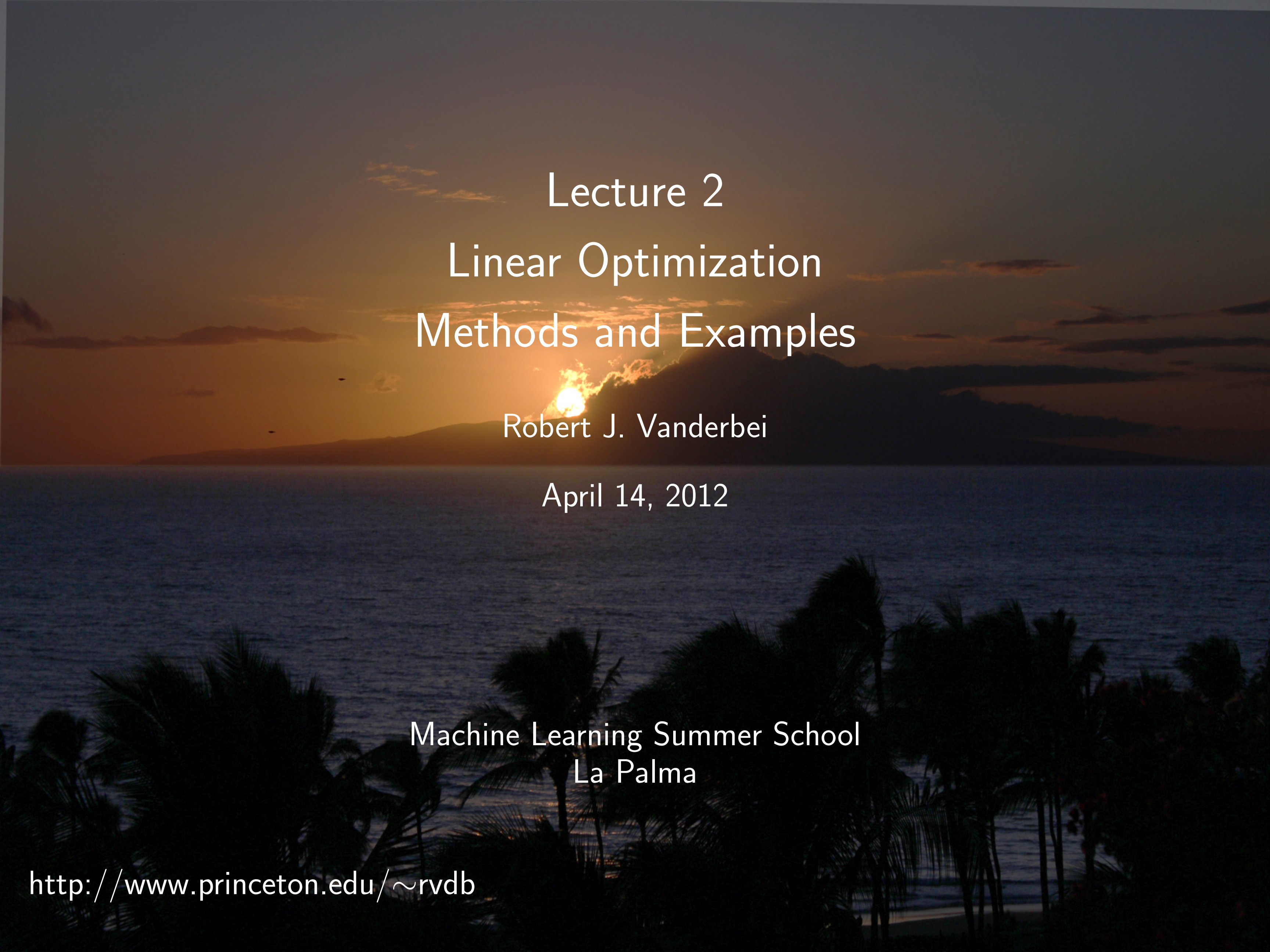


Lecture 2

Linear Optimization
Methods and Examples

Robert J. Vanderbei

April 14, 2012

Machine Learning Summer School
La Palma

http://www.princeton.edu/∼rvdb

Parametric Self-Dual Simplex Method

An Example

$$\begin{array}{rrcrcrcr} \text{maximize} & -3\,x_1 & + & 11\,x_2 & + & 2\,x_3 & & \\ \text{subj. to} & -x_1 & + & 3\,x_2 & & & \leq & 5 \\ & 3\,x_1 & + & 3\,x_2 & & & \leq & 4 \\ & & & 3\,x_2 & + & 2\,x_3 & \leq & 6 \\ & -3\,x_1 & & & - & 5\,x_3 & \leq & -4 \\ & & & x_1, & x_2, & x_3 & \geq & 0. \end{array}$$

Initial Dictionary:

$$\begin{array}{rcrcrcrcr} \zeta & = & & & -3\,x_1 & + & 11\,x_2 & + & 2\,x_3 \\ \hline w_1 & = & 5 & + & x_1 & - & 3\,x_2 & & \\ w_2 & = & 4 & - & 3\,x_1 & - & 3\,x_2 & & \\ w_3 & = & 6 & & & - & 3\,x_2 & - & 2\,x_3 \\ w_4 & = & -4 & + & 3\,x_1 & & & + & 5\,x_3 \end{array}$$

Note: neither primal nor dual feasible.

Perturb

Introduce a parameter μ and perturb:

$$\begin{array}{rcrcrcrcrcr}
\zeta & = & & & & & -3\,x_1 & + & 11\,x_2 & + & 2\,x_3 \\
 & & & & & & -\mu x_1 & - & \mu x_2 & - & \mu x_3 \\
\hline
w_1 & = & 5 & + & \mu & + & x_1 & - & 3\,x_2 & & \\
w_2 & = & 4 & + & \mu & - & 3\,x_1 & - & 3\,x_2 & & \\
w_3 & = & 6 & + & \mu & & & - & 3\,x_2 & - & 2\,x_3 \\
w_4 & = & -4 & + & \mu & + & 3\,x_1 & & & + & 5\,x_3
\end{array}$$

For μ large, dictionary is *optimal*.

Question: For which μ values is dictionary optimal?

Answer:

$$\begin{array}{rcccll}
-3 & - & \mu & \leq & 0 & \\
11 & - & \mu & \leq & 0 & * \\
2 & - & \mu & \leq & 0 & * \\
\hline
5 & + & \mu & \geq & 0 & \\
4 & + & \mu & \geq & 0 & \\
6 & + & \mu & \geq & 0 & \\
-4 & + & \mu & \geq & 0 & *
\end{array}$$

Note: only those marked with (*) give inequalities that omit $\mu = 0$.

Tightest:

$$\mu \geq 11$$

Achieved by: objective row perturbation on x_2.

