The Ascendance of the Dual Simplex Method: A Geometric View

Robert Fourer

4er@ampl.com

AMPL Optimization Inc.
www.ampl.com — +1 773-336-AMPL

U.S.-Mexico Workshop on Optimization and Its Applications

Huatulco — 8-12 January 2018

The Ascendance of the Dual Simplex Method: A Geometric View

First described in the 1950s, the dual simplex evolved in the 1990s to become the method most often used in solving linear programs. Factors in the ascendance of the dual simplex method include Don Goldfarb's proposal for a steepest-edge variant, and an improved understanding of the bounded-variable extension. The ways that these come together to produce a highly effective algorithm are still not widely appreciated, however. This talk employs a geometric approach to the dual simplex method to provide a unified and straightforward description of the factors that work in its favor.

Motivation

Comput Optim Appl (2007) 37: 49–65
DOI 10.1007/s10589-007-9022-3

Progress in the dual simplex method for large scale LP problems: practical dual phase 1 algorithms

Achim Koberstein · Uwe H. Suhl

---

1 Introduction

Lemke [18] developed the dual simplex method in 1954 but it was not found to be an alternative to the primal simplex method for nearly forty years. This changed in the early Nineties mainly due to the contributions of Forrest and Goldfarb [7] and Fourer [8]. During the last decade commercial solvers have made great progress in

7. Forrest, J.J., Goldfarb, D.: Steepest edge simplex algorithms for linear programming. Math. Program. **57**(3), 341–374 (1992)
8. Fourer, R.: Notes on the dual simplex method. Draft report (1994)

Primal Linear Program

Minimize	cx
Subject to	$Ax = b,\ x \geq 0$

Basic variables $\mathcal{B}$, nonbasic variables $\mathcal{N}$

- Coefficient columns of basic variables form a nonsingular matrix B

Basic solution

- $\bar{x}_{\mathcal{N}} = 0, B\bar{x}_{\mathcal{B}} = b$

Feasible basic solution

- $\bar{x}_{\mathcal{B}} \geq 0$

Dual Linear Program

Maximize	πb
Subject to	$\pi A \le c$

Binding constraints $\mathcal{B}$, nonbinding constraints $\mathcal{N}$

- Coefficient rows of binding constraints form a nonsingular matrix B

Vertex solution

- $\bar{\pi} B = c_{\mathcal{B}}$

Feasible vertex solution

- $\bar{\pi} A_{\mathcal{N}} \le c_{\mathcal{N}}$

Dual Linear Program

Maximize	πb
Subject to	$\pi A + \sigma = c,\ \sigma \geq 0$

Binding constraints $\mathcal{B}$, *nonbinding constraints* $\mathcal{N}$

- Coefficient rows of binding constraints form a nonsingular matrix B

Vertex solution

- $\bar{\sigma}_{\mathcal{B}} = 0, \bar{\pi} B = c_{\mathcal{B}}$

Feasible vertex solution

- $\bar{\sigma}_{\mathcal{N}} = c_{\mathcal{N}} - \bar{\pi} A_{\mathcal{N}} \geq 0$

Primal Simplex Method

Given

- feasible basic solution $\bar{x}$ and corresponding basis matrix B

Choose a nonbasic variable to enter

- solve $\pi B = c_{\mathcal{B}}$
- select $p \in \mathcal{N}$: $\sigma_p = c_p - \pi a_p < 0$

Choose a basic variable to leave

- solve $B y_{\mathcal{B}} = a_p$
- select $q \in \mathcal{B}$: $\Theta = \bar{x}_q / y_q = \min_{y_i > 0} \bar{x}_i / y_i$

Update

- $\bar{x}_p \leftarrow \Theta$
- $\bar{x}_i \leftarrow \bar{x}_i - \Theta\, y_i$ for all $i \in \mathcal{B}$

Dual Simplex Method

Given

- feasible vertex solution $\bar{\sigma}$ and corresponding basis matrix B

Choose a binding constraint to leave

- solve $Bx_{\mathcal{B}} = b$
- select $q \in \mathcal{B}$: $x_q < 0$

Choose a nonbinding constraint to enter

- solve $\delta B = e_q$
- select $p \in \mathcal{N}$: $\Phi = \bar{\sigma}_p / \delta a_p = \min_{\delta a_j > 0} \bar{\sigma}_j / \delta a_j$

