New Directions in Linear Programming

Robert Vanderbei

November 5, 2001

INFORMS Miami Beach

NOTE: This is a talk mostly on pedagogy. There will be some new results. It is not a talk on state-of-the-art implementations of algorithms for LP.

Operations Research and Financial Engineering, Princeton University

http://www.princeton.edu/∼rvdb

1 Outline

- Themes: Symmetry & Parametric Methods & Segues
- Strong Duality Theorem: Negative Transpose Property
- Simplex Methods: Primal, Dual, Two-Phase
- Efficiency: A Parametric Klee–Minty Problem
- The Parametric Self-Dual Simplex Method: a Path-Following Method
- Average Performance
- Two Algorithms for L^1 Regression: Reweighted Least Squares
- Symmetry in Network Flow Problems: Geometric Dual $\neq$ Algebraic Dual
- An Interior-Point Method (IPM): Path-Following
- Another IPM: Affine-Scaling=Reweighted Least Squares

2 LPs in Inequality Form

A Primal–Dual Symmetric Setting

Primal Problem:

$$\begin{array}{ll} \text{max} & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \end{array}$$

Dual Problem:

$$\begin{array}{ll} \text{min} & \boldsymbol{b}^T \boldsymbol{y} \\ \text{s.t.} & A^T\boldsymbol{y} \ge \boldsymbol{c} \\ & \boldsymbol{y} \ge \boldsymbol{0} \end{array}$$

$$\Updownarrow$$

Dual in "Canonical"

Form:

$$\begin{array}{ll} -\text{max} & -\boldsymbol{b}^T \boldsymbol{y} \\ \text{s.t.} & -A^T\boldsymbol{y} \le -\boldsymbol{c} \\ & \boldsymbol{y} \ge \boldsymbol{0} \end{array}$$

3 Dictionary Notation & Basic Solutions

Primal–Dual Symmetry

$$\begin{aligned} \zeta &= 0 + c^T x \\ w &= b - Ax \end{aligned} \quad \overset{\text{after pivoting}}{\Longrightarrow} \quad \begin{aligned} \zeta &= \bar{\zeta} + \bar{c}^T x_{\mathcal{N}} \\ x_{\mathcal{B}} &= \bar{b} - B^{-1} N x_{\mathcal{N}} \end{aligned}$$

$\Downarrow$ dual — Note: neg transp $\Downarrow$

$\Downarrow$ dual — Theorem: dual=neg transp $\Downarrow$

$$\begin{aligned} -\xi &= 0 \quad - b^T y \\ z &= -c + A^T y \end{aligned} \quad \overset{\text{after pivoting}}{\Longrightarrow} \quad \begin{aligned} -\xi &= -\bar{\zeta} + \bar{b}^T z_{\mathcal{B}} \\ z_{\mathcal{N}} &= -\bar{c} + (B^{-1} N)^T z_{\mathcal{B}} \end{aligned}$$

Corollary (Strong Duality Theorem): When primal is optimal, so is dual and objective function values agree, i.e., no duality gap.

4 Proof of Neg. Transp. Property

b	a	
d	c	

$\xrightarrow{\text{pivot}}$

$-\frac{b}{a}$	$\frac{1}{a}$	
$d - \frac{bc}{a}$	$\frac{c}{a}$	

Now, if we start with a dual dictionary that is the negative transpose of the primal and apply one pivot operation, we get

	$-b$	$-d$
	$-a$	$-c$

$\xrightarrow{\text{pivot}}$

	$\frac{b}{a}$	$-d + \frac{bc}{a}$
	$-\frac{1}{a}$	$-\frac{c}{a}$

.

5 Simplex Method and Duality

A Primal Problem:

obj	=	0	+	−3	x1	+	2	x2	+	1	x3
w1	=	0	−	0	x1	−	−1	x2	−	2	x3
w2	=	3	−	−3	x1	−	4	x2	−	1	x3

Its Dual:

obj	=	0	+	0	y1	+	−3	y2
z1	=	3	−	0	y1	−	3	y2
z2	=	−2	−	1	y1	−	−4	y2
z3	=	−1	−	−2	y1	−	−1	y2

Notes:

- Dual is negative transpose of primal.
- Primal is feasible, dual is not.

Use primal to choose pivot: x_2 enters, w_2 leaves.

Make analogous pivot in dual: z_2 leaves, y_2 enters.

6 Second Iteration

After First Pivot:

Primal (still feasible):

obj	=	3/2	+	−3/2	x1	+	−1/2	w2	+	1/2	x3
w1	=	3/4	−	−3/4	x1	−	1/4	w2	−	9/4	x3
x2	=	3/4	−	−3/4	x1	−	1/4	w2	−	1/4	x3

Dual (still not feasible):

obj	=	−3/2	+	−3/4	y1	+	−3/4	z2
z1	=	3/2	−	3/4	y1	−	3/4	z2
y2	=	1/2	−	−1/4	y1	−	−1/4	z2
z3	=	−1/2	−	−9/4	y1	−	−1/4	z2

Note: *negative transpose property intact.*

Again, use primal to pick pivot: x_3 enters, w_1 leaves.

Make analogous pivot in dual: z_3 leaves, y_1 enters.

7 After Second Iteration

Primal:

obj	=	5/3	+	-4/3	x1	+	-5/9	w2	+	-2/9	w1
x3	=	1/3	-	-1/3	x1	-	1/9	w2	-	4/9	w1
x2	=	2/3	-	-2/3	x1	-	2/9	w2	-	-1/9	w1

- Is optimal.

Dual:

obj	=	-5/3	+	-1/3	z3	+	-2/3	z2
z1	=	4/3	-	1/3	z3	-	2/3	z2
y2	=	5/9	-	-1/9	z3	-	-2/9	z2
y1	=	2/9	-	-4/9	z3	-	1/9	z2

- Negative transpose property remains intact.
- Is optimal.

Conclusion

Simplex method applied to primal problem (two phases, if necessary), solves both the primal and the dual.

Negative-transpose property implies zero duality gap.

8 Dual Simplex Method

When: dual feasible, primal infeasible (i.e., pinks on the left, not on top).

An Example. Showing both primal and dual dictionaries:

obj	=	0.0	+	-2.0	x1	+	-4.0	x2	+	0.0	x3	+	-6.0 x4
w1	=	-3.0	-	-1.0	x1	-	2.0	x2	-	0.0	x3	-	-1.0 x4
w2	=	-5.0	-	2.0	x1	-	-3.0	x2	-	0.0	x3	-	-2.0 x4
w3	=	8.0	-	2.0	x1	-	3.0	x2	-	3.0	x3	-	2.0 x4

obj	=	0.0	+	3.0	y1	+	5.0	y2	+	-8.0 y3
z1	=	2.0	-	1.0	y1	-	-2.0	y2	-	-2.0 y3
z2	=	4.0	-	-2.0	y1	-	3.0	y2	-	-3.0 y3
z3	=	0.0	-	0.0	y1	-	0.0	y2	-	-3.0 y3
z4	=	6.0	-	1.0	y1	-	2.0	y2	-	-2.0 y3

Looking at dual dictionary: y_2 enters, z_2 leaves.

On the primal dictionary: w_2 leaves, x_2 enters.

After pivot...

9 Dual Simplex Method: Second Pivot

Going in, we have:

obj	=	-6.6667	+	-4.6667	x1	+	-1.3333	w2	+	0.0	x3	+	-3.3333	x4		
w1	=	-6.3333	-	0.3333	x1	-	0.6667	w2	-	0.0	x3	-	-2.3333	x4		
x2	=	1.6667	-	-0.6667	x1	-	-0.3333	w2	-	0.0	x3	-	0.6667	x4		
w3	=	3.0	-	4.0	x1	-	1.0	w2	-	3.0	x3	-	0.0	x4		

obj	=	6.6667	+	6.3333	y1	+	-1.6667	z2	+	-3.0	y3	
z1	=	4.6667	-	-0.3333	y1	-	0.6667	z2	-	-4.0	y3	
y2	=	1.3333	-	-0.6667	y1	-	0.3333	z2	-	-1.0	y3	
z3	=	0.0	-	0.0	y1	-	0.0	z2	-	-3.0	y3	
z4	=	3.3333	-	2.3333	y1	-	-0.6667	z2	-	0.0	y3	

Looking at dual: y_1 enters, z_4 leaves.

Looking at primal: w_1 leaves, x_4 enters.

10 Dual Simplex Method Pivot Rule

obj	=	-6.6667	+	-4.6667	x1	+	-1.3333	w2	+	0.0	x3	+	-3.3333	x4
w1	=	-6.3333	−	0.3333	x1	−	0.6667	w2	−	0.0	x3	−	-2.3333	x4
x2	=	1.6667	−	-0.6667	x1	−	-0.3333	w2	−	0.0	x3	−	0.6667	x4
w3	=	3.0	−	4.0	x1	−	1.0	w2	−	3.0	x3	−	0.0	x4

Refering to the primal dictionary:

- Pick leaving variable from those rows that are infeasible.
- Pick entering variable from a box with a negative value and which can be increased the least (on the dual side).