Let x_2 *enter.*

Who Leaves?

Do ratio test using current lowest μ value, i.e. $\mu = 11$:

$$\begin{array}{rcccccc} 5 & + & 11 & - & 3\,x_2 & \geq & 0 \\ 4 & + & 11 & - & 3\,x_2 & \geq & 0 \\ 6 & + & 11 & - & 3\,x_2 & \geq & 0 \\ -4 & + & 11 & & & \geq & 0 \end{array}$$

Tightest:

$$4 + 11 - 3\,x_2 \geq 0.$$

Achieved by: constraint containing basic variable w_2.

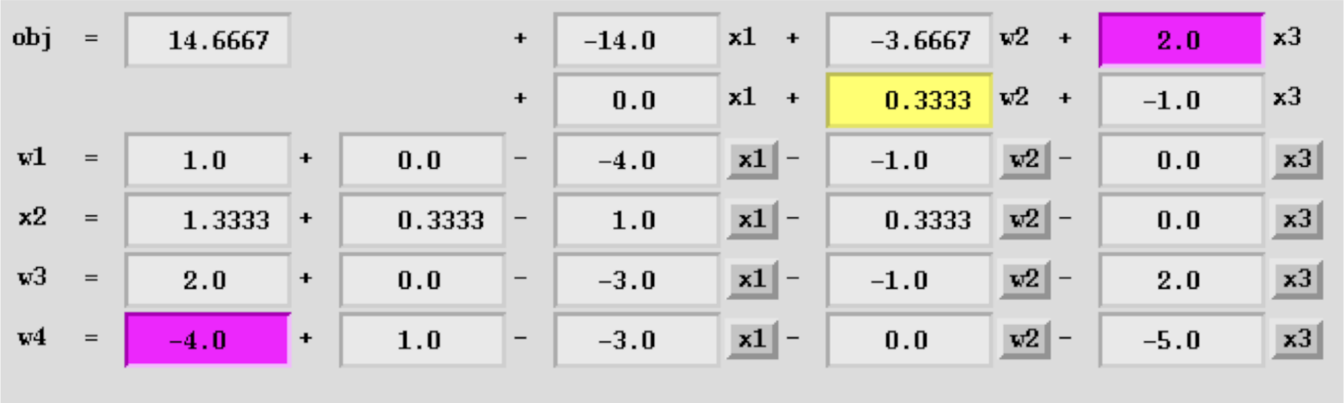
Let w_2 *leave.*

After the pivot:

$$
\begin{array}{rcrcrcrcrcr}
\zeta & = & 14.67 & & & - & 14\,x_1 & - & 3.67\,w_2 & + & 2\,x_3 \\
 & & & & & & & + & 0.33\mu w_2 & - & \mu x_3 \\
\hline
w_1 & = & 1 & & & + & 4\,x_1 & + & w_2 & & \\
x_2 & = & 1.33 & + & 0.33\mu & - & x_1 & - & 0.33\,w_2 & & \\
w_3 & = & 2 & & & + & 3\,x_1 & + & w_2 & - & 2\,x_3 \\
w_4 & = & -4 & + & \mu & + & 3\,x_1 & & & + & 5\,x_3
\end{array}
$$

Second Pivot

Using the *advanced* pivot tool, the current dictionary is:



obj	=	14.6667			+	-14.0	x1	+	-3.6667	w2	+	2.0	x3
					+	0.0	x1	+	0.3333	w2	+	-1.0	x3
w1	=	1.0	+	0.0	−	-4.0	x1	−	-1.0	w2	−	0.0	x3
x2	=	1.3333	+	0.3333	−	1.0	x1	−	0.3333	w2	−	0.0	x3
w3	=	2.0	+	0.0	−	-3.0	x1	−	-1.0	w2	−	2.0	x3
w4	=	-4.0	+	1.0	−	-3.0	x1	−	0.0	w2	−	-5.0	x3

Note: the parameter μ is not shown. *But it is there!*

Question: For which μ values is dictionary optimal? Answer:

$$
\begin{array}{rcll|rcll}
-14 & \leq & 0 & & 1 & \geq & 0 & \\
-3.67+0.33\mu & \leq & 0 & & 1.33+0.33\mu & \geq & 0 & \\
2- \quad \mu & \leq & 0 & * & 2 & \geq & 0 & \\
 & & & & -4+ \quad \mu & \geq & 0 & *
\end{array}
$$

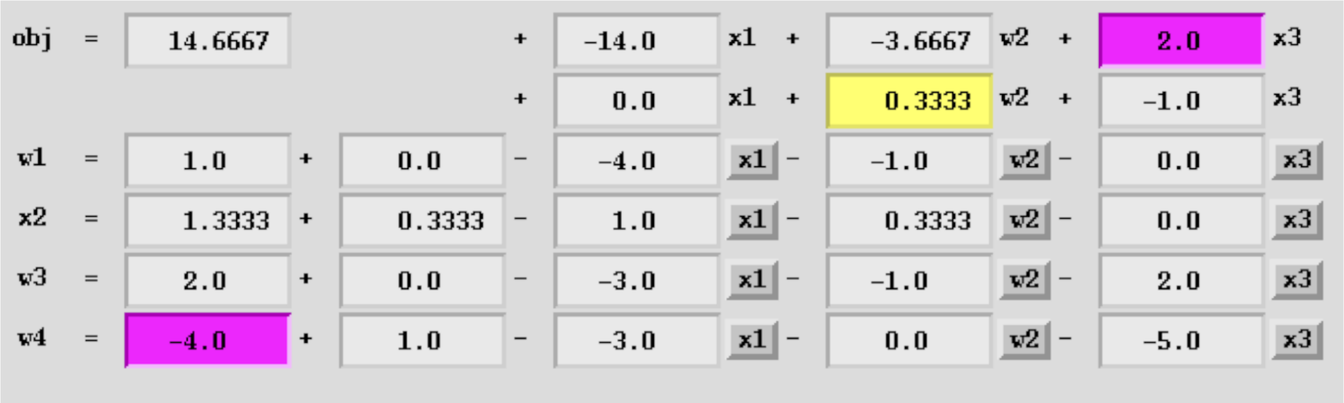
Tightest lower bound: $\mu \geq 4$.

Achieved by: constraint containing basic variable w_4. Let w_4 *leave.*

Second Pivot–Continued

Who shall enter?

