The Parametric Self-Dual Simplex Method

A Modern Perspective

Robert J. Vanderbei

2019 November 5

INFORMS Student Chapter
Lehigh University

http://vanderbei.princeton.edu

Linear Optimization in Symmetric Form

Here's a *"primal"* problem in standard form:

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \leq b \\ & x \geq 0 \end{array}$$

This is it's *"dual"*:

$$\begin{array}{ll} \text{minimize} & b^T y \\ \text{subject to} & A^T y \geq c \\ & y \geq 0 \end{array}$$

Writing the dual in standard form, we see that it's the *negative transpose* of the primal problem:

$$\begin{array}{ll} -\text{maximize} & -b^T y \\ \text{subject to} & -A^T y \leq -c \\ & y \geq 0 \end{array}$$

Theorem 1: The dual of the dual is the primal.

Theorem 2: If x is feasible for the primal and y is feasible for the dual, then $c^T x \leq b^T y$.

Theorem 3: If x is optimal for the primal, then there exists a dual-feasible y such that $c^T x = b^T y$.

An Example

Primal Problem:

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

Dual Problem:

$$
\begin{array}{lrcrcrcrcr}
-\text{maximize} & -5y_1 & - & 4y_2 & - & 6y_3 & + & 4y_4 & & \\
\text{subj. to} & y_1 & - & 3y_2 & & & + & 3y_4 & \leq & 3 \\
 & -3y_1 & - & 3y_2 & - & 3y_3 & & & \leq & -11 \\
 & & & & - & 2y_3 & + & 5y_4 & \leq & -2 \\
 & & & & & y_1, \; y_2, & y_3, & y_4 & \geq & 0
\end{array}
$$

Written in *Dictionary Form*:

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

Written in *Dictionary Form*:

$$
\begin{array}{rcrcrcrcrcr}
-\xi & = & & & -5y_1 & - & 4y_2 & - & 6y_3 & + & 4\,y_4 \\
\hline
z_1 & = & 3 & - & y_1 & + & 3y_2 & & & - & 3y_4 \\
z_2 & = & -11 & + & 3y_1 & + & 3y_2 & + & 3y_3 & & \\
z_3 & = & -2 & & & & & + & 2y_3 & - & 5y_4
\end{array}
$$

Dictionary Solution:

$$x_1 = 0, \; x_2 = 0, \; x_3 = 0,$$
$$w_1 = 5, \; w_2 = 4, \; w_3 = 6, \; w_4 = -4$$

Dictionary Solution:

$$y_1 = 0, \; y_2 = 0, \; y_3 = 0, \; y_4 = 0,$$
$$z_1 = 3, \; z_2 = -11\,, \; z_3 = -2$$

An Example

Primal Problem:

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

Dual Problem:

$$\begin{array}{lrcrcrcrcr}
-\text{maximize} & -5y_1 & - & 4y_2 & - & 6y_3 & + & 4y_4 & & \\
\text{subj. to} & y_1 & - & 3y_2 & & & + & 3y_4 & \le & 3 \\
& -3y_1 & - & 3y_2 & - & 3y_3 & & & \le & -11 \\
& & & & - & 2y_3 & + & 5y_4 & \le & -2 \\
& & & & & y_1, & y_2, & y_3, y_4 & \ge & 0
\end{array}$$

Written in *Dictionary Form*:

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

Written in *Dictionary Form*:

$$\begin{array}{rcrcrcrcrcr}
-\xi & = & & & -5y_1 & - & 4y_2 & - & 6y_3 & + & 4y_4 \\
\hline
z_1 & = & 3 & - & y_1 & + & 3y_2 & & & - & 3y_4 \\
z_2 & = & -11 & + & 3y_1 & + & 3y_2 & + & 3y_3 & & \\
z_3 & = & -2 & & & & & + & 2y_3 & - & 5y_4
\end{array}$$

Dictionary Solution:

$$x_1 = 0,\ x_2 = 0,\ x_3 = 0,$$
$$w_1 = 5,\ w_2 = 4,\ w_3 = 6,\ w_4 = -4$$

Dictionary Solution:

$$y_1 = 0,\ y_2 = 0,\ y_3 = 0,\ y_4 = 0,$$
$$z_1 = 3,\ z_2 = -11,\ z_3 = -2$$

Note: Current "solution" is neither primal nor dual feasible.

Parametric Self-Dual Simplex Method

Introduce a parameter μ and perturb:

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

Here's how it looks in my online pivot tool:

$$
\begin{array}{lrcl}
\text{maximize} & \zeta & = & 0 + 0\,\mu + (-3)\,x_1 + 11\,x_2 + 2\,x_3 \\
 & & & +\ 0\,\mu + 0\,\mu^2 + (-1)\,\mu\, x_1 + (-1)\,\mu\, x_2 + (-1)\,\mu\, x_3 \\
\text{subject to:} & w_1 & = & 5 + 1\,\mu - (-1)\,x_1 - 3\,x_2 - 0\,x_3 \\
 & w_2 & = & 4 + 1\,\mu - 3\,x_1 - 3\,x_2 - 0\,x_3 \\
 & w_3 & = & 6 + 1\,\mu - 0\,x_1 - 3\,x_2 - 2\,x_3 \\
 & w_4 & = & -4 + 1\,\mu - (-3)\,x_1 - 0\,x_2 - (-5)\,x_3
\end{array}
$$

$$x_1,\ x_2,\ x_3,\ \ w_1,\ w_2,\ w_3,\ w_4 \ \geq\ 0$$

$$11 \leq \mu \leq \infty$$

For $\mu \geq 11$, dictionary is optimal. x_2 is the *entering variable* and w_2 is the *leaving variable*.