Update

- $\bar{\sigma}_q \leftarrow \Phi$
- $\bar{\sigma}_j \leftarrow \bar{\sigma}_j - \Phi(\delta a_j)$ for all $j \in \mathcal{N}$

Inner Products with a_j: Column-Wise

Work of $v \cdot a_j$ is the same for any v

- For each nonzero a_{ji} in column j of A, add $v_i a_{ji}$ to sum
- Need all v_i in an m-vector

Primal simplex

- Select one $\sigma_p = c_p - \pi a_p < 0$ $(\pi B = c_{\mathcal{B}})$ for $p \in \mathcal{N}$
- $\leq |\mathcal{N}|$ inner products, but often $\ll |\mathcal{N}|$

Dual simplex

- Compute $\min_{\delta_q a_j > 0} \bar{\sigma}_j / \delta_q a_j$ $(\delta_q B = e_q)$
- Always $|\mathcal{N}|$ inner products

Inner Products with a_j: Row-Wise

Store A by row as well as by column

- Accumulate n inner products together

Work of $v \cdot A$ depends on sparsity of v

- For each nonzero v_i,
  - for each nonzero a_{ji} in row i of A, add $v_i a_{ji}$ to sum for a_j

Faster $\delta_q a_j$ in dual simplex

- $\delta_q = e_q B^{-1}$ tends to be especially sparse

Faster $\sigma_j = c_j - \pi a_j$ in primal if you update them all

- $\sigma_q \leftarrow \sigma_p / \delta_q a_p$
- $\sigma_j \leftarrow \sigma_j - \sigma_q(\delta_q a_j)$ for all $j \in \mathcal{N}$

Primal Steepest Edge

Mathematical Programming 12 (1977) 361–371.
North-Holland Publishing Company

A PRACTICABLE STEEPEST-EDGE SIMPLEX ALGORITHM

D. GOLDFARB
The City College of The City University of New York, New York, U.S.A.

J.K. REID
A.E.R.E., Harwell, Oxon, Great Britain

Received 1 August 1975

It is shown that suitable recurrences may be used in order to implement in practice the steepest-edge simplex linear programming algorithm. In this algorithm each iteration is along an edge of the polytope of feasible solutions on which the objective function decreases most rapidly with respect to distance in the space of all the variables. Results of computer comparisons on medium-scale problems indicate that the resulting algorithm requires less iterations but about the same overall time as the algorithm of Harris [8], which may be regarded as approximating the steepest-edge algorithm. Both show a worthwhile advantage over the standard algorithm.

Key words: Simplex Method, Linear Programming, Steepest-edge, LU Factorization.

Primal Steepest Edge

Simplex step

- If x_j enters, solution $\bar{x}$ changes by θy_j
  - Entering variable $\bar{x}_p$ increases to θ
  - Basic variables $\bar{x}_{\mathcal{B}}$ change by $\theta y_{\mathcal{B}}$ (where $B y_{\mathcal{B}} = a_p$)
- Objective is reduced by $\theta \sigma_j$

Steepness of step

- $\sigma_j / \|y_j\| =$
  reduction of objective per unit change in solution

Main steepest-edge computations

- Choose largest $\sigma_j^2 / y_j^T y_j$ over all $j \in \mathcal{N}$ with $\sigma_j < 0$
- Update σ_j for $j \in \mathcal{N}$
- Update $y_j^T y_j$ for $j \in \mathcal{N} \ldots$