Recall the current dictionary:



obj	=	14.6667			+	-14.0	x1	+	-3.6667	w2	+	2.0	x3
					+	0.0	x1	+	0.3333	w2	+	-1.0	x3
w1	=	1.0	+	0.0	-	-4.0	x1	-	-1.0	w2	-	0.0	x3
x2	=	1.3333	+	0.3333	-	1.0	x1	-	0.3333	w2	-	0.0	x3
w3	=	2.0	+	0.0	-	-3.0	x1	-	-1.0	w2	-	2.0	x3
w4	=	-4.0	+	1.0	-	-3.0	x1	-	0.0	w2	-	-5.0	x3

Do *dual-type* ratio test using current lowest μ value, i.e. $\mu = 4$:

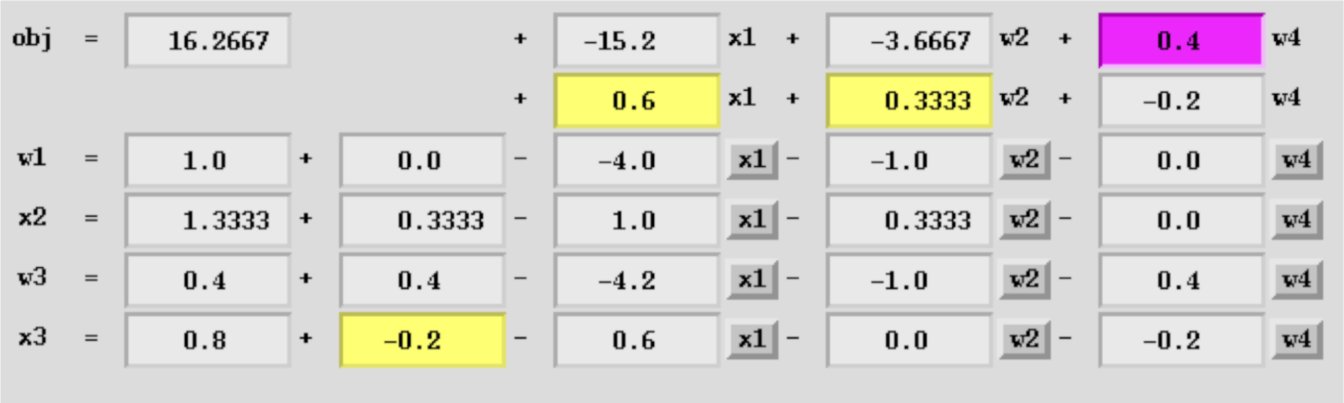
$$\begin{array}{rcrcrcl} 14 & + & 0 * 4 & - & 3y_4 & \geq & 0 \\ 3.67 & - & 0.33 * 4 & & & \geq & 0 \\ -2 & + & 1 * 4 & - & 5y_4 & \geq & 0 \end{array}$$

Tightest: $-2 + 1 * 4 - 5y_4 \geq 0$.

Achieved by: objective term containing nonbasic variable x_3. Let x_3 *enter.*

Third Pivot

The current dictionary is:



obj	=	16.2667			+	-15.2	x1	+	-3.6667	w2	+	0.4	w4
					+	0.6	x1	+	0.3333	w2	+	-0.2	w4
w1	=	1.0	+	0.0	−	-4.0	x1	−	-1.0	w2	−	0.0	w4
x2	=	1.3333	+	0.3333	−	1.0	x1	−	0.3333	w2	−	0.0	w4
w3	=	0.4	+	0.4	−	-4.2	x1	−	-1.0	w2	−	0.4	w4
x3	=	0.8	+	-0.2	−	0.6	x1	−	0.0	w2	−	-0.2	w4

Question: For which μ is dictionary optimal? Answer:

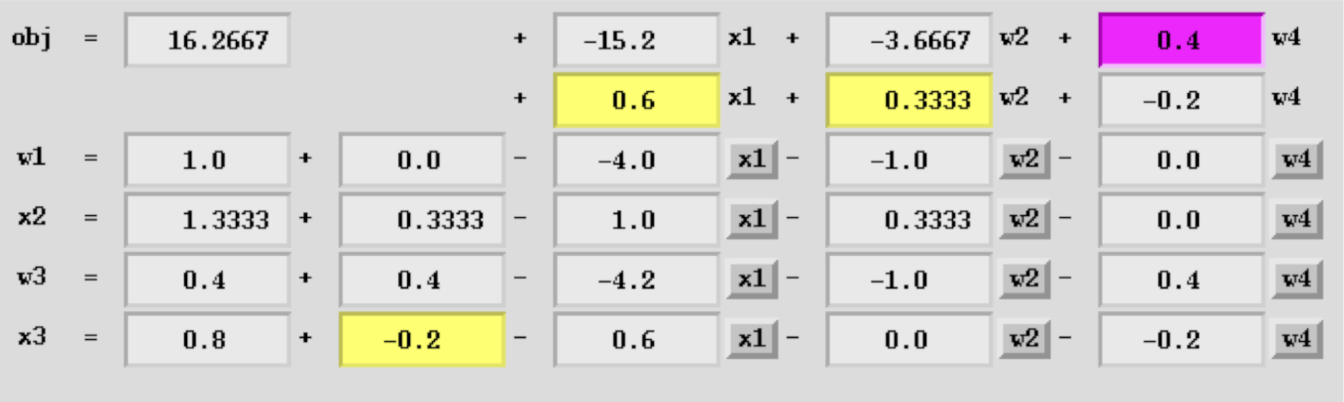
$$\begin{array}{rcrcll|rcrcl} -15.2 & + & 0.6\mu & \leq & 0 & & 1 & & & \geq & 0 \\ -3.67 & + & 0.33\mu & \leq & 0 & & 1.33 & + & 0.33\mu & \geq & 0 \\ 0.4 & - & 0.2\mu & \leq & 0 & * & 0.4 & + & 0.4\mu & \geq & 0 \\ & & & & & & 0.8 & - & 0.2\mu & \geq & 0 \end{array}$$

Tightest lower bound: $\mu \geq 2$.