Before and After the First Pivot

$$
\begin{aligned}
\text{maximize} \quad \zeta &= 0 + 0\,\mu + (-3)\,x_1 + 11\,x_2 + 2\,x_3 \\
&\quad + 0\,\mu + 0\,\mu^2 + (-1)\,\mu\,x_1 + (-1)\,\mu\,x_2 + (-1)\,\mu\,x_3 \\
\text{subject to:} \quad w_1 &= 5 + 1\,\mu - (-1)\,x_1 - 3\,x_2 - 0\,x_3 \\
w_2 &= 4 + 1\,\mu - 3\,x_1 - 3\,x_2 - 0\,x_3 \\
w_3 &= 6 + 1\,\mu - 0\,x_1 - 3\,x_2 - 2\,x_3 \\
w_4 &= -4 + 1\,\mu - (-3)\,x_1 - 0\,x_2 - (-5)\,x_3
\end{aligned}
$$

$$x_1,\ x_2,\ x_3,\ \ w_1,\ w_2,\ w_3,\ w_4 \geq 0$$

$$11 \leq \mu \leq \infty$$

$$
\begin{aligned}
\text{maximize} \quad \zeta &= 44/3 + 11/3\,\mu + (-14)\,x_1 + (-11/3)\,w_2 + 2\,x_3 \\
&\quad + (-4/3)\,\mu + (-1/3)\,\mu^2 + 0\,\mu\,x_1 + 1/3\,\mu\,w_2 + (-1)\,\mu\,x_3 \\
\text{subject to:} \quad w_1 &= 1 + 0\,\mu - (-4)\,x_1 - (-1)\,w_2 - 0\,x_3 \\
x_2 &= 4/3 + 1/3\,\mu - 1\,x_1 - 1/3\,w_2 - 0\,x_3 \\
w_3 &= 2 + 0\,\mu - (-3)\,x_1 - (-1)\,w_2 - 2\,x_3 \\
w_4 &= -4 + 1\,\mu - (-3)\,x_1 - 0\,w_2 - (-5)\,x_3
\end{aligned}
$$

$$x_1,\ x_2,\ x_3,\ \ w_1,\ w_2,\ w_3,\ w_4 \geq 0$$

$$4 \leq \mu \leq 11$$

Before and After the Second Pivot

$$\begin{array}{lrcl} \text{maximize} & \zeta & = & \frac{44}{3} + \frac{11}{3}\mu - 14 x_1 - \frac{11}{3} w_2 + 2 x_3 \\ & & & + \left(-\frac{4}{3}\mu - \frac{1}{3}\mu^2\right) + 0\,\mu\, x_1 + \frac{1}{3}\mu\, w_2 - 1\,\mu\, x_3 \\ \text{subject to:} & w_1 & = & 1 + 0\,\mu - (-4) x_1 - (-1) w_2 - 0\, x_3 \\ & x_2 & = & \frac{4}{3} + \frac{1}{3}\mu - 1\, x_1 - \frac{1}{3} w_2 - 0\, x_3 \\ & w_3 & = & 2 + 0\,\mu - (-3) x_1 - (-1) w_2 - 2\, x_3 \\ & w_4 & = & -4 + 1\,\mu - (-3) x_1 - 0\, w_2 - (-5) x_3 \end{array}$$

$$x_1,\ x_2,\ x_3,\quad w_1,\ w_2,\ w_3,\ w_4 \geq 0$$

$$4 \leq \mu \leq 11$$

$$\begin{array}{lrcl} \text{maximize} & \zeta & = & \frac{244}{15} + \frac{49}{15}\mu + \left(-\frac{76}{5}\right) x_1 - \frac{11}{3} w_2 + \frac{2}{5} w_4 \\ & & & + \left(-\frac{32}{15}\mu - \frac{2}{15}\mu^2\right) + \frac{3}{5}\mu\, x_1 + \frac{1}{3}\mu\, w_2 - \frac{1}{5}\mu\, w_4 \\ \text{subject to:} & w_1 & = & 1 + 0\,\mu - (-4) x_1 - (-1) w_2 - 0\, w_4 \\ & x_2 & = & \frac{4}{3} + \frac{1}{3}\mu - 1\, x_1 - \frac{1}{3} w_2 - 0\, w_4 \\ & w_3 & = & \frac{2}{5} + \frac{2}{5}\mu - \left(-\frac{21}{5}\right) x_1 - (-1) w_2 - \frac{2}{5} w_4 \\ & x_3 & = & \frac{4}{5} + \left(-\frac{1}{5}\right)\mu - \frac{3}{5} x_1 - 0\, w_2 - \left(-\frac{1}{5}\right) w_4 \end{array}$$

$$x_1,\ x_2,\ x_3,\quad w_1,\ w_2,\ w_3,\ w_4 \geq 0$$

$$2 \leq \mu \leq 4$$

Before and After the Third Pivot

$$
\begin{array}{llrcrcrcrcr}
\text{maximize} & \zeta = & 244/15 & + & 49/15\,\mu & + & -76/5\,x_1 & + & -11/3\,w_2 & + & 2/5\,w_4 \\
& + & -32/15\,\mu & + & -2/15\,\mu^2 & + & 3/5\,\mu\,x_1 & + & 1/3\,\mu\,w_2 & + & -1/5\,\mu\,w_4 \\
\text{subject to:} & w_1 = & 1 & + & 0\,\mu & - & -4\,x_1 & - & -1\,w_2 & - & 0\,w_4 \\
& x_2 = & 4/3 & + & 1/3\,\mu & - & 1\,x_1 & - & 1/3\,w_2 & - & 0\,w_4 \\
& w_3 = & 2/5 & + & 2/5\,\mu & - & -21/5\,x_1 & - & -1\,w_2 & - & 2/5\,w_4 \\
& x_3 = & 4/5 & + & -1/5\,\mu & - & 3/5\,x_1 & - & 0\,w_2 & - & -1/5\,w_4
\end{array}
$$