Next primal dictionary shown on next page...

11 Dual Simplex Method: Third Pivot

Going in, we have:

obj	=	-15.7143	+	-5.1429	x1	+	-2.2857	w2	+	0.0	x3	+	-1.4286	w1
x4	=	2.7143	−	-0.1429	x1	−	-0.2857	w2	−	0.0	x3	−	-0.4286	w1
x2	=	-0.1429	−	-0.5714	x1	−	-0.1429	w2	−	0.0	x3	−	0.2857	w1
w3	=	3.0	−	4.0	x1	−	1.0	w2	−	3.0	x3	−	0.0	w1

x_2 leaves, x_1 enters.

Resulting dictionary is OPTIMAL:

obj	=	-17.0	+	-9.0	x2	+	-1.0	w2	+	0.0	x3	+	-4.0	w1
x4	=	2.75	−	-0.25	x2	−	-0.25	w2	−	0.0	x3	−	-0.5	w1
x1	=	0.25	−	-1.75	x2	−	0.25	w2	−	0.0	x3	−	-0.5	w1
w3	=	2.0	−	7.0	x2	−	0.0	w2	−	3.0	x3	−	2.0	w1

A Two-Phase Method: Dual-Based Phase I

An Example:

obj	=	0.0			+	-4.0	x1	+	2.0	x2	+	3.0	x3
					+	-1.0	x1	+	-1.0	x2	+	-1.0	x3
w1	=	0.0	+	1.0	−	2.0	x1	−	-1.0	x2	−	3.0	x3
w2	=	0.0	+	1.0	−	3.0	x1	−	-3.0	x2	−	-4.0	x3
w3	=	-3.0	+	1.0	−	-1.0	x1	−	-1.0	x2	−	1.0	x3
w4	=	-1.0	+	1.0	−	-2.0	x1	−	0.0	x2	−	0.0	x3

Notes:

- Two objective functions: the true objective (on top), and a fake one (below it).
- For Phase I, use the fake objective—it's dual feasible.
- Two right-hand sides: the real one (on the left) and a fake (on the right).
- Ignore the fake right-hand side—we'll use it in another algorithm later.

Phase I—First Pivot: w_3 leaves, x_1 enters.

After first pivot...

Dual-Based Phase I Method—Second Pivot

Current dictionary:

obj	=	-12.0			+	-4.0	w3	+	6.0	x2	+	-1.0	x3
					+	-1.0	w3	+	0.0	x2	+	-2.0	x3
w1	=	-6.0	+	3.0	−	2.0	w3	−	-3.0	x2	−	5.0	x3
w2	=	-9.0	+	4.0	−	3.0	w3	−	-6.0	x2	−	-1.0	x3
x1	=	3.0	+	-1.0	−	-1.0	w3	−	1.0	x2	−	-1.0	x3
w4	=	5.0	+	-1.0	−	-2.0	w3	−	2.0	x2	−	-2.0	x3

Dual pivot: w_2 leaves, x_2 enters.

After pivot:

obj	=	-3.0			+	-1.0	w3	+	1.0	w2	+	-2.0	x3
					+	-1.0	w3	+	0.0	w2	+	-2.0	x3
w1	=	-1.5	+	1.0	−	0.5	w3	−	-0.5	w2	−	5.5	x3
x2	=	1.5	+	-0.6667	−	-0.5	w3	−	-0.1667	w2	−	0.1667	x3
x1	=	1.5	+	-0.3333	−	-0.5	w3	−	0.1667	w2	−	-1.1667	x3
w4	=	2.0	+	0.3333	−	-1.0	w3	−	0.3333	w2	−	-2.3333	x3

Dual-Based Phase I Method—Third Pivot

Current dictionary:

obj	=	-3.0			+	-1.0	w3	+	1.0	w2	+	-2.0	x3
					+	-1.0	w3	+	0.0	w2	+	-2.0	x3
w1	=	-1.5	+	1.0	−	0.5	w3	−	-0.5	w2	−	5.5	x3
x2	=	1.5	+	-0.6667	−	-0.5	w3	−	-0.1667	w2	−	0.1667	x3
x1	=	1.5	+	-0.3333	−	-0.5	w3	−	0.1667	w2	−	-1.1667	x3
w4	=	2.0	+	0.3333	−	-1.0	w3	−	0.3333	w2	−	-2.3333	x3

Dual pivot:

w_1 leaves,

w_2 enters.

After pivot:

obj	=	0.0			+	0.0	w3	+	2.0	w1	+	9.0	x3
					+	-1.0	w3	+	0.0	w1	+	-2.0	x3
w2	=	3.0	+	-2.0	−	-1.0	w3	−	-2.0	w1	−	-11.0	x3
x2	=	2.0	+	-1.0	−	-0.6667	w3	−	-0.3333	w1	−	-1.6667	x3
x1	=	1.0	+	0.0	−	-0.3333	w3	−	0.3333	w1	−	0.6667	x3
w4	=	1.0	+	1.0	−	-0.6667	w3	−	0.6667	w1	−	1.3333	x3

It's **feasible!**

15 Fourth Pivot—Phase II

Current dictionary:

obj	=	0.0			+	0.0	w3	+	2.0	w1	+	9.0	x3
					+	-1.0	w3	+	0.0	w1	+	-2.0	x3
w2	=	3.0	+	-2.0	−	-1.0	w3	−	-2.0	w1	−	-11.0	x3
x2	=	2.0	+	-1.0	−	-0.6667	w3	−	-0.3333	w1	−	-1.6667	x3
x1	=	1.0	+	0.0	−	-0.3333	w3	−	0.3333	w1	−	0.6667	x3
w4	=	1.0	+	1.0	−	-0.6667	w3	−	0.6667	w1	−	1.3333	x3

It’s feasible.

Ignore fake objective—Use the real thing (top row).

Primal pivot: x_3 enters, w_4 leaves.

After pivot:

Unbounded!

obj	=	6.75			+	4.5	w3	+	-2.5	w1	+	-6.75	w4
					+	-2.0	w3	+	1.0	w1	+	1.5	w4
w2	=	11.25	+	6.25	−	-6.5	w3	−	3.5	w1	−	8.25	w4
x2	=	3.25	+	0.25	−	-1.5	w3	−	0.5	w1	−	1.25	w4
x1	=	0.5	+	-0.5	−	0.0	w3	−	0.0	w1	−	-0.5	w4
x3	=	0.75	+	0.75	−	-0.5	w3	−	0.5	w1	−	0.75	w4

Note: could use a primal phase-I (with the fake right-hand side) and then a dual method phase-II.

16 Efficiency: The Klee–Minty Problem

via Parametric Methods

$$
\begin{array}{llr}
\text{maximize} & \displaystyle\sum_{j=1}^{n} 2^{n-j} x_j & \\
\text{subject to} & \displaystyle 2 \sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \leq 100^{i-1} & i = 1, 2, \ldots, n \\
& x_j \geq 0 & j = 1, 2, \ldots, n.
\end{array}
$$

Example $n = 3$:

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 4x_1 & + & 2x_2 & + & x_3 & & \\
\text{subj. to} & x_1 & & & & & \leq & 1 \\
& 4x_1 & + & x_2 & & & \leq & 100 \\
& 8x_1 & + & 4x_2 & + & x_3 & \leq & 10000 \\
& & & x_1, & x_2, & x_3 & \geq & 0.
\end{array}
$$

17 Efficiency: A Parametric Klee–Minty Problem

Replace

$$1, 100, 10000, \ldots,$$

with

$$1 = b_1 \ll b_2 \ll b_3 \ll \ldots .$$

Then, make following replacements to rhs:

$$\begin{array}{rcl} b_1 & \longrightarrow & b_1 \\ b_2 & \longrightarrow & 2b_1 + b_2 \\ b_3 & \longrightarrow & 4b_1 + 2b_2 + b_3 \\ b_4 & \longrightarrow & 8b_1 + 4b_2 + 2b_3 + b_4 \\ & \vdots & \end{array}$$

Hardly a change!

Make a similar constant adjustment to objective function.