Achieved by: objective term containing nonbasic variable w_4. Let w_4 *enter.*

Third Pivot–Continued

Who shall leave? Recall the current dictionary:



obj	=	16.2667			+	-15.2	x1	+	-3.6667	w2	+	0.4	w4
					+	0.6	x1	+	0.3333	w2	+	-0.2	w4
w1	=	1.0	+	0.0	-	-4.0	x1	-	-1.0	w2	-	0.0	w4
x2	=	1.3333	+	0.3333	-	1.0	x1	-	0.3333	w2	-	0.0	w4
w3	=	0.4	+	0.4	-	-4.2	x1	-	-1.0	w2	-	0.4	w4
x3	=	0.8	+	-0.2	-	0.6	x1	-	0.0	w2	-	-0.2	w4

Do *primal-type* ratio test using current lowest μ value, i.e. $\mu = 2$:

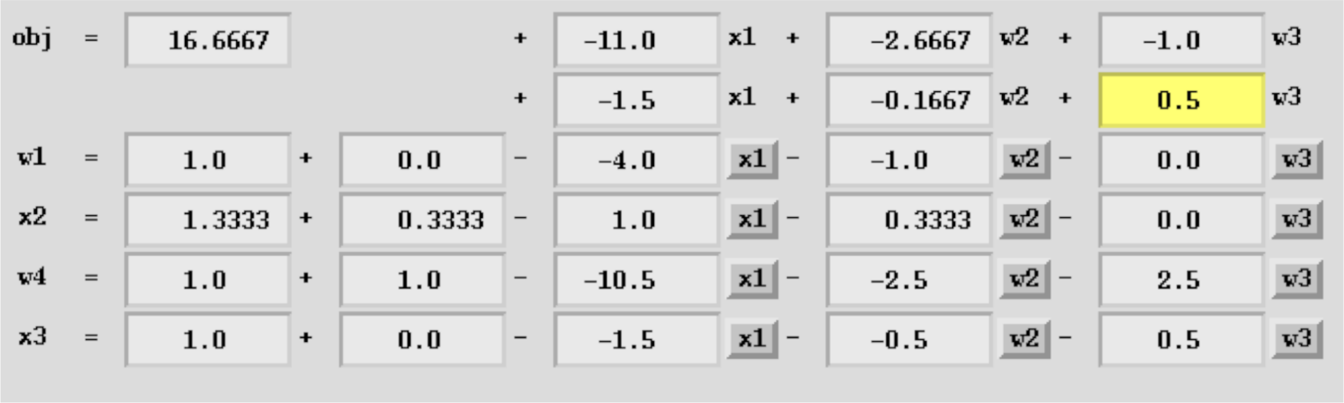
$$\begin{array}{rcrcrcl} 1 & + & 0 * 2 & & & \geq & 0 \\ 1.33 & + & 0.33 * 2 & & & \geq & 0 \\ 0.4 & + & 0.4 * 2 & - & 0.4\, w_4 & \geq & 0 \\ 0.8 & - & 0.2 * 2 & + & 0.2\, w_4 & \geq & 0 \end{array}$$

Tightest: $0.4 + 0.4 * 2 - 0.4\, w_4 \geq 0$.

Achieved by: constraint containing basic variable w_3. Let w_3 *leave.*

Fourth Pivot

The current dictionary is:



obj	=	16.6667			+	-11.0	x1	+	-2.6667	w2	+	-1.0	w3
					+	-1.5	x1	+	-0.1667	w2	+	0.5	w3
w1	=	1.0	+	0.0	-	-4.0	x1	-	-1.0	w2	-	0.0	w3
x2	=	1.3333	+	0.3333	-	1.0	x1	-	0.3333	w2	-	0.0	w3
w4	=	1.0	+	1.0	-	-10.5	x1	-	-2.5	w2	-	2.5	w3
x3	=	1.0	+	0.0	-	-1.5	x1	-	-0.5	w2	-	0.5	w3

It's *optimal!* Also, the range of μ values includes $\mu = 0$:

$$\begin{array}{rcrcl|rcrcl} -11 & - & 1.5\mu & \leq & 0 & 1 & & & \geq & 0 \\ -2.67 & - & 0.167\mu & \leq & 0 & 1.33 & + & 0.33\mu & \geq & 0 \\ -1 & + & 0.5\mu & \leq & 0 & 1 & + & 1\mu & \geq & 0 \\ & & & & & 1 & & & \geq & 0 \end{array}$$

That is, $-1 \leq \mu \leq 2$.

Range of μ values is shown at bottom of pivot tool. Invalid ranges are highlighted in yellow.

Top Ten Reasons to Like this Method

- Freedom to pick perturbation as you like.
- Randomizing perturbation completely solves the degeneracy problem.
- Perturbations don't have to be "small".
- In the optimal dictionary, perturbation is completely gone—no need to remove it.
- In some real-world problems, a "natural" perturbation exists (next lecture).
- The average-case performance can be analyzed (lecture after that).

Okay, there are only 6 items in the list. SORRY.

An Example: Structural Optimization

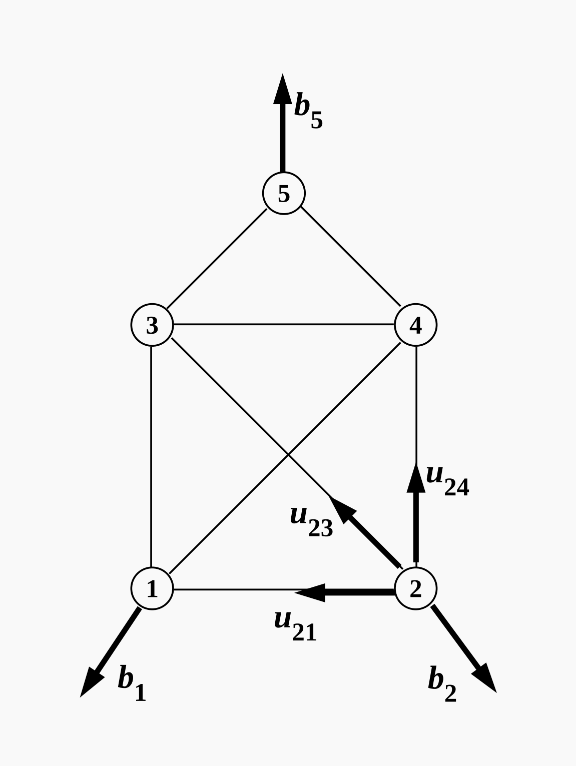
Forces: x_{ij} = tension in *beam* (aka *member*) $\{i, j\}$.

- $x_{ij} = x_{ji}$.
- 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:

$$\begin{aligned} p_i &= \text{position vector for joint } i \\ u_{ij} &= \frac{p_j - p_i}{\|p_j - p_i\|} \qquad (\text{ Note } u_{ji} = -u_{ij}) \end{aligned}$$

Constraints:

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

Matrix Form

$$Ax = -b$$

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

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

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:

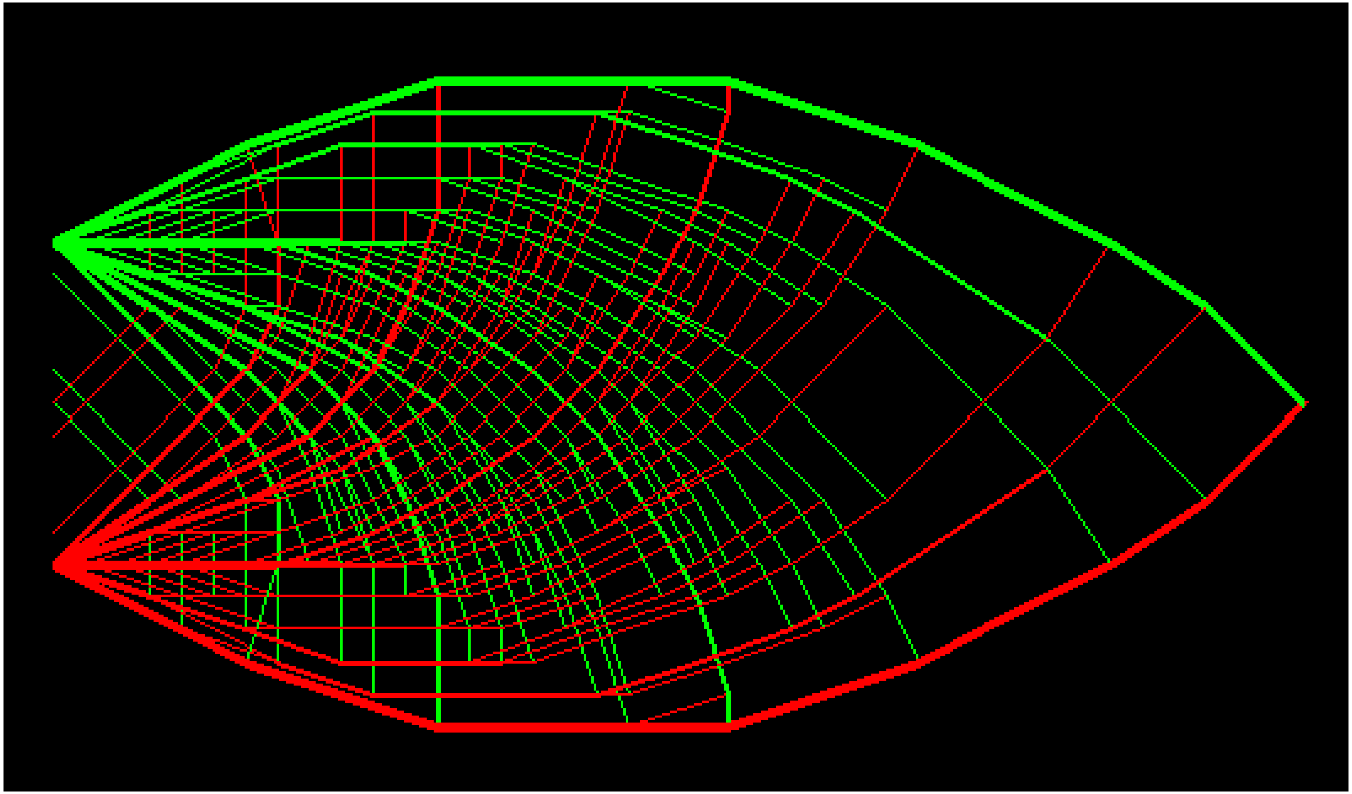
$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{\{i,j\}\in\mathcal{A}} (l_{ij}x_{ij}^+ + l_{ij}x_{ij}^-) \\ \text{subject to} & \displaystyle\sum_{\substack{j: \\ \{i,j\}\in\mathcal{A}}} (u_{ij}x_{ij}^+ - u_{ij}x_{ij}^-) = -b_i \qquad i = 1, 2, \ldots, m \\ & \qquad\qquad x_{ij}^+, \; x_{ij}^-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