Primal Steepest Edge

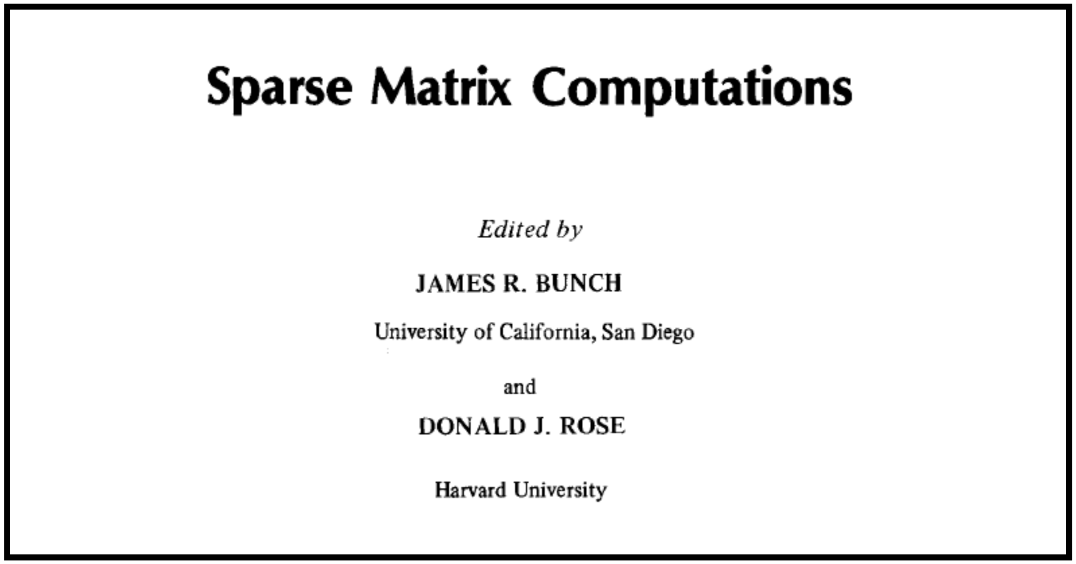
Updating $y_j^T y_j$

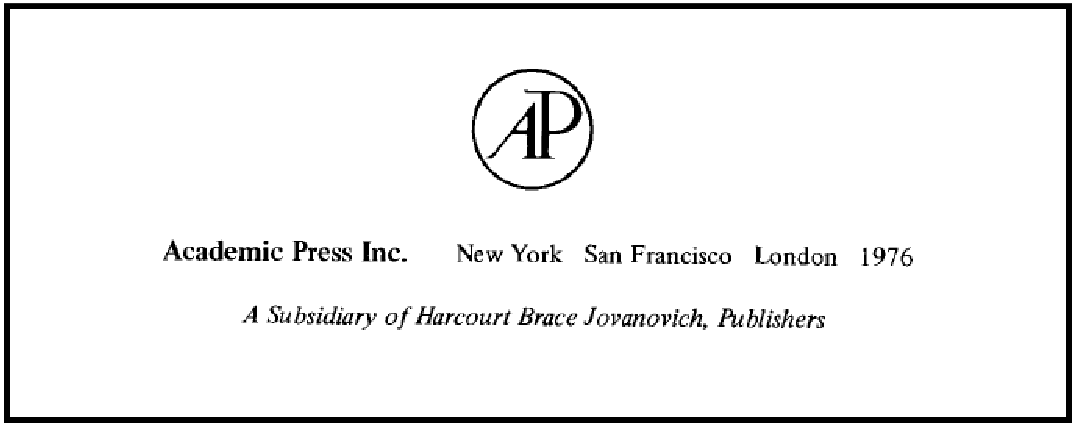
- $y_j \leftarrow y_j - \alpha_j y_p \;\; (\alpha_j = y_{jq}/y_{pq})$
  - α_j known after updating σ_j
  - y_j not known except for y_p, but $y_j^T y_j$ is known
- $y_j^T y_j \leftarrow (y_j - \alpha_j y_p)^T (y_j - \alpha_j y_p)$
- $y_j^T y_j \leftarrow y_j^T y_j - 2\alpha_j y_j^T y_p + \alpha^2 \, y_p^T y_p$

Hard part is $y_j^T y_p = a_j^T B^{-T} y_p$

- Solve $wB = y_p$
- Then compute $y_j^T y_p$ as $a_j^T w$ for each $j \in \mathcal{N}$
- One extra solve and $|\mathcal{N}|$ extra inner products

Dual Steepest Edge





Dual Steepest Edge

USING THE STEEPEST-EDGE SIMPLEX ALGORITHM
TO SOLVE SPARSE LINEAR PROGRAMS.

By D. Goldfarb
The City College of The City University of New York

Sometimes the solutions of the LP problem (1.1)-(1.3) are required for several different vectors b in (1.2). In such a situation or when a constraint is added to an LP problem whose solution is already known, it is advantageous to use a dual feasible algorithm. Thus we consider briefly a dual steepest-edge simplex algorithm. One might better call such an algorithm a maximal distance algorithm since at each step the pivot row selected is the one which in the transformed set of equations

$$B^{-1}Ax = B^{-1}b, \tag{5.4}$$

has a negative right hand side and whose corresponding hyperplane is furthest from the origin. Algebraically, one considers the elements of $B^{-1}b$ weighted by the norms of the corresponding rows.