Parametric Klee–Minty Problem: Case $n = 3$

obj	=	−2	b1	+	−1	b2	+	0	b3	+	4	x1	+	2	x2	+	1	x3
w1	=	1	b1	+	0	b2	+	0	b3	−	1	x1	−	0	x2	−	0	x3
w2	=	2	b1	+	1	b2	+	0	b3	−	4	x1	−	1	x2	−	0	x3
w3	=	4	b1	+	2	b2	+	1	b3	−	8	x1	−	4	x2	−	1	x3

Now watch the pivots...

obj	=	2	b1	+	−1	b2	+	0	b3	+	−4	w1	+	2	x2	+	1	x3
x1	=	1	b1	+	0	b2	+	0	b3	−	1	w1	−	0	x2	−	0	x3
w2	=	−2	b1	+	1	b2	+	0	b3	−	−4	w1	−	1	x2	−	0	x3
w3	=	−4	b1	+	2	b2	+	1	b3	−	−8	w1	−	4	x2	−	1	x3

obj	=	−2	b1	+	1	b2	+	0	b3	+	4	w1	+	−2	w2	+	1	x3
x1	=	1	b1	+	0	b2	+	0	b3	−	1	w1	−	0	w2	−	0	x3
x2	=	−2	b1	+	1	b2	+	0	b3	−	−4	w1	−	1	w2	−	0	x3
w3	=	4	b1	+	−2	b2	+	1	b3	−	8	w1	−	−4	w2	−	1	x3

obj	=	2	b1	+	1	b2	+	0	b3	+	−4	x1	+	−2	w2	+	1	x3
w1	=	1	b1	+	0	b2	+	0	b3	−	1	x1	−	0	w2	−	0	x3
x2	=	2	b1	+	1	b2	+	0	b3	−	4	x1	−	1	w2	−	0	x3
w3	=	−4	b1	+	−2	b2	+	1	b3	−	−8	x1	−	−4	w2	−	1	x3

obj	=	−2	b1	+	−1	b2	+	0	b3	+	4	x1	+	2	w2	+	−1	w3
w1	=	1	b1	+	0	b2	+	0	b3	−	1	x1	−	0	w2	−	0	w3
x2	=	2	b1	+	1	b2	+	0	b3	−	4	x1	−	1	w2	−	0	w3
x3	=	−4	b1	+	−2	b2	+	1	b3	−	−8	x1	−	−4	w2	−	1	w3

obj	=	2	b1	+	−1	b2	+	0	b3	+	−4	w1	+	2	w2	+	−1	w3
x1	=	1	b1	+	0	b2	+	0	b3	−	1	w1	−	0	w2	−	0	w3
x2	=	−2	b1	+	1	b2	+	0	b3	−	−4	w1	−	1	w2	−	0	w3
x3	=	4	b1	+	−2	b2	+	1	b3	−	8	w1	−	−4	w2	−	1	w3

obj	=	-2	b1	+	1	b2	+	0	b3	+	4	w1	+	-2	x2	+	-1	w3
x1	=	1	b1	+	0	b2	+	0	b3	−	1	w1	−	0	x2	−	0	w3
w2	=	-2	b1	+	1	b2	+	0	b3	−	-4	w1	−	1	x2	−	0	w3
x3	=	-4	b1	+	2	b2	+	1	b3	−	-8	w1	−	4	x2	−	1	w3

obj	=	2	b1	+	1	b2	+	0	b3	+	-4	x1	+	-2	x2	+	-1	w3
w1	=	1	b1	+	0	b2	+	0	b3	−	1	x1	−	0	x2	−	0	w3
w2	=	2	b1	+	1	b2	+	0	b3	−	4	x1	−	1	x2	−	0	w3
x3	=	4	b1	+	2	b2	+	1	b3	−	8	x1	−	4	x2	−	1	w3

Observe: only sign changes–no other numerical changes.

Put pink=0, gray=1. Pivots count from $\mathbf{0}$ to $\mathbf{2^{n-1}}$ in base 2.

Degeneracy

The lexicographic method for resolving degeneracy can be thought of as method with several parametric right-hand sides:

$$\epsilon_1 \gg \epsilon_2 \gg \epsilon_3 \gg \ldots .$$

No time to consider this further.

19 The Parametric Self-Dual Simplex Method

A Symmetric Method. A Parametric Method. A Path-Following Method too.

In general, one expects both c and b to contain negative elements.

Consider a perturbation by a real parameter μ:

$$\begin{aligned} \zeta &= \bar{\zeta} && + (c + \mu e)^T x_{\mathcal{N}} \\ x_{\mathcal{B}} &= (b + \mu e) && - B^{-1} N x_{\mathcal{N}} \end{aligned}$$

For μ large, this dictionary is optimal.

Decrease μ to the brink of nonoptimality.

Two cases:

1. an objective coefficient is going negative: do a primal pivot.
2. a right-hand side is going negative: do a dual pivot.

Continue pivoting and reducing the interval of optimal μ values until $\mu = 0$ is optimal.

Parametric Self-Dual Simplex Method with the Pivot Tool

It's easy to use the pivot tool to do the parametric self-dual simplex method.
Just think of the artificial objective row and the artificial right-hand side column as being multiplied by μ and added to the true objective and right-hand side:

obj	=	0.0			+	-3.0	x1	+	11.0	x2	+	2.0	x3
					+	-1.0	x1	+	-1.0	x2	+	-1.0	x3
w1	=	5.0	+	1.0	−	-1.0	x1	−	3.0	x2	−	0.0	x3
w2	=	4.0	+	1.0	−	3.0	x1	−	3.0	x2	−	0.0	x3
w3	=	6.0	+	1.0	−	0.0	x1	−	3.0	x2	−	2.0	x3
w4	=	-4.0	+	1.0	−	-3.0	x1	−	0.0	x2	−	-5.0	x3

21 Remarks

- It suffices to perturb just the infeasible (i.e. negative) coefficents.
- It is not necessary to use e as the initial perturbation vector. It's better to randomize this vector.
- If one perturbs all coefficients randomly, then the probability of encountering a degenerate dictionary (with $\mu \neq 0$) is zero.
- An example: on a randomly generated (600x90,000) assignment problem this variant took 1655 iterations compared to 9951 iterations for a two-phase simplex method.

Average Performance

An Unexpected Result.

Thought Experiment:

- Consider parametric self-dual method with random initial perturbations.
- The parameter μ starts at ∞.
- Via pivoting, we reduce μ to zero.
- There are $n + m$ barriers to reducing μ.
- With each pivot one barrier is passed—of course, all of the other barriers move.
- For a "random problem" there is no propensity for the jumping barriers to "intervene".
- Hence, we expect

$$\text{Expected number of iters} = (m + n)/2.$$

Empirical Study:

- Using the netlib suite of test problems, we found

$$\text{iters} \approx 0.517(m + n)^{1.05}.$$

- Remarkably close!

23 Regression

Speaking of regression, L^1 regression problems can be formulated as LPs:

Least Squares Regression (L^2):

$$\bar{x} = \text{argmin}_x \|b - Ax\|_2^2$$

$\Longrightarrow$

Solve a system of equations:

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

Least Absolute Deviation Regression (L^1):

$$\hat{x} = \text{argmin}_x \|b - Ax\|_1$$

$\Longrightarrow$

Solve an LP:

$$\min \sum_i t_i$$
$$-t_i \le b_i - \sum_j a_{ij} x_j \le t_i$$

Two Algorithms for L^1 Regression

Simplex Method and Reweighted Least Squares

L^1 regressions can be computed by solving an LP. Here's another method:

- Set the derivative of the L^1 objective function to zero.

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

- After rearranging, we get:

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

where

$$E(\hat{x}) = \text{Diag}(\epsilon(\hat{x}))^{-1} \qquad \text{and} \qquad \epsilon_i(\hat{x}) = \left|b_i - \sum_j a_{ij}\hat{x}_j\right|$$

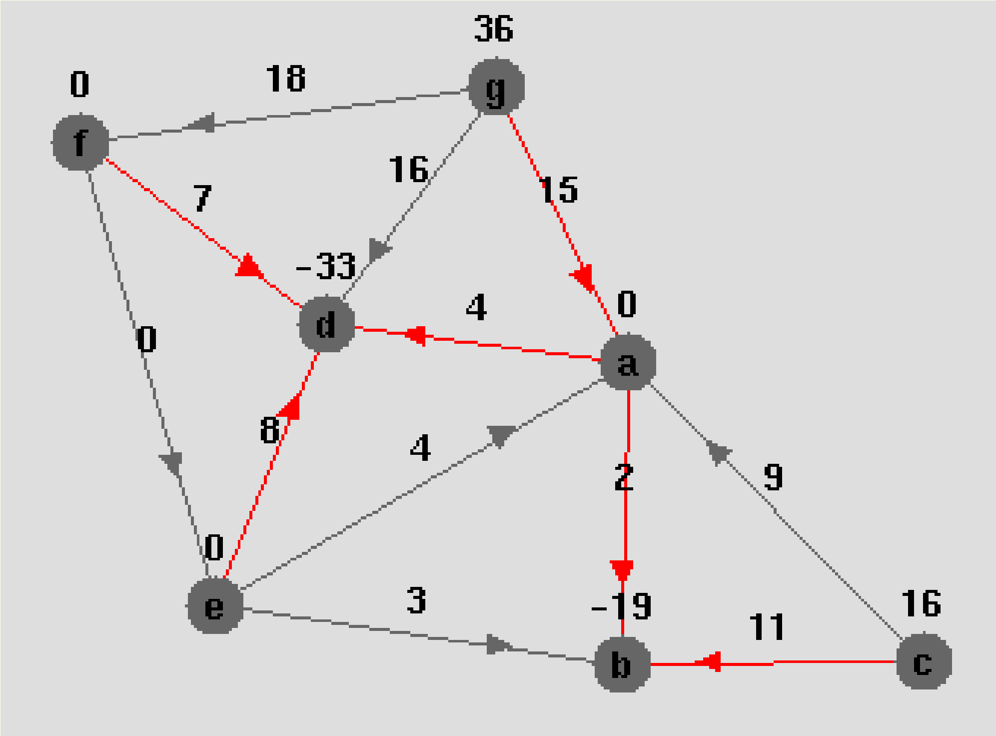
- This suggests an iterative reweighted least squares algorithm:

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

- This is the affine-scaling method applied to the LP (as first noted by M. Meketon).

