The Revised Simplex Method

Combinatorial Problem Solving (CPS)

Javier Larrosa Albert Oliveras Enric Rodríguez-Carbonell

April 17, 2020

Tableau Simplex Method

- The simplex method we have seen so far is called tableau simplex method
- Some observations:
  - At each iteration we update the full tableau
    $x_{\mathcal{B}} = B^{-1}b - B^{-1}Rx_{\mathcal{R}}$
    for the new basis
  - But ...
    - For pricing only one negative reduced cost is needed
    - For ratio test, only
      - the column of the chosen non-basic variable in the tableau, and
      - the current basic solution

      are needed

Revised Simplex Method

- Idea: do not keep a representation of the full tableau, only B^{-1}
- Advantages over the tableau version:
  - Time and space are saved
  - Errors due to floating-point arithmetic are easier to control

Revised Simplex Method

- Idea: do not keep a representation of the full tableau, only B^{-1}
- Advantages over the tableau version:
  - Time and space are saved
  - Errors due to floating-point arithmetic are easier to control
- We will revise the algorithm and express it in terms of B^{-1}
- First for LP's of the form

$$\begin{aligned} &\min\ z = c^T x \\ &Ax = b \\ &x \geq 0 \end{aligned}$$

Basic Solution

- Let us see how the basic solution is expressed in terms of B^{-1}
- For any basis B,
  values of basic variables can be expressed in terms of non-basic variables:

$$\begin{aligned} Bx_{\mathcal{B}} + Rx_{\mathcal{R}} &= b \\ Bx_{\mathcal{B}} &= b - Rx_{\mathcal{R}} \\ x_{\mathcal{B}} &= B^{-1}b - B^{-1}Rx_{\mathcal{R}} \end{aligned}$$

Basic Solution

- Let us see how the basic solution is expressed in terms of B^{-1}

- For any basis B,
  values of basic variables can be expressed in terms of non-basic variables:

$$\begin{aligned} Bx_{\mathcal{B}} + Rx_{\mathcal{R}} &= b \\ Bx_{\mathcal{B}} &= b - Rx_{\mathcal{R}} \\ x_{\mathcal{B}} &= B^{-1}b - B^{-1}Rx_{\mathcal{R}} \end{aligned}$$

- By definition, the basic solution corresponds to
  assigning null values to all non-basic variables: $x_{\mathcal{R}} = 0$
  Then $x_{\mathcal{B}} = B^{-1}b$

- We will denote the basic solution (projected on basic variables) with
  $\beta := B^{-1}b$

Optimality Condition

- Let us see now how to express the reduced costs in terms of B^{-1}
- Recall the equation of basic variables in terms of non-basic variables:

$$x_{\mathcal{B}} = B^{-1}b - B^{-1}Rx_{\mathcal{R}}$$

- Cost function can be split: $c^Tx = c^T_{\mathcal{B}}x_{\mathcal{B}} + c^T_{\mathcal{R}}x_{\mathcal{R}}$, where
  $c^T_{\mathcal{B}}$ are the costs of basic variables,
  $c^T_{\mathcal{R}}$ are the costs of non-basic variables

Optimality Condition

- Let us see now how to express the reduced costs in terms of B^{-1}

- Recall the equation of basic variables in terms of non-basic variables:

$$x_{\mathcal{B}} = B^{-1}b - B^{-1}Rx_{\mathcal{R}}$$

- Cost function can be split: $c^T x = c_{\mathcal{B}}^T x_{\mathcal{B}} + c_{\mathcal{R}}^T x_{\mathcal{R}}$, where
  $c_{\mathcal{B}}^T$ are the costs of basic variables,
  $c_{\mathcal{R}}^T$ are the costs of non-basic variables

- We can express the cost function in terms of non-basic variables:

$$\begin{aligned}
&c^T x = \\
&c_{\mathcal{B}}^T x_{\mathcal{B}} + c_{\mathcal{R}}^T x_{\mathcal{R}} = \\
&c_{\mathcal{B}}^T (B^{-1}b - B^{-1}Rx_{\mathcal{R}}) + c_{\mathcal{R}}^T x_{\mathcal{R}} = \\
&c_{\mathcal{B}}^T B^{-1}b - c_{\mathcal{B}}^T B^{-1}Rx_{\mathcal{R}} + c_{\mathcal{R}}^T x_{\mathcal{R}} = \\
&c_{\mathcal{B}}^T B^{-1}b + (c_{\mathcal{R}}^T - c_{\mathcal{B}}^T B^{-1}R)x_{\mathcal{R}}
\end{aligned}$$

Optimality Condition

- We found that $c^T x = c_{\mathcal{B}}^T B^{-1} b + (c_{\mathcal{R}}^T - c_{\mathcal{B}}^T B^{-1} R) x_{\mathcal{R}}$

- The part that depends on non-basic variables is $(c_{\mathcal{R}}^T - c_{\mathcal{B}}^T B^{-1} R) x_{\mathcal{R}}$

- Let a_j be the column in A corresponding to variable $x_j \in x_{\mathcal{R}}$.

