Myths and Counterexamples in Linear Programming

Harvey J. Greenberg

hjgreenberg@gmail.com

February 20, 2010
(Previous: September 16, 2009)

Contact the author to be notified of updates.

List of LP Myths

Acknowledgments

I thank the authors of the many papers I used in this compilation. I especially thank Hossein Arsham, András A. Benczúr, Charles Blair, Michael A.H. Dempster, Saul I. Gass, Fred Glover, Richard C. Grinold, Julian Hall, Alan J. Hoffman, David S.P. Hopkins, T.C. Hu, Santosh N. Kabadi, Ed Klotz, Nimrod Megiddo, Katta G. Murty, James B. Orlin, Romeo Rizzi, Tim Roughgarden, Suvrajeet Sen, Craig Tovey, Roman L. Weil, Jr., H. Paul Williams, and Henry Wolkowicz for their feedback.

The general form of a linear program (LP) is the optimization of a linear function subject to a system of linear equations and inequalities. The *standard form* is

$$\min\ cx : \ Ax = b,\ x \geq 0,$$

where $\text{rank}(A) = m =$ number of equations. This form is particularly useful when considering the simplex method.

When talking about duality, I use the *canonical form*:

$$\min\ cx : \ Ax \geq b,\ x \geq 0.$$

(No rank condition on A.) This renders the dual prices non-negative, giving the dual canonical form:

$$\max\ \pi b : \ \pi A \leq c,\ \pi \geq 0.$$

Unless stated otherwise, or implied from context, the LP in question could be any linear system; it need not be in standard or canonical form.

The *standard simplex method* is the original pivot-selection rule by Dantzig, applied to the standard form — a variable with the greatest reduced cost (rate of improvement) is chosen to enter the basis. An alternative is the *best-gain* criterion, which evaluates the actual gain of each candidate to enter the basis by computing its change in level and multiplying by the rate of improvement.

A constraint is *redundant* if its removal does not change the set of feasible points. An inequality is an *implied equality* if it must hold with equality in every feasible solution. Consult the *Mathematical Programming Glossary*[44] for other terms and concepts, not defined here.

LP Myth 1. *All redundant constraints can be removed.*

The reason this is incorrect is that once a redundancy is removed, the other constraints may no longer be redundant.

Counterexample. $x, y \geq 0$ and $x - y = 0$. Each non-negativity constraint is redundant, but they cannot both be removed. The redundancy of $x \geq 0$ follows from the equation and the non-negativity of y: $x = y \geq 0$.

Practical use was first reported by Tomlin and Welch[76], and that led to a theory of *common dependency sets* by Greenberg[35].

LP Myth 2. *A degenerate basis implies there is a (weakly) redundant constraint.*

Counterexample. Consider $y \geq 0$, $x \geq 1$, $x + y \leq 1$. The only feasible point is $(x, y) = (1, 0)$ with slack and surplus variables both 0. Thus, each of the possible feasible bases is degenerate, but no constraint is redundant.

Sierksma and Tijssen[71] generalized this: If a face of dimension $n - 1$ or $n - 2$ is degenerate, the defining linear inequalities are not minimal — that is, the system must contain either a redundant inequality or an implied equality. Note the special conditions on dimension. For $n \geq 3$, it cannot apply generally to an extreme point (face of 0 dimension). A pyramid is a counterexample for $n = 3$. The pyramid's top extreme point is degenerate because it is the intersection of 4 planes, but none of the defining inequalities is redundant or an implied equality.

LP Myth 3. *If an LP has an optimal solution, there is an extreme point of the feasible region that is optimal.*

Counterexample. Arsham[2, #9] provides the following: $\max x_1 + x_2 : x_1 + x_2 \leq 5$. The feasible set is a polyhedron with no extreme point. This occurs because we do not require the variables to be non-negative.

The myth's statement is true when the LP is in standard form. Converting the example to standard form increases the dimension:

$$\begin{gathered} \max u_1 - v_1 + u_2 - v_2 : \\ u_1 - v_1 + u_2 - v_2 + x_3 = 5, \\ u_1, v_1, u_2, v_2, x_3 \geq 0, \end{gathered}$$

where we have augmented the slack variable, x_3, and we have partitioned each of the original variables into their positive and negative parts:

$$x_1 = u_1 - v_1 \text{ and } x_2 = u_2 - v_2.$$

(Be sure to see LP Myth 13.)

In this higher-dimensional space, it is true that an extreme point is optimal — in particular, $(u_1, v_1, u_2, v_2, x_3) = (5, 0, 0, 0, 0)$. In fact, there are three extreme points; the other two are (0,0,5,0,0) and (0,0,0,0,5). Each of these three extreme points is optimal for some objective value coefficients, spanning all that render the LP optimal (vs. unbounded).

LP Myth 4. *If one knows that an inequality constraint must hold with equality in every optimal solution, it is better to use the equality in the constraint because it will reduce the solution time.*

First, it is not necessarily the case that it will reduce the solution time — the solver could get a first feasible solution faster with the inequality formulation. Second, even if the tighter version solves faster (perhaps by pre-solve reduction), it is generally better to let the model tell you the answer than for you to wire the result. Your intuition could be wrong, or there could be a data entry error that goes undetected with the equality constraint. A better approach is to attach a back-end report to examine all things "known" to be true and flag the violations. Thus, if an inequality is slack and you expected it to be tight, you can investigate why the model did what it did.

LP Myth 5. *In a dynamic LP, each period should be the same duration.*

This is tacitly implied in many textbook examples. The reality is that we know more about what is likely to happen tomorrow than next year. In general, data can provide forecasts for demands, supplies, and other model parameters, but the accuracy tends to be less as the time is further into the future. One may have, for example, a 5-year planning model with the first 12 time periods being months, the next 4 periods being quarters, and the last 3 being years.

LP Myth 6. *Maximizing an absolute value can be converted to an equivalent LP.*

Consider the conversion of the NLP with free variables:

$$\max \sum_j c_j \, |x_j| : \; Ax = b$$

to a standard LP:

$$\max \sum_j c_j x_j^+ + \sum_j c_j x_j^- \; : \; Ax^+ - Ax^- = b, \; x^+, x^- \geq 0.$$

Shanno and Weil[70] point out that this equivalence is not correct if $c \not\leq 0$.

Counterexample. max $|x| : -4 \leq x \leq 2$, where x is a free variable. (Add slack variables to put into equality form.) The associated LP is

$$\max \; x^+ + x^- \; : \; -x^+ + x^- + s_1 = 4, \; x^+ - x^- + s_2 = 2, \; x^+, x^-, s \geq 0.$$

The LP is unbounded (let $x^+ = 4 + \theta$, $x^- = \theta \to \infty$), but the original NLP is optimized by $x = -4$.

Shanno and Weil note that the unboundedness problem is avoided with the simplex method by adding the restricted basis entry condition: $x_j^+ x_j^- = 0$ for all j. When $c \leq 0$, this condition is satisfied anyway, but for $c_j > 0$, it must be forced.

Rao[60] points out that $c \geq 0$ means the objective function is convex, which implies there is an extreme point that is optimal, but there could be (and generally are) local maxima. On the other hand, $c \leq 0$ means the objective function is concave, so local maxima is not an issue.

Kaplan[48] proposed the following modification. Bound the variables by a single constraint:

$$\sum_j x_j^+ + \sum_j x_j^- \leq M,$$

where M is large enough to make this redundant when the NLP has a solution. Then, he purported that if this constraint is not active at the LP optimum (that is, if the slack variable is basic), it solves the NLP. If it is active (that is, if the slack variable is nonbasic), the NLP is unbounded. Unfortunately, this simple fix does not always work.

Counterexample. Ravindran and Hill[61] provide the following:

$$\max\ |x_1| : x_1 - x_2 = 2.$$

Kaplan's LP is:

$$\begin{array}{ll} \max\ x_1^+ - x_1^- : x^+, x^-, s \geq 0, & \\ x_1^+ - x_1^- - x_2^+ + x_2^+ & = 2 \\ x_1^+ + x_1^- + x_2^+ + x_2^+ + s & = M. \end{array}$$

The simplex method obtains the basic solution with $x_1^+ = 2$ and $s = M - 2$ (and all other variables zero). Thus, this does not solve the NLP. The problem here is that the LP can have only two basic variables, and the original polyhedron has no extreme points.

The unboundedness is not the real issue. Ravindran and Hill note that we could add the constraint $-6 \leq x_1 \leq 4$. Then, the LP solution is the same, but the original problem is solved by $x = (-6, -8)$.

For $c \leq 0$, the NLP is equivalent to minimization of the form:

$$\min\ \textstyle\sum_j |\alpha_j x_j - \beta_j| : x \in P,$$

where P is the polyhedron. This is equivalent to the LP:

$$\min\ \textstyle\sum_j v_j : x \in P,\ v_j \geq \alpha_j x_j - \beta_j,\ v_j \geq -\alpha_j x_j + \beta_j.$$

This is the common LP equivalent, and it uses two properties: $|z| = \max\{z, -z\}$ and $\min\{v : v = |z|\} = \min\{v : v \geq |z|\}$. This latter property fails for maximization. The Shanno-Weil example would become

$$\max\ v : v \geq x,\ v \geq -x,\ -6 \leq x \leq 4,$$

which is unbounded.

Opportunity Knocks

There remains the issue of how we can use LP to maximize a linear function of absolute values, where the coefficients (c) could be positive. For $c > 0$, we know this is an instance of the hard problem of maximizing a convex function on a polyhedron, and there can be local maxima at some vertices. However, is there some special structure to exploit?

LP Myth 7. *The expected value of the second-stage of a stochastic linear program with recourse is a differentiable function, provided that the random variable is continuous.*

My thanks to Suvrajeet Sen for suggesting this.

The 2-stage recourse LP model is defined here as:

$$\min\ cx + \mathbf{E}_\theta[h(x, \theta)] : x \geq 0,\ Ax = b,$$

where θ is a random variable, and the *recourse function* is the LP value:

$$h(x, \theta) = \min\{Cy : y \geq 0,\ By = \theta - Tx\}.$$

The myth asserts that h is differentiable in x, provided the probability distribution function of θ is continuous. (It is obvious that h is not generally differentiable for a discrete distribution function since then h is piece-wise linear.)

Counterexample. Sen[69] provides the following: let $\theta = (d_1, d_2, d_3)$ be demands in the second stage for three destinations, and let the first stage determine supplies from two sources, so h is the optimal value of a transportation problem:

$$\begin{aligned} h(x, \theta) = \min\ \textstyle\sum_{i,j} C_{ij} y_{ij} &: y \geq 0, \\ y_{i1} + y_{i2} + y_{i3} \leq x_i &\quad \text{for } i = 1, 2 \\ y_{1j} + y_{2j} \geq d_j &\quad \text{for } j = 1, 2, 3. \end{aligned}$$

Suppose d_1, d_2 are deterministic and $d_3 \in (0, D)$ for some finite $D > 0$. Let the unit shipping cost matrix be

$$C = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 2 & 2 \end{bmatrix}.$$

Suppose $\bar{x} = (d_1, d_2 + D)$. Then, the following are alternative dual-optimal solutions:

$$\lambda = (-3, 0,\ 3, 2, 2) \text{ and } \lambda' = (-1, 0,\ 1, 2, 2).$$

(Supply prices are $-(\lambda_1, \lambda_2)$, and demand prices are $(\lambda_3, \lambda_4, \lambda_5)$.) Sen proves that these are optimal for all $d_3 \in (0, D)$. The subgradient of h thus includes subgradients $(-3, 0)$ and $(-1, 0)$, so the recourse function is not differentiable at $\bar{x}$.

Sen extends earlier works to establish necessary and sufficient conditions for h to be differentiable.

LP Myth 8. **new** *For a multistage stochastic program with non-anticipativity constraints, there exist optimal dual multipliers that are also non-anticipative.* *next new* ▷

My thanks to Suvrajeet Sen for suggesting this.

Non-anticipativity constraints require recourse variables to be independent of the history of outcomes. (See Beasley[6] for a succinct introduction and example.)

Counterexample. Higle and Sen[42] consider a 3-stage LP:

$$\min \sum_{t=1}^{3} c_t x_t : -1 \leq x_t \leq 1,\ t = 1, 2, 3,\ x_t \geq x_{t+1},\ t = 1, 2.$$

Let c be random with four equally-likely values:

$$c \in \{(1, 1, 1),\ (1, 1, -1),\ (1, -1, 1),\ (1, -1, -1)\}.$$

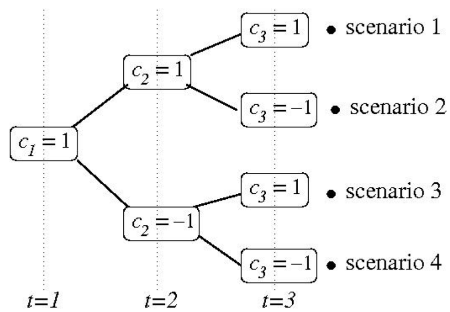
Indexing the objective coefficients as c_{it} for scenario i at stage t, let x_{it} be the associated decision variable. Thus, the recourse LP is

$$\min \tfrac{1}{4}\sum_{i=1}^{4}\sum_{t=1}^{3} c_{it}x_{it} : -1 \le x_{it} \le 1,\ t = 1,2,3,\ x_{it} \ge x_{i\,t+1},\ t = 1,2$$

$$x_{11} = x_{21} = x_{31} = x_{41},\ x_{12} = x_{22},\ x_{32} = x_{42},\ x_{13} = x_{33},\ x_{23} = x_{43}, \qquad \text{(LP.1)}$$

where (LP.1) comprise the non-anticipativity constraints. These are due to the commonality: $c_{i1} = 1$ for all $i = 1,\dots,4$, $c_{12} = c_{22}$, and $c_{32} = c_{42}$.

The scenario tree, shown on the right, illustrates that each path through time corresponds to a scenario. There is a bifurcation at a node when there is an event that changes the cost coefficient. For example, at $t = 1$, events can cause $c_2 = 1$ or $c_2 = -1$. However, to avoid clairvoyance, the decision variable, x_2, must be the same for scenarios 1 and 2, and for scenarios 3 and 4, since the cost is the same within each of those groupings. That is the "commonality" that yields the non-anticipativity constraints.



Let $u = (u_1, u_2, u_3)$ denote the dual variables for the non-anticipativity constraints associated with $t = 1$, and let $v = (v_1, v_2)$ be those associated with $t = 2$. The myth asserts that the dual variables associated with the non-anticipativity constraints are themselves non-anticipative — that is, $u_1 = u_2 = u_3$. However, the dual solution has $u = (\,^1/_4\,, -\,^1/_2\,, -\,^1/_4\,)$, giving a contradiction.

Higle and Sen model the non-anticipativity constraints differently, but primal-equivalent to (LP.1):

$$x_{i1} - \tfrac{1}{4}\textstyle\sum_{k=1}^{4} x_{k1} = 0 \qquad \text{for } i = 1,\dots,4 \qquad \text{(LP.2a)}$$

$$x_{i2} - \tfrac{1}{2}(x_{12} + x_{22}) = 0 \qquad \text{for } i = 1,2 \qquad \text{(LP.2b)}$$

$$x_{i2} - \tfrac{1}{2}(x_{32} + x_{42}) = 0 \qquad \text{for } i = 3,4 \qquad \text{(LP.2c)}$$

The dual variables now measure the rate of deviation from a group's average. Intuition may suggest that this averaging induces a non-anticipative dual stochastic process.

However, an optimal dual value has $u = (0, 0, \,^3/_8\,, \,^1/_8\,)$ and $v = (0, \,^1/_4\,, 0, \,^1/_2\,)$, which contradict non-anticipativity. (Higle and Sen obtain different optimal dual values, but they show all optimal dual values are non-anticipative.)

Higle and Sen prove that the optimal non-anticipativity dual variables are non-anticipative if, and only if, the expected value of perfect information equals zero. In the example, $EVPI = -1\frac{1}{2}$.