Network Flows: A Symmetric View

Data and Decision Vars



Data:

- A network $(\mathcal{N}, \mathcal{A})$ with nodes $\mathcal{N}$ and arcs $\mathcal{A}$.
- $b_i, \; i \in \mathcal{N}$, supply at node i (demands are recorded as negative supplies).
- $c_{ij}, \; (i,j) \in \mathcal{A}$, cost of shipping 1 unit along arc (i,j).

Decision Variables:

- $x_{ij}, \; (i,j) \in \mathcal{A}$, quantity to ship along arc (i,j).

Network Flow Problem

No Apparent Primal-Dual Symmetry

Primal Problem

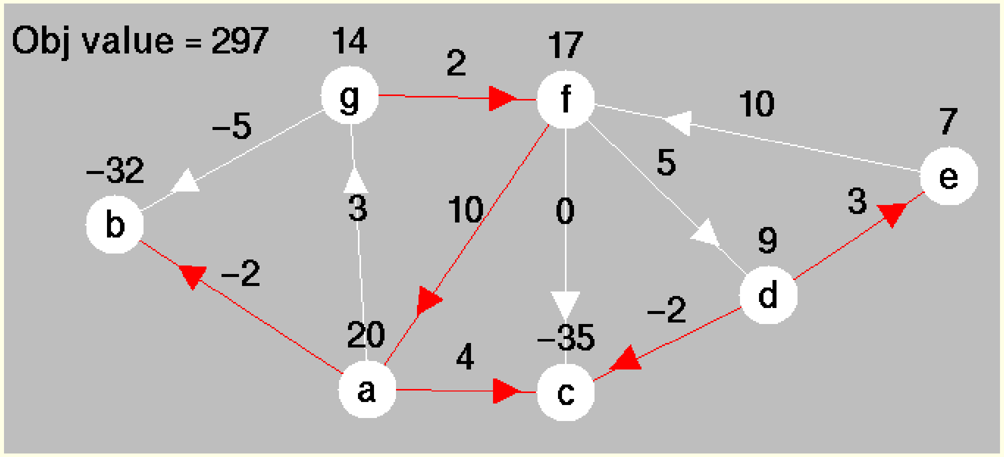
$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} & \\ \text{subject to} & \displaystyle\sum_{\substack{i\,:\\ (i,k)\in\mathcal{A}}} x_{ik} - \sum_{\substack{j\,:\\ (k,j)\in\mathcal{A}}} x_{kj} = -b_k, & k \in \mathcal{N} \\ & x_{ij} \geq 0, & (i,j)\in\mathcal{A} \end{array}$$

Dual Problem

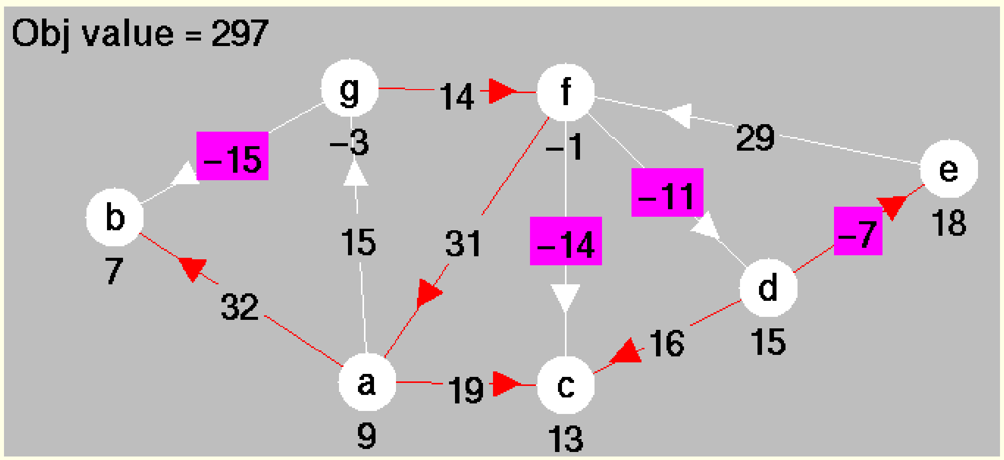
$$\begin{array}{lll} \text{maximize} & \displaystyle -\sum_{i\in\mathcal{N}} b_i y_i & \\ \text{subject to} & y_j - y_i + z_{ij} = c_{ij} & (i,j)\in\mathcal{A} \\ & z_{ij} \geq 0 & (i,j)\in\mathcal{A} \end{array}$$

Tree Solution = Basic Solution

data



variables



- Fix a root node, say a.
- Primal flows on tree arcs calculated recursively from leaves inward.
- Dual variables at nodes calculated recursively from root node outward along tree arcs using:

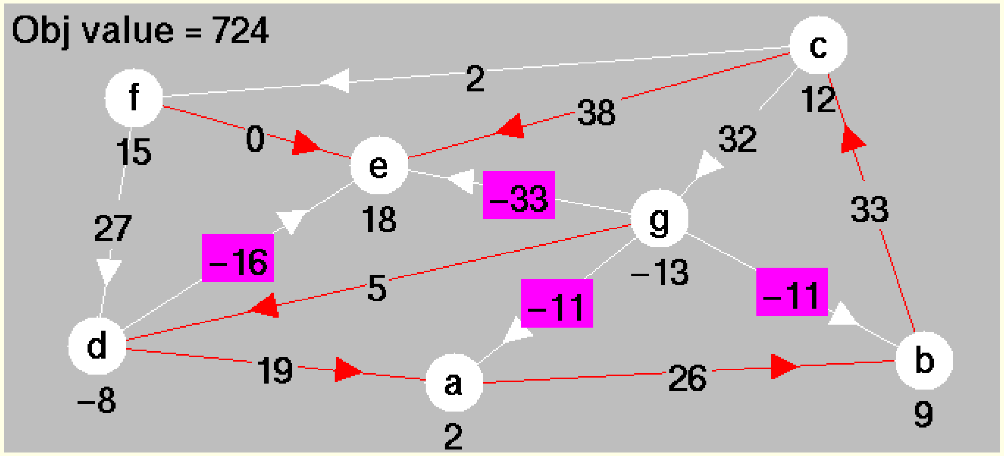
$$y_j - y_i = c_{ij}$$

- Dual slacks on nontree arcs calculated using:

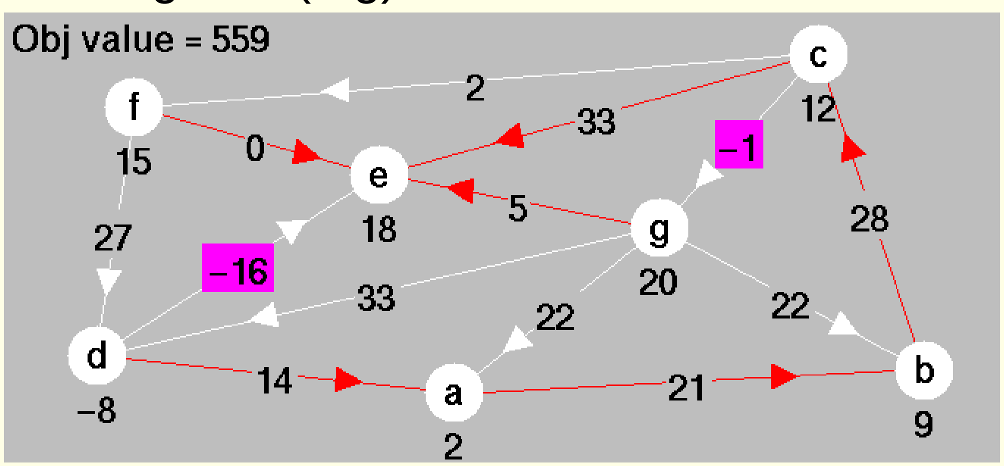
$$z_{ij} = y_i - y_j + c_{ij}.$$

Primal Network Simplex Method

Used when all primal flows are nonnegative (i.e., primal feasible).