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;
```

The Michell Bracket (1904)



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.

Regression

- Means and Medians

- Least Squares Regression

- Least Absolute Deviation (LAD) Regression

- LAD via LP

- Average Complexity of Parametric Self-Dual Simplex Method

1995 Adjusted Gross Incomes

Consider 1995 Adjusted Gross Incomes on Individual Tax Returns:

Individual	AGI
b_1	\$25,462
b_2	\$45,110
b_3	\$15,505
$\vdots$	$\vdots$
b_{m-1}	\$33,265
b_m	\$75,420

Real summary data is shown on the next slide...

Table 1.--1995, Individual Income Tax Returns

[All figures are estimates based on samples--money amounts are in thousands of dollars]

Size of adjusted gross income	Number of returns	Adjusted gross income
	(1)	(2)
All returns	**118,218,327**	**4,189,353,615**
No adjusted gross income	944,141	-55,253,648
$1 under $5,000	14,646,131	37,604,828
$5,000 under $10,000	13,982,404	104,603,365
$10,000 under $15,000	13,562,088	169,317,443
$15,000 under $20,000	11,385,632	198,418,324
$20,000 under $25,000	9,970,099	223,400,219
$25,000 under $30,000	7,847,862	215,200,244
$30,000 under $40,000	12,380,339	430,491,242
$40,000 under $50,000	9,098,760	406,638,597
$50,000 under $75,000	13,679,023	828,349,278
$75,000 under $100,000	5,374,489	458,505,650
$100,000 under $200,000	4,074,852	532,030,480
$200,000 under $500,000	1,007,136	292,117,517
$500,000 under $1,000,000	178,374	120,347,093
$1,000,000 or more	86,998	227,582,987
Taxable returns	**89,252,989**	**4,007,580,441**
Nontaxable returns	**28,965,338**	**181,773,174**

Means and Medians

Median:

$$\hat{x} = b_{\frac{1+m}{2}} \approx \$22{,}500.$$

Mean:

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} b_i = \$4{,}189{,}353{,}615{,}000/118{,}218{,}327 = \$35{,}437.$$

Mean's Connection with Optimization

$$\bar{x} = \text{argmin}_x \sum_{i=1}^{m} (x - b_i)^2 .$$

Proof:

$$\begin{aligned}
f(x) &= \sum_{i=1}^{m} (x - b_i)^2 \\
f'(x) &= \sum_{i=1}^{m} 2(x - b_i) \\
f'(\bar{x}) &= 0 \qquad \Longrightarrow \qquad \bar{x} = \frac{1}{m} \sum_{i=1}^{m} b_i \\
\lim_{x \to \pm\infty} f(x) &= +\infty \qquad \Longrightarrow \qquad \bar{x} \text{ is a minimum}
\end{aligned}$$

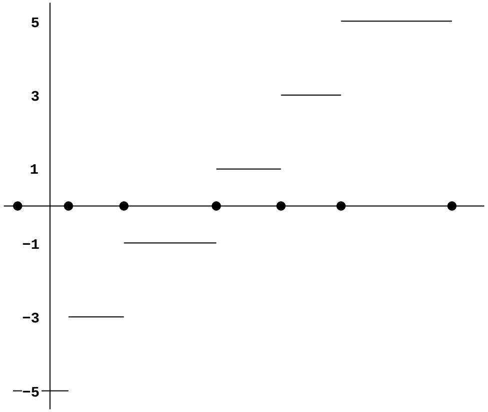
Median's Connection with Optimization

$$\hat{x} = \mathrm{argmin}_x \sum_{i=1}^{m} |x - b_i|.$$

Proof:

$$\begin{aligned} f(x) &= \sum_{i=1}^{m} |x - b_i| \\ f'(x) &= \sum_{i=1}^{m} \mathrm{sgn}\,(x - b_i) \qquad \text{where } \mathrm{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \\ &= (\#\text{ of } b_i\text{'s smaller than } x) - (\#\text{ of } b_i\text{'s larger than } x)\,. \end{aligned}$$

If m is odd:



Parametric Self-Dual Simplex Method: Data

Name	m	n	iters	Name	m	n	iters
25fv47	777	1545	5089	nesm	646	2740	5829
80bau3b	2021	9195	10514	recipe	74	136	80
adlittle	53	96	141	sc105	104	103	92
afiro	25	32	16	sc205	203	202	191
agg2	481	301	204	sc50a	49	48	46
agg3	481	301	193	sc50b	48	48	53
bandm	224	379	1139	scagr25	347	499	1336
beaconfd	111	172	113	scagr7	95	139	339
blend	72	83	117	scfxm1	282	439	531
bnl1	564	1113	2580	scfxm2	564	878	1197
bnl2	1874	3134	6381	scfxm3	846	1317	1886
boeing1	298	373	619	scorpion	292	331	411
boeing2	125	143	168	scrs8	447	1131	783
bore3d	138	188	227	scsd1	77	760	172
brandy	123	205	585	scsd6	147	1350	494
czprob	689	2770	2635	scsd8	397	2750	1548
d6cube	403	6183	5883	sctap1	284	480	643
degen2	444	534	1421	sctap2	1033	1880	1037
degen3	1503	1818	6398	sctap3	1408	2480	1339
e226	162	260	598	seba	449	896	766

Data Continued

Name	m	n	iters	Name	m	n	iters
etamacro	334	542	1580	share1b	107	217	404
fffff800	476	817	1029	share2b	93	79	189
finnis	398	541	680	shell	487	1476	1155
fit1d	24	1026	925	ship04l	317	1915	597
fit1p	627	1677	15284	ship04s	241	1291	560
forplan	133	415	576	ship08l	520	3149	1091
ganges	1121	1493	2716	ship08s	326	1632	897
greenbea	1948	4131	21476	ship12l	687	4224	1654
grow15	300	645	681	ship12s	417	1996	1360
grow22	440	946	999	sierra	1212	2016	793
grow7	140	301	322	standata	301	1038	74
israel	163	142	209	standmps	409	1038	295
kb2	43	41	63	stocfor1	98	100	81
lotfi	134	300	242	stocfor2	2129	2015	2127
maros	680	1062	2998				

A Regression Model for Algorithm Efficiency

Observed Data:

$$\begin{aligned} t &= \#\text{ of iterations} \\ m &= \#\text{ of constraints} \\ n &= \#\text{ of variables} \end{aligned}$$

Model:

$$t \approx 2^{\alpha}(m+n)^{\beta}$$

Linearization: Take logs:

$$\log t = \alpha \log 2 + \beta \log(m+n) + \underset{\substack{\uparrow \\ \text{error}}}{\epsilon}$$

Regression Model Continued

Solve several instances (say k of them):

$$\begin{bmatrix} \log t_1 \\ \log t_2 \\ \vdots \\ \log t_k \end{bmatrix} = \begin{bmatrix} \log 2 & \log(m_1 + n_1) \\ \log 2 & \log(m_2 + n_2) \\ \vdots & \vdots \\ \log 2 & \log(m_k + n_k) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_k \end{bmatrix}$$

In matrix notation:

$$b = Ax + \epsilon$$

Goal: find x that “minimizes” ϵ.

Least Squares Regression

Euclidean Distance: $\|x\|_2 = \left(\sum_i x_i^2\right)^{1/2}$

Least Squares Regression: $\bar{x} = \mathsf{argmin}_x \|b - Ax\|_2^2$

Calculus:

$$f(x) = \|b - Ax\|_2^2 = \sum_i \left(b_i - \sum_j a_{ij} x_j \right)^2$$

$$\frac{\partial f}{\partial x_k}(\bar{x}) = \sum_i 2 \left(b_i - \sum_j a_{ij} \bar{x}_j \right) (-a_{ik}) = 0, \qquad k = 1, 2, \dots, n$$

Rearranging,

$$\sum_i a_{ik} b_i = \sum_i \sum_j a_{ik} a_{ij} \bar{x}_j, \qquad k = 1, 2, \dots, n$$

In matrix notation,

$$A^T b = A^T A \bar{x}$$

Assuming $A^T A$ is invertible,

$$\bar{x} = \left(A^T A \right)^{-1} A^T b$$

Least Absolute Deviation Regression

Manhattan Distance: $\|x\|_1 = \sum_i |x_i|$

Least Absolute Deviation Regression: $\hat{x} = \text{argmin}_x \|b - Ax\|_1$

Calculus:

$$f(x) = \|b - Ax\|_1 = \sum_i \left| b_i - \sum_j a_{ij} x_j \right|$$

$$\frac{\partial f}{\partial x_k}(\hat{x}) = \sum_i \frac{b_i - \sum_j a_{ij}\hat{x}_j}{\left|b_i - \sum_j a_{ij}\hat{x}_j\right|}(-a_{ik}) = 0, \qquad k = 1, 2, \ldots, n$$

Rearranging,

$$\sum_i \frac{a_{ik} b_i}{\epsilon_i(\hat{x})} = \sum_i \sum_j \frac{a_{ik} a_{ij} \hat{x}_j}{\epsilon(\hat{x})}, \qquad k = 1, 2, \ldots, n$$

In matrix notation,

$$A^T E(\hat{x}) b = A^T E(\hat{x}) A \hat{x}, \qquad \text{where } E(\hat{x}) = \text{Diag}(\epsilon(\hat{x}))^{-1}$$

Assuming $A^T E(\hat{x}) A$ is invertible,

$$\hat{x} = \left(A^T E(\hat{x}) A\right)^{-1} A^T E(\hat{x}) b$$

Least Absolute Deviation Regression—Continued

An implicit equation.

Can be solved using *successive approximations*:

$$
\begin{aligned}
x^0 &= 0 \\
x^1 &= \left(A^T E(x^0) A\right)^{-1} A^T E(x^0) b \\
x^2 &= \left(A^T E(x^1) A\right)^{-1} A^T E(x^1) b \\
&\vdots \\
x^{k+1} &= \left(A^T E(x^k) A\right)^{-1} A^T E(x^k) b \\
&\vdots \\
\hat{x} &= \lim_{k \to \infty} x^k
\end{aligned}
$$

Least Absolute Deviation Regression via Linear Programming

$$\min \sum_i \left| b_i - \sum_j a_{ij} x_j \right|$$

Equivalent Linear Program:

$$\begin{aligned} &\min \sum_i t_i \\ &-t_i \ \le \ b_i - \sum_j a_{ij} x_j \ \le \ t_i \qquad i = 1, 2, \ldots, m \end{aligned}$$

AMPL Model