LP Myth 9. *A feasible infinite-horizon LP can be approximated by truncating to a finite horizon.*

The infinite-horizon model has the form:

$$f^* = \max \sum_{t=0}^{\infty} c^t x^t : x \geq 0,\ A^0 x^0 = b^0,\ A^{t+1} x^{t+1} - B^t x^t = b^t, \text{ for } t = 0, 1, \ldots$$

One associated finite-horizon model is the truncation:

$$\begin{aligned} f^*(T) = \max\ \textstyle\sum_{t=0}^{T} c^t x^t : x \geq 0,\ A^0 x^0 = b^0, \\ A^{t+1} x^{t+1} - B^t x^t = b^t, \text{ for } t = 0, 1, \ldots, T-1, \\ x^T \in \mathcal{T}, \end{aligned}$$

where $\mathcal{T}$ is an *end condition.*

Consider a stationary model, where $A^t = A$, $B^t = B$, and $b^t = b$ for all t. One could define $x^t = x^T$ for all $t > T$, in which case the end condition simply requires $(A - B)x^T = b$. For this case, Grinold[38] provides the following:

Counterexample. Let $A = 1$, $B = 1.6$, $b = 1$, and $c^t = \left(\frac{1}{4}\right)^t$. Then, $x^t = \frac{1.6^{t+1}-1}{0.6}$ is feasible, and $\sum_{t=0}^{\infty} c^t x^t = 2.222\ldots$ However, $(A - B)x = 1,\ x \geq 0$ has no solution.

Grinold provides another counterexample, where x^T is not required to satisfy $(A - B)x^T = b$.

Counterexample. Let $c^t = \left(\left(\frac{1}{2}\right)^t, \left(\frac{1}{2}\right)^t, 0, 0 \right)$, $b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$,

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The truncated-horizon model is

$$\max \textstyle\sum_{t=0}^{T} \left(\frac{1}{2}\right)^t (x_1^t + x_2^t) : x \geq 0$$

$$\begin{array}{lllllllll} x_1^0 & & + & x_3^0 & & & = & 1 & \\ & - x_2^0 & & & + & x_4^0 & = & 0 & \\ x_1^t & & + & x_3^t & & + x_2^{t-1} & = & 1 & \text{for } t = 1, \ldots, T \\ & - x_2^t & & & + & x_4^t - x_2^{t-1} & = & 0 & \text{for } t = 1, \ldots, T \end{array}$$

Let $x^t = (1, 0, 0, 0)^\intercal$ for $t = 0, \ldots, T-1$ and $x^T = (1, \theta, 0, \theta)$. This is feasible for all $\theta > 0$, and the objective value satisfies

$$f^*(T) \geq \sum_{t=0}^{T} c^t x^t = \sum_{t-0}^{T} \left(\tfrac{1}{2}\right)^t + \theta \left(\tfrac{1}{2}\right)^T = (\theta - 1) \left(\tfrac{1}{2}\right)^T + 2.$$

Letting $\theta \to \infty$, we conclude that the truncated LP is unbounded for all finite T. However, the infinite-horizon objective is bounded, with optimal value 2.

Grinold provides greater analysis and conditions for finite-horizon approximations. He extends his work by analyzing four methods to correct end effects[39]. Also see Evers[23].

Even when the myth's statement holds, a software anomaly can arise with discounting.

Counterexample. The following infinite-horizon model is unbounded:

$$\max \sum_{t=0}^{\infty} \left(\tfrac{1}{2}\right)^t x_t : x \geq 0, \; x_t + \sum_{s=0}^{t-1} x_s \leq 2^{t+1}, \;\; \text{for } t = 0, 1, \ldots$$

Letting $x_t = 2^t$, each constraint holds and the objective equals ∞. An associated finite-horizon model is the truncation:

$$f^*(T) = \max \sum_{t=0}^{T} \left(\tfrac{1}{2}\right)^t x_t : x \geq 0, \; x_t + \sum_{s=0}^{t-1} x_s \leq 2^{t+1}, \;\; \text{for } t = 0, 1, \ldots, T.$$

For T sufficiently large, the objective coefficient, $\left(\tfrac{1}{2}\right)^t$, becomes zero in the computer. Thus, the computed value of $f^*(T)$ is bounded. In particular, both MATLAB® and CPLEX® reach this at $T = 20$, giving $f^*(T) = 43$ for all $T \geq 20$.

We can add stationary bounds, $x^t \leq U$ for $t > \tau$, so the infinite-horizon model is bounded. For $\tau > 20$, the problem persists: the truncated optima converge to the incorrect limit due to the computer's limitation of precision.

LP Myth 10. *The duality theorem applies to infinite LPs.*

An infinite LP is one with an infinite number of variables and constraints, which arises naturally in infinite-horizon models. The duality theorem includes the following implications:

1. If x is primal-feasible, y is dual-feasible, and they satisfy complementary slackness, they are optimal in their respective LPs.
2. If the primal and dual have optimal solutions, their objective values are equal.

Counterexample. Hopkins[46] rebukes the first implication with the following:

$$\begin{array}{rcrcrcl} & & \min\ x_1 : x \geq 0, & & & & \\ x_1 & - & x_2 & & & \geq & 1 \\ & & x_2 & - & x_3 & \geq & 0 \\ & & & \ddots & & & \end{array}$$

Its dual is given by:

$$\begin{array}{rcrcrcl} & & \max\ y_1 : y \geq 0, & & & & \\ y_1 & & & & & \leq & 1 \\ -y_1 & + & y_2 & & & \leq & 0 \\ & - & y_2 & + & y_3 & \leq & 0 \\ & & & \ddots & & & \end{array}$$

A primal-feasible solution is $x = (2, 1, 1, \ldots)$; a dual-feasible solution is $y = (1, 1, \ldots)$. They satisfy complementary slackness, but x is not optimal for the primal since $x = (1, 0, 0, \ldots)$ is also primal-feasible with lower objective value.

Hopkins identifies the cause: the sequence $\left\{\sum_{i=1}^T \sum_{j=1}^T y_i A_{ij} x_j\right\}_{T\to\infty}$ is not absolutely convergent. (Hopkins proves that absolute convergence is a sufficient condition for duality to hold.)

Counterexample. Grinold and Hopkins[40] rebuke the second implication with the following:

$$\begin{aligned} \min\ \textstyle\sum_{t=0}^{\infty} \left(\tfrac{1}{2}\right)^t z_t :\ x_0 = 1,\ y_0 + z_0 = 1 \\ -2y_{t-1} + x_t = 0,\ -2x_{t-1} + y_t + z_t = 0 \quad \text{for } t = 1, 2, \ldots \\ x_t, y_t, z_t \ge 0 \quad \text{for } t = 0, 1, \ldots \end{aligned}$$

The objective is bounded below by zero. A feasible solution is $x_t = y_t = 2^t$, $z_t = 0$, and it is optimal since its objective value is zero.

The dual is

$$\begin{aligned} \max\ u_0 + v_0 : \\ u_t - 2v_{t+1} \le 0,\ v_t - 2u_{t+1} \le 0,\ v_t \le \left(\tfrac{1}{2}\right)^t, \quad \text{for } t = 0, 1, 2, \ldots \end{aligned}$$

Since $v_0 \le 1$ and $u_0 \le 2v_1 \le 1$, the objective value is bounded above by 2. A feasible solution is $u_t = v_t = \left(\frac{1}{2}\right)^t$, and it is optimal since its objective value is 2.

Moreover, the complementary slackness conditions are satisfied:

$$(u_t - 2v_{t+1})x_t = 0,\ (v_t - 2u_{t+1})y_t = 0,\ (v_t + (\tfrac{1}{2})^t)z_t = 0, \text{ for } t = 0, 1, 2, \ldots$$

The failure of equal objective values can be attributed to the correction by u_{t+1}, v_{t+1} in the dual. The truncation yields an optimal value of zero because the last constraints do not have that correction:

$$u_T \le 0,\ v_T \le 0.$$

This back-propagates to render $u_0 = v_0 = 0$.

Also see Evers[23, §6.9,§6.20].

LP Myth 11. *If the optimal value of a slack variable is zero, the associated constraint is binding.*

As suggested by H. P. Williams, this myth reflects confusion in terminology. An inequality constraint is *active* at a point if it holds with equality; it is *binding* if its removal changes the solution.

Counterexample. $\max\ x_1 :\ x \ge 0,\ x_1 + 2x_2 \le 3,\ 2x_1 + x_2 \le 3,\ x_1 + x_2 \le 2.$

The (unique) optimal solution is at $x^* = (1, 1)$, and all slack variables are zero. Although the last constraint is active, it is not binding (it is redundant).

LP Myth 12. *If the primal and dual are both degenerate, they cannot both have alternative optima.*

Suggested by H. P. Williams, this myth violates the established fact:

> *If the primal and dual LPs have optimal solutions, they have a strictly complementary optimal solution.*

Counterexample.

Primal	Dual
$\max\ 0x : x \geq 0,\ x_1 \leq 0,\ x_2 \geq 0.$	$\min\ 0\pi : \pi \geq 0,\ \pi_1 \geq 0,\ \pi_2 \leq 0.$

Primal optima are of the form $(0, x_2) : x_2 \geq 0$; dual optima are of the form $(\pi_1, 0) : \pi_1 \geq 0$.

LP Myth 13. *It is a good idea to convert free variables to the standard form by the expression: $x = u - v$, where u is the positive part and v is the negative part of x.*

Too often students (and new graduates) do this, perhaps thinking it is necessary due to the text they used. However, all solvers handle free variables directly.

For a simplex method, the conversion requires a change in basis whenever x needs to change sign. This is an unnecessary pivot, wasting time and space. Recognition of free variables allows the solver to put all free variables into the basis at the start (dealing with linear dependence, if that should be a problem). Once in the basis, a free variable cannot block an entrant, so it simply stays there. Some solvers also use the free variable to eliminate a row (and restore it after a solution is obtained). Thus, it is never a good idea to perform this conversion when using a simplex method.

For an interior method, this causes the optimality region to be unbounded (if it is not empty). Whatever the value of x^*, there is an infinite number of values of u^* and v^* that yield the same difference, $u^* - v^*$. During the iterations, it is not unusual for u and v to diverge, while maintaining a constant difference, and this divergence can cause numerical problems for the algorithm (especially for convergence detection).

LP Myth 14. *The standard simplex method does not select a dominated column to enter the basis.*

Consider LP in canonical form:

$$\max\ cx : x \geq 0,\ Ax \leq b.$$

A column, j, is *dominated* if there exists $k \neq j$ such that

$$c_k \geq c_j \text{ and } A_k \leq A_j.$$

Counterexample. Blair[12] provides the following:

$$\begin{array}{lrcrcrcrcl} \max & 5x_1 & + & 3x_2 & + & x_3 & + & x_4 & & \\ & x_1 & - & x_2 & + & 5x_3 & + & 3x_4 & \le & 10 \\ & 3x_1 & + & x_2 & + & x_3 & + & x_4 & \le & 40 \\ & -2x_1 & + & x_2 & - & 3x_3 & - & 3x_4 & \le & 10 \\ & & & & & x \ge 0. & & & & \end{array}$$

After adding slack variables to convert to standard form, the first simplex tableau is:

	Level	x_1	x_2	x_3	x_4	s_1	s_2	s_3
←	10	1	−1	5	3	1	0	0
	40	3	1	1	1	0	1	0
	10	−2	1	3	3	0	0	1
	0	5	3	1	1	0	0	0
		↑						

The first pivot exchange is $s_1 \leftarrow x_1$:

	Level	x_1	x_2	x_3	x_4	s_1	s_2	s_3
	10	1	−1	5	3	1	0	0
←	10	0	4	−14	−8	−3	1	0
	30	0	−1	13	9	2	0	1
	50	0	8	−24	−14	−5	0	0
			↑					

Column 3 is dominated by column 4, but it enters the basis next:

Level	x_1	x_2	x_3	x_4	s_1	s_2	s_3
$12\frac{1}{2}$	1	0	$1\frac{1}{2}$	1	$\frac{1}{4}$	$\frac{1}{4}$	0
$2\frac{1}{2}$	0	1	$-3\frac{1}{2}$	−2	$-\frac{3}{4}$	$\frac{1}{4}$	0
$32\frac{1}{2}$	0	0	$9\frac{1}{2}$	7	$1\frac{1}{4}$	$\frac{1}{4}$	1
69	0	0	4	2	1	−2	0
			↑				

One way to look at Blair's example is that the dominance conditions are not generally preserved as the basis changes.

Another view is to drop the first two columns entirely and consider a 2-variable LP with an initial basis that is slack. The values of A do not affect the selection of the basis entrant. With equal costs, the first variable (x_3) is selected, which is dominated by the second (x_4).

Level	x_3	x_4	s_1	s_2	s_3
10	5	3	1	0	0
40	1	1	0	1	0
10	3	3	0	0	1
0	1	1	0	0	0
	↑				

LP Myth 15. *The affine scaling algorithm converges to an optimum extreme point.*

Counterexample. Mascarenhas[53] provides the following:

$$\begin{aligned} \min\ x_1 : x_1, x_2 \geq 0 \\ \alpha x_1 + \beta x_2 - x_3 \geq 0 \\ \beta x_1 + \alpha x_2 - x_3 \geq 0 \\ -x_1 - x_2 + x_3 \geq -1, \end{aligned}$$

where $\alpha = 0.39574487$ and $\beta = 0.91836049$. The optimal solution is at the extreme point $x^* = (0, 0, -1)$. The essence of the counterexample is Mascarenhas' proof that there exists a half-line such that starting there and using a step size of 0.999, causes all even iterates to be in the half-line, and they converge to zero.

LP Myth 16. *At optimality, $\pi^* b = cx^*$ — that is, the inner product of the optimal dual variables on the constraints and the right-hand side values equals the optimal primal objective value.*

While this is true for standard and canonical forms, it fails when primal bounds are handled directly. Consider the primal-dual LPs:

Primal	Dual
$\min\ cx : 0 \leq x \leq U,\ Ax \geq b.$	$\max\ \pi b - \mu U : \pi, \mu \geq 0,\ \pi A - \mu \leq c.$

At optimality, $cx^* = \pi^* b - \mu^* U$, so one must be careful to subtract $\mu^* U$ from $\pi^* b$ to obtain the correct equation.

Support for handling bounds directly, rather than including them in other constraints, is an example of how optimization software may use different conventions than in the theory. Such deviations from theory in the world of optimization software include reporting dual prices and/or reduced costs as the negative of their theoretically-correct values. One must check the manual or run a small test case to see how they are reported in any particular solver. (ANALYZE[33] reports theoretically-correct values, changing solver-values as needed.)

LP Myth 17. *Once the simplex method reaches an optimal vertex, it terminates.*

The fallacy is that the Basic Feasible Solution (BFS) reached must be both primal and dual optimal for the tableau to be terminal.

Counterexample. Gerard Sierksma provided the following (converted to standard form):

$$\begin{array}{ccccccc} \multicolumn{7}{c}{\max\ x_1 + x_2 : x, s \geq 0} \\ x_1 & & & + & s_1 & = & 1 \\ & + & x_2 & + & s_2 & = & 1 \\ x_1 & + & x_2 & - & s_3 & = & 2 \end{array}$$

The extreme point $(1,1)$ is optimal and corresponds to three BFSs:

basic	level	s_2	s_3
x_1	1	−1	−1
x_2	1	1	0
s_1	0	1	1
$-z$	2	0	1
			↑

basic	level	s_1	s_3
x_1	1	1	0
x_2	1	−1	−1
s_2	0	1	1
$-z$	2	0	1
			↑

basic	level	s_1	s_2
x_1	1	1	0
x_2	1	0	1
s_3	0	1	1
$-z$	2	−1	−1
	Terminal		

Only the third of these is both primal and dual optimal; the other two are not terminal. The reason is the myopic nature of rates, oblivious to the degeneracy:

Tableau 1

$$\Delta x_1 = \Delta s_3,\quad \Delta x_2 = 0,\quad \Delta s_1 = -\Delta s_3,\quad \Delta z = \Delta s_3$$

Tableau 2

$$\Delta x_1 = 0,\quad \Delta x_2 = \Delta s_3,\quad \Delta s_2 = -\Delta s_3,\quad \Delta z = \Delta s_3$$

Tableau 3

$$\Delta x_1 = -\Delta s_1,\quad \Delta x_2 = -\Delta s_2,\quad \Delta s_3 = -\Delta s_1 - \Delta s_2,\quad \Delta z = -\Delta s_1 - \Delta s_2$$

Tableau 1 sees a rate of change in the objective value as +1 per unit of increase in s_3 (keeping $s_2 = 0$). The linear equations show that the net rate of change in the objective value (z) is +1, which is its reduced cost. Similarly, tableau 2 sees a rate of change in the objective value as +1 per unit of increase in s_3 (keeping $s_1 = 0$). The linear equations show that the net rate of change in the objective value (z) is +1, which is its reduced cost. The third tableau has s_3 in the basis, so it responds to changes in either of the first two slack variables. Any increase in one slack value causes a decreases in its corresponding variable while keeping the other primary variable at 1 — for example,

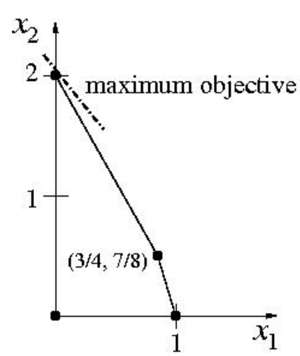
$$\Delta s_1 > 0 \Rightarrow \Delta x_1 = -\Delta s_1 < 0 \text{ and } \Delta x_2 = 0.$$

(The value of s_3 also decreases at the same rate, which does not affect the objective value.) The net effect is that the objective value decreases at that same unit rate, as indicated by the reduced cost. The same analysis applies to increasing s_2.