Entering arc: (g,e)
Leaving arc: (d,g)

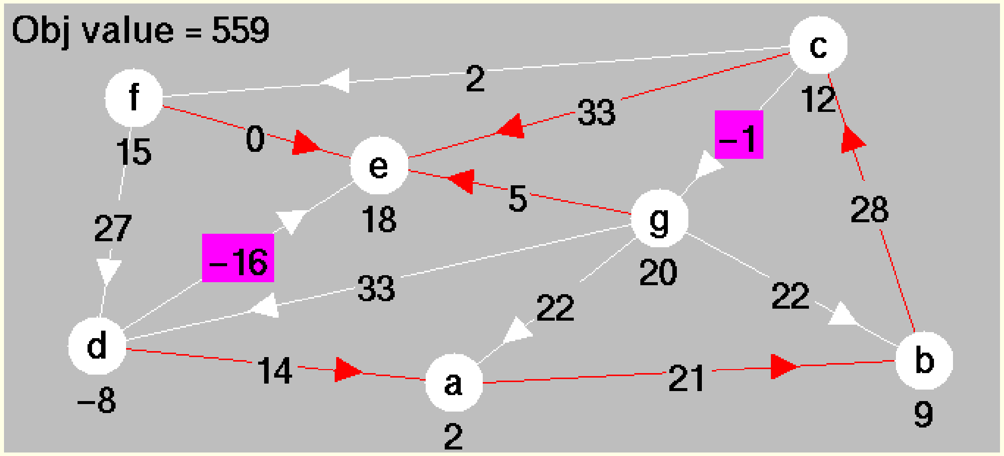


Pivot Rules:

Entering arc: Pick a nontree arc having a negative (i.e. infeasible) dual slack.

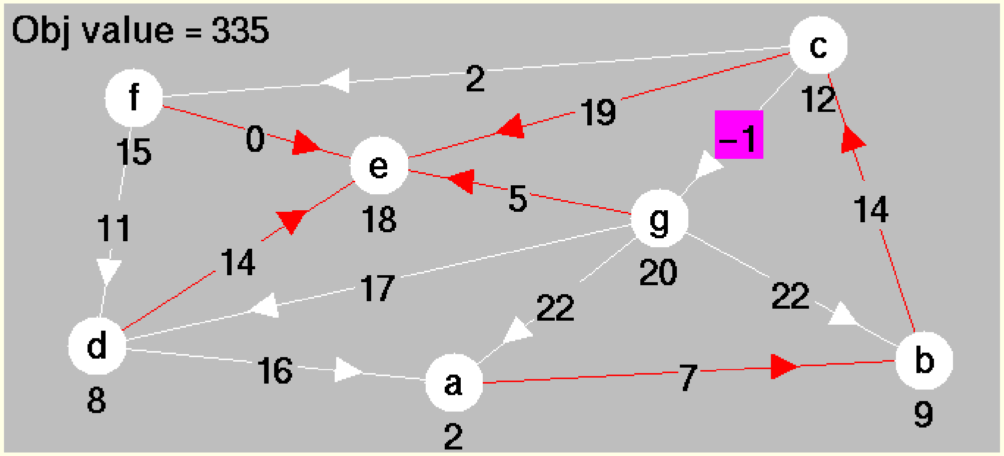
Leaving arc: Add entering arc to make a cycle.
Leaving arc is an arc on the cycle, pointing in the opposite direction to the entering arc, and
of all such arcs, it is the one with the smallest primal flow.

Primal Method—Second Pivot



Entering arc: (d,e)

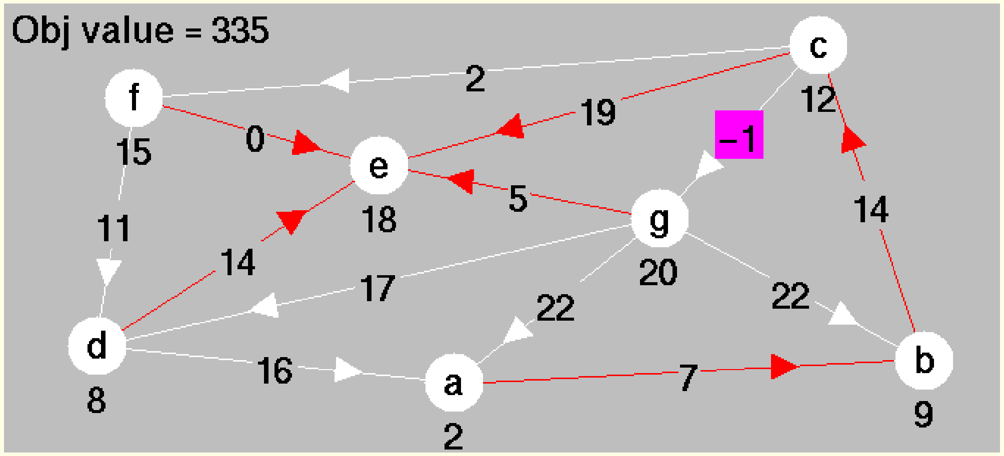
Leaving arc: (d,a)



Explanation of leaving arc rule:

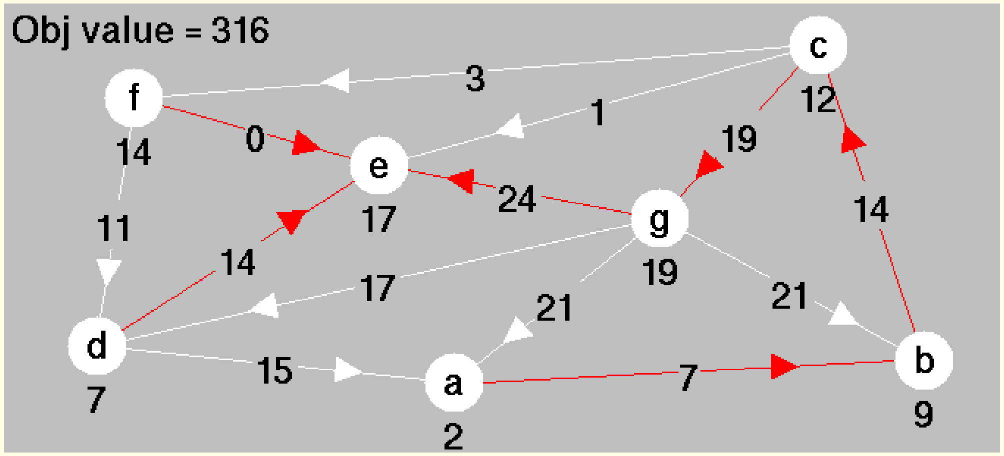
- Increase flow on (d,e).
- Each unit increase produces a unit increase on arcs pointing in the same direction.
- Each unit increase produces a unit decrease on arcs pointing in the opposite direction.
- The first to reach zero will be the one pointing in the opposite direction and having the smallest flow among all such arcs.

Primal Method—Third Pivot



Entering arc: (c,g)

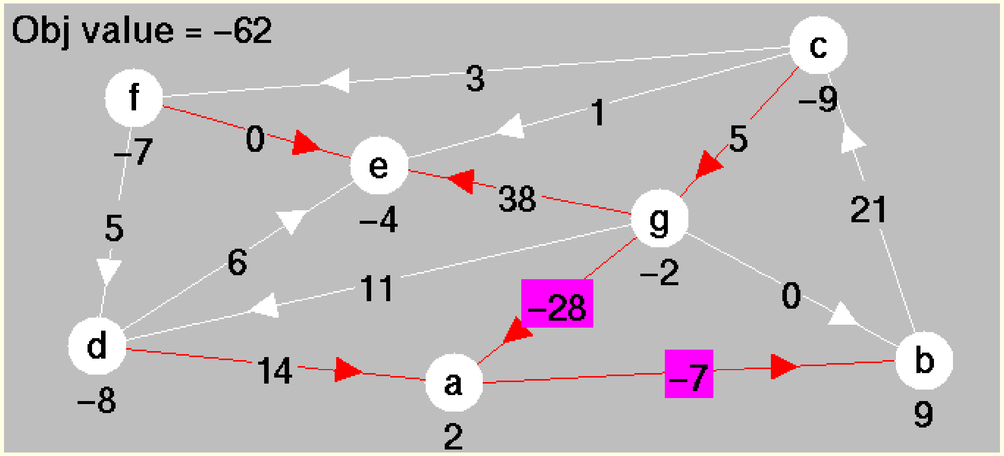
Leaving arc: (c,e)