  The coefficient of x_j in $(c_{\mathcal{R}}^T - c_{\mathcal{B}}^T B^{-1} R) x_{\mathcal{R}}$ is $c_j - c_{\mathcal{B}}^T B^{-1} a_j$

  We will denote the reduced cost of x_j with $d_j := c_j - c_{\mathcal{B}}^T B^{-1} a_j$

- Optimality condition: $d_j \geq 0$ for all $j \in \mathcal{R}$

Cost at Basic Solution

- Let's see how to express the value of the cost function at the basic solution
- We found that $c^T x = c_{\mathcal{B}}^T B^{-1} b + d_{\mathcal{R}}^T x_{\mathcal{R}}$, where $d_j = c_j - c_{\mathcal{B}}^T B^{-1} a_j$
- We will denote the value of the cost function at the basic solution with z
- Taking $x_{\mathcal{R}} = 0$ in the above equation: $z := c_{\mathcal{B}}^T B^{-1} b$

Cost at Basic Solution

- Let's see how to express the value of the cost function at the basic solution
- We found that $c^T x = c_{\mathcal{B}}^T B^{-1} b + d_{\mathcal{R}}^T x_{\mathcal{R}}$, where $d_j = c_j - c_{\mathcal{B}}^T B^{-1} a_j$
- We will denote the value of the cost function at the basic solution with z
- Taking $x_{\mathcal{R}} = 0$ in the above equation: $z := c_{\mathcal{B}}^T B^{-1} b$
- To avoid repeating computations:

  Let us define the simplex multiplier as $\pi := (B^T)^{-1} c_{\mathcal{B}}$

  Then $\pi^T = c_{\mathcal{B}}^T B^{-1}$

  So $d_j = c_j - \pi^T a_j$
  (and $z = \pi^T b$)

Improving a Non-Optimal Solution

- Let us assume the optimality condition is violated
- Let x_q be a non-basic variable such that its reduced cost is $d_q < 0$
- Current value of x_q is 0.
  We can improve by increasing only this value
  while non-negativity constraints of basic variables are satisfied.

Improving a Non-Optimal Solution

- Let us assume the optimality condition is violated
- Let x_q be a non-basic variable such that its reduced cost is $d_q < 0$
- Current value of x_q is 0.
  We can improve by increasing only this value
  while non-negativity constraints of basic variables are satisfied.
- Let $t \geq 0$ be the new value for x_q.

  Let $x_{\mathcal{B}}(t)$ be the values of basic variables in terms of t

  Let $x_{\mathcal{R}}(t)$ be the values of non-basic variables in terms of t

  Note that $x_q(t) = t$, and $x_p(t) = 0$ if $p \in \mathcal{R}$ and $p \neq q$

Improving a Non-Optimal Solution

- Let us assume the optimality condition is violated
- Let x_q be a non-basic variable such that its reduced cost is $d_q < 0$
- Current value of x_q is 0.
  We can improve by increasing only this value
  while non-negativity constraints of basic variables are satisfied.
- Let $t \geq 0$ be the new value for x_q.

  Let $x_{\mathcal{B}}(t)$ be the values of basic variables in terms of t

  Let $x_{\mathcal{R}}(t)$ be the values of non-basic variables in terms of t

  Note that $x_q(t) = t$, and $x_p(t) = 0$ if $p \in \mathcal{R}$ and $p \neq q$

  So $x_{\mathcal{B}}(t) = B^{-1}b - B^{-1}Rx_{\mathcal{R}}(t) = B^{-1}b - B^{-1}a_q t = \beta - t\alpha_q$

  where $\beta = B^{-1}b$ is the basic solution
  and we denote the column in the tableau of x_q as $\alpha_q := B^{-1}a_q$

Improving a Non-Optimal Solution

- How much do we improve?