```
param m;
param n;

set I := {1..m};
set J := {1..n};

param A {I,J};
param b {I};

var x{J};
var t{I};

minimize sum_dev:
        sum {i in I} t[i];

subject to lower_bound {i in I}:
    -t[i] <= b[i] - sum {j in J} A[i,j]*x[j];

subject to upper_bound {i in I}:
             b[i] - sum {j in J} A[i,j]*x[j] <= t[i];
```

Parametric Self-Dual Simplex Method

Thought experiment:

- μ starts at ∞.
- In reducing μ, there are $n + m$ barriers.
- At each iteration, one barrier is passed—the others move about randomly.
- To get μ to zero, we must on average pass half the barriers.
- Therefore, on average the algorithm should take $(m + n)/2$ iterations.

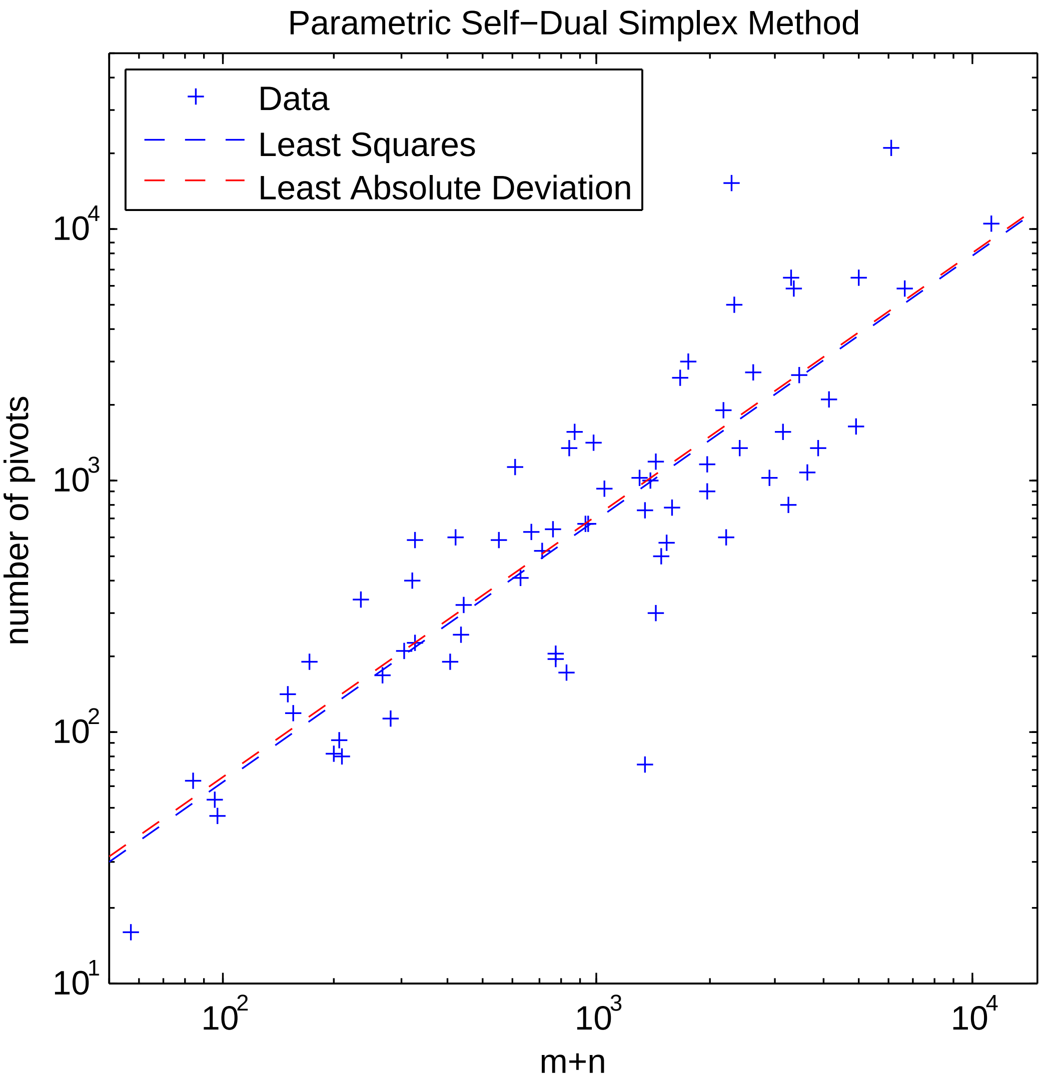