Under the assumptions of section 2 it is simple to show that the following recurrences hold for the rows of $B^{-1}A$, $\rho_i = e_i^T B^{-1}A$, and the square of their norms $\beta_i = \rho_i^T \rho_i$

Dual Steepest Edge

$$\bar{\rho}_q = \rho_p / w_p \tag{5.5a}$$

$$\bar{\rho}_i = \rho_i - w_i \bar{\rho}_p \qquad i \le m,\ i \ne p \tag{5.5b}$$

and

$$\bar{\beta}_q = \beta_p / w_p^{\,2} \tag{5.6a}$$

$$\bar{\beta}_i = \beta_i - 2(w_i/w_p) e_i^T B^{-1} \hat{A} \hat{A}^T B^{-T} e_p + w_i^{\,2} \bar{\beta}_p \quad \begin{cases} i \le m \\ i \ne p \end{cases} \tag{5.6b}$$

where $A = [B:\hat{A}]$.

As in the primal algorithm the pivot column $w = B^{-1} a_q$ and all elements of the pivot row $\alpha_i = e_p^{\,T} B^{-1} a_i$ must be computed. Note that $\bar{\beta}_i \geq 1 + w_i^{\,2}/w_p^{\,2}$. In the primal algorithm $B^{-T}w$ is needed whereas here we require $B^{-1}y$ where $y = \sum_{i>m} \alpha_i a_i$. y can be computed in the same loop as the α_i's. As in the primal case most of these will be zero in sparse problems with a consequent reduction in the work required to compute y. Small savings also result from zeros in the pivot column.

The vector of row weights β can also be initialized economically by setting all of its components to one and then adding to these the square of the respective components of $B^{-1}a_j$ for $j > m$. Thus it is clear that it is possible to implement a practicable dual steepest-edge algorithm for large sparse LP problems.

Dual Steepest Edge

Simplex step

- If constraint $i \in \mathcal{B}$ is relaxed,
  - Slack variable σ_i increases to ϕ
  - Variables π change by $\phi\delta_i$ (where $\delta_i B = e_i$)
- Objective is reduced by ϕx_i

Steepness of step

- $x_i/\|\delta_i\| =$
  reduction of objective per unit change in solution

Main steepest-edge computations

- Choose largest $x_i^2/\delta_i^T\delta_i$ over all $i \in \mathcal{B}$ with $x_i < 0$
- Update x_i for $i \in \mathcal{B}$
- Update $\delta_i^T\delta_i$ for $i \in \mathcal{B}$ . . .

Dual Steepest Edge

Updating $\delta_i^T \delta_i$

- $\delta_i \leftarrow \delta_i - \beta_i \delta_q \ \ (\beta_i = y_{pi}/y_{pq})$
  - β_i known after updating x_i
  - δ_i not known except for δ_q, but $\delta_i^T \delta_i$ is known
- $\delta_i^T \delta_i \leftarrow (\delta_i - \beta_i \delta_q)^T (\delta_i - \beta_i \delta_q)$
- $\delta_i^T \delta_i \leftarrow \delta_i^T \delta_i - 2\beta\,_j^T \delta_i^T \delta_q + \beta^2 \delta_q^T \delta_q$

Hard part is $\delta_i^T \delta_q = e_i B^{-1} \delta_q$

- Solve $Bv = \delta_q$
- Then $\delta_i^T \delta_q$ is v_i for each $i \in \mathcal{B}$
- One extra solve but ***no extra inner products***

Bounded Variables

Notes on the Dual Simplex Method

0. The standard dual simplex method
1. A more general and practical dual simplex method
2. Phase I for the dual simplex method
3. Degeneracy in the dual simplex method
4. A generalized ratio test for the dual simplex method

Robert Fourer — 4er@iems.nwu.edu — *March 14, 1994*

Bounded Variables

Generalize $x \geq 0$ *to* $\ell \leq x \leq u$

- State simplex methods for ℓ, u finite
- Extend to allow some $\ell_j = -\infty$ and/or $u_j = +\infty$
- Check that $\ell = 0, u = \infty$ reduces to previous case

Further improve the dual simplex method

- Take longer steps
- Adapt to degeneracy

Primal LP with Bounded Variables

Minimize	cx
Subject to	$Ax = b,\ \ell \le x \le u$

Basic variables $\mathcal{B}$, nonbasic variables $\mathcal{N} = \mathcal{L} \cup \mathcal{U}$