  How does the objective value change as a function of t?

$$
\begin{aligned}
&z(t) = \\
&c^T x(t) = \\
&c_{\mathcal{B}}^T x_{\mathcal{B}}(t) + c_{\mathcal{R}}^T x_{\mathcal{R}}(t) = \\
&c_{\mathcal{B}}^T x_{\mathcal{B}}(t) + c_q t = \\
&c_{\mathcal{B}}^T \beta - t c_{\mathcal{B}}^T \alpha_q + c_q t = \\
&c_{\mathcal{B}}^T \beta - t c_{\mathcal{B}}^T B^{-1} a_q + c_q t = \\
&z + t d_q
\end{aligned}
$$

- As expected, the improvement in cost is $\Delta z = z(t) - z = t d_q$

Improving a Non-Optimal Solution

- Recall that $x_{\mathcal{B}}(t) = \beta - t\alpha_q$
- How can we satisfy the non-negativity constraints of basic variables?

Improving a Non-Optimal Solution

- Recall that $x_{\mathcal{B}}(t) = \beta - t\alpha_q$
- How can we satisfy the non-negativity constraints of basic variables?
- Basic variables have indices $\mathcal{B} = (k_1, ..., k_m)$
- Let $i \in \{1, ..., m\}$. The i-th basic variable is x_{k_i}
- Value of x_{k_i} as a function of t is the i-th component of $x_{\mathcal{B}}(t)$: $\beta_i - t\alpha_q^i$, where β_i is the i-th component of β and α_q^i is the i-th component of α_q

Improving a Non-Optimal Solution

- Recall that $x_{\mathcal{B}}(t) = \beta - t\alpha_q$
- How can we satisfy the non-negativity constraints of basic variables?
- Basic variables have indices $\mathcal{B} = (k_1, ..., k_m)$
- Let $i \in \{1, ..., m\}$. The i-th basic variable is x_{k_i}
- Value of x_{k_i} as a function of t is the i-th component of $x_{\mathcal{B}}(t)$: $\beta_i - t\alpha_q^i$, where β_i is the i-th component of β and α_q^i is the i-th component of α_q
- We need $\beta_i - t\alpha_q^i \geq 0 \iff \beta_i \geq t\alpha_q^i$
    - If $\alpha_q^i \leq 0$ the constraint is satisfied for all $t \geq 0$
    - If $\alpha_q^i > 0$ we need $\frac{\beta_i}{\alpha_q^i} \geq t$

Improving a Non-Optimal Solution

- Recall that $x_{\mathcal{B}}(t) \; = \; \beta - t\alpha_q$
- How can we satisfy the non-negativity constraints of basic variables?
- Basic variables have indices $\mathcal{B} = (k_1, ..., k_m)$
- Let $i \in \{1, ..., m\}$. The i-th basic variable is x_{k_i}
- Value of x_{k_i} as a function of t is the i-th component of $x_{\mathcal{B}}(t)$: $\beta_i - t\alpha_q^i$, where β_i is the i-th component of β and α_q^i is the i-th component of α_q
- We need $\quad \beta_i - t\alpha_q^i \geq 0 \quad \Longleftrightarrow \quad \beta_i \geq t\alpha_q^i$
  - If $\alpha_q^i \leq 0$ the constraint is satisfied for all $t \geq 0$
  - If $\alpha_q^i > 0$ we need $\frac{\beta_i}{\alpha_q^i} \geq t$
- The best improvement is achieved with the strongest of the upper bounds:

$$\theta := \min\{\frac{\beta_i}{\alpha_q^i} \mid \alpha_q^i > 0\}$$

- We say the p-th basic variable x_{k_p} is blocking or tight when $\theta = \frac{\beta_p}{\alpha_q^p}$.

  Then α_q^p is the pivot

Improving a Non-Optimal Solution

1. If $\theta = +\infty$
   (there is no upper bound, i.e., no i such that $1 \leq i \leq m$ and $\alpha_q^i > 0$):

   Value of objective function can be decreased infinitely.
   LP is unbounded.

Improving a Non-Optimal Solution

1. If $\theta = +\infty$
   (there is no upper bound, i.e., no i such that $1 \leq i \leq m$ and $\alpha_q^i > 0$):

   Value of objective function can be decreased infinitely.
   LP is unbounded.

2. If $\theta < +\infty$ and the p-th basic variable x_{k_p} is blocking:

   When setting $x_q = \theta$, the non-negativity of basic variables is respected

   In particular the value of x_{k_p}, i.e. the p-th component of $x_{\mathcal{B}}(t)$, is
   $\beta_p - \theta\alpha_q^p = 0$

   We can make a basis change:
   x_q enters the basis and x_{k_p} leaves, where $\mathcal{B} = (k_1, \dots, k_m)$

Update

- New basic indices: $\bar{\mathcal{B}} = (k_1, \ldots, k_{p-1}, q, k_{p+1} \ldots, k_m)$

  Before the p-th basic variable was x_{k_p}, now it is x_q

- New basis: $\bar{B} = B + (a_q - a_{k_p})e_p^T$

  where $e_p^T = \underbrace{(0, ..., 0, \overbrace{\;1\;}^{p}, 0, ..., 0)}_{m}$

  The p-th column of the basis (which was a_{k_p}) is replaced by a_q.