LP Myth 18. *In the absence of degeneracy, the standard simplex method does not repeat a basis exchange.*

Saaty[68] presented this conjecture with some supporting intuition. In the absence of degeneracy, this has a unique choice of departing variable for the exchange. However, Goldman and Kleinman[31] found the following:

Counterexample. This is a special case of the family of counterexamples in [31]:



$$\max\ 3x_1 + 2x_2 :$$
$$7x_1 + 2x_2 \le 7$$
$$3x_1 + 2x_2 \le 4$$
$$x_1, x_2 \ge 0.$$

Adding slack variables $s = (s_1, s_2)$, and starting at $x = (0, 0)$, the standard simplex iterations are:

Iteration	Vertex	Basic Variables	Basis Exchange
0	$(0, 0)$	s_1, s_2	$s_1 \leftarrow x_1$
1	$(1, 0)$	x_1, s_2	$s_2 \leftarrow x_2$
2	$(3/4, 7/8)$	x_1, x_2	$x_1 \leftarrow s_1$
3	$(0, 2)$	s_1, x_2	

LP Myth 19. *The standard simplex method does not revisit a basic feasible solution (that is, cycle) as it pivots to an optimum.*

Hoffman[43] gave the first example of cycling in the standard simplex method, which has 11 variables and 3 equations.

Counterexample. The following is due to Beale[5], with only 7 variables and 3 equations.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
$(1/4)$	-60	$-1/25$	9	1			0
$1/2$	-90	$-1/50$	3		1		0
		1				1	1
$-3/4$	150	$-1/50$	6	•	•	•	0
↑							
1	-240	$-4/25$	36	4			0
	(30)	$3/50$	-15	-2	1		0
		1				1	1
•	-30	$-7/50$	33	3	•	•	0
	↑						
1		$(8/25)$	-84	-12	8		0
	1	$1/500$	$-1/2$	$-1/15$	$1/30$		0
		1				1	1
•	•	$-2/25$	18	1	1	•	0
		↑					
$25/8$		1	$-525/2$	$-75/2$	28		0
$-1/160$	1		$(1/40)$	$1/120$	$-1/60$		0
$-25/8$			$525/2$	$75/2$	-25	1	1
$1/4$	•	•	-3	-2	3	•	0
			↑				

$-{}^{125}/_{2}$	10500	1		(**50**)	−150		0
$-{}^{1}/_{4}$	40		1	$^{1}/_{3}$	$-{}^{2}/_{3}$		0
$-{}^{125}/_{2}$	−10500			−50	150	1	1
$-{}^{1}/_{2}$	120	•	•	−1	1	•	0
				↑			

$-{}^{5}/_{4}$	210	$^{1}/_{50}$		1	−3		0
$^{1}/_{6}$	−30	$-{}^{1}/_{150}$	1		($^{1}/_{3}$)		0
		1				1	1
$-{}^{7}/_{4}$	330	$^{1}/_{50}$	•	•	−2	•	0
					↑		

Next tableau is same as first.

Hall and McKinnon[41] established the following form for a class of cycling examples with the same dimensions as Beale's example — four variables, three inequality constraints, one of which is just for bounding:

$$\max\ cx : x \geq 0,\ A^1 x + A^2 y \leq 0,\ x_1 + x_2 \leq 1,$$

where $c = (a, b)$ such that $a > 0 > b$, and A^1, A^2 and are 2×2 blocks such that $A^i_{11} + A^i_{22} = A^i_{21}A^i_{12} - A^i_{11}A^i_{22} = -1$ for $i = 1, 2$. In particular, they provide the following:

Counterexample.

$$\begin{array}{rcrcrcrl} \max\ 2.3\,x_1 & + & 2.15\,x_2 & - & 13.55\,x_3 & - & 0.4\,x_4 & : x \geq 0 \\ 0.4\,x_1 & + & 0.2\ \ x_2 & - & 1.4\ \ x_3 & - & 0.2\,x_4 & \leq 0 \\ -7.8\,x_1 & - & 1.4\ \ x_2 & + & 7.8\ \ x_3 & + & 0.4\,x_4 & \leq 0 \\ x_1 & + & x_2 & & & & & \leq 1. \end{array}$$

The optimal solution is $(0, 1, 0, 1)$.

Using the standard max reduced cost for entry, Hall and McKinnon use the largest pivot value to select the variable to leave the basis (among those with minimum ratio). Starting with the basis of surplus variables, $\{x_5, x_6, x_7\}$, the example cycles after six iterations. An important difference with Beale's example is that Hall and McKinnon establish a *family* of smallest examples, for which the above is one instance.

Hall and McKinnon also provide a library of test problems at http:\www.maths.ed.ac.uk/hall/PublicLP/. The above example is called HAMCK26E. The library also includes examples of a related phenomenon, called *stalling*, where the objective remains constant for a large number of iterations.

Also see Gass and Vinjamuri[27] for more cycling examples.

LP Myth 20. *A simplex method using steepest-edge column selection does not cycle.*

Counterexample. Using the same construction approach as in LP Myth 19, Hall and McKinnon[41] provide the following:

$$
\begin{array}{rcrcrcrl}
\max\ x_1 & + & 1.75\,x_2 & - & 12.25\,x_3 & - & 0.5\,x_4 & : x \geq 0 \\
0.4\,x_1 & + & 0.2\ \ x_2 & - & 1.4\ \ x_3 & - & 0.2\,x_4 & \leq 0 \\
-7.8\,x_1 & - & 1.4\ \ x_2 & + & 7.8\ \ x_3 & + & 0.4\,x_4 & \leq 1 \\
 & - & 20\ \ x_2 & + & 156\ \ x_3 & + & 8\ \ x_4 & \leq 0.
\end{array}
$$

Here are the tableaux that form the 6-cycle, where the last row in each tableau is the reduced cost divided by the Euclidean norm of the tableau column vector. (This is the initial rate of change in the objective value with respect to change in total distance. Further, it is scale-free and accounts for the geometry of the basis in the sense that $T_j = B^{-1}A_j$. See Greenberg and Kalan[37] for how this measure can be computed without solving $B\,T_j = A_j$ explicitly.) The steepest-edge rule chooses the maximum of these to enter the basis. (The departing variable remains chosen by largest pivot.)

x_1	x_2	x_3	x_4	x_5	x_6	x_7	*RHS*
(**0.4**)	0.2	−1.4	−0.2	1			0
−7.8	−1.4	7.8	0.4		1		0
	−20	156	8.0			1	1
1	1.75	−12.25	−0.5	•	•	•	0
0.128	0.09	−0.08	−0.06				0

1	0.5	−3.5	−0.5	2.5			0
	(**2.5**)	−19.5	−3.5	19.5	1		0
	−20	156	8	0		1	1
•	1.25	−8.75	0	−2.5	•	•	0
	0.06	−0.06	0	−0.13			0

1		(**0.4**)	0.2	−1.4	−0.2		0
	1	−7.8	−1.4	7.8	0.4		0
		0	−20	156	8	1	1
•	•	1	1.75	−12.25	−0.5	•	0
		0.13	0.09	−0.08	−0.06		0

19.5	1		(**2.5**)	−19.5	−3.5		0
2.5		1	0.5	−3.5	−0.5		0
0			−20	156	8	1	1
−2.5	•	•	1.25	−8.75	0	•	0
−0.13			0.06	−0.06	0		0

−1.4	−0.2	1		(**0.4**)	0.2		0
7.8	0.4		1	−7.8	−1.4		0
156	8			0	−20	1	1
−12.25	−0.5	•	•	1	1.75	•	0
−0.08	−0.06			0.13	0.09		0

−19.5	−3.5	19.5	1		(**2.5**)		0
−3.5	−0.5	2.5		1	0.5		0
156	8	0			−20	1	1
−8.75	0	−2.5	•	•	1.25	•	0
−0.06	0	−0.13			0.06		0

The next pivot exchange is $x_4 \leftarrow x_6$, which returns to the initial tableau.

The odd iterates have two candidates to enter the basis (that is, two reduced costs are positive). The one with greatest steepest-edge is opposite the one with greatest reduced cost. Then, there is only one positive entry in the column (0.4), which dictates the variable to leave the basis. The even iterates have only one candidate to enter the basis but two candidates to leave. The greatest pivot element is 2.5 (vs. 0.5).

LP Myth 21. *A simplex method does not cycle for an assignment problem.*

"A simplex method" is taken to mean any sequence of (adjacent) basic feasible solutions that enters a basic variable with negative reduced cost. This need not be the standard simplex method, which selects one with the most negative reduced cost.

Counterexample. Gassner[28] provides a 4×4 with costs:

$$c = \begin{bmatrix} 3 & 5 & 5 & 11 \\ 9 & 7 & 9 & 15 \\ 7 & 7 & 11 & 13 \\ 13 & 13 & 13 & 17 \end{bmatrix}.$$

Begin with the diagonal assignment: $x_{11} = x_{22} = x_{33} = x_{44} = 1$. Let the additional 3 basic (degenerate) variables be x_{12}, x_{23}, and x_{34}. Here is the initial (abbreviated) tableau:

			Nonbasic								
	Basic	level	x_{13}	x_{14}	x_{21}	x_{24}	x_{31}	x_{32}	x_{41}	x_{42}	x_{43}
	x_{11}	1	0	0	1	0	1	0	1	0	0
	x_{22}	1	−1	−1	1	0	1	1	1	1	0
	x_{33}	1	0	−1	0	−1	1	1	1	1	1
	x_{44}	1	0	0	0	0	0	0	1	1	1
	x_{12}	0	1	1	−1	0	−1	0	−1	0	0
←	x_{23}	0	**1**	1	0	1	−1	−1	−1	−1	0
	x_{34}	0	0	1	0	1	0	0	−1	−1	−1
		38	−2	2	4	4	0	−2	2	0	−2
			↑								

There are three candidates for entering the basis; select x_{13}. Then, there are two candidates to leave the basis; select x_{23}. The pivot results in the following tableau:

	Basic	level	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{41}	x_{42}	x_{43}
			Nonbasic								
	x_{11}	1	0	1	0	0	1	0	1	0	0
	x_{22}	1	0	1	1	1	0	0	0	0	0
	x_{33}	1	−1	0	0	−1	1	1	1	1	1
	x_{44}	1	0	0	0	0	0	0	1	1	1
←	x_{12}	0	0	−1	−1	−1	0	1	0	**1**	0
	x_{13}	0	1	0	1	1	−1	−1	−1	−1	0
	x_{34}	0	1	0	0	1	0	0	−1	−1	−1
		38	4	4	2	6	−2	−4	0	−2	−2
										↑	

The next entering variable is x_{42}, which has reduced cost $= -2$ (not the most negative). In each of the subsequent tableaux, Gassner selects an entrant with reduced cost $= -2$, although some have a reduced cost $= -4$, which would be selected by the standard simplex method.

	Basic	level	x_{12}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{41}	x_{43}
			Nonbasic								
	x_{11}	1	0	0	1	0	0	1	0	1	0
	x_{22}	1	0	0	1	1	1	0	0	0	0
	x_{33}	1	−1	−1	1	1	0	1	0	1	1
	x_{44}	1	−1	0	1	1	1	0	−1	1	1
	x_{42}	0	1	0	−1	−1	−1	0	1	0	0
	x_{13}	0	1	1	−1	0	0	−1	0	−1	0
←	x_{34}	0	1	1	−1	−1	0	0	**1**	−1	−1
		38	2	4	2	0	4	−2	−2	0	−2
									↑		

	Basic	level	x_{12}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{34}	x_{41}	x_{43}
			Nonbasic								
	x_{11}	1	0	0	1	0	0	1	0	1	0
	x_{22}	1	0	0	1	1	1	0	0	0	0
	x_{33}	1	−1	−1	1	1	0	1	0	1	1
	x_{44}	1	0	1	0	0	1	0	1	0	0
←	x_{42}	0	0	−1	0	0	−1	0	−1	**1**	1
	x_{13}	0	1	1	−1	0	0	−1	0	−1	0
	x_{32}	0	1	1	−1	−1	0	0	1	−1	−1
		38	4	6	0	−2	4	−2	2	−2	−4
										↑	

	Basic	level	x_{12}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{34}	x_{42}	x_{43}
			Nonbasic								
	x_{11}	1	0	1	1	0	1	1	1	−1	−1
	x_{22}	1	0	0	1	1	1	0	0	0	0
	x_{33}	1	−1	0	1	1	1	1	1	−1	0
	x_{44}	1	0	1	0	0	1	0	1	0	0
	x_{41}	0	0	−1	0	0	−1	0	−1	1	1
←	x_{13}	0	1	0	−1	0	−1	−1	−1	1	**1**
	x_{32}	0	1	0	−1	−1	−1	0	0	1	0
		38	4	4	0	−2	2	−2	0	2	−2
											↑

	Basic	level	Nonbasic								
			x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{34}	x_{42}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	0	0	0	1	1	1	0	0	0
	x_{33}	1	−1	0	0	1	1	1	1	1	−1
	x_{44}	1	0	0	1	0	0	1	0	1	0
←	x_{41}	0	−1	−1	−1	**1**	0	0	1	0	0
	x_{43}	0	1	1	0	−1	0	−1	−1	−1	1
	x_{32}	0	1	0	0	−1	−1	−1	0	0	1
		38	6	2	4	−2	−2	0	−4	−2	4
						↑					

Standard simplex enters x_{31}

	Basic	level	Nonbasic								
			x_{12}	x_{13}	x_{14}	x_{23}	x_{24}	x_{31}	x_{34}	x_{41}	x_{42}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	1	1	1	1	1	−1	0	−1	0
	x_{33}	1	0	1	1	1	1	0	1	−1	−1
	x_{44}	1	0	0	1	0	1	0	1	0	0
	x_{21}	0	−1	−1	−1	0	0	1	0	1	0
	x_{43}	0	0	0	−1	0	−1	0	−1	1	1
←	x_{32}	0	0	−1	−1	−1	−1	**1**	0	1	1
		38	4	0	2	−2	0	−2	−2	2	4
								↑			

	Basic	level	Nonbasic								
			x_{12}	x_{13}	x_{14}	x_{23}	x_{24}	x_{32}	x_{34}	x_{41}	x_{42}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	1	0	0	0	0	1	0	0	1
	x_{33}	1	0	1	1	1	1	0	1	−1	−1
	x_{44}	1	0	0	1	0	1	0	1	0	0
	x_{21}	0	−1	0	0	1	1	−1	0	0	−1
	x_{43}	0	0	0	−1	0	−1	0	−1	1	1
←	x_{31}	0	0	−1	−1	−1	−1	1	0	1	1
		38	4	−2	0	−4	−2	2	−2	4	6
							↑				

	Basic	level	Nonbasic								
			x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{32}	x_{34}	x_{41}	x_{42}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	1	0	0	0	0	1	0	0	1
	x_{33}	1	1	1	1	−1	0	1	1	−1	0
	x_{44}	1	1	0	1	−1	−1	1	1	0	1
	x_{24}	0	−1	0	0	1	1	−1	0	0	−1
←	x_{43}	0	−1	0	−1	1	1	−1	−1	1	0
	x_{31}	0	−1	−1	−1	1	0	0	0	1	0
		38	2	−2	0	2	−2	0	−2	4	4
							↑				

	Basic	level	Nonbasic x_{12}	x_{13}	x_{14}	x_{21}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	1	0	0	0	1	0	0	1	0
	x_{33}	1	1	1	1	−1	1	1	−1	0	0
	x_{44}	1	0	0	0	0	0	0	1	1	1
←	x_{24}	0	0	0	1	0	0	1	−1	−1	−1
	x_{23}	0	−1	0	−1	1	−1	−1	1	0	1
	x_{31}	0	−1	−1	−1	1	0	0	1	0	0
		38	0	−2	−2	4	−2	−4	6	4	2
					↑						

	Basic	level	Nonbasic x_{12}	x_{13}	x_{21}	x_{24}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
	x_{11}	1	1	1	0	−1	0	−1	1	1	1
	x_{22}	1	1	0	0	0	1	0	0	1	0
	x_{33}	1	1	1	−1	−1	1	0	0	1	1
	x_{44}	1	0	0	0	0	0	0	1	1	1
	x_{14}	0	0	0	0	1	0	1	−1	−1	−1
	x_{23}	0	−1	0	1	1	−1	0	0	−1	0
←	x_{31}	0	−1	−1	1	1	0	1	0	−1	−1
		38	0	−2	4	2	−2	−2	4	2	0
								↑			

	Basic	level	Nonbasic x_{12}	x_{13}	x_{21}	x_{24}	x_{31}	x_{32}	x_{41}	x_{42}	x_{43}
	x_{11}	1	0	0	1	0	1	0	1	0	0
	x_{22}	1	1	0	0	0	0	1	0	1	0
	x_{33}	1	1	1	−1	−1	0	1	0	1	1
	x_{44}	1	0	0	0	0	0	0	1	1	1
←	x_{14}	0	1	1	−1	0	−1	0	−1	0	0
	x_{23}	0	−1	0	1	1	0	−1	0	−1	0
	x_{34}	0	−1	−1	1	1	1	0	0	−1	−1
		38	−2	−4	6	4	2	−2	4	0	−2
			↑								

The next pivot brings us back to the initial tableau, thus completing the cycle. (Also see Gass[26, Chap. 10].)