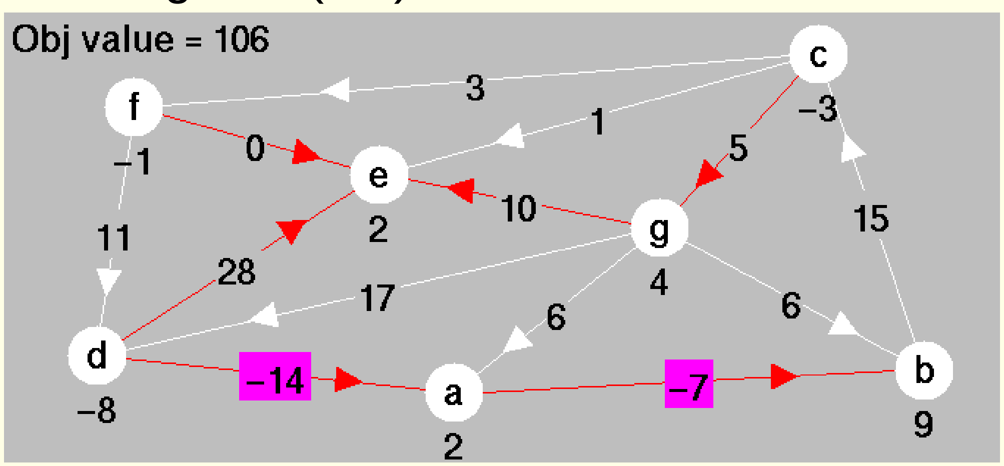
Optimal!

Dual Network Simplex Method

Used when all dual slacks are nonnegative (i.e., dual feasible).



Leaving arc: (g,a)
Entering arc: (d,e)

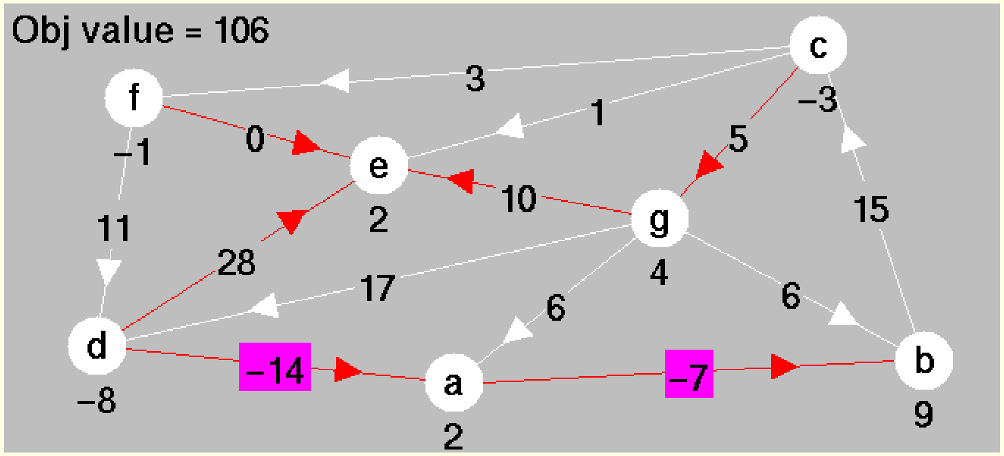


Pivot Rules:

Leaving arc: Pick a tree arc having a negative (i.e. infeasible) primal flow.

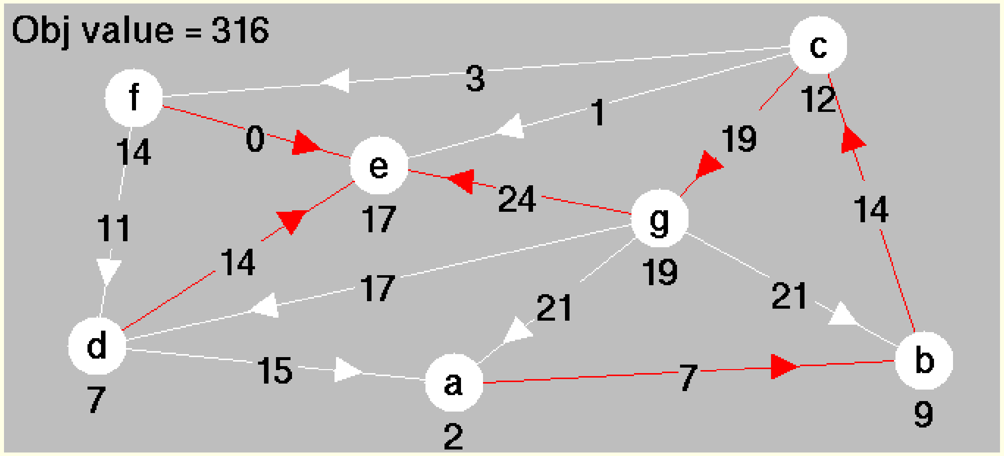
Entering arc: Remove leaving arc to split the spanning tree into two subtrees. Entering arc is an arc reconnecting the spanning tree with an arc in the opposite direction, and, of all such arcs, is the one with the smallest dual slack.

Dual Network Simplex Method—Second Pivot



Leaving arc: (d,a)

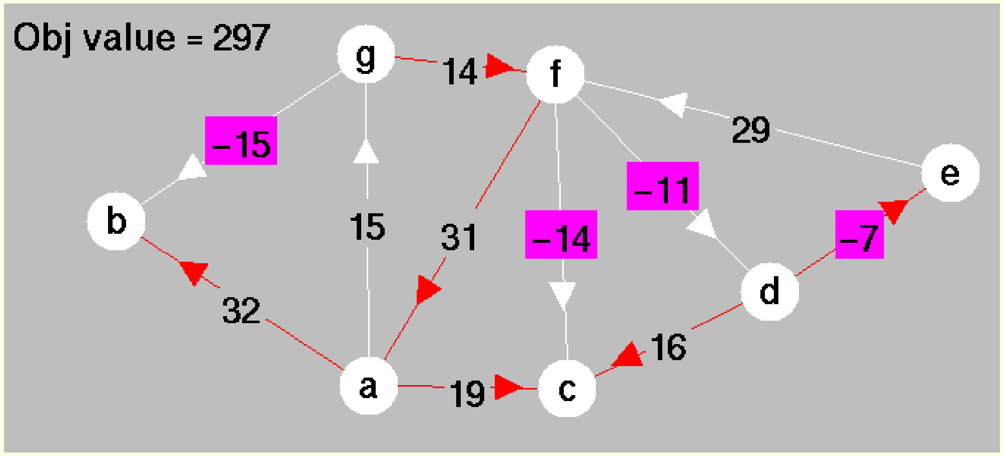
Entering arc: (b,c)



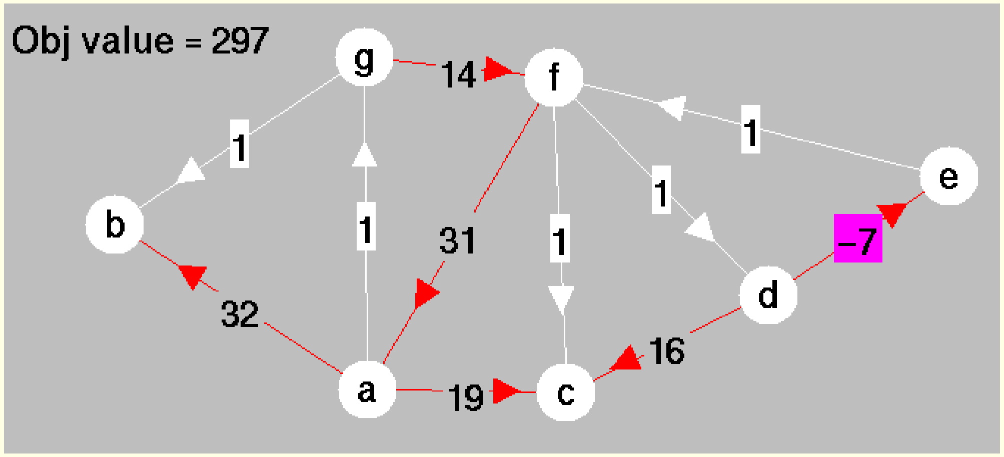
Optimal!