$$x_1,\ x_2,\ x_3,\quad w_1,\ w_2,\ w_3,\ w_4 \ge 0$$

$$2 \le \mu \le 4$$

$$
\begin{array}{llrcrcrcrcr}
\text{maximize} & \zeta = & 50/3 & + & 11/3\,\mu & + & -11\,x_1 & + & -8/3\,w_2 & + & -1\,w_3 \\
& + & -7/3\,\mu & + & -1/3\,\mu^2 & + & -3/2\,\mu\,x_1 & + & -1/6\,\mu\,w_2 & + & 1/2\,\mu\,w_3 \\
\text{subject to:} & w_1 = & 1 & + & 0\,\mu & - & -4\,x_1 & - & -1\,w_2 & - & 0\,w_3 \\
& x_2 = & 4/3 & + & 1/3\,\mu & - & 1\,x_1 & - & 1/3\,w_2 & - & 0\,w_3 \\
& w_4 = & 1 & + & 1\,\mu & - & -21/2\,x_1 & - & -5/2\,w_2 & - & 5/2\,w_3 \\
& x_3 = & 1 & + & 0\,\mu & - & -3/2\,x_1 & - & -1/2\,w_2 & - & 1/2\,w_3
\end{array}
$$

$$x_1,\ x_2,\ x_3,\quad w_1,\ w_2,\ w_3,\ w_4 \ge 0$$

$$-1 \le \mu \le 2$$

We're done! It's *optimal*.

Top 10 Reasons to Like this Method

- Freedom to pick perturbation as you like.

Top 10 Reasons to Like this Method

- Freedom to pick perturbation as you like.
- Randomizing perturbation completely solves the degeneracy problem.

Top 10 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".

Top 10 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.

Top 10 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.
- The average-case performance can be analyzed.

Top 10 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.
- The average-case performance can be analyzed.
- In some real-world problems, a "natural" perturbation exists.

Top 10 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.
- The average-case performance can be analyzed.
- In some real-world problems, a "natural" perturbation exists.

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

Expected Number of Pivots

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 we expect the algorithm to take $(m + n)/2$ iterations.

Real-World Data (from Netlib)

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

Netlib 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:

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

Parametric Self-Dual Simplex Method

Recall the 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...

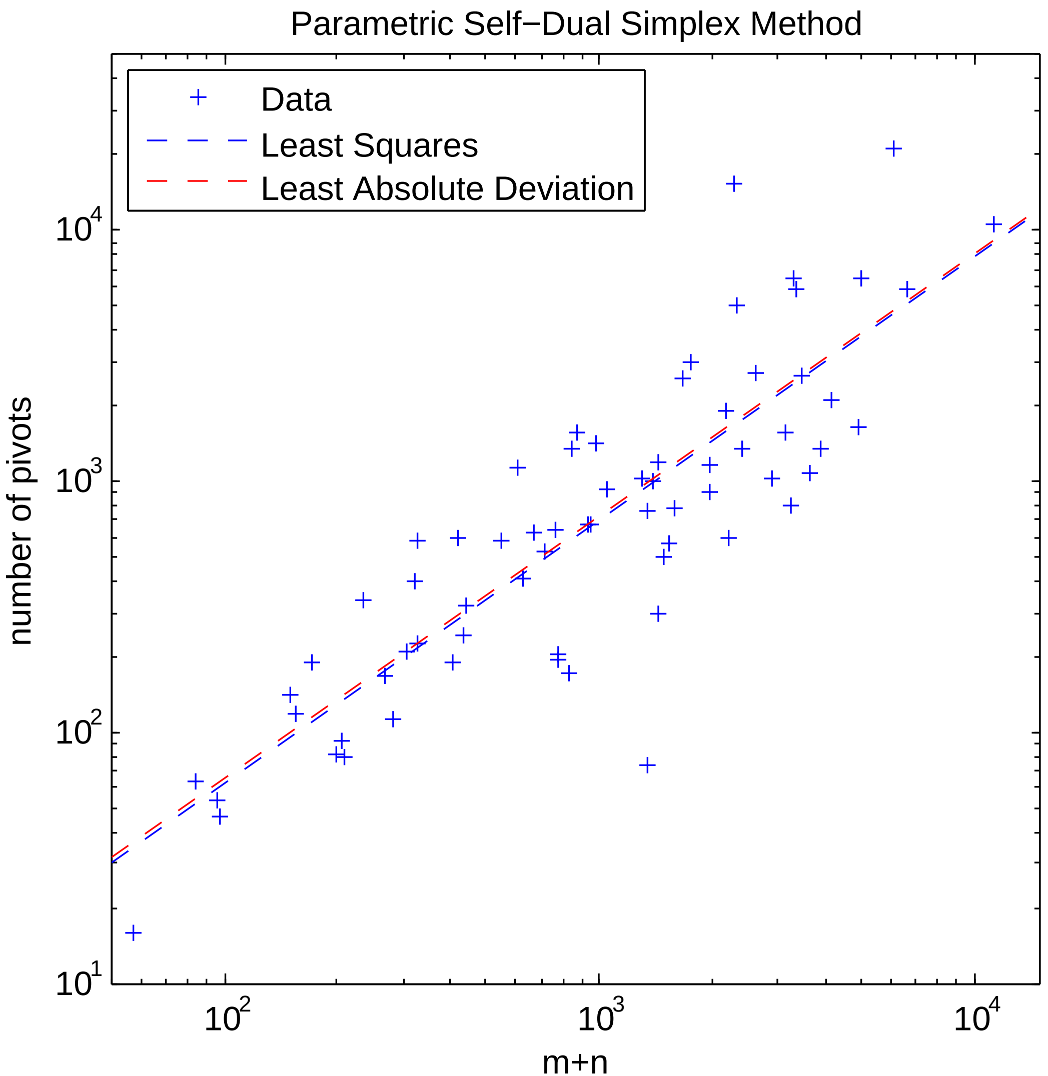
Least Squares Regression:

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

Least Absolute Deviation Regression:

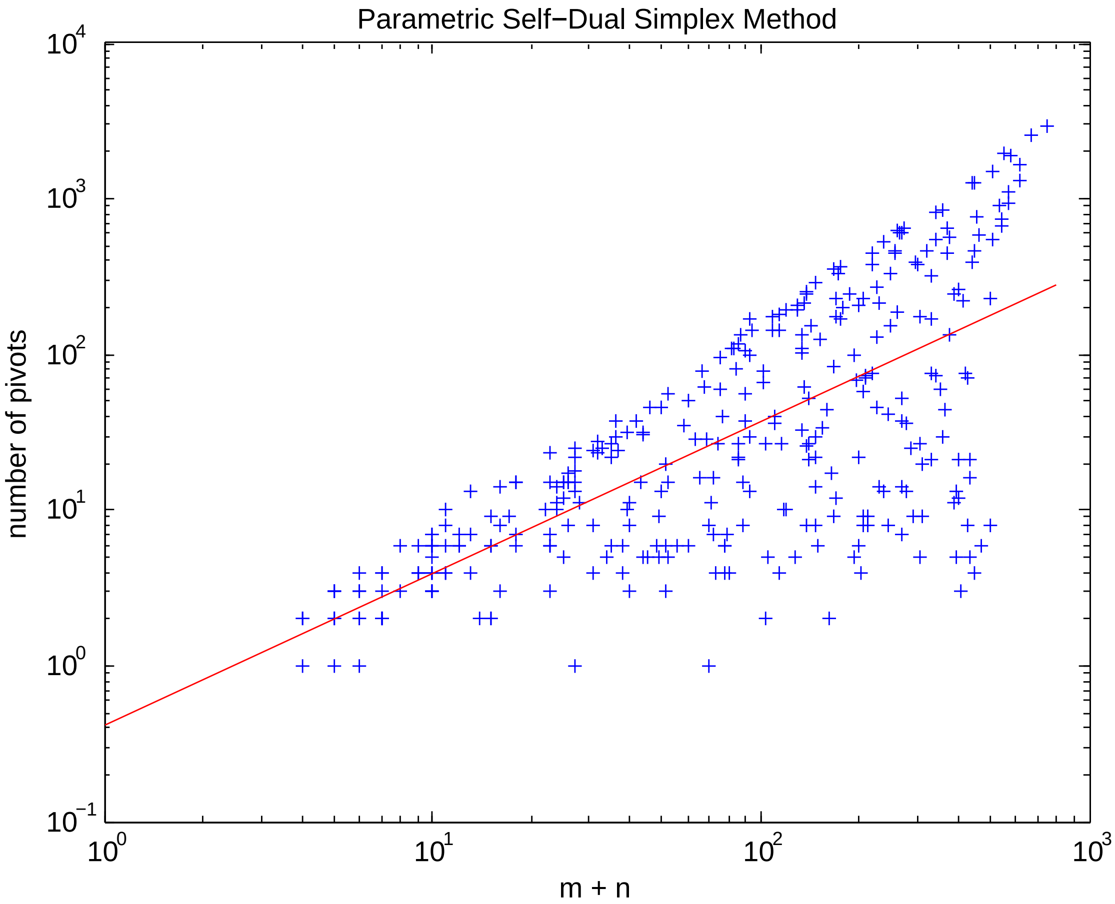
$$\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} -0.9508 \\ 1.0491 \end{bmatrix} \qquad \Longrightarrow \qquad t \approx 0.517(m+n)^{1.049}$$

Netlib Data



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

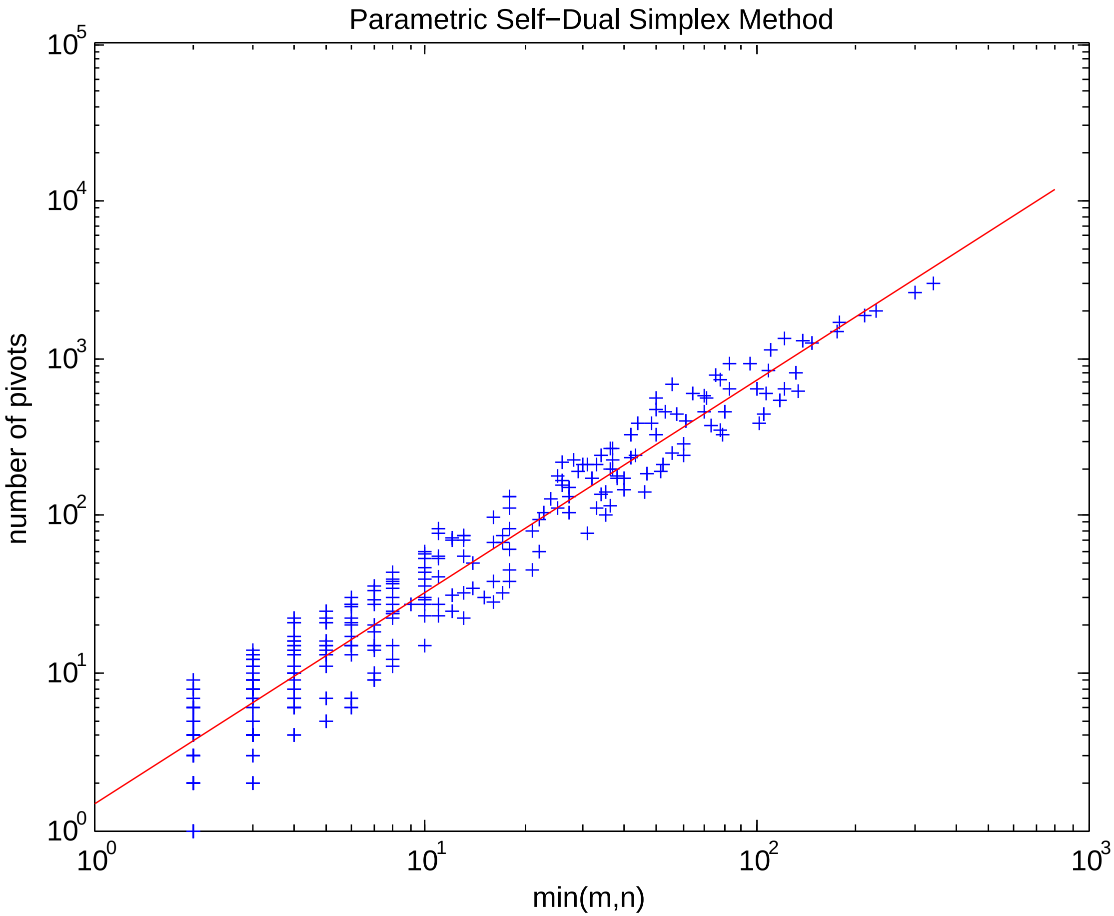
Randomly Generated Feasible/Bounded Problems