- Coefficient columns of basic variables form a nonsingular matrix B

Basic solution

- $\bar{x}_j = \ell_j$ for $j \in \mathcal{L},\ \bar{x}_j = u_j$ for $j \in \mathcal{U}$
- $B\bar{x}_{\mathcal{B}} = b - \sum_{j\in\mathcal{L}} \ell_j a_j - \sum_{j\in\mathcal{U}} u_j a_j,$

Feasible basic solution

- $\ell_{\mathcal{B}} \le \bar{x}_{\mathcal{B}} \le u_{\mathcal{B}}$

Dual LP with Bounded Variables

Nonlinear
Programming 4

Edited by Olvi L. Mangasarian
Robert R. Meyer
Stephen M. Robinson

Computer Sciences Department
University of Wisconsin—Madison
Madison, Wisconsin

Academic Press 1981
A SUBSIDIARY OF HARCOURT BRACE JOVANOVICH, PUBLISHERS
New York London Toronto Sydney San Francisco

Dual LP with Bounded Variables

Dual LP with Bounded Variables

Obviously there is no difficulty in passing from f_j to Γ_j to g_j by the method above when Γ_j is of such type.

To drive this point home and make apparent the directness and flexibility of this duality scheme in monotropic programming, we turn to the example of (P) as a general linear programming problem with both upper and lower bounds for each variable. Linear programming theory is incapable of producing a dual without first subjecting the problem to a transformation into one of the canonical forms where no single variable is bounded in both directions. In the monotropic programming context, however, we can regard a problem of this sort as

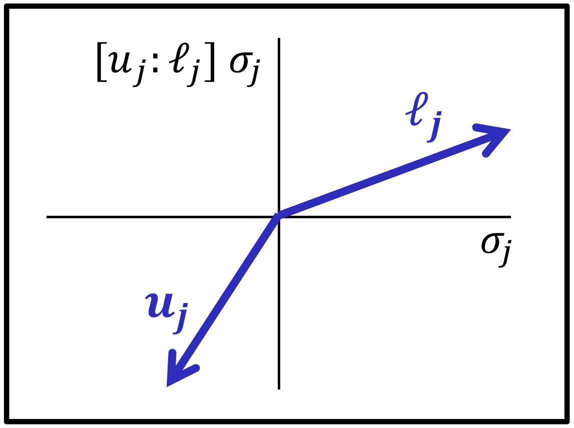
Note that in this example the dual of a linear programming problem turns out, in general, to be merely piecewise linear. It is no wonder, then, that linear programming theory cannot fully capture such duality. Other forms of linear programming problems can be handled similarly. In essence, one

Dual LP with Bounded Variables

Maximize	$\pi b + [u\!:\ell]\,\sigma$
Subject to	$\pi A + \sigma = c$

What is $[u\!:\ell]\,\sigma$ ***??***

- Sum of concave piecewise-linear functions $[u_j\!:\ell_j]\,\sigma_j$
  - Slope of u_j for $\sigma_j \le 0$
  - Slope of ℓ_j for $\sigma_j \ge 0$
- Example for $0 \le \ell_j < u_j < \infty$:



Dual LP with Bounded Variables

Maximize	$\pi b + [u\!:\!\ell]\, \sigma$
Subject to	$\pi A + \sigma = c$

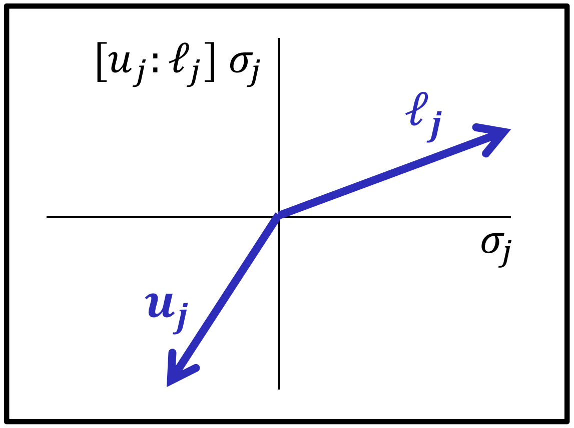
Binding constraints $\mathcal{B}$,
nonbinding constraints $\mathcal{N} = \mathcal{L} \cup \mathcal{U}$

- Coefficient rows of binding constraints form a nonsingular matrix B

"Vertex" solution

- $\bar{\sigma}_{\mathcal{B}} = 0, \bar{\pi} B = c_{\mathcal{B}}$
- $j \in \mathcal{L}$ for $\bar{\sigma}_j = c_j - \bar{\pi} a_j > 0$
- $j \in \mathcal{U}$ for $\bar{\sigma}_j = c_j - \bar{\pi} a_j < 0$

Always feasible!