Two-Phase Network Simplex Method

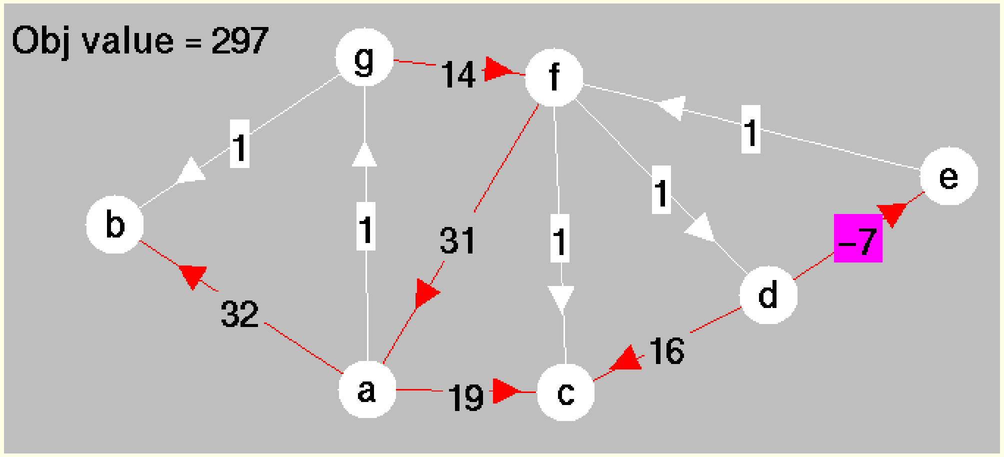
Example.



- Turn off display of dual slacks.
- Turn on display of artificial dual slacks.

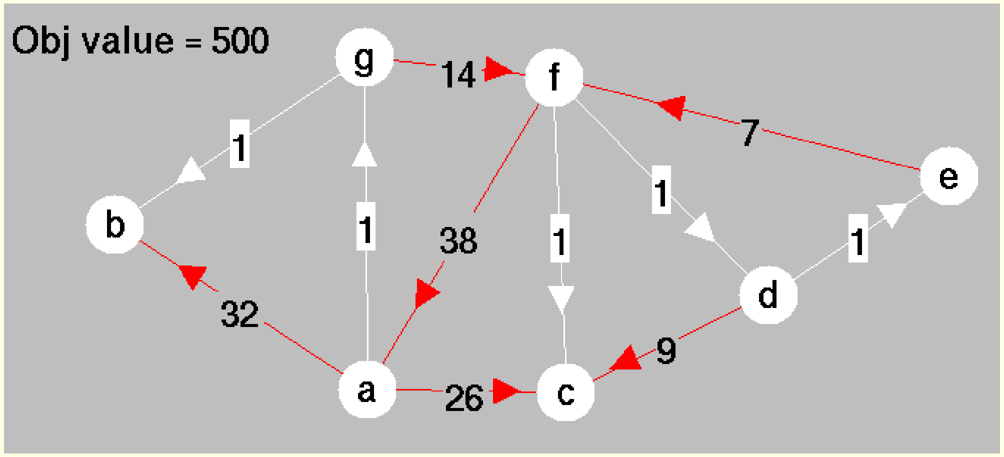


Two-Phase Method–First Pivot



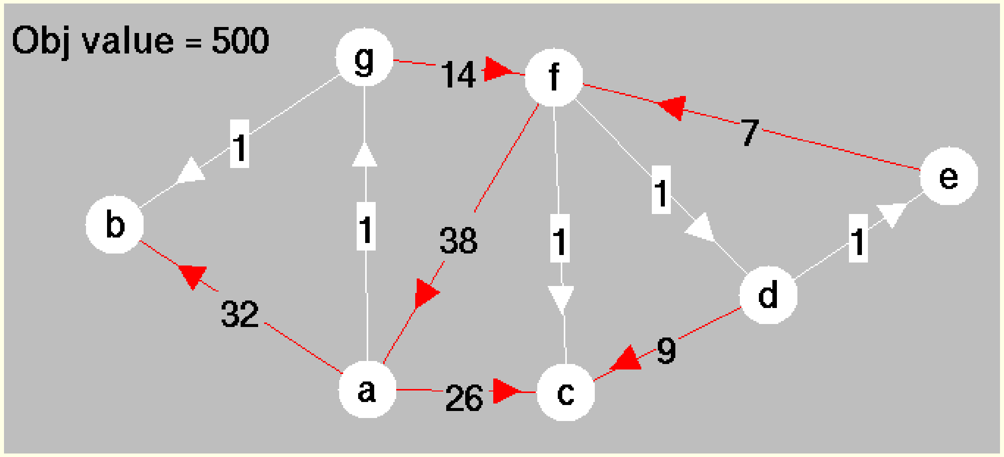
Use dual network simplex method.

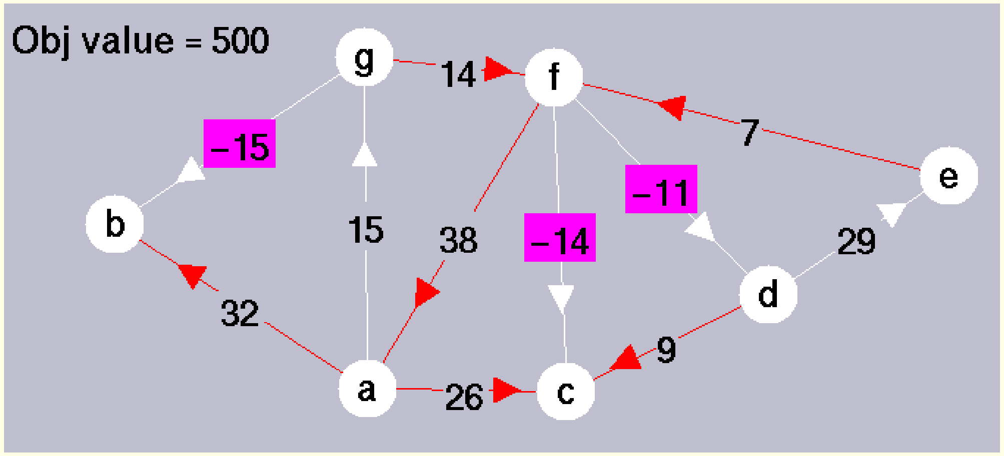
Leaving arc: (d,e) Entering arc: (e,f)



Dual Feasible!

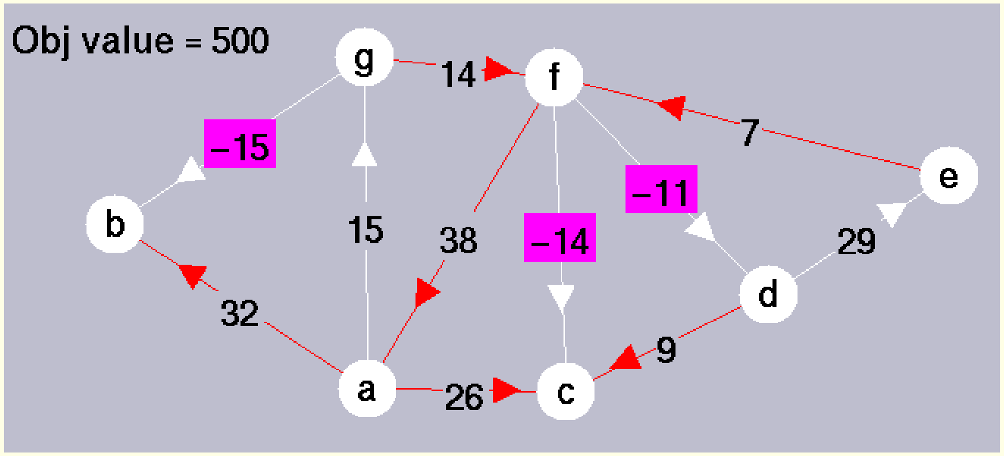
Two-Phase Method–Phase II





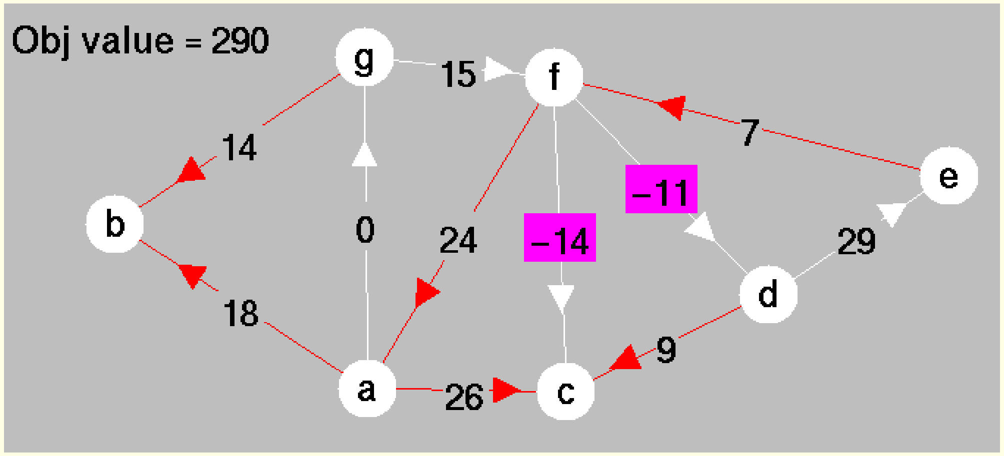
- Turn off display of artificial dual slacks.
- Turn on display of dual slacks.