Using 69 real-world problems from the *Netlib* suite...

Least Squares Regression:

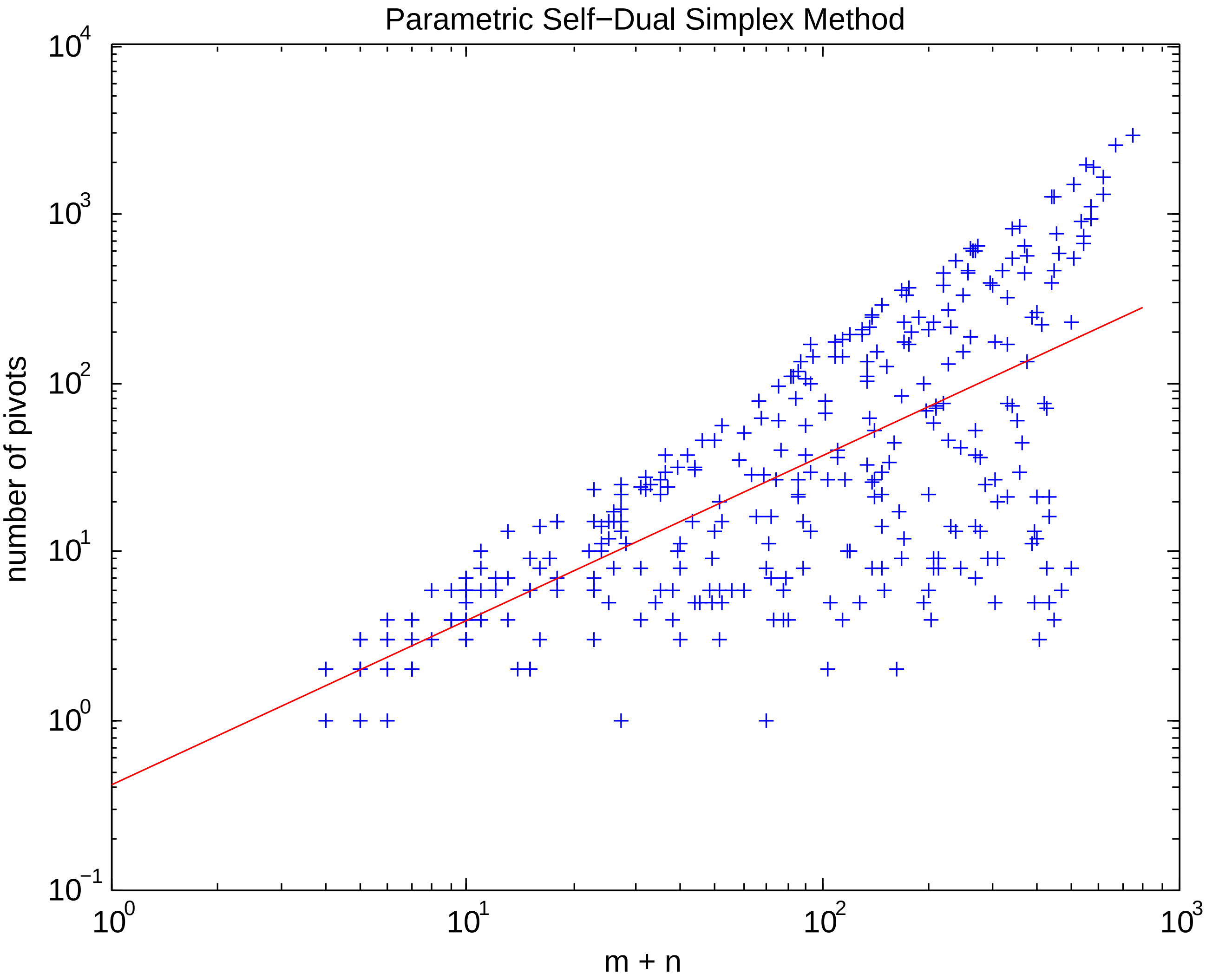
$$\begin{bmatrix} \bar{\alpha} \\ \bar{\beta} \end{bmatrix} = \begin{bmatrix} -1.03561 \\ 1.05152 \end{bmatrix} \qquad \Longrightarrow \qquad T \approx 0.488(m + n)^{1.052}$$

Least Absolute Deviation Regression:

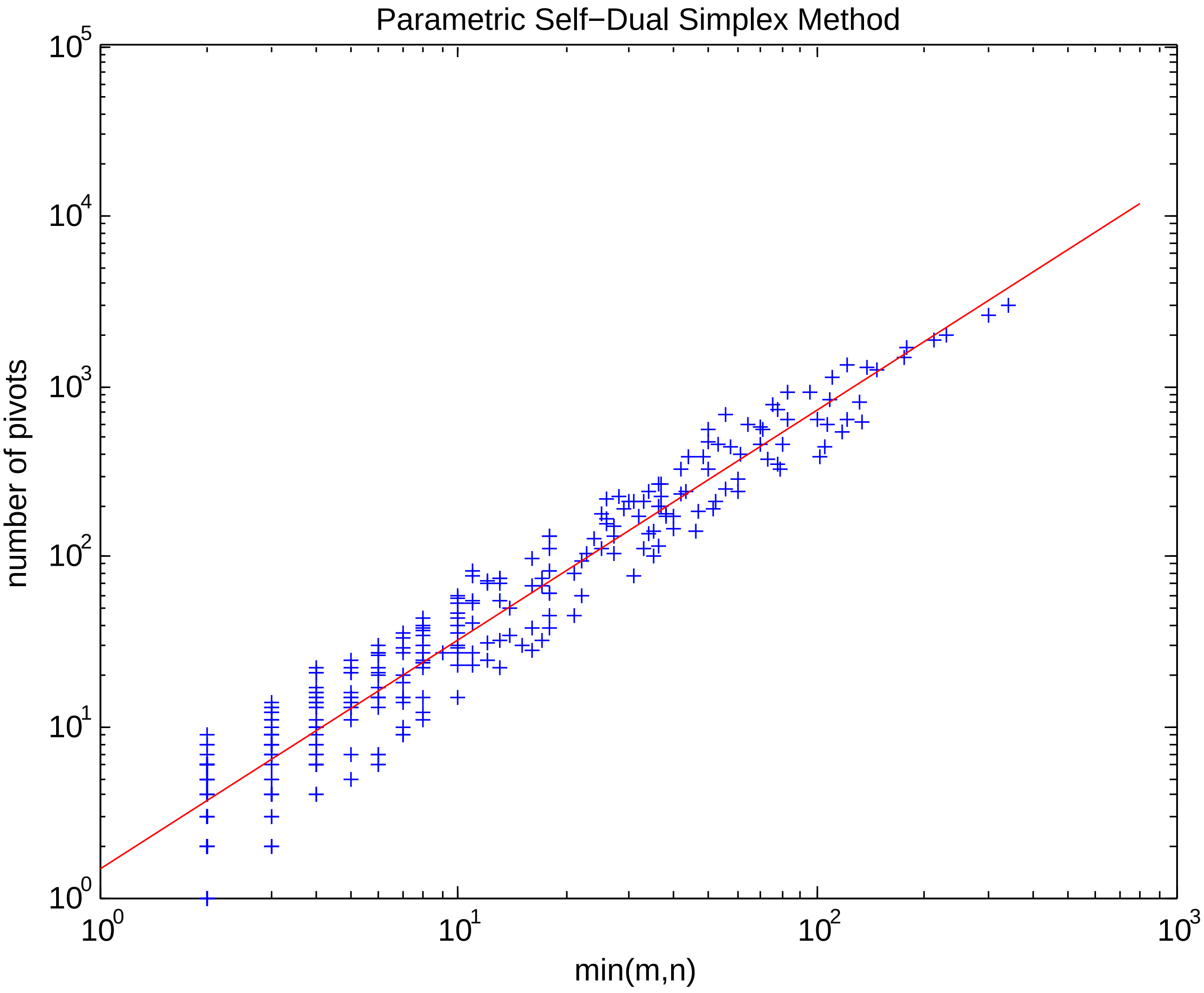
$$\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} -0.9508 \\ 1.0491 \end{bmatrix} \qquad \Longrightarrow \qquad T \approx 0.517(m + n)^{1.049}$$



A log–log plot of T vs. $m+n$ and the L^1 and L^2 regression lines.



$$\mathsf{iters} = 0.4165(\mathsf{m}+\mathsf{n})^{0.9759}$$



$$\textsf{iters} = 1.4880\,\min(\textsf{m},\textsf{n})^{1.3434}$$