Primal Simplex, Bounded Variables

Given

- feasible basic solution $\bar{x}$ and corresponding basis matrix B

Choose a nonbasic variable to enter

- solve $\pi B = c_{\mathcal{B}}$
- select $p \in \mathcal{L}$: $\sigma_p = c_p - \pi a_p < 0$ or
  select $p \in \mathcal{U}$: $\sigma_p = c_p - \pi a_p > 0$

Choose a basic variable to leave

- solve $By_{\mathcal{B}} = a_p$ $(p \in \mathcal{L})$ or $By_{\mathcal{B}} = -a_p$ $(p \in \mathcal{U})$
- select $\Theta = \min\,(\Theta_{\mathcal{L}}, \Theta_{\mathcal{U}}, \Theta_p)$:
  - ∗ $q \in \mathcal{B}$: $\Theta_{\mathcal{L}} = (\bar{x}_q - \ell_q)/y_q = \min\limits_{y_i > 0}\,(\bar{x}_i - \ell_i)/y_i$
  - ∗ $q \in \mathcal{B}$: $\Theta_{\mathcal{U}} = (u_q - \bar{x}_q)/y_q = \min\limits_{y_i < 0}\,(\bar{x}_i - u_i)/y_i$
  - ∗ $q = p$: $\Theta_p = u_p - \ell_p$

Update . . .

Dual Simplex, Bounded Variables

Given

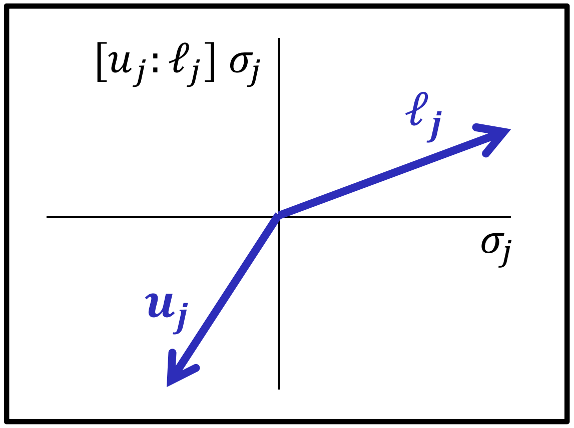
- feasible vertex solution $\bar{\sigma}$ and corresponding basis matrix B

Choose a binding constraint to leave

- solve $Bx_{\mathcal{B}} = b - \sum_{j\in\mathcal{L}} \ell_j a_j - \sum_{j\in\mathcal{U}} u_j a_j$
- select $q \in \mathcal{B}$: $x_q < \ell_q$ or $x_q > u_q$

Choose a nonbinding constraint to enter

- solve $\delta B = e_q$ (if $x_q < \ell_q$) or $\delta B = -e_q$ (if $x_q > u_q$)
- select $\Phi = \min\,(\Phi_{\mathcal{L}}, \Phi_{\mathcal{U}})$:
  - $\ast$ $j \in \mathcal{L}$: $\Phi_{\mathcal{L}} = \bar{\sigma}_p/\delta a_p = \min\limits_{\delta a_j > 0} \bar{\sigma}_j/\delta a_j$
  - $\ast$ $j \in \mathcal{U}$: $\Phi_{\mathcal{U}} = \bar{\sigma}_p/\delta a_p = \min\limits_{\delta a_j < 0} \bar{\sigma}_j/\delta a_j$

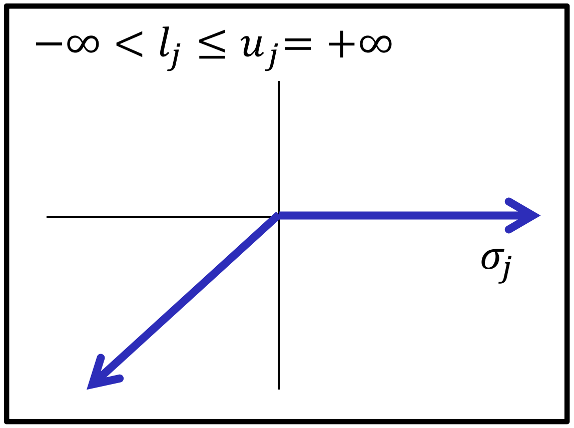


Update . . .