Gassner proved that a simplex method cannot cycle for $n < 4$, so the above is an example of a smallest assignment problem for which a simplex method cycles.

Opportunity Knocks

To my knowledge, there is no example of an assignment problem that cycles with the standard simplex method. You may want to construct one or prove that no such counterexample exists.

LP Myth 22. *When applying the simplex method to minimum-cost flows on a directed, generalized network, the strongly convergent pivot rule out-performs the lexicographic rule for selecting a departing variable from the basis.*

The *strongly convergent pivot rule* was introduced by Elam, Glover, and Klingman[22] for the LP model:

$$\min\ cx :\ Ax = b,\ 0 \le x \le U,$$

where A is the node-arc incidence matrix (with weights), and x is the arc flow. Orlin[58] proves it is equivalent to the lexicographic rule (though not at all obvious). He also cites related works.

LP Myth 23. *Suppose LP is solved and π_i is the dual price associated with the i^{th} constraint. Then, the same solution is obtained when removing the constraint and subtracting $\pi_i A_{i\bullet} x$ from the objective.*

The reason this incorrect is because other solutions might exist to the revised LP. This error has caused some to say that a tax is equivalent to a prohibition in the sense that the dual price can be used as a tax in an LP that adds the tax to the objective and removes the prohibition constraint.

Counterexample. min $x + 2y : 0 \le x, y \le 10,\ x + y = 1$. The solution is $(x^*, y^*) = (1, 0)$ with dual price, $\pi = 1$ for the last constraint. Then, the *tax equivalent* is:

$$\min y : 0 \le x, y \le 10.$$

The solutions are of the form $(x, 0)$, where x is arbitrary in $[0, 10]$. Using a simplex method, the solution obtained will be one of the extremes: $x = 0$ or $x = 10$, neither of which is the original solution. In fact, the basic solution $(10, 0)$ violates the original constraint.

A motivating application is the control of emissions of some pollutant. In an LP, there may be a prohibition constraint:

$$\max\ cx :\ x \ge 0,\ Ax = b,\ dx \le \delta,$$

where d_j is the rate of emission caused by activity j, and δ is the limit. The tax model has the form:

$$\max\ cx - \tau dx :\ x \ge 0,\ Ax = b,$$

where τ is the shadow price associated with the prohibition constraint (equal to an extreme dual-variable value). Although the prohibition solution is optimal in this tax model, there may be other optimal solutions that violate the limit.

Consider a numerical example for electricity generation by three sources: scrubbed coal, oil, and uranium. The variables are fuel purchases and generation. The prohibition is a limit on sulfur emissions (LSU) while satisfying electricity demand (DEL). The B-rows balance fuels.

	Purchase			*Generate*					*Dual*
	PCL	*POL*	*PUR*	*GSC*	*GOL*	*GUR*			*Price*
COST	18	15	20	0.9	0.6	0.4	$=$	min	
BCL	1			-1			$\geq$	0	18
BOL		1			-1		$\geq$	0	15
BUR			1			-1	$\geq$	0	20
DEL				0.3	0.3	0.4	$\geq$	10	67.5
LSU				0.2	0.6		$\leq$	6	-8.25
bound					25	10			
level	15	5	10	15	5	10			

The solution to this LP generates all the electricity it can from uranium, which is 4 units, and the remaining 6 units from the only combination of oil and scrubbed coal to satisfy both the demand and the sulfur limit: $GSC = 15$ and $GOL = 5$. The issue is whether the sulfur-limit constraint can be replaced by a tax on sulfur emissions.

The tax model adds 8.25 times the LSU coefficients to the objective:

$$COST + 8.25(0.2\,GSC + 0.6\,GOL).$$

The tax model and its two optimal solutions are:

	Purchase			*Generate*					*Dual*
	PCL	*POL*	*PUR*	*GSC*	*GOL*	*GUR*			*Price*
COST	18	15	20	2.55	5.55	0.4	$=$	min	
BCL	1			-1			$\geq$	0	18
BOL		1			-1		$\geq$	0	15
BUR			1			-1	$\geq$	0	20
DEL				0.3	0.3	0.4	$\geq$	10	67.5
bound					25	10			
level^1	20	0	10	20	0	10			
level^2	0	20	10	0	20	10			

The tax LP has alternative solutions with extremes that contain the original limit of 6 units of sulfur emissions. At one extreme (level^1), the company uses no oil; it generates the 6 units of remaining electricity (after nuclear generation) by scrubbed coal. This complies with the sulfur limit with slack: the amount of sulfur emitted is only 4 units. At the other extreme (level^2), the company uses no scrubbed coal. This violates the sulfur limit: the amount emitted is 12 units. (This is the solution to the original model without the sulfur limit constraint; the prohibition was specified to disallow this.)

Because the 'equivalent' tax model could result in a violation, the tax might be levied at slightly more than the dual price of \$8.25. In that case, however, the result is overly conservative, resulting in much less sulfur emission than was deemed necessary for good health while raising the cost above its minimum.

The problem is the bang-bang phenomenon with linear models: solutions respond to data changes by an all-or-nothing principle. This reflects the fact that constant rates of substitution cause trade-offs that are marginally beneficial to be globally beneficial; only a constraint can stop the negotiation.

LP Myth 24. *Let $z(t) = \min\{cx : x \geq 0,\ Ax = b + th\}$, where h is a (fixed) m-vector. Then, z is piece-wise linear, where the break-points occur wherever there must be a basis change.*

The fallacy is the last sentence. The reason that this is not correct is that not every change in basis implies the slope must change.

Counterexample. $\min x - y : x, y \geq 0,\ x - y = t$. Because $z(t) = t$ for all t, there is only one linearity interval (no breakpoints). However, for t positive, we must have x basic, and for t negative, we must have y basic. At $t = 0$ there are two optimal bases, and the basis must change as t varies in either of the two directions. Thus, although the basis must change (to be feasible), the point at which this occurs (namely, at $t = 0$) is not a breakpoint of z.

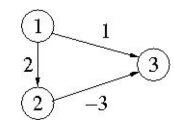
Note: the interior approach gives the correct answer (that is, the slope changes when the optimal partition changes). In the example, the optimal support has both $x > 0$ and $y > 0$, no matter what the value of t. Thus, the optimal partition does not change.

LP Myth 25. *Dijkstra's shortest path algorithm is correct, even with negative arc-costs, as long as there are no negative cycles.*

The usual counterexample to the correctness of Dijkstra's algorithm is with a negative cycle, for which there is no shortest path. What if there is no cycle?

Counterexample. Yen[78] provides the following:

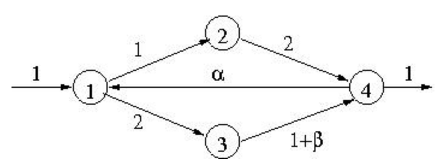
Dijkstra's algorithm obtains the path $(1, 3)$, whereas the shortest path from 1 to 3 is $(1, 2, 3)$.



LP Myth 26. *Alternative, integer-valued optima in a shortest path problem correspond to alternative shortest paths.*

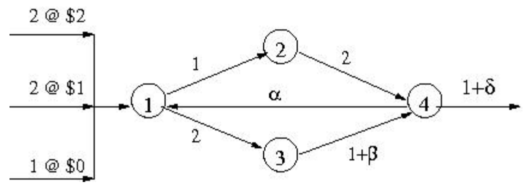
Counterexample. Consider the following network, where the LP is to ship one unit from node 1 to node 4 along the least costly route. An optimal solution is the shortest path, $1 \to 2 \to 4$, with a cost of \$3. There are two parameters, α, β, whose values can create alternative optima. We assume $\alpha \geq -3$ to avoid a negative cycle, and we assume $\beta \geq 0$.

If $\beta = 0$, another shortest path is $1 \to 3 \to 4$. The two shortest paths correspond to two basic optima in the LP formulation, consistent with the myth. However, when $\alpha = -3$, we have a zero-cost cycle: $1 \to 2 \to 4 \to 1$. Any solution can be augmented by an arbitrary amount of flow around the cycle without changing the total cost.



The essence of the myth rings true — there are two *simple* paths corresponding to two basic optima. However, the alternative optima with positive flow around the cycle spoils the result being literally true. One must consider zero-cost cycles as a caveat in how the statement is worded. The issue runs deeper in separating true alternative optima from *frivolous* ones. In particular, the dual always has alternative optima of the form $\pi' = \pi + K$, where π is any dual solution and $K > 0$. This is frivolous because they do not convey any true alternatives in the underlying economics.

To illustrate the difference between true versus frivolous alternative dual optima, consider a 3-tier supply, shown on the right. The dual price at node 4 depends on the demand parameter $\delta \geq 0$.



For $\delta = 0$, the initial supply step can be basic, giving a basic dual price of $\pi_4 = 3$ (and $\pi_1 = 0$). Another basic optimum has the initial supply step out of the basis at its upper bound of one unit, and the second supply step is in the basis (at zero level), giving $\pi_1 = 1$. The price at node 4 then becomes $\pi_4 = 4$. We have another interval of optimal prices at $\delta = 2$. Optimal dual prices are never unique, but when $\delta \neq 0, 2, 4$, alternatives are frivolous in that we could simply add any constant to all of them to obtain an alternative optimum. That notion of "alternative" does not correspond to a real alternative; it is an artifact of the modeling.

To summarize, we have the following cases (for $\alpha \geq -3$, $\beta \geq 0$, $\delta \geq 0$):

	Primal	Dual
unique	$\alpha > -3$, $\beta > 0$	never
frivolous	$\alpha = -3$, $\beta > 0$	$\delta \neq 0, 2, 4$
true alternatives	$\alpha > -3$, $\beta = 0$	$\delta = 0, 2, 4$

Opportunity Knocks

The distinction between true and frivolous alternative optima can be difficult to represent precisely. There is practical benefit to doing so. Besides ruling out some solutions as frivolous, one may want to know some *generating set* that brings an exponential number of alternatives down to a linear number in terms of more basic dimensions. For example, suppose an m-regional model has two alternatives within each region (but distributions among regions are completely determined by specifying one of the 2^m alternative optima). The total number of alternative optima is 2^m, but I suggest that there are circumstances where the distributions associated with combinations are not of much interest compared to knowing each of the $2m$ alternatives. Syntactically, a modeling language could allow some notion of *blocks* or *submodels* that make this practical.

LP Myth 27. *In a standard assignment problem, it is always optimal to assign the person-to-job that has the least cost.*

If this were true, we would have a greedy algorithm that recursively assigns the pair of least cost among unassigned pairs. As illustrated with the following counterexample, the optimality of an assignment depends upon *relative* costs. The one with least cost may eliminate an alternative savings that is greater when considering second-least costs.

Counterexample.

1	2
10	15

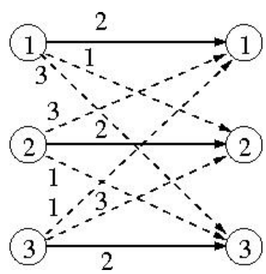
This is a 2×2 problem, and the issue is whether to assign Person 1 to Job 1 since that is the least cost.

If we assign Person 1 to Job 1, that cost is only 1, but we must then assign Person 2 to Job 2. That yields a total cost of 16. The optimal assignment is to assign Person 1 to Job 2 and Person 2 to Job 1, for a total cost of 12.

LP Myth 28. *Given an assignment problem with a non-optimal (but feasible) assignment, its cost can be reduced by swapping some pair of assignments.*

The following counterexample is adapted from Bertsekas[10].

Counterexample. There are 3 people to be assigned to 3 jobs. The current assignment is shown below with the solid arcs, having total cost = 6.



Numbers next to arcs are costs.

Here are the possible pair-wise swaps:

Old	New	Δcost
{1-1, 2-2}	{1-2, 2-1}	0
{1-1, 3-3}	{1-3, 3-1}	0
{2-2, 3-3}	{2-3, 3-2}	0

Every pair of swaps leaves the cost unchanged, but an optimal assignment is {1-2, 2-3, 3-1}, having total cost = 3.

LP Myth 29. *A transportation problem with unique shipping costs has a uniquely optimal shipment.*

Counterexample. Rubin and Wagner[66] pointed this out after noticing that managers apply this myth in practice. They provided the following:

	Supplier 1	Supplier 2	Demand
Market 1	55 0 10	5 10 0	10
Market 2	65 5 5	15 10 10	15
Market 3	75 10 0	25 0 10	10
Supply	20	20	

The upper number in each cell is the unit shipping cost. For example, each unit shipped from Supplier 1 to Market 1 is \$55. The lower-left number is the shipment in one optimal solution, and the lower-right number is the shipment in another optimal solution.

Note that the unit costs are all different, yet there are alternative optimal shipments. (The minimum total cost is \$1,275.)

LP Myth 30. *The optimal dual price of a demand constraint equals the increase in the minimum total cost if that demand is increased one unit.*

This fails if the solution is not at a *compatible basis*[36] (in the presence of primal degeneracy).

Counterexample. The following is taken from Rubin and Wagner[66].

	Supplier 1	Supplier 2	Demand	Price
Market 1	55 10	10 $0^{\ddagger}$	10	$55^{\dagger}$,$55^{\ddagger}$
Market 2	65 $0^{\dagger}$	15 10	10	$65^{\dagger}$,$60^{\ddagger}$
Market 3	80 0	25 10	10	$75^{\dagger}$,$70^{\ddagger}$
Supply	20	20		†Basis 1
Price	0, 0	50, 45		‡Basis 2

The cell values are unit costs and the (unique) optimal shipment levels. Two (basic) dual prices are shown.

If Market 2 demand increases, the first basis is compatible, and the change in the minimum total cost is indeed \$65. This can be achieved by sending one unit from Supplier 1 (which has excess). The basis is compatible with this change because the shipment level, x_{12}, can increase from its degenerate basic value, 0. On the other hand, if the solver obtains Basis 2, the \$60 dual price understates the increase in minimum total cost.

However, if we want to know the rate of savings from decreasing the demand in Market 2, we obtain the minimum optimal dual price (among the alternative optima) of the demand constraint. It is given by Basis 2 by letting the basic shipment level, x_{21}, increase by 1, balanced by decreasing x_{11} and x_{22} to 9.

The importance of using the wrong dual price for a marginal demand change is that the computed change in the minimum total cost may not be correct. One must have the maximum dual price to compute the effect of a demand increase, and one must have the minimum dual price to compute the effect of a demand decrease. (More details are in [34].)

For non-network LPs the myth can fail by having the correct slope (that is, $\partial f^*(b)/\partial b_i = \pi_i$), but the slope changes at $\Delta b_i < 1$, so the effect of a full unit change cannot be measured precisely with the shadow price.

LP Myth 31. *An increase in a demand requirement (with concomitant increase in supply) increases the minimum total cost.*

This is called the "more-for-less paradox." The following transportation problem is from Charnes and Klingman[20] (also see [74]).

Counterexample. There are 3 suppliers, with supplies shown in the last column, and 4 destinations, with demands shown in the last row. The cell values are optimal flows (blank is zero) and the boxed cell values in the NW corner are costs. The modified problem is to increase demand 1 and supply 2 by 9 units. The new optimal flow is shown on the right, and the total cost has decreased from \$152 to \$143, despite the increase in total flow, from 55 to 64.