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

https://vanderbei.princeton.edu/307/python/psd_simplex_pivot.ipynb

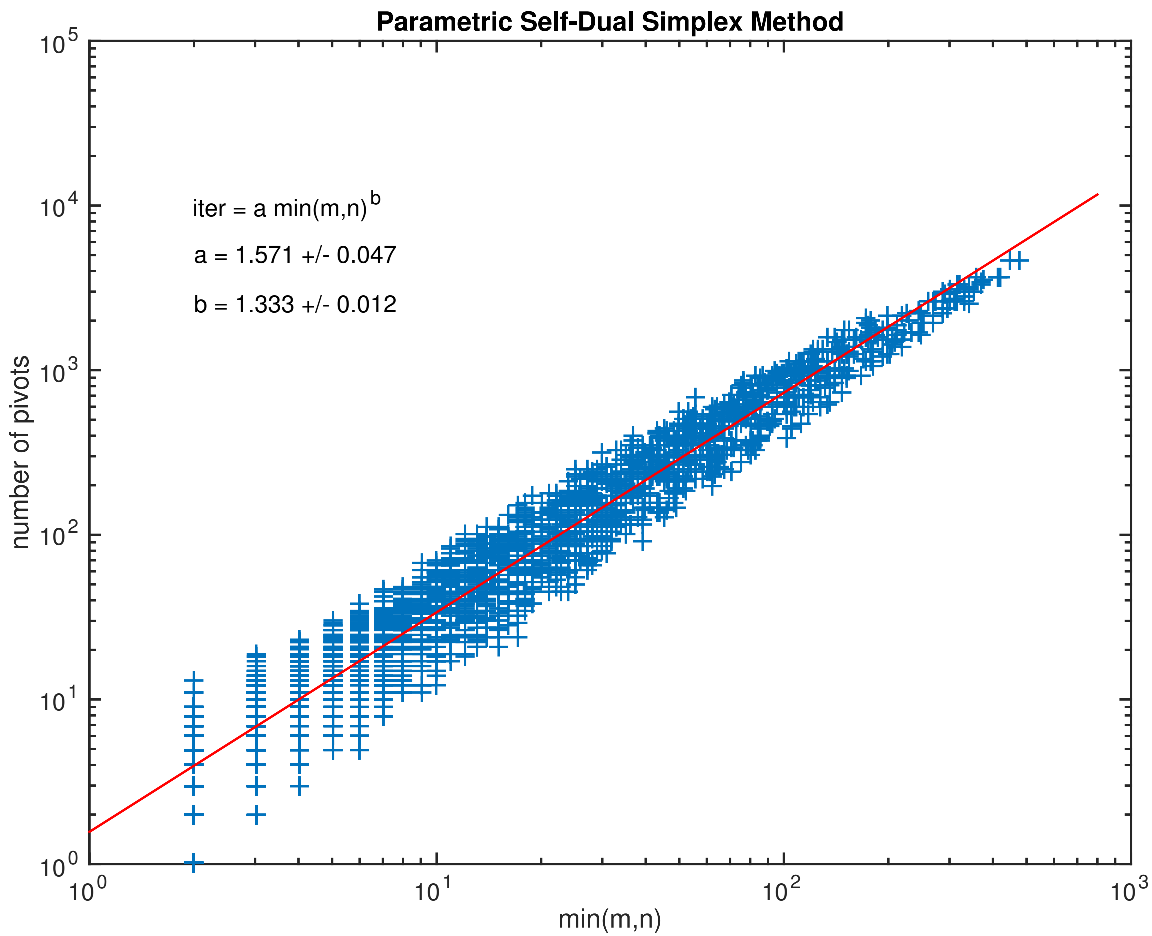
Randomly Generated Feasible/Bounded Problems



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

https://vanderbei.princeton.edu/307/python/psd_simplex_pivot.ipynb

Lots of Randomly Generated Problems



$$\text{iters} = 1.571 \min(\mathsf{m}, \mathsf{n})^{1.3333}$$

Portfolio Optimization

Historical Data:

$R_j(t)$ = return on asset j in time period t

$\Longrightarrow$

Derived Data:

$$r_j = \frac{1}{T}\sum_{t=1}^{T} R_j(t)$$

$$D_{tj} = R_j(t) - r_j.$$

Decision Variables:

x_j = fraction of portfolio to invest in asset j

Decision Criteria:

$$\text{reward}(x) = \sum_j r_j x_j$$

$$\text{risk}(x) = \frac{1}{T}\sum_{t=1}^{T}\left|\sum_j D_{tj} x_j\right|$$

Optimization Problem

Set a value for *risk affinity* parameter μ (risk affinity is the reciprocal of risk aversion)

and maximize a combination of *reward minus risk*:

$$\begin{array}{ll} \text{maximize} & \mu \sum_j r_j x_j - \frac{1}{T} \sum_{t=1}^{T} \left| \sum_j D_{tj} x_j \right| \\ \text{subject to} & \sum_j x_j = 1 \\ & x_j \geq 0 \qquad \text{for all investments } j \end{array}$$

Because of absolute values not a linear programming problem.

Easy to convert...

A Linear Programming Formulation

$$\begin{aligned}
\text{maximize} \quad & \mu \sum_j r_j x_j - \frac{1}{T} \sum_{t=1}^{T} y_t \\
\text{subject to} \quad & -y_t \le \sum_j D_{tj} x_j \le y_t && \text{for all times } t \\
& \sum_j x_j = 1 \\
& x_j \ge 0 && \text{for all investments } j \\
& y_t \ge 0 && \text{for all times } t
\end{aligned}$$

Note: The y_t's are the absolute values of the deviations from the average reward.
To be clear: they are *not the dual variables.*

Adding Slack Variables w_t^+ and w_t^-

$$
\begin{array}{lll}
\text{maximize} & \mu \sum_j r_j x_j - \frac{1}{T} \sum_{t=1}^{T} y_t & \\
\text{subject to} & -y_t - \sum_j D_{tj} x_j + w_t^- = 0 & \text{for all times } t \\
& -y_t + \sum_j D_{tj} x_j + w_t^+ = 0 & \text{for all times } t \\
& \sum_j x_j = 1 & \\
& x_j \geq 0 & \text{for all investments } j \\
& y_t,\ w_t^-,\ w_t^+ \geq 0 & \text{for all times } t
\end{array}
$$

A dictionary will have $2T+1$ equations and $3T+n$ variables, $2T+1$ of which will be basic and rest will be nonbasic (here, n denotes the number of assets).

The Solution for Large μ

Varying the risk bound $0 \leq \mu < \infty$ produces the *efficient frontier*.

Large values of μ favor reward maximization whereas small values favor minimizing risk.

Beyond some finite (but perhaps large) value for μ, the optimal solution will be a portfolio consisting of just one asset—the asset j^* with the largest average return:

$$r_{j^*} \geq r_j \qquad \text{for all } j.$$

For this case, it's easy to identify *basic* (nonzero) vs. *nonbasic* (i.e. zero) variables:

- Variable x_{j^*} is basic whereas the remaining x_j's are nonbasic.
- All of the y_t's are basic.
- If $D_{tj^*} > 0$, then w_t^- is basic and w_t^+ is nonbasic. Otherwise, it is switched.

The algebra is tedious, but we can now write down a starting dictionary...

The Optimal Dictionary for Large μ

Let

$$T^+ = \{t : D_{tj^*} > 0\}, \quad T^- = \{t : D_{tj^*} < 0\}, \quad \text{and} \quad \epsilon_t = \begin{cases} 1, & \text{for } t \in T^+ \\ -1, & \text{for } t \in T^- \end{cases}$$

Here's the optimal dictionary (for μ large):

$$\begin{array}{lclllll}
\zeta & = & \frac{1}{T}\sum_{t=1}^{T}\epsilon_t D_{tj^*} & -\frac{1}{T}\sum_{j\neq j^*}\sum_{t=1}^{T}\epsilon_t(D_{tj}-D_{tj^*})x_j & -\frac{1}{T}\sum_{t\in T^-} w_t^- & -\frac{1}{T}\sum_{t\in T^+} w_t^+ & \\
 & & +\mu r_{j^*} & +\mu\sum_{j\neq j^*}(r_j - r_{j^*})x_j & & & \\
\hline
y_t & = & -D_{tj^*} & -\sum_{j\neq j^*}(D_{tj}-D_{tj^*})x_j & +w_t^- & & t\in T^- \\
w_t^- & = & 2D_{tj^*} & +2\sum_{j\neq j^*}(D_{tj}-D_{tj^*})x_j & & +w_t^+ & t\in T^+ \\
y_t & = & D_{tj^*} & +\sum_{j\neq j^*}(D_{tj}-D_{tj^*})x_j & & +w_t^+ & t\in T^+ \\
w_t^+ & = & -2D_{tj^*} & -2\sum_{j\neq j^*}(D_{tj}-D_{tj^*})x_j & +w_t^- & & t\in T^- \\
x_{j^*} & = & 1 & -\sum_{j\neq j^*}x_j & & &
\end{array}$$

An Example

Collected data for 719 stocks (and bonds, etc.) from January 1, 1990, to March 18, 2002.