Two-Phase Method–Second Pivot

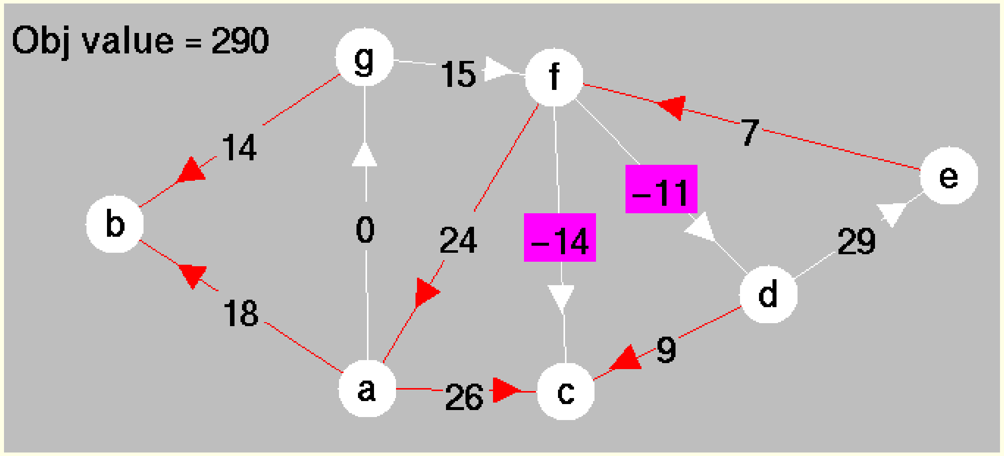


Entering arc: (g,b)

Leaving arc: (g,f)

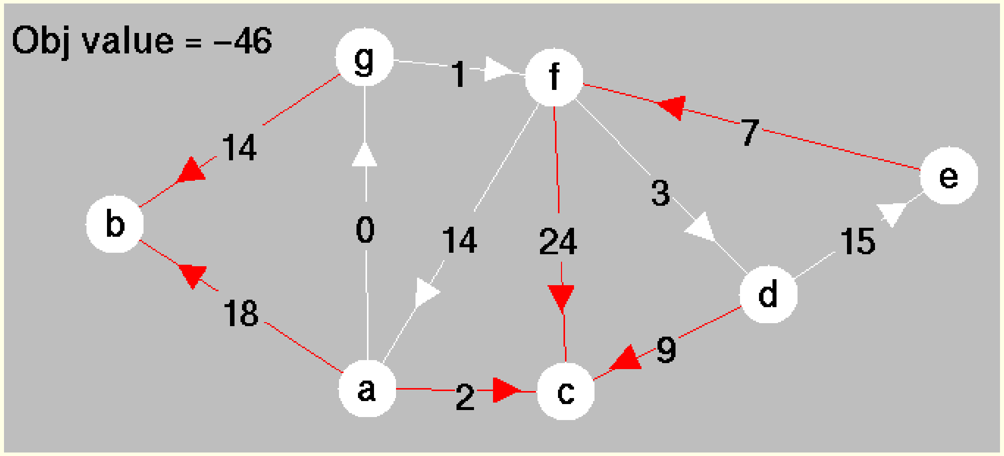


Two-Phase Method–Third Pivot



Entering arc: (f,c)

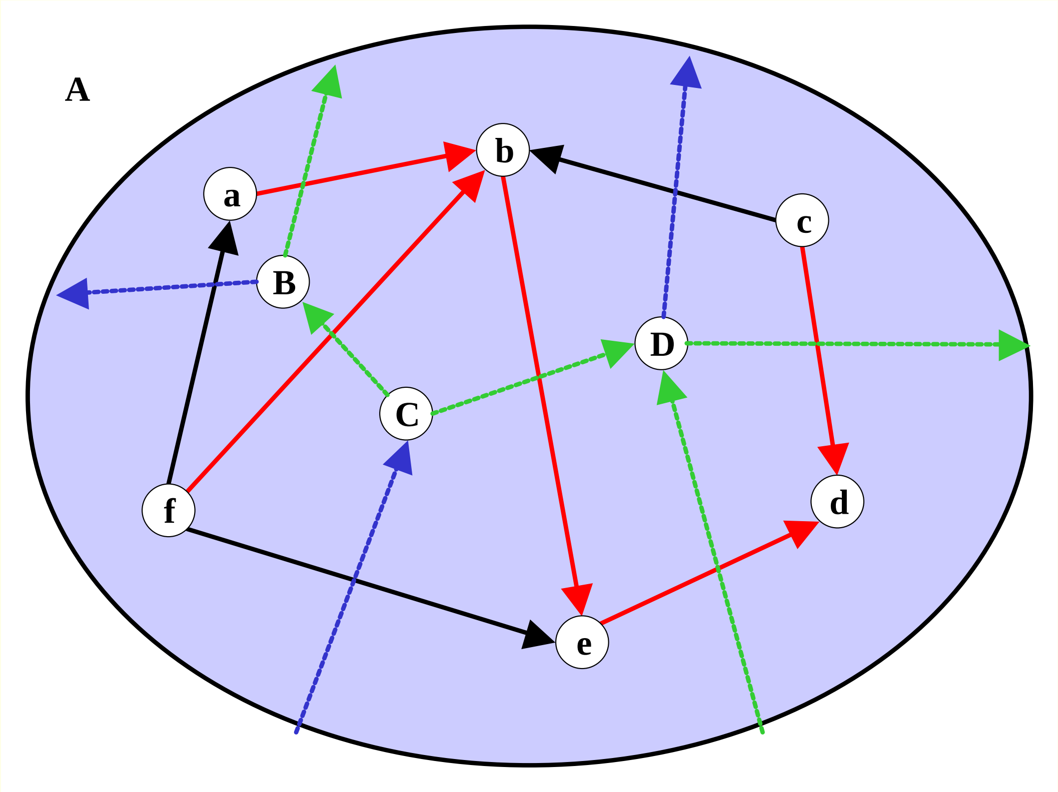
Leaving arc: (a,f)



Optimal!

Planar Networks

Symmetry Returns



Definition. *Network is called planar if can be drawn on a plane without intersecting arcs.*

Theorem. *Every planar network has a dual—dual nodes are faces of primal network.*

Notes:

- Dual node A is "node at infinity".
- Primal spanning tree shown in red.
- Dual spanning tree shown in blue (don't forget node A).

Theorem. *A dual pivot on the primal network is exactly a primal pivot on the dual network.*

An Interior-Point Method (IPM)

Another Path-Following Method

Central Path

For each $\mu \geq 0$ there exists a unique solution to:

$$\begin{aligned} Ax + w &= b \\ A^T y - z &= c \end{aligned} \qquad \begin{aligned} XZe &= \mu e \\ YWe &= \mu e \end{aligned} \qquad x, y, w, z \geq 0.$$

We start with $x, y, w, z > 0$ and use Newton's method to find search directions to a new point:

$$\begin{aligned} A\Delta x + \Delta w &= \rho && := b - Ax - w \\ A^T \Delta y - \Delta z &= \sigma && := c - A^T y + z \\ Z\Delta x + X\Delta z &= \mu e - XZe \\ W\Delta y + Y\Delta w &= \mu e - YWe. \end{aligned}$$

The algorithm is summarized on the next page...

initialize $(x, w, y, z) > 0$

while (not optimal) $\{$

$$\rho = b - Ax - w$$

$$\sigma = c - A^T y + z$$

$$\gamma = z^T x + y^T w$$

$$\mu = 0.1 \frac{\gamma}{n+m}$$

solve:

$$\begin{aligned} A\Delta x + \Delta w &= \rho \\ A^T \Delta y - \Delta z &= \sigma \\ Z\Delta x + X\Delta z &= \mu e - XZe \\ W\Delta y + Y\Delta w &= \mu e - YWe \end{aligned}$$

$$\theta = r \left(\max_{ij} \left\{ -\frac{\Delta x_j}{x_j}, -\frac{\Delta w_i}{w_i}, -\frac{\Delta y_i}{y_i}, -\frac{\Delta z_j}{z_j} \right\} \right)^{-1} \wedge 1$$

$$x \leftarrow x + \theta \Delta x, \qquad w \leftarrow w + \theta \Delta w$$

$$y \leftarrow y + \theta \Delta y, \qquad z \leftarrow z + \theta \Delta z$$

$\}$

Another IPM: Affine-Scaling = Reweighted Least Squares

The Affine-Scaling algorithm is obtained by putting $\mu = 0$.

For problems with x free, we can also set $z = 0$.

In this case the equation for the search direction Δx reduces to:

$$(x + \Delta x) = \left(A^T E A\right)^{-1} (\sigma + A^T E b),$$

where

$$E = W Y^{-1}.$$

For L^1 regression it is possible to initialize so that $\sigma = 0$.

It then remains zero throughout and the new x is related to the old x by a weighted least-squares formula:

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