				Supply
[1] 11	[6]	[3] 7	[5] 2	20
[7]	[3]	[1] 10	[6]	10
[9]	[4] 13	[5]	[4] 12	25
11	13	17	14	55

Original Problem
Min Cost = \$152

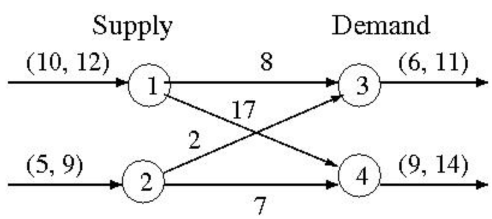
				Supply
[1] 20	[6]	[3]	[5]	20
[7]	[3] 2	[1] 17	[6]	19
[9]	[4] 11	[5]	[4] 14	25
20	13	17	14	64

Modified Problem
Min Cost = \$143

The underlying economics is that the greater flow can take advantage of low-cost activities. In the transportation example, shipments from supplier 1 to destination 1 have the lowest cost, but the original demand is not enough to ship all of the availability supply; supplier 1 must ship to other destinations. In the revised problem, supplier 1 can ship all of its units to destination 1, and the other destinations can meet their requirements from other suppliers less expensively.

Dĭneko, B. Klinz, and G. J. Woeginger[21] provide the following 3×3 transportation problem: supply: $s = (0, 1, 1)$, demand: $d = (1, 1, 0)$, and cost: $c_{ij} = 2^{|i-j|}$. The minimum total cost is 4. Increasing the first supply and last demand to $s' = d' = (1, 1, 1)$, the minimum total cost is only 3. They proceed to develop a key condition under which this paradox cannot occur: *there does not exist* i, j, p, q *such that* $c_{ij} + c_{pq} < c_{iq}$. If this condition does not hold, the more-for-less paradox may apply, depending on the data.

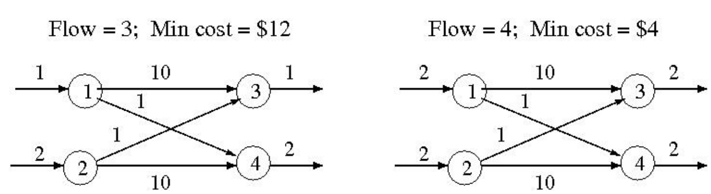
Glover[29, p. 37] gives another example:



The supplies and demands are required ranges, and the arc numbers are unit flow costs.

The minimum feasible flow is 15 units, and the least costly way to send that minimum is $x_{13} = 6$, $x_{14} = 4$, and $x_{24} = 5$, for a total cost of \$151. However, we can ship $x_{13} = 10$ and $x_{24} = 9$, for a total cost of \$143. We thus ship more for less!

Michael Hennebry provided the small example shown at the right.



Another form of the more-for-less paradox also arises with modeling requirement constraints as equations, rather than with inequalities. The problem need not be a network.

Counterexample. The following is a diet problem with 3 foods and 2 nutrient requirements, given by Arsham[4, 1]:

$$\begin{aligned} \min 40x_1 + 100x_2 + 150x_3 &: \\ x_1 + 2x_2 + 2x_3 &= 10 \\ 3x_1 + \ x_2 + 2x_3 &= 20 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

The optimal diet is $x = (6, 2, 0)$ with a minimum total cost of \$440. If we increase the second nutrient requirement to 30, the optimal diet becomes $x = (10, 0, 0)$ with a minimum total cost of \$400.

The diet problem usually has the canonical form:

$$\min\ cx : \ Ax \geq b,\ x \geq 0$$

(perhaps with bounds on the levels of foods, as $L \leq x \leq U$). To require $Ax = b$ does not give the flexibility of allowing over-satisfaction of nutrient requirements, even though it could be quite healthy to do so. This principle carries over to other situations, where modeling with equations is not the appropriate representation. (Also see Charnes, Duffuaa, and Ryan[17].)

Arsham[3] provides another vantage, with some focus on production problems. Ryan[67] addresses economies of scale and scope, using goal programming for multiple market structures.

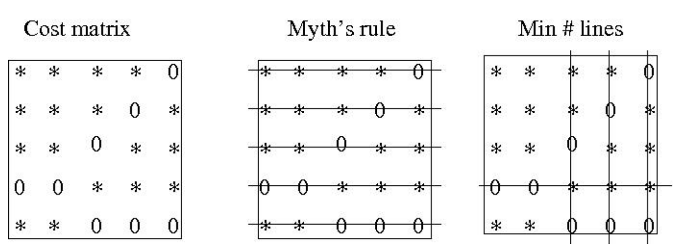
Opportunity Knocks

Does the more-for-less paradox extend to generalized networks? What about nonlinear costs?

LP Myth 32. *The line-drawing step of the Hungarian method for the assignment problem can be replaced by:* cover as many zeroes as possible with each line.

There have been several variants of the Hungarian algorithm — see Kuhn[52]. The original Hungarian method is to cover the zeroes with a minimum number of lines. This myth suggests another criterion, which turns out not to guarantee an optimal solution.

Counterexample. Storøy and Sørevik[77] provide the following 5×5 (* denotes non-zero):



The line-drawing rule starts by covering the three zeroes in row 5, followed by covering the two zeroes in row 4. Thus, a total of five lines must be drawn to cover all zeroes. Since this equals the number of rows (and columns), the Hungarian method's next step is to create an optimal solution from the covered zeroes. This is not possible.

The minimum number of lines is four, and the Hungarian method continues to subtract the minimum uncovered element (adding it to those covered by two lines).

LP Myth 33. *The Stepping Stone Method always produces an optimal distribution.*

This clever, early algorithm by Charnes and Cooper[16] specifically requires equality constraints (with total supply equal to total demand). It was extended to the general *node-bounded problem* by Charnes and Klingman[19]:

$$\begin{aligned} \min\ \textstyle\sum_{i,j} c_{ij}x_{ij} &: x \geq 0 \\ \underline{s}_i \leq \textstyle\sum_j x_{ij} &\leq \overline{s}_i,\ \forall i \\ \underline{d}_j \leq \textstyle\sum_i x_{ij} &\geq \overline{d}_j,\ \forall j, \end{aligned}$$

where $0 \leq \underline{s} \leq \overline{s}$ (supply out-flow bounds) and $0 \leq \underline{d} \leq \overline{d}$ (demand in-flow bounds).

Charnes, Glover, and Klingman[18] illustrated that the Stepping Stone Method need not terminate with an optimal solution if the constraints are the following special case of the node-bounded problem:

$$\min \textstyle\sum_{i,j} c_{ij}x_{ij} : x \geq 0,\ \sum_j x_{ij} \geq a_i,\ \sum_i x_{ij} \geq b_j.$$

Counterexample. Charnes, Glover, and Klingman gave a counterexample for each case:

$\sum_i a_i = \sum_j b_j$

1	6	3	5	20
7	3	1	6	10
8	3	4	3	25
11	13	17	14	

$\sum_i a_i < \sum_j b_j$

2	4	3
1	1	1
2	5	1
3	4	

$\sum_i a_i > \sum_j b_j$

1	1	2	5
6	5	1	6
2	7	1	

Each table gives the data in the form:

c_{11}	$\ldots$	c_{1n}	a_1
$\vdots$		$\vdots$	$\vdots$
c_{m1}	$\ldots$	c_{mn}	a_m
b_1	$\ldots$	b_m	

The solutions given by the Stepping Stone Method are the associated x_{ij}:

$\sum_{i,j} c_{ij}x_{ij} = 127$

11	0	9	0
0	2	8	0
0	11	0	14

$\sum_{i,j} c_{ij}x_{ij} = 13$

2	1
0	3
1	0

$\sum_{i,j} c_{ij}x_{ij} = 27$

2	3	0
0	4	2

Here are feasible solutions with lower costs:

$\sum_{i,j} c_{ij}x_{ij} = 118$

20	0	0	0
0	2	17	0
0	11	0	14

$\sum_{i,j} c_{ij}x_{ij} = 12$

3	0
0	4
1	0

$\sum_{i,j} c_{ij}x_{ij} = 15$

2	7	0
0	0	6

LP Myth 34. *The standard free-float formula for an activity in an activity-on-arc network equals the maximum leeway for scheduling the activity without affecting any the earliest start time of any later activity.*

The standard formula for the free float (FF) activity (i,j) is:

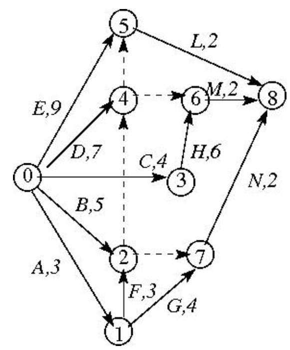
$$FF_{ij} = ES_j - EC_i \tag{LP.3}$$

where ES = earliest start time, EC = earliest completion time.

The statement is true in the absence of dummy arcs, but it can be an underestimate when all successors of some activity in the activity-on-arc network are dummy arcs.

Counterexample. Zhao and Tseng[81] provide the following (numbers on arcs are activity durations):

Activity	(i,j)	incorrect FF_{ij}	correct FF_{ij}
B	$(0,2)$	1	2
D	$(0,4)$	0	2
F	$(1,2)$	0	1



The incorrect values are from (LP.3). For example, $FF_{02} = ES_2 - (ES_0 + 5) = 6 - (0+5) = 1$. The maximum leeway, however, is 2. If we delay starting activity B by 2 time units, that will delay reaching node 2 by 2 time units. But since all arcs out of node 2 are dummy arcs, no activity is immediately affected. Instead, the float limit of 2 comes from tracing the paths out of node 2. Path $2 \to 7 \to 8$ gives a limit of 2 time units — that is, increasing the start of activity B by t delays the start of activity N by $t-2$ for $t \geq 2$. Similarly, the path $2 \to 4 \to 5 \to 8$ reveals that the start of activity L will be delayed by $t-9$, and the path $2 \to 4 \to 6 \to 8$ reveals that the start of activity M will be delayed by $t-10$. The binding limit is from the first path, which yields the correct float value of 2.

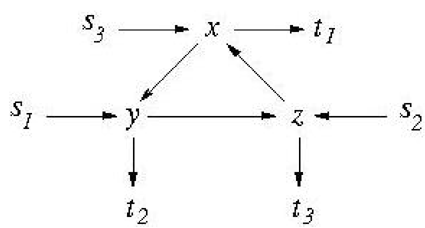
Similarly, applying (LP.3) to arc $(0,4)$, we have the incorrect value: $FF_{04} = ES_4 - (ES_0 + 7) = 7 - (0+7) = 0$. The correct value is obtained by tracing the paths $4 \to 5 \to 8$ and $4 \to 6 \to 8$. The former path yields a float limit of 2 time units (since activity L earliest start time $= ES_5 = 9$); the latter yields a float limit of 3 time unit (since activity M earliest start time $= ES_6 = 10$). The least of these limits is 2, which is the correct float value.

Zhao and Tseng developed this into an algorithm that follows dummy arcs from a rooted tree to obtain the correct free float values.

LP Myth 35. *The maximum flow of commodities through a network equals the capacity of a minimum disconnecting set.*

This is correct when there is only one commodity and for special cases of more than one. The failure for general numbers of commodities on networks of arbitrary topology was recognized in the 1950's — see Zullo[83] and her bibliography through 1995. The following example is from Ford and Fulkerson[24], and is further discussed by Bellmore, Greenberg, and Jarvis[7].

Counterexample. In the following network, all capacities are 3.



The max-flow is to send $^3/_2$ units along each path from its source to its sink, for a total of $^9/_2$ units. Here are the (unique) paths for each commodity: $s_1 \to y \to z \to x \to t_1$; $s_2 \to z \to x \to y \to t_2$; $s_3 \to x \to y \to z \to t_3$.

The minimum disconnecting is just to break the cycle, say with arc $(x,\ y)$, and the supply arc for the one remaining commodity, which is $(s_1,\ y)$, for a total of 6 units of capacity. There is no 1-arc disconnecting set, so this is a minimum, which implies max-flow < min-cut.

LP Myth 36. **new** *A maximum dynamic-flow on a network defined by a static network with stationary data is temporally-repetitive.* *next new* ▷

The maximum dynamic-flow problem is to find the maximum total flow that reaches the sink(s) within a specified number of time periods, N. The time-expanded network is defined by the given, static network, $G = [V, A]$, with specified sources, $S = \{s_1, \ldots, s_m\} \subset V$, and destinations, $D = \{d_1, \ldots, d_m\} \subset V$. The data are capacities, c_a, and traversal time, τ_a, for each $a \in A$. For each $v \in V$ define $N + 1$ nodes, $\{v(t)\}_{t=0}^{N}$. For each arc, $a \in A$, with endpoints (u, v), define the arcs $\{a(t)\}_{t=0}^{N-\tau_a}$ with (time-independent) data (c_a, τ_a) and endpoints $(u(t), v(t + \tau_a))$.

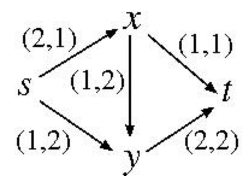
A flow is defined over a set of simple paths, each being from a source to a sink. Let the j^{th} path be $P_j = (a_{i_1}, \ldots, a_{i_{L_j}})$, where the tail of a_{i_1} is in S and the head of $a_{i_{L_j}}$ is in D. The path's total travel time is

$$\sigma_j = \sum_{k=1}^{L_j} \tau_{a_{i_k}}.$$

Let $f_j(t)$ be the flow along path j, starting at time t (restricted to j such that $\sigma_j \le N$), and let $\{P_j\}_{j=1}^{N_p}$ be the set of paths in G. To satisfy capacity constraints, we must sum flow across each arc at each time period. Let β_{aj} be the time that flow reaches arc $a \in P_j$ along path j, starting at time 0. Then, $\beta_{aj} + t$ is the time it reaches a for flow $f_j(t)$.

For example, the network on the right has one commodity. Dropping the commodity subscripts, there are three paths:

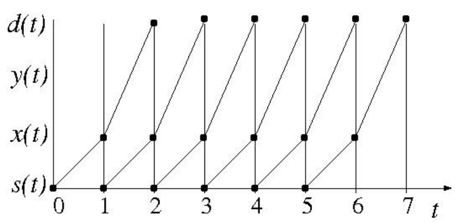
$$
\begin{aligned}
P_1 &= (s \to x \to d), & \sigma_1 &= 2;\\
P_2 &= (s \to y \to d), & \sigma_2 &= 4;\\
P_3 &= (s \to x \to y \to d), & \sigma_3 &= 5.
\end{aligned}
$$



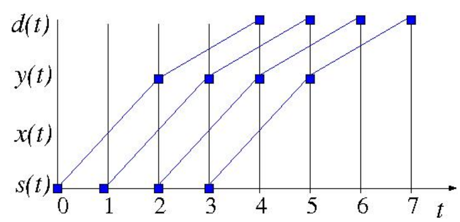
Arc values (c, τ) equal the capacity and traversal time.

The time-expanded network for $N = 7$ has 13 paths:

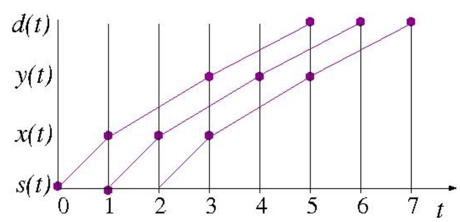
$$
\begin{aligned}
P_1(t) &= (s(t) \to x(t+1) \to d(t+2)), && \text{for t=0,...,5}\\
P_2(t) &= (s(t) \to y(t+2) \to d(t+4)), && \text{for t=0,...,3}\\
P_3(t) &= (s(t) \to x(t+1) \to y(t+3) \to d(t+5)), && \text{for t=0,...,2}
\end{aligned}
$$



Expansion of P_1



Expansion of P_2



Expansion of P_3

The arc-chain formulation of the *multi-commodity maximum dynamic-flow problem* is thus:

$$
\max \sum_{j=1}^{N_p} \sum_{t=0}^{N-\sigma_j} f_j(t) : \; x \geq 0,
$$

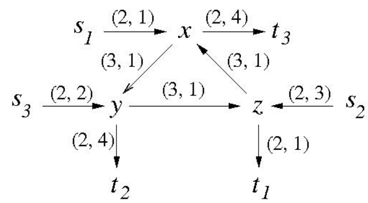
$$
\sum_{j:\, a \in P_j} f_j(t - \beta_{aj}) \leq c_a, \qquad t = \underline{t}_a, \dots, \bar{t}_a, \; a \in A,
$$

where the time range is $\underline{t}_a = \max_j \beta_{aj}$, $\bar{t}_a = N + \min_j \{\beta_{aj} - \sigma_j\}$.