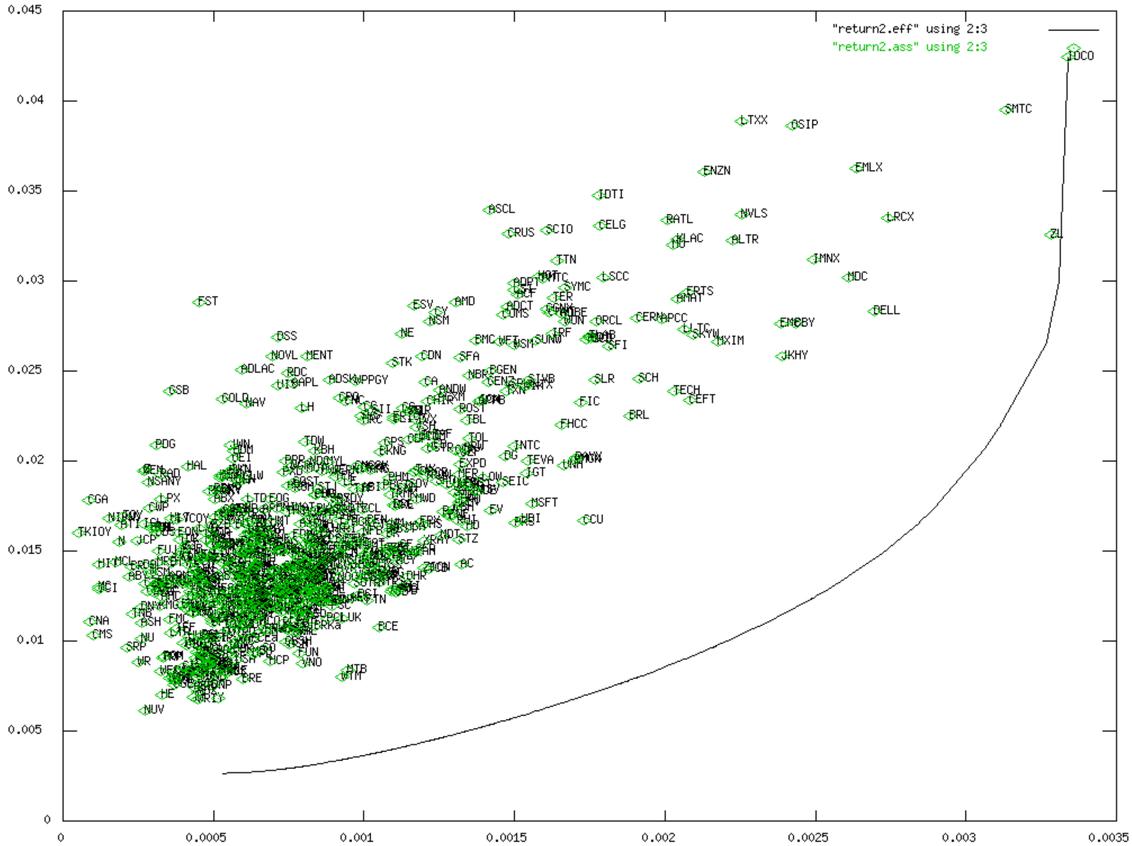
Hence,

$$n = 719$$

and

$$T = 3080.$$

Efficient Frontier



Click here for an expanded browser view.

Computing the Efficient Frontier

Using a reasonably efficient code for the parametric self-dual simplex method (simpo), it took *22,000* pivots to solve for *one point* on the efficient frontier.

Customizing this same code to solve parametrically for every point on the efficient frontier, it took *20,500* pivots to compute *every point* on the frontier.

The efficient frontier consists of *1308* distinct portfolios. Click here for a complete list.

Lasso Regression / Basis Pursuit Denoising

The problem is to solve a sparsity-encouraging "regularized" regression problem:

$$\text{minimize} \quad \|Ax - b\|_2^2 + \lambda \|x\|_1$$

My reaction:

Why not replace *least squares* (LS) with *least absolute deviations* (LAD)?

LAD is to LS as median is to mean. Median is a more robust statistic (i.e., insensitive to outliers).

The LAD version can be recast as a *linear programming* (LP) problem.

If the solution is expected to be sparse, then the *simplex method* can be expected to solve the problem very quickly.

No one knows the "correct" value of the parameter λ. The *parametric simplex method* can solve the problem for *all values of λ* from $\lambda = \infty$ to a small value of λ in the same (fast) time it takes the standard simplex method to solve the problem for one choice of λ.

Linear Programming Formulation

Here is the reformulated linear programming problem:

$$\begin{array}{llcll} \text{minimize} & \mu\, 1^T(x^+ + x^-) & + & 1^T(\varepsilon^+ + \varepsilon^-) & \\ \text{subject to} & A(x^+ - x^-) & + & (\varepsilon^+ - \varepsilon^-) & = y \\ & & & x^+, x^-, \varepsilon^+, \varepsilon^- & \geq 0. \end{array}$$

For μ large, the *optimal solution* has $x^+ = x^- = 0$, and $\varepsilon^+ - \varepsilon^- = y$.

And, given that these variables are required to be nonnegative, it follows that

$$\begin{array}{lcl} y_i > 0 & \Longrightarrow & \varepsilon_i^+ > 0 \text{ and } \varepsilon_i^- = 0 \\ y_i < 0 & \Longrightarrow & \varepsilon_i^+ = 0 \text{ and } \varepsilon_i^- > 0 \\ y_i = 0 & \Longrightarrow & \varepsilon_i^+ = 0 \text{ and } \varepsilon_i^- = 0 \end{array}$$

With these choices, the solution is feasible for all μ and is optimal for large μ.

Furthermore, declaring the nonzero variables to be *basic* variables and the zero variables to be *nonbasic*, we see that this optimal solution is also a basic solution and can therefore serve as a starting point for the *parametric simplex method*.