In Principle, Bounds Can Be Infinite

Some $\ell_j = -\infty$ *and/or* $u_j = +\infty$ *??*

- A basis may be infeasible
  - $\bar{\sigma}_j > 0$ but $\ell_j = -\infty$
  - $\bar{\sigma}_j < 0$ but $u_j = +\infty$
- Minimize "sum of infeasible variables" to get feasible
  - Replace each $\ell_j = -\infty$ by -1
  - Replace each $u_j = +\infty$ by $+1$
  - Replace all finite bounds by 0



All $\ell_j = 0$ *and* $u_j = +\infty$ *??*

- Then $x_j \ge 0$
- Primal and dual algorithms reduce to their simpler forms

In Practice, Bounds Tend to be Finite

Decisions are bounded

- Variables are bounded
- Slacks on inequality constraints are bounded

Integer-valued decisions have small values

- Many are zero-one!

Standard presolve routines compute bounds

- Compute bounds where none given by user
- Tighten bounds in multiple passes

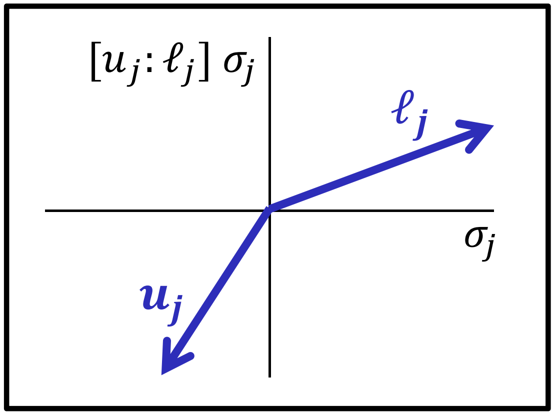
Long Steps

Standard iteration

- $\bar{\sigma}_q$ increases to Φ (if $x_q < \ell_q$) or decreases to $-\Phi$ (if $x_q > u_q$)
- $\bar{\sigma}_p$ moves to 0

Suppose $\bar{\sigma}_q$ increases/decreases further . . .

- $\bar{\sigma}_p$ moves past 0, but solution remains feasible
- Rate of improvement in objective degrades by $|\delta a_p|(u_p - \ell_p)$
  - If $\bar{\sigma}_p$ increasing, objective slope changes from u_p to ℓ_p
  - If $\bar{\sigma}_p$ decreasing, objective slope changes from $-\ell_p$ to $-u_p$
- If rate still positive, can continue until next $\bar{\sigma}_j$, $j \in \mathcal{N}$ reaches 0



Long Step Iteration

Replace single ratio test by a loop

- Set initial objective improvement rate to
  $r = \ell_q - x_q$ (if $x_q < \ell_q$) or $r = x_q - u_q$ (if $x_q > u_q$)
- Form set of all ratios that may be reached:
  $\mathcal{Q} = \{\bar{\sigma}_j/\delta a_j\colon\ j \in \mathcal{L}, \delta a_j > 0\} \cup \{\bar{\sigma}_j/\delta a_j\colon\ j \in \mathcal{U}, \delta a_j < 0\}$
- Repeat for increasing $\bar{\sigma}_j/\delta a_j \in \mathcal{Q}$:
  - Let $r \leftarrow r - |\delta a_j|(u_j - \ell_j)$
  - Let $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{\bar{\sigma}_j/\delta a_j\}$

  until $r \leq 0$

Equivalent to "weighted selection"

- In theory, faster than sorting
- In practice, a small part of iteration cost

Re-Optimization for MIPs

Change bounds for some fractional $\bar{x}_i$, $i \in \mathcal{B}$

- Increase ℓ_i to $\lceil \bar{x}_i \rceil$, resulting in $\bar{x}_i < \ell_i$
- Decrease u_i to $\lfloor \bar{x}_i \rfloor$, resulting in $\bar{x}_i > u_i$
- Either way, binding constraint $i \in \mathcal{B}$ can leave
- Continue with dual simplex steps until optimal again