Let f^* be a maximum (static) flow in G. A solution is *temporally repetitive* if $f_j(t) = f_j^*$ for $t = 0, 1, \dots, N - \sigma_j$ and $f_j(t) = 0$ otherwise. The myth's statement asserts that there is a maximum dynamic-flow solution that is temporally repetitive. Ford and Fulkerson[24] proved that the myth's statement is true for one commodity, but Bellmore and Vemuganti[8] provide the following

Counterexample. Consider three commodities ($m = 3$) and $N = 20$.

Path (P_j)	σ_j
$s_1 \to x \to y \to z \to d_1$	4
$s_2 \to z \to x \to y \to d_2$	9
$s_3 \to y \to z \to x \to d_3$	8



There are 42 paths in the dynamic network:

$$P_1(t) = s_1(t) \to x(t+1) \to y(t+2) \to z(t+3) \to d_1(t+4), \quad \text{for } t = 0, \dots, 16$$
$$P_2(t) = s_2(t) \to z(t+3) \to x(t+4) \to y(t+5) \to d_2(t+9), \quad \text{for } t = 0, \dots, 11$$
$$P_3(t) = s_3(t) \to y(t+2) \to z(t+3) \to x(t+4) \to d_3(t+8), \quad \text{for } t = 0, \dots, 12$$

The path-arc arrival times, β (shown on right), determine the potentially binding capacity constraints from the inner arcs:

$$(x, y): \quad f_1(t-1) + f_2(t-4) \le 3 \quad \text{for } t = 4, \dots, 15$$
$$(y, z): \quad f_1(t-2) + f_3(t-2) \le 3 \quad \text{for } t = 2, \dots, 14$$
$$(z, x): \quad f_2(t-3) + f_3(t-3) \le 3 \quad \text{for } t = 3, \dots, 14$$

	arc		
j	(x, y)	(y, z)	(z, x)
1	1	2	na
2	4	na	3
3	na	2	3

The maximum dynamic-flow solution is:

	path			
t	1	2	3	
0	2	2	2	
1	1	1	1	
2	2	2	2	
3	1	1	1	
4	2	2	2	
5	1	1	1	
6	2	1	2	
7	2	1	1	
8	2	2	1	
9	1	2	1	
10	1	2	2	
11	1	1	2	
12	2	0	2	
13	2	0	0	
14	2	0	0	
15	2	0	0	
16	2	0	0	
$\ge$17	0	0	0	
total	28	18	20	66

The total flow is 66. The maximum temporally-repeated flow is:

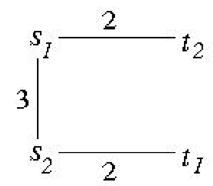
$$\begin{aligned} f_1(t) &= 2 \quad \text{for } t = 0, \dots, 16 \\ f_2(t) &= 1 \quad \text{for } t = 0, \dots, 11 \\ f_3(t) &= 1 \quad \text{for } t = 0, \dots, 12 \\ f_j(t) &= 0 \quad \text{otherwise.} \end{aligned}$$

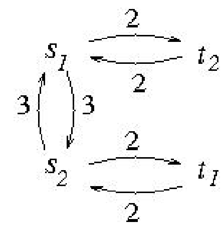
The value of this flow is 59. (Bellmore and Vemuganti give the maximum temporally-repeated flow value as 63, but I cannot see it.)

LP Myth 37. *Undirected arcs can be replaced by a pair of oppositely oriented arcs, and there is no loss in generality in obtaining a max-flow or a min-cut.*

This is true for a single-commodity network[24], but it generally fails for multi-commodity networks. The following is given by Bellmore, Greenberg, and Jarvis[7].

Counterexample. In the following network, capacities are shown next to each edge.





In the undirected graph, the max-flow is only 3, sending $^3/_2$ units of each commodity (the min-cut is also 3). After the replacement of each edge with opposite arcs, the max-flow becomes 4 units (also the min-cut value).

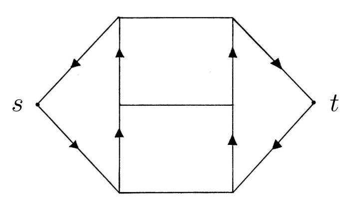
(Note: for a single commodity there is no advantage to sending flow across both arcs since they would cancel out in computing the total flow.)

LP Myth 38. *The maximum two-way flow of a commodity through a network equals its min-cut.*

In this variation of capacitated network flow, some links may be directed (arcs) and some may be undirected (edges). The flow on edges may be in either direction. Two-way flow from node s to node t, denoted $s \leftrightarrow t$, means two paths, one from s to t, denoted $s \to t$, and one from t to s, denoted $t \to s$. A two-way flow is a pair of paths, one in each direction, and the value of the flow is the minimum of all capacities of the links in the paths. A two-way cut for (s, t) is a set of links whose removal removes all paths in both directions, $s \to t$ and $t \to s$.

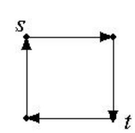
Rothschild and Whinston[63] provide the following:

Counterexample. In the following network (taken from [63]), all capacities are one. We have: two-way max-flow = $1 <$ two-way min-cut = 2.



Counterexample. T.C. Hu sent me the following:

All capacities are 1, so the two-way max flow = 1, and the two-way min cut = 2.



LP Background — Gomory-Hu Cut Trees

Consider an undirected graph with distinguished nodes s, t. Each edge e has a capacity, c_e, so there is a maximum flow from s to t, which equals the minimum cut that disconnects s from t. The *multi-terminal max-flow/min-cut problem* is to find the max-flow/min-cut between each

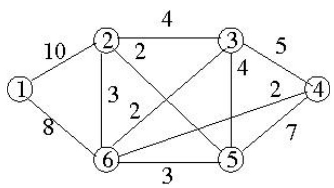
s, t in the graph. This could be done by solving each of the $\binom{n}{2}$ min-cut problems, but the Gomory-Hu algorithm[32] does this with only $n-1$ min-cut solutions.

Let V_{st} denote the max-flow/min-cut value between s and t. The Gomory-Hu algorithm produces a *cut-tree* (sometimes called a *Gomory-Hu tree*), whose nodes are those of the original graph and whose edges satisfy:

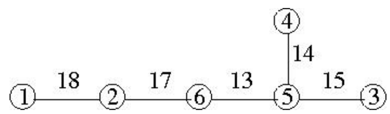
$$V_{st} = \min_{(i,j) \in P_{st}} V_{ij}, \qquad \text{(LP.4)}$$

where P_{st} = edges in s-t path. The Gomory-Hu algorithm computes the $n-1$ cuts, from which (LP.4) yields all of the $\binom{n}{2}$ min-cut values in the original graph.

Example (taken from [32]):



Capacitated network



Gomory-Hu cut-tree

For example, $V_{14} = 13 = \min\{18, 17, 13, 14\}$. The cut set is $\{(2,3),\ (2,5)\,(6,3),\ (6,4),\ (6,5)\}$, with graph partition $= \{1, 2, 6 \,|\, 3, 4, 5\}$.

A cut-tree has two key properties:

1. Each max-flow/min-cut value in the original graph equals the minimum of the edge values along the unique path connecting them in the cut-tree (that is, equation (LP.4)).
2. Removal of any edge from the cut-tree partitions the original graph into two sets of nodes that comprise a cut set whose value equals the cut-tree edge value.

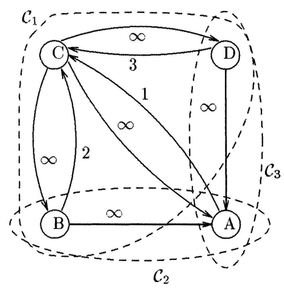
The first property gives the correct value of the min-cut, and hence the max-flow, and the second property gives the actual cut-set for any pair of nodes.

LP Myth 39. *Every connected network has a cut-tree.*

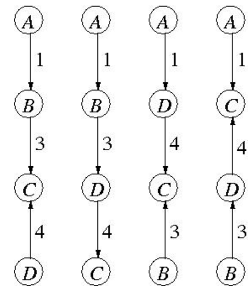
The classical algorithm by Gomory and Hu[32] constructively establishes the existence of a cut-tree for every connected, ***undirected*** graph. This was allegedly extended to directed graphs for the *symmetric case*: the min-cut between two nodes is the lesser of the min-cut from one to the other:

$$\mathcal{V}_{st} = \min\{V_{st}, V_{ts}\}$$

Counterexample. Benczúr[9] provides the following:



Only 3 (of 7) cut-sets are min



Flow trees do not encode min cut-set

Here are the min-cut values:

$$V = \begin{bmatrix} 0 & 1 & 1 & 1 \\ \infty & 0 & 3 & 3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & 4 & 0 \end{bmatrix} \Rightarrow \mathcal{V} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 3 & 3 \\ 1 & 3 & 0 & 4 \\ 1 & 3 & 4 & 0 \end{bmatrix}.$$

Since the min-cut value of A is 1 and all other min-cut values are greater than 1, any cut-tree must have A as a leaf. That leaves 9 trees to consider. Of these, 4 are shown with the edge values equal to the associated min-cut values: $V(\mathcal{C}_1 = (A \mid B, C, D)) = 1$, $V(\mathcal{C}_2 = (A, B \mid C, D)) = 3$, and $V(\mathcal{C}_3 = (A, D \mid B, C)) = 4$. Each tree violates the second property to be a cut-tree: the cut-set obtained upon breaking an edge of minimum value in the path between two nodes is not their min-cut.

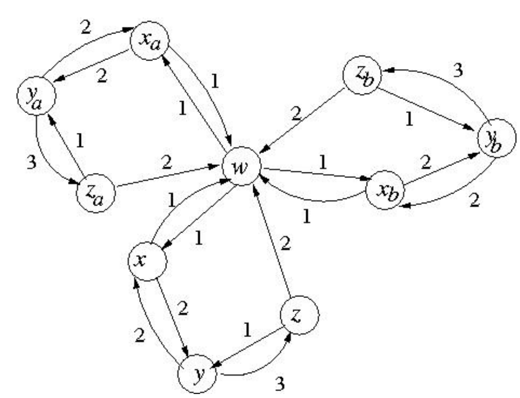
Going from left-to-right, the first two trees' violation is with (D, C). The cut-sets obtained from the edge is $(D \mid C, B, A)$ and $(C \mid D, B, A)$, respectively, but the min cut-set between D and C is $\mathcal{C}_3$. The third tree's violation is with (B, C). The cut-set obtained from the edge is $(B \mid C, D, A)$, but the min cut-set between B and C is $\mathcal{C}_2$. The fourth tree's violation is with (B, D). The cut-set obtained from the edge is $(B \mid D, C, A)$, but the min-cut is $\mathcal{C}_2$.

Now consider the other possible trees. Separating C and D makes their path value 3, which is not the value of their min-cut. The four shown are the only ones satisfying the first property of a cut-tree, showing the correct values of the min-cut using equation (LP.4). Since min-cut = max-flow, these are called *flow trees*.

Rizzi[62] provides the following with additional insight.

Counterexample.

In any tree there must be a leaf. Any cut-tree for this network must therefore have a star cut, $(v \mid \{u \neq v\})$. Suppose z is a leaf and its neighbor is y. The edge value of (z, y) is the star cut value $V(z \mid x, y, w, x_a, \dots) = 3$. If it were a cut-tree, this partition must be the min-cut between z and y. This is not the case, as the min-cut between z and y is $V(z, x \mid y, w, x_a, \dots) = 2$.

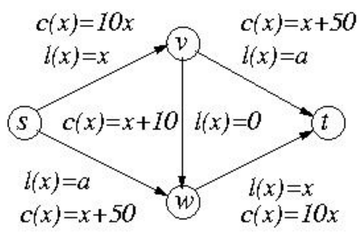


The key property identified by Rizzi is the notion of a *good pair*: (s, t) such that the star cut at t is a min-cut of (s, t), or there is no min-cut of (s, t) (that is, no path $s \to t$ or $t \to s$). Rizzi's network has no good pair, and that is why a cut-tree does not exist.

LP Myth 40. *Removing an arc in a network cannot decrease users' latency or cost.*

This is *Braess' Paradox*[13] applied to traffic flow.

Counterexample. The following is the classical example[14] — also see http://supernet.som.umass.edu/facts/braess.html.



$\ell(x)$ is the latency function of flow, x;
$c(x)$ is the cost.

The equilibrium flow is determined by each driver using the min-latency path. For n users, such that $n < a$, this is $s \to v \to w \to t$. (The users are indifferent among the three paths if $n = a$.) This results in each user experiencing $2n$ units of latency. If we remove arc (v, w), the drivers evenly split the use of the two different paths: $s \to v \to t$ and $s \to w \to t$. Their latencies thus reduce to $\frac{1}{2}n + a$ each.

Using the same graph, Steinberg and Zangwill[73] provide the rest of the counterexample, using the cost functions shown. With arc (v, w), 6 users evenly split each of the three paths from s to t, so that $x_{sv} = x_{wt} = 4$, while the other arc flows are 2. Thus, each user pays \$92, and the system cost is \$552. Without arc (v, w), 6 users split evenly between the two paths. Thus, each user pays \$83, and the system cost is \$498.

A great deal of literature has developed since Braess introduced his paradox in 1968. It has become a cornerstone of traffic equilibrium, as reflected in modern books by Nagurney[54, 57, 55] and Roughgarden[64]. Also see Nagurney[56] and Roughgarden[65] for focus on the Braess paradox and its relatives.

LP Myth 41. *Given strict improvement in the objective value, the standard simplex method does not visit an exponential number of vertices of the feasible polyhedron.*

The falsity of this was first demonstrated by Klee and Minty[51]. The so-called *Klee-Minty polytope* causes the standard simplex method to visit every extreme point, which grows exponentially with the number of variables.

Counterexample. The LP has n variables, n constraints, and 2^n extreme points. The elementary simplex method, starting at $x = 0$, goes through each of the extreme points before reaching the optimum solution at $(0, 0, \ldots, 0, 5^n)$.

$$\begin{array}{rcrcccrcrcr}
\max\ 2^{n-1}x_1 & + & 2^{n-2}x_2 & + & \dots & + & 2x_{n-1} & + & x_n : & & \\
x_1 & & & & & & & & & \le & 5 \\
4x_1 & + & x_2 & & & & & & & \le & 25 \\
8x_1 & + & 4x_2 & + & x_3 & & & & & \le & 125 \\
\vdots & & & & & & & & & & \vdots \\
2^n x_1 & + & 2^{n-1}x_2 & + & \dots & + & 4x_{n-1} & + & x_n & \le & 5^n \\
\end{array}$$
$$x \ge 0.$$

Another interesting example of exponential growth is due to Blair[11].

Jeroslow[47] was the first to present the construction of a class of examples for the best-gain basis entrance rule to visit an exponential number of vertices. (Also see Blair[11].)

LP Myth 42. *The worst-case time complexity of the simplex method is exponential and hence worse than the worst-case time complexity of the interior-point method.*

There are several things wrong with this statement. The first thing to note is that there is no "***the*** simplex method" and there is no "***the*** interior-point method." We know that both the standard simplex method and the best-gain rule have exponential time complexity (see LP Myth 41). However, the *Hirsch Conjecture*[82] leaves open the prospect for some simplex method to be linear in the numbers of variables and constraints. Also, there are interior-point methods that behave better than Karmarkar's original[49] in practice, but have no proof of polynomial complexity.

The second thing to note is the perturbation analysis by Spielman and Teng[72]. In fact, many coefficients are subjected to "random" perturbation due to rounding in their computations from other data.

Now suppose we are talking about the standard simplex method and one of the interior-point methods with a proof of polynomial complexity in the length of the data. Then, the third thing to consider is that the length of the data could be an exponential function of the number of variables. One example of this is a *Linear Programming Relaxation* (LPR) whose coefficients are computed from an aggregation algorithm[30]. The length of the coefficients (number of digits) can be an exponential function of the numbers of variables and constraints.

Thus, one must be careful in how to compare the (theoretical) worst-case time complexities of simplex versus interior methods.