Numerical Results

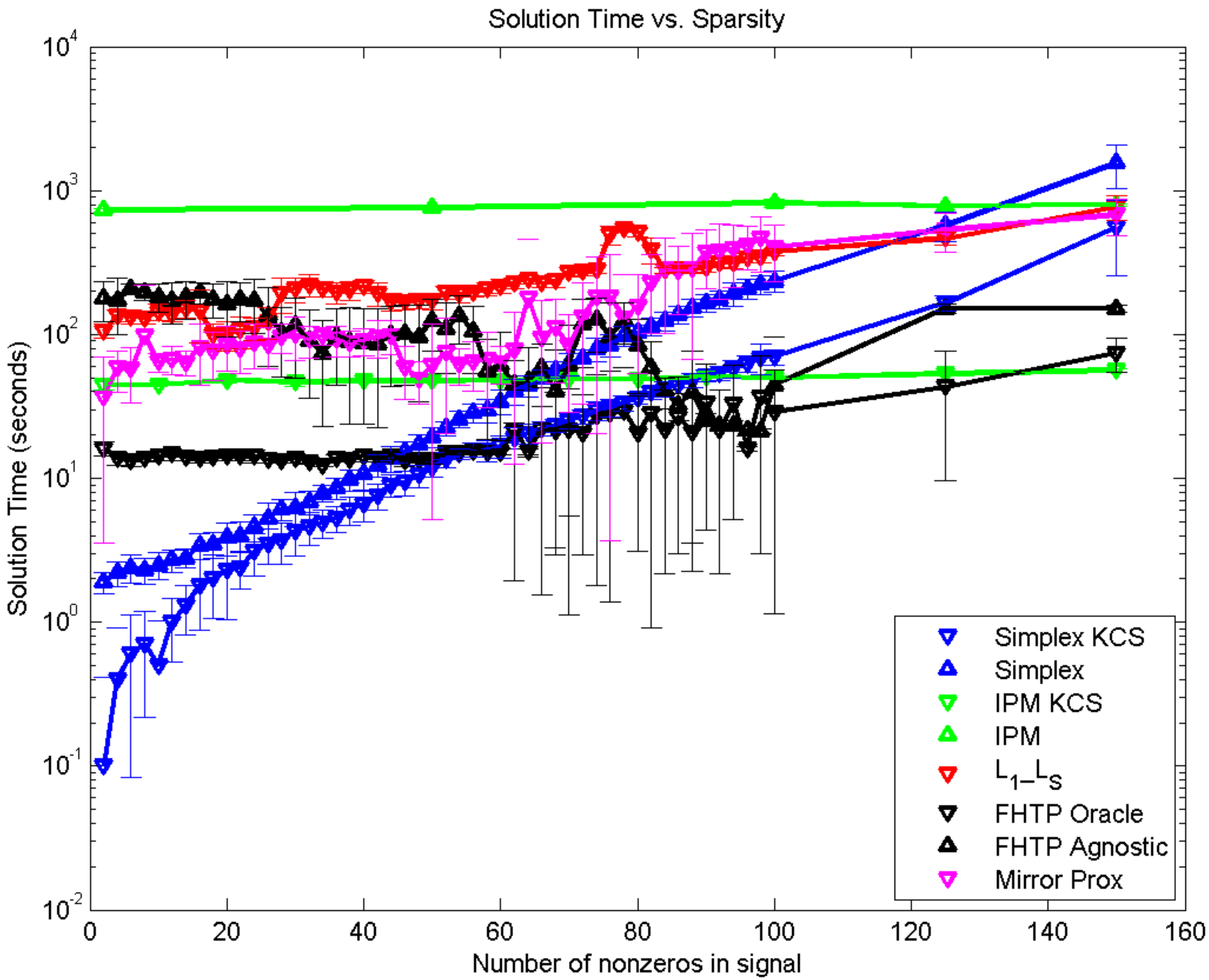
We generated random problems using $m = 1{,}122$ and $n = 20{,}022$.

We varied the number of nonzeros k in signal x^0 from 2 to 150.

We solved the straightforward linear programming formulations of these instances using an interior-point solver called LOQO.

We also solved a large number of instances of the parametrically formulated problem using the parametric simplex method as outlined above.

And we ran some publicly-available, state-of-the-art codes: $L_1\_L_s$, *FHTP*, and *Mirror Prox*.



$m = 1{,}222$, $n = 20{,}022$. Error bars represent one standard deviation.

References

[1] I. Adler and N. Megiddo. A simplex algorithm whose average number of steps is bounded between two quadratic functions of the smaller dimension. *Journal of the ACM*, 32:871–895, 1985.

[2] K.-H. Borgwardt. The average number of pivot steps required by the simplex-method is polynomial. *Zeitschrift für Operations Research*, 26:157–177, 1982.

[3] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.

[4] S.I. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2:39–45, 1955.

[5] C.E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Science*, 11:681–689, 1965.

[6] I.J. Lustig. The equivalence of Dantzig's self-dual parametric algorithm for linear programs to Lemke's algorithm. Technical Report SOL 87-4, Department of Operations Research, Stanford University, 1987.

[7] J.L. Nazareth. Homotopy techniques in linear programming. *Algorithmica*, 1:529–535, 1986.

[8] B. Rudloff, F. Ulus, and R.J. Vanderbei. A parametric simplex algorithm for linear vector optimization problems. *Mathematical Programming, Series A*, 163(1):213–242, 2017.

[9] A. Ruszczyński and R.J. Vanderbei. Frontiers of Stochastically Nondominated Portfolios. *Econometrica*, 71(4):1287–1297, 2003.

[10] S. Smale. On the average number of steps of the simplex method of linear programming. *Mathematical Programming*, 27:241–262, 1983.

[11] R.J. Vanderbei. *Linear Programming: Foundations and Extensions*. Springer, 4th edition, 2013.

[12] R.J. Vanderbei, Kevin Lin, Han Liu, and Lie Wang. Revisiting Compressed Sensing: Exploiting the Efficiency of Simplex and Sparsification Methods. *Math. Prog. C*, 8:253–269, 2016.

Thank You!

Questions?

Let's Party!