Fix some fractional binary $\bar{x}_i$, $i \in \mathcal{B}$

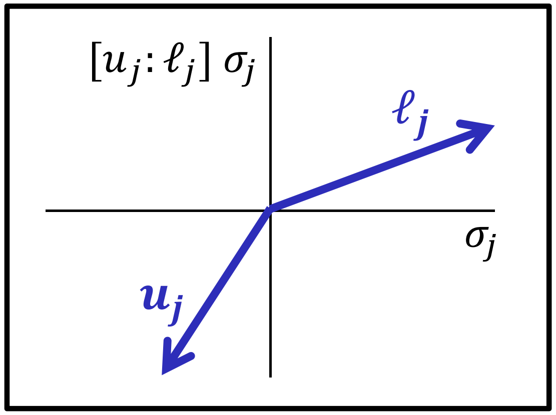
- Increase ℓ_i to $u_i = 1$, resulting in $\bar{x}_i < \ell_i$
- Decrease u_i to $\ell_i = 0$, resulting in $\bar{x}_i > u_i$
- Either way, binding constraint $i \in \mathcal{B}$ can leave
- Since now $\ell_i = u_i$, using long steps
  the constraint will never become binding again

Degeneracy

Choosing a binding constraint q to relax

- For $\bar{\sigma}_j > 0$, place $j \in \mathcal{L}$
- For $\bar{\sigma}_j < 0$, place $j \in \mathcal{U}$
- For $\bar{\sigma}_j = 0$???
  - Guess $j \in \mathcal{L}$ if you think $\bar{\sigma}_j$ is likely to increase
  - Guess $j \in \mathcal{U}$ if you think $\bar{\sigma}_j$ is likely to decrease

  . . . can't be sure though until you choose q
- solve $B x_{\mathcal{B}} = b - \sum_{j \in \mathcal{L}} \ell_j a_j - \sum_{j \in \mathcal{U}} u_j a_j$
- select $q \in \mathcal{B}$: $x_q < \ell_q$ or $x_q > u_q$



Degeneracy

Choosing a nonbinding constraint p to add

- Set initial objective improvement rate to

  $r = \ell_q - x_q$ (if $x_q < \ell_q$) or $r = x_q - u_q$ (if $x_q > u_q$)

- Collect all ratios that may be reached:

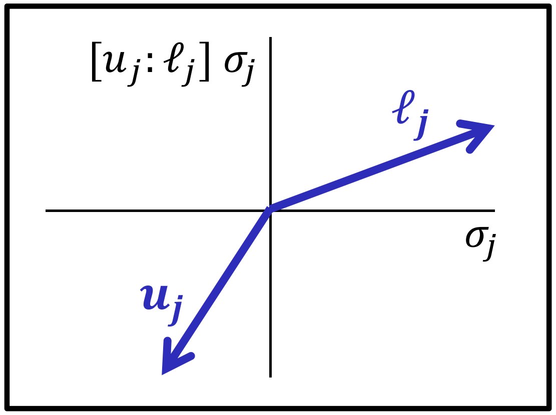
  $\mathcal{Q} = \left[\bar{\sigma}_j/\delta a_j\colon\ j \in \mathcal{L}, \delta a_j > 0\right] \cup \left[\bar{\sigma}_j/\delta a_j\colon\ j \in \mathcal{U}, \delta a_j < 0\right]$

- ***Some of these ratios may be zero!***
  - For $j \in \mathcal{L},\ \bar{\sigma}_j = 0$ and $\delta a_j > 0$
  - For $j \in \mathcal{U},\ \bar{\sigma}_j = 0$ and $\delta a_j < 0$

- Repeat for every $\bar{\sigma}_j/\delta a_j = 0 \in \mathcal{Q}$:
  - Let $r \leftarrow r - \left|\delta a_j\right|(u_j - \ell_j)$
  - Let $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \left\{\bar{\sigma}_j/\delta a_j\right\}$

  while $r > 0$

- If $r \leq 0$, iteration is degenerate
- If still $r > 0$, continue with nondegenerate iteration



Degeneracy: Benefit of Long Steps

For some $\bar{\sigma}_j = 0$, you "guess wrong"

- ❖ $j \in \mathcal{L}$, but $\bar{\sigma}_j$ decreases
- ❖ $j \in \mathcal{U}$, but $\bar{\sigma}_j$ increases

As a result, you are overly optimistic about the objective improvement rate r

Long-step ratio test corrects for your wrong guesses

- ❖ If the corrected $r > 0$ then
  you can take a nondegenerate step after all

None of this changes the optimality condition

- ❖ $\ell_q \leq x_q \leq u_q$ for all $q \in \mathcal{B}$

Ascendance of the Dual Simplex

Fast inner products

Dual steepest edge

Bounded-variable extension

- Feasibility for finite bounds
- Long steps
- More nondegenerate steps