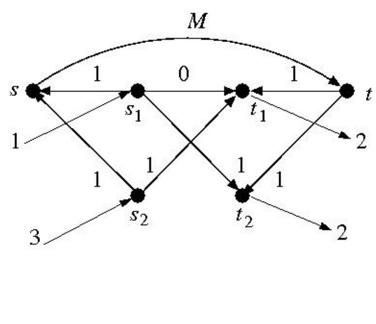
LP Myth 43. *The standard simplex method is polynomial time for min-cost network flow problems.*

Counterexample. Zadeh[79] provides the following. Consider a network, denoted N_n, with $2n+2$ nodes such that n nodes are sources, n are sinks, one node is a super-source, and one is a super-sink; arcs are (s,t), (s_1,t_1), (s_i,s), (t,t_i), $\{(s_i,t_j) : j \ne i\}$ for $i = 1,\dots,n$. The $2n^2+2$ arc capacities are infinite, but supplies and demands are forced flow values for each (s_i,s) and (t,t_i) from the external supplies (S) and demands (D) for $n > 1$:

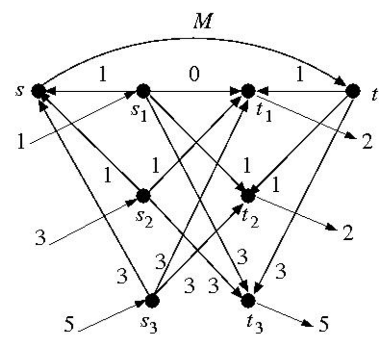
$$
\begin{aligned}
S_1 &= 1 \quad D_1 = 2\\
S_2 &= 3 \quad D_2 = 2\\
S_i &= D_i = 2^{i-1} + 2^{i-3} \;\text{ for } i = 3, \dots, n
\end{aligned}
$$

The arc costs are $c(s,t) = M$, $c(s_1,t_1) = 0$, $c(s_i,s) = c(t,t_i) = c(s_i,t_j) = 2^{i-1} - 1$ for $j < i$ and $c(s_i,t_j) = 2^i - 1$ for $j \geq i$ for $i = 1, \dots, n$. The value of M is sufficiently large to render the use of arc (s,t) prohibitive (that is, $x(s,t)$ is its minimum feasible value in every optimal solution).

Here are N_2 and N_3 — supplies are tailless arcs into s_i; demands are headless arcs out of t_i; and, arc numbers are costs.

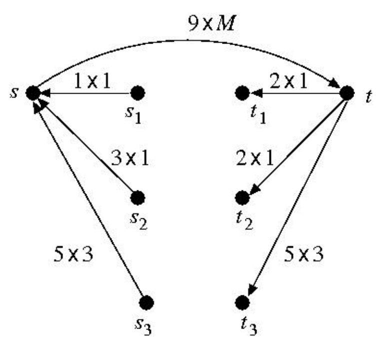


N_2

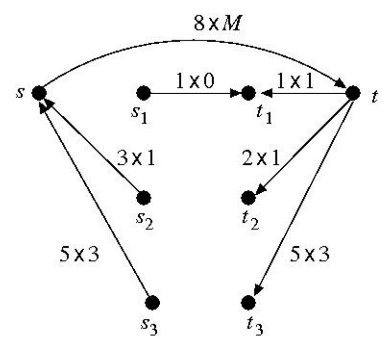


N_3

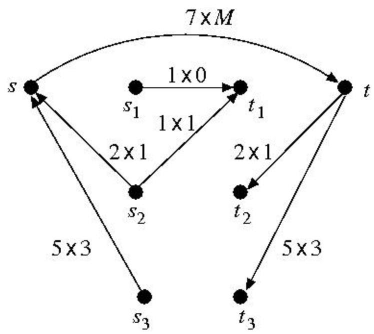
Let the initial feasible basis be the tree with arcs (s,t), $\{(s_i,s), (t,t_i) : i = 1, \dots, n\}$. The basic levels are the associated supplies and demands. The following shows the initial basis for N_3 and the new basis after one iteration. Arc numbers are flow×cost.



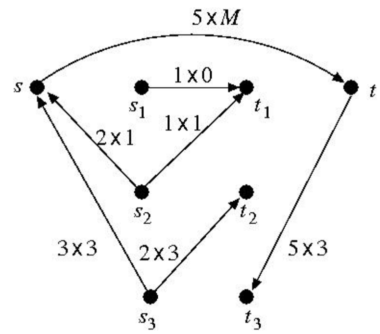
Initial basis for N_3
Total cost $= 38 + 9M$



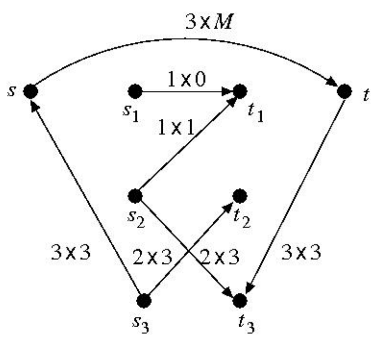
Basis after one iteration
Total cost $= 36 + 8M$



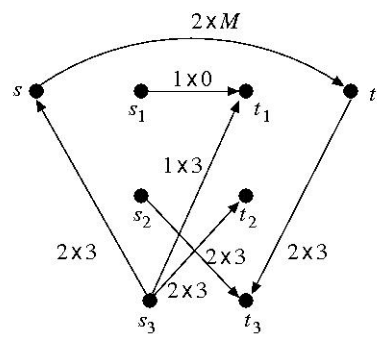
Basis after two iterations
Total cost $= 35 + 7M$



Basis after three iterations
Total cost $= 33 + 5M$

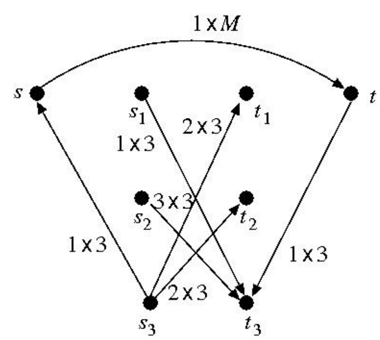


Basis after four iterations
Total cost $= 31 + 3M$



Basis after five iterations
Total cost $= 30 + 2M$

The optimal solution for N_3, shown to the right, has Total cost $= 30 + M$, which is reached in six iterations. (One unit of flow must flow across (s, t) to satisfy demand at t_3).



The choice of arc (s_1, t_1) to enter the basis is because its reduced cost, $-(M+1)$, yields the greatest rate of cost decrease. Every arc of the form (s_i, t_j) reduces a unit of flow across (s, t), so the most negative reduced cost is the one with the least cost. Thus, $c(s_1, t_1) = 0$ is the one. Arc (s_2, t_1), for example, has reduced cost $1 - (M + 1)$.

While each iteration selects the arc to enter the basis that has the greatest rate of decrease in flow across (s, t), the actual reduction is limited. For N_3, the reduction is either 1 or 2 each iteration. For $n \geq 3$, the initial flow across (s, t) is

$$\begin{aligned} \sum_{i=1}^n S_i = \sum_{i=1}^n D_i &= 4 + \sum_{i=3}^n \left(2^{i-1} + 2^{i-3}\right) \\ &= 4 + 5 \sum_{i=0}^{n-3} 2^i = 4 + 5(2^{n-2} - 1) = 5 \times 2^{n-2} - 1 \end{aligned}$$

Since we do not have arc (s_n, t_n) and the total supply from $s_1, \dots, s_{n-1}$ is less than D_n, the difference must go across (s, t). That is, the minimum value of $x(s, t)$ is

$$\begin{aligned} D_n - \textstyle\sum_{i=1}^{n-1} S_i &= 2^{n-1} + 2^{n-3} - 1 - 3 - \textstyle\sum_{i=3}^{n-1} \left(2^{i-1} + 2^{i-3}\right) \\ &= 2^{n-1} + 2^{n-3} - 4 - 5(2^{n-3} - 1) \\ &= 2^{n-1} + 2^{n-3} + 1 - 5 \times 2^{n-3} \\ &= 2^{n-1} + 1 - 4 \times 2^{n-3} = 1. \end{aligned}$$

I believe Zadeh's argument claims that the standard simplex method reduces the flow across (s, t) by no more than K each iteration. Since the initial feasible bases has $x(s, t) = O(2^n)$, the number of iterations for N_n to reach the optimal flow of $x(s, t) = 1$ is $O(2^n)$.

Opportunity Knocks

My interpretation of Zadeh's argument could be wrong, and I am unable to present a complete proof that the reduction of $x(s, t)$ in one iteration is limited by some constant, K. I have been unable to reach Zadeh to obtain clarification (and his approach is different). It would be useful to have this completed (or some other counterexample).

Zadeh analyzes other network problems and algorithms using this type of construction. He also provides more pathological examples in [80]. (See Orlin[59] for a polynomial-time simplex algorithm for min-cost network flows.)

LP Myth 44. *The c-diameter of a non-empty polytope of dimension d with f facets cannot exceed* $f - d$.

Let P denote the polytope (that is, bounded polyhedron), and let $V^*(P, c)$ denote the set of vertices that minimize a linear form, cx, on P. The *c-diameter* from a vertex $v \in P$ for a given linear form is defined as the maximum distance from v to $V^*(P, c)$. The *distance* is defined to be the minimum number of edges in a path joining v to $V^*(P, c)$ along which cx is non-increasing. (In terms of LP, the c-diameter is an upper bound on how many vertices the simplex method visits before reaching an optimal vertex.) Denote the c-diameter from v by $\Delta(v, c)$, and the myth asserts $\Delta(v, c) \le f - d$. This is known as the *monotonic bounded Hirsch conjecture*.

Counterexample. Todd[75] provides the following:

$$P = \{x \in \mathbb{R}^4_+ : Ax \le b\}, \text{ where } A = \begin{bmatrix} 7 & 4 & 1 & 0 \\ 4 & 7 & 0 & 1 \\ 43 & 53 & 2 & 5 \\ 53 & 43 & 5 & 2 \end{bmatrix}, \; b = \begin{pmatrix} 1 \\ 1 \\ 8 \\ 8 \end{pmatrix}.$$

This is a 4-dimensional polytope with 8 facets, so the myth asserts that the c-diameter cannot exceed 4 for any linear form. Let $c = (1, 1, 1, 1)$, so $V^*(P, c) = \{(0, 0, 0, 0)^\intercal\}$. Todd proves that all non-increasing paths from $v = \frac{1}{19}(1, 1, 8, 8)^\intercal$ to 0 have a distance of 5.

See Klee and Kleinschmidt[50] for an extensive survey of the Hirsch conjecture and related properties of polytopes. Also see Holt and Klee[45] for a counterexample to the strong d-step conjecture.

LP Myth 45. *Determining whether an LP has a degenerate basis has the same complexity as solving the LP.*

LP has a polynomial algorithm, but Chandrasekaran, Kabadi, and Murty[15] prove that the *degeneracy testing problem* is NP-complete. It is easily seen that degeneracy testing is in NP, so it remains to construct a polynomial reduction of an NP-complete problem to degeneracy testing. They use the NP-complete *subset sum problem*[25, SP13]:

$$\text{SS: Given } a_1, \dots, a_n, b \in \mathbb{Z}_+, \text{ find } x \in \{0,\, 1\}^n : \, ax = b.$$

Consider an $n \times 2$ transportation problem with supplies $s = a$ and demands $d = (b, \sum_{i=1}^n a_i - b)$. Then, checking whether SS has a solution is equivalent to checking if the usual algebraic representation of the transportation polytope is degenerate:

$$\sum_j x_{ij} = s_i, \; \sum_i x_{ij} = d_j, \; x \geq 0.$$

Chandrasekaran et al. note that the transportation problem is degenerate whenever there are subsets with total supply equal to total demand — that is, there exist non-empty, proper subsets I, J ($\emptyset \neq I \subset \{1, \dots, m\}$, $\emptyset \neq J \subset \{1, \dots, n\}$) such that $\sum_{i \in I} s_i = \sum_{j \in J} d_j$.

My thanks to Katta Murty for providing clarification and the following additional examples of NP-complete (or NP-hard) problems.

1. Find a BFS with the fewest number of positive variables.
2. Find a maximum-cardinality subset of minimally linearly dependent vectors.
3. Find a minimum-cardinality subset of linearly dependent vectors containing a given vector.
4. Find a singular principal submatrix of a square matrix.

LP Myth 46. *In employing successive bound reduction in a presolve, we can fix a variable when its bounds are within a small tolerance of each other.*

The myth is that we can pick a tolerance, say $\tau > 0$, such that if we infer $L \leq x \leq U$ and $U - L \leq \tau$, we can fix x to some value in the interval, such as the midpoint, $\frac{1}{2}(L + U)$. There are a few things wrong with this, as reported by Greenberg[33].

Counterexample. Consider $x \geq 0$ and

$$\begin{array}{rcrcl} \frac{1}{2}x_1 & + & x_2 & = & 1 \\ x_1 & + & x_2 & = & 2. \end{array}$$

This has the unique solution, $x = (2, 0)$, and it is this uniqueness that causes a problem with greater implications.

In successive bound reduction, the most elementary tests evaluate rows to see if just one row alone can tighten a bound on a variable. Initially, the bounds are the original ones: $L^0 = L = (0, 0)$ and $U^0 = U = (\infty, \infty)$. The first iteration results in the inference that

$x_1 \le 2$, from the first equation and the fact that $x_2 \ge 0$. It similarly produces an upper bound, $x_2 \le 1$, so $U^1 = (2, 1)$. Still in iteration 1, the second equation causes the inference, $x_1 \ge 1$, because we already have $x_2 \le 1$ when we get there. Thus, $L^1 = (1, 0)$.

At a general iteration, we will have inferred $L_1^k \le x_1 \le 2$ and $0 \le x_2 \le U_2^k$, where $L_1^k < 2$ and $U_2^k > 0$. At the end of iteration k, the inferred bounds are:

$$2 - \left(\tfrac{1}{2}\right)^k \le x_1 \le 2 \quad \text{and} \quad 0 \le x_2 \le \left(\tfrac{1}{2}\right)^k.$$

This converges to the unique solution, but it does not reach it finitely. If the iterations go far enough, the bounds become within the tolerance $\tau > 0$. At that point, suppose x is fixed to the interval's midpoint: $x = \frac{1}{2}(L^k + U^k)$.

To see a consequence of this, suppose that the presolve tests feasibility with another tolerance, μ. Let the constraints be of the form $Ax = b$. The rule is: *Declare infeasibility if, for some equation, i,*

$$y_i^{\max} = \max_{L^k \le x \le U^k} A_{i\bullet} x < b_i - \mu \quad \textit{or}$$
$$y_i^{\min} = \min_{L^k \le x \le U^k} A_{i\bullet} x > b_i + \mu.$$

In our example, when $k = \lceil -\log_2 \tau \rceil$, both variables are fixed:

$$x_1 = 2 - \left(\tfrac{1}{2}\right)^{k+1}, \; x_2 = \left(\tfrac{1}{2}\right)^{k+1}.$$

Equation 2 passes the feasibility test, but equation 1 has

$$y_1^{\max} = y_1^{\min} = 1 - \left(\tfrac{1}{2}\right)^{k+2} + \left(\tfrac{1}{2}\right)^{k+1} = 1 + \left(\tfrac{1}{2}\right)^{k+2}.$$

Thus, $y_1^{\min} = 1 + \left(\frac{1}{2}\right)^{k+2}$, so we declare infeasibility if $\left(\frac{1}{2}\right)^{k+2} > \mu$. Taking logs, this is equivalent to $-(k+2) > \log_2 \mu$. Replacing k, we have that a false infeasibility is declared if

$$-\lceil -\log_2 \tau \rceil - 2 > \log_2 \mu.$$

For example, if $\tau = 2^{-20}$, we declare a *false infeasibility* if $\mu < 2^{-22}$.

This example highlights two things:

1. Tolerances are related. The tolerance to fix a variable should not be substantially less than the infeasibility tolerance.
2. Fix a variable judiciously. When having inferred $x_j \in [L_j, U_j]$, such that $U_j - L_j$ is within tolerance of fixing x_j, do so in the following order of choice:
   (1) If L_j is an original bound, fix $x_j = L_j$;
   (2) If U_j is an original bound, fix $x_j = U_j$;
   (3) If $[L_j, U_j]$ contains an integer, p, fix $x_j = p$;
   (4) If all of the above fail, fix $x_j = \frac{1}{2}(L_j + U_j)$.

LP Myth 47. *A factored form of the basis contains less error for FTRAN after reinversion.*

The Forward Transformation (FTRAN) algorithm solves the forward system, $Bx = b$, by factoring B and updating it after each basis change. Consider the elementary product form: $B = E_1 E_2 \cdots E_k$, where each E_i is an elementary matrix.

Algorithm: Forward Transformation with PFI

Initialize. Set $x^0 = b$.
for $i = 1 : k$ **do**
  Solve $E_i x^i = x^{i-1}$
end for
Exit with x^k the (computed) solution to $Bx = b$.

During the pivoting process, k increases and there are more factors than needed. *Reinversion* is the process of restarting to obtain the minimum number of factors, which equals the number of variables in the basis (except slacks). One reason to reinvert is to "cleanup" the errors that accumulate, which affects the accuracy of solving $Bx_B = b$. (Another reason is to reduce the FTRAN time.)

The essence of the counterexample is cancelation of errors in the first factors that does not cancel in the reinverted factorization.

Counterexample. Consider the 2×3 system:

$$\begin{array}{ccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \end{array}$$

Pivoting x_1 on equation 1, then x_2 on equation 2 into the basis, then replacing x_1 with x_3 yields the following elementary factors:

$$E_1 = \begin{bmatrix} a_{11} & 0 \\ a_{21} & 1 \end{bmatrix};\ E_2 = \begin{bmatrix} 0 & a_{12}/a_{11} \\ 1 & a_{22} - a_{21}a_{12}/a_{11} \end{bmatrix}$$

$$E_3 = \begin{bmatrix} a_{13}/a_{11} - a_{22} - a_{12}a)21/a_{11})((a_{23} - a_{13}a)21/a_{11})/a_{12}/a_{11} & 0 \\ \big(a_{23} - a_{13}a_{21}/a_{11}\big)/(a_{12}/a_{11}) & 1 \end{bmatrix}.$$

Collecting computed values and substituting c with a new index whenever there is a new computation, we obtain:

$$E_1 = \begin{bmatrix} a_{11} & 0 \\ a_{21} & 1 \end{bmatrix};\ E_2 = \begin{bmatrix} 0 & c_1 \\ 1 & c_2 \end{bmatrix};\ E_3 = \begin{bmatrix} c_3 & 0 \\ c_4 & 1 \end{bmatrix}.$$

Then, executing FTRAN for b (to get basic levels):

$$x^1 = \begin{pmatrix} b_1/a_{11} \\ b_2 - (b_1/a_{11})a_{12} \end{pmatrix} = \begin{pmatrix} c_5 \\ c_6 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} x_1^1 - (x_2^1/c_2)c_1 \\ x_2^1/c_2 \end{pmatrix} = \begin{pmatrix} c_7 \\ c_8 \end{pmatrix}$$

$$x^3 = \begin{pmatrix} x_1^2/c_3 \\ x_2^2 - (x_1^2/c_3)c_4 \end{pmatrix} = \begin{pmatrix} c_9 \\ c_{10} \end{pmatrix}$$

After reinversion, the elementary matrices have the form:

$$E_1 = \begin{bmatrix} a_{13} & 0 \\ a_{23} & 1 \end{bmatrix}; \; E_2 = \begin{bmatrix} 0 & c_{11} \\ 1 & c_{12} \end{bmatrix}.$$

Now the FTRAN algorithm yields computed levels:

$$\widehat{B^{-1}}b = \begin{pmatrix} c_{13} \\ c_{14} \end{pmatrix}.$$

Suppose $\beta = B^{-1}b$, the true value of the levels. The issue is whether

$$\left|\left| \begin{pmatrix} \beta_1 - c_{13} \\ \beta_2 - c_{14} \end{pmatrix} \right|\right| = ||\beta - \zeta'|| < \left|\left| \begin{pmatrix} \beta_1 - c_9 \\ \beta_2 - c_{10} \end{pmatrix} \right|\right| = ||\beta - \zeta||,$$

where ζ the accumulated error before reinversion, and ζ' is the accumulated error after reinversion.

It is possible that $\zeta = 0$ while $\zeta' \neq 0$ — that is, that we obtain an error-free solution with the original factorization and reinversion introduces error. This can happen by error cancelation. However, even if $||\zeta'|| < ||\zeta||$, the computed levels could have less error, at least for some particular b. For example, let $\beta = (100, 100)^\intercal$, $\zeta = (2, 2)^\intercal$, and $\zeta' = (1, -1)^\intercal$. Then, $||\zeta|| > ||\zeta'||$, yet $||\beta - \zeta|| \approxeq 138.6 < 141.4 \approxeq ||\beta - \zeta'||$.

LP References

[1] H. Arsham. On the more-for-less paradoxical situations in linear programs: A parametric optimization approach. *Journal of Information & Optimization Sciences*, 17(3):485–500, 1996.

[2] H. Arsham. Tools for modeling validation process: The dark side of LP. Web note, Merrick School of Business, University of Baltimore, http://home.ubalt.edu/ntsbarsh/opre640A/partv.htm, 1996–2008.

[3] H. Arsham. Construction of the largest sensitivity region for general linear programs. *Applied Mathematics and Computation*, 189(2):1435–1447, 2007.

[4] H. Arsham and M. Oblak. Perturbation analysis of general LP models: A unified approach to sensitivity, parametric, tolerance, and more-for-less analysis. *Mathematical and Computer Modelling*, 13(8):79–102, 1990.

[5] E. M. L. Beale. Cycling in the dual simplex method. *Naval Research Logistics Quarterly*, 2(4), 1955.

[6] J. E. Beasley. *OR-Notes*. Brunel University, http://people.brunel.ac.uk/~mastjjb/jeb/or/sp.html, 1999.

[7] M. Bellmore, H. J. Greenberg, and J. J. Jarvis. Multi-commodity disconnecting sets. *Management Science*, 16(6):427–433, 1970.

[8] M. Bellmore and R. R. Vemuganti. On multi-commodity maximal dynamic flows. *Operations Research*, 21(1):10–21, 1973.

[9] A. A. Benczúr. Counterexamples for directed and node capacitated cut-trees. *SIAM Journal on Computing*, 24(3):505–510, 1995.

[10] D. P. Bertsekas. *Linear Network Optimization: Algorithms and Codes*. MIT Press, Cambridge, MA, 1991.

[11] C. Blair. Some linear programs requiring many pivots. Technical Report 867, College of Commerce and Business Administration, University of Illinois, Champaign, IL, 1982. Available at https://netfiles.uiuc.edu/c-blair/www/papers/papers.html.

[12] C. Blair. Random linear programs with many variables and few constraints. *Mathematical Programming*, 34(1):62–71, 1986.

[13] D. Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968. English translation in [14]

[14] D. Braess, A. Nagurney, and T. Wakolbinger. On a paradox of traffic planning. *Transportation Science*, 39(4):446–450, 2005. English translation of [13]

[15] R. Chandrasekaran, S. N. Kabadi, and K. G. Murty. Some NP-complete problems in linear programming. *Operations Research Letters*, 1(3):101–104, 1982.

[16] A. Charnes and W. W. Cooper. The stepping stone method of explaining linear programming calculations in transportation problems. *Management Science*, 1(1):49–69, 1954.

[17] A. Charnes, S. Duffuaa, and M. Ryan. Degeneracy and the more-for-less paradox. *Journal of Information & Optimization Sciences*, 1(1):52–56, 1980.

[18] A. Charnes, F. Glover, and D. Klingman. A note on a distribution problem. *Operations Research*, 18(6):1213–1216, 1970.

[19] A. Charnes and D. Klingman. The distribution problem with upper and lower bounds on the node requirements. *Management Science*, 16(9):638–642, 1970.

[20] A. Charnes and D. Klingman. The "more for less" paradox in the distribution model. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, 13(1):11–22, 1971.

[21] V. G. Dǐneko, B. Klinz, and G. J. Woeginger. Which matrices are immune against the transportation paradox. *Discrete Applied Mathematics*, 130(3):495–501, 2003.

[22] J. Elam, F. Glover, and D. Klingman. A strongly convergent primal simplex algorithm for generalized networks. *Mathematics of Operations Research*, 4(1):39–59, 1979.

[23] J. J. M. Evers. *Linear programming over an infinite horizon*, volume 8 of *Tilburg studies on economics*. Tilburg University Press, The Netherlands, 1973.

[24] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1963.

[25] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. freeman, New York, NY, 1979.

[26] S. I. Gass. *Linear Programming: Methods and Applications*. McGraw-Hill, New York, NY, fifth edition, 1975.

[27] S. I. Gass and S. Vinjamuri. Cycling in linear programming problems. *Computers & Operations Research*, 31(2):303–311, 2004.

[28] B. J. Gassner. Cycling in the transportation problem. *Naval Research Logistics Quarterly*, 11(1):43–58, 1964.

[29] F. Glover. *Netforms in Management Science and Decision Support*. University of Colorado, School of Business, Boulder, CO, 1983.

[30] F. Glover and D. A. Babayev. New results for aggregating integer-valued equations. *Annals of Operations Research*, 58:227–242, 1995.

[31] A. J. Goldman and D. Kleinman. Examples relating to the simplex method. *Operations Research*, 12(1):159–161, 1964.

[32] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[33] H. J. Greenberg. *A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions: A User's Guide for ANALYZE*, volume 1 of *Operations Research/Computer Science Interfaces Series*. Springer, New York, NY, 1993.

[34] H. J. Greenberg. How to analyze the results of linear programs — Part 2: Price interpretation. *Interfaces*, 23(5):97–114, 1993.

[35] H. J. Greenberg. Consistency, redundancy, and implied equalities in linear systems. *Annals of Mathematics and Artificial Intelligence*, 17:37–83, 1996.

[36] H. J. Greenberg. Chapter 3: Linear programming 1: Basic principles. In T. Gal and H. J. Greenberg, editors, *Advances in Sensitivity Analysis and Parametric Programming*. Kluwer Academic Press, 1997.

[37] H. J. Greenberg and J. E. Kalan. An exact update for Harris' tread. *Mathematical Programming Study*, 4:26–29, 1975.

[38] R. C. Grinold. Finite horizon approximations of infinite horizon linear programs. *Mathematical Programming*, 12(1):1–17, 1977.

[39] R. C. Grinold. Model building techniques for the correction of end effects in multistage convex programs. *Operations Research*, 31(3):407–431, 1983.

[40] R. C. Grinold and D. S. P. Hopkins. Duality overlap in infinite linear programs. *Journal of Mathematical Analysis and Applications*, 41(2):333–335, 1973.

[41] J. A. J. Hall and K. I. M. McKinnon. The simplest examples where the simplex method cycles and conditions where EXPAND fails to prevent cycling. *Mathematical Programming, Series B*, 100(1):133–150, 2004.

[42] J. L. Higle and S. Sen. Duality for multistage convex stochastic programs. *Annals of Operations Research*, 142(1):129–146, 2006.

[43] A. J. Hoffman. Cycling in the simplex algorithm. Report 2974, National Bureau of Standards, Gaithersburg, MD, 1953.

[44] A. Holder, editor. *Mathematical Programming Glossary*. INFORMS Computing Society, http://glossary.computing.society.informs.org, 2006–10. (Originally authored by Harvey J. Greenberg, 1999–2006.)

[45] F. Holt and V. Klee. Counterexamples to the strong d-step conjecture for $d \geq 5$. *Discrete & Computational Geometry*, 19(1):33–46, 1998.

[46] D. S. P. Hopkins. *Sufficient conditions for optimality in infinite horizon linear economic models*. Ph. D. thesis, Stanford University, Stanford, CA, 1969.

[47] R. G. Jeroslow. The simplex algorithm with the pivot rule of maximizing criterion improvement. *Discrete Mathematics*, 4(4):367–377, 1973.

[48] S. Kaplan. Comment on a Précis by Shanno and Weil. *Management Science*, 17(11):778–780, 1971.

[49] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[50] V. Klee and P. Kleinschmidt. The d-step conjecture and its relatives. *Mathematics of Operations Research*, 12(4):718–755, 1987.

[51] V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities III*, pages 159–175, New York, NY, 1972. Academic Press.

[52] H. W. Kuhn. Variants of the Hungarian method for assignment problem. *Naval Research Logistics Quarterly*, 3:253–258, 1956.

[53] W. F. Mascarenhas. The affine scaling algorithm fails for stepsize 0.999. *SIAM Journal on Optimization*, 7(1):34–46, 1997.

[54] A. Nagurney. *Network Economics: A Variational Inequality Approach*, volume 10 of *Advances in Computational Economics*. Springer, Dordrecht, The Netherlands, revised edition, 1999.

[55] A. Nagurney. *Supernetworks: Decision-Making for the Information Age*. Edward Elgar Publishing, Cheltenham, England, 2002.

[56] A. Nagurney. Supernetworks: Paradoxes, challenges, and new opportunities. In W. H. Dutton, B. Kahin, R. O'Callaghan, and A. W. Wyckoff, editors, *Transforming Enterprise*, pages 229–254, Cambridge, MA, 2004. MIT Press.

[57] A. Nagurney. *Supply Chain Network Economics: Dynamics of Prices, Flows And Profits*. New Dimensions in Networks. Edward Elgar Publishing, Cheltenham, England, 2006.

[58] J. B. Orlin. On the simplex algorithm for networks and generalized networks. *Mathematical Programming Study*, 24:166–178, 1985.

[59] J. B. Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(1):109–129, 1997.

[60] M. R. Rao. Some comments on 'linear' programming with absolute-value functionals. *Operations Research*, 21(1):373–374, 1973.

[61] A. Ravindran and T. W. Hill, Jr. A comment on the use of simplex method for absolute value problems. *Management Science*, 19(5):581–582, 1973.

[62] R. Rizzi. Excluding a simple good pair approach to directed cuts. *Graphs and Combinatorics*, 17(4):741–744, 2001.

[63] B. Rothschild and A. Whinston. Maximal two-way flows. *SIAM Journal on Applied Mathematics*, 5(5):1228–1238, 1967.

[64] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, Cambridge, MA, 2005.

[65] T. Roughgarden. On the severity of Braess's paradox: Designing networks for selfish users is hard. *Journal of Computer and System Sciences*, 72(5):922–953, 2006.

[66] D. S. Rubin and H. M. Wagner. Shadow prices: Tips and traps for managers. *Interfaces*, 20(4):150–157, 1990.

[67] M. J. Ryan. The distribution problem, the more for less (nothing) paradox and economies of scale and scope. *European Journal of Operational Research*, 121(1):92–104, 2000.

[68] T. L. Saaty. A conjecture concerning the smallest bound on the iterations in linear programming. *Operations Research*, 11(1):151–153, 1963.

[69] S. Sen. Subgradient decomposition and the differentiability of the recourse function of a two stage stochastic LP with recourse. *Operations Research Letters*, 13(3):143–148, 1993.

[70] D. F. Shanno and R. L. Weil, Jr. 'Linear' programming with absolute value functionals. *Operations Research*, 19(1):120–124, 1971.

[71] G. Sierksma and G. A. Tijssen. Degeneracy degrees of constraint collections. *Mathematical Methods of Operations Research*, 57(3):437–448, 2003.

[72] D. A. Spielman and S.H. Teng. Smoothed analysis: why the simplex algorithm usually takes polynomial time. *Journal of the Association for Computing Machinery*, 51(3):385–463, 2004.

[73] R. Steinberg and W. I. Zangwill. The prevalence of braess' paradox. *Transportation Science*, 17(3):301–318, 1983.

[74] W. Szwarc. The transportation paradox. *Naval Research Logistics Quarterly*, 18:185–202, 1971.

[75] M. J. Todd. The monotonic bounded Hirsch conjecture is false for dimension at least 4. *Mathematics of Operations Research*, 5(4):599–601, 1980.

[76] J. A. Tomlin and J. S. Welch. A pathological case in the reduction of linear programs. *Operations Research Letters*, 2(2):53–57, 1983.

[77] S. Storøy and T. Sørevik. A note on an orthogonal systolic design for the assignment problem. *Parallel Computing*, 17(4–5):523–525, 1991.

[78] J. Y. Yen. On Elmaghraby's "The theory of networks and management science". *Management Science*, 18(1):84–86, 1971.

[79] N. Zadeh. A bad network problem for the simplex method and other minimum cost flow algorithms. *Mathematical Programming*, 5(1):255–266, 1973.

[80] N. Zadeh. More pathological examples for network flow problems. *Mathematical Programming*, 5:217–224, 1973.

[81] T. Zhao and C-L. Tseng. A note on activity floats in activity-on-arrow networks. *Journal of the Operational Research Society*, 54(12):1296–1299, 2003.

[82] G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer, 1995. See pp. 83–93.

[83] H. Zullo. *Feasibile Flows in Multicommodity Graphs*. Ph.D. thesis, University of Colorado at Denver, Mathematics Department, Denver, CO, 1995. Available at http://www-math.cudenver.edu/graduate/thesis/hzullo.pdf.