- New basic solution: $\bar{\beta}_p = \theta$, $\bar{\beta}_i = \beta_i - \theta\alpha_q^i$ if $i \neq p$

  Note that before the p-th component of β corresponded to x_{k_p}, now to x_q

- New objective value: $\bar{z} = z + \theta d_q$

Algorithmic Description

1. `Initialization:` Find an initial feasible basis B
   Compute $B^{-1}, \beta = B^{-1}b, z = c_{\mathcal{B}}^T\beta$

2. `Pricing:` Compute $\pi^T = c_{\mathcal{B}}^T B^{-1}$ and $d_j = c_j - \pi^T a_j$.
   If for all $j \in \mathcal{R}, d_j \geq 0$ then return OPTIMAL
   Else let q be such that $d_q < 0$. Compute $\alpha_q = B^{-1}a_q$

3. `Ratio test:` Compute $\mathcal{I} = \{i \mid 1 \leq i \leq m, \alpha_q^i > 0\}$.
   If $\mathcal{I} = \emptyset$ then return UNBOUNDED
   Else compute $\theta = \min_{i\in\mathcal{I}}(\frac{\beta_i}{\alpha_q^i})$ and p such that $\theta = \frac{\beta_p}{\alpha_q^p}$

4. `Update:`
   $\bar{\mathcal{B}} = \mathcal{B} - \{k_p\} \cup \{q\}$ $\qquad$ $\bar{B} = B + (a_q - a_{k_p})e_p^T$
   $\bar{\beta}_p = \theta$, $\bar{\beta}_i = \beta_i - \theta\alpha_q^i$ if $i \neq p$ $\qquad$ $\bar{z} = z + \theta d_q$
   Go to 2.

Updating Matrix Inverse

- Actually what we really care about is B^{-1}, not B

  We need it for computing $\pi = c_{\mathcal{B}}^T B^{-1}$ and $\alpha_q = B^{-1} a_q$ at each step

  (and also $\beta = B^{-1} b$ in the initialization)

- Recomputing B^{-1} at each iteration is too expensive
  (e.g. $O(m^3)$ arithmetic operations with Gaussian elimination!)

- Next slides: a more efficient way of computing $\bar{B}^{-1}$ using B^{-1}

Updating Matrix Inverse

- Let us make a diversion into linear algebra
- Let $b_1, ..., b_m$ be the columns of an invertible matrix B
  Let a, α be such that $a = B\alpha = \sum_{i=1}^{m} \alpha_i b_i$
  Let p be such that $1 \leq p \leq m$
- $B_a = (b_1, \ldots, b_{p-1}, a, b_{p+1}, \ldots, b_m)$. Want to compute B_a^{-1}

Updating Matrix Inverse

- Let us make a diversion into linear algebra
- Let $b_1, ..., b_m$ be the columns of an invertible matrix B
  Let a, α be such that $a = B\alpha = \sum_{i=1}^{m} \alpha_i b_i$
  Let p be such that $1 \leq p \leq m$
- $B_a = (b_1, \ldots, b_{p-1}, a, b_{p+1}, \ldots, b_m)$. Want to compute B_a^{-1}
- Note $\alpha_p \neq 0$ as otherwise $\mathrm{rank}(B_a) < m$.
  Then $a = \alpha_p b_p + \sum_{i \neq p} \alpha_i b_i \quad \Rightarrow \quad b_p = \left(\frac{1}{\alpha_p}\right) a + \sum_{i \neq p} \left(\frac{-\alpha_i}{\alpha_p}\right) b_i$
- Let $\eta^T = \left(\left(\frac{-\alpha_1}{\alpha_p}\right), \ldots, \left(\frac{-\alpha_{p-1}}{\alpha_p}\right), \frac{1}{\alpha_p}, \left(\frac{-\alpha_{p+1}}{\alpha_p}\right), \ldots, \left(\frac{-\alpha_m}{\alpha_p}\right) \right)^T$.
  Then $b_p = B_a \eta$
- Let $E = (e_1, \ldots, e_{p-1}, \eta, e_{p+1}, \ldots, e_m)$.
  Then $B_a E = B \implies E^{-1} B_a^{-1} = B^{-1} \implies B_a^{-1} = E B^{-1}$

Updating Matrix Inverse

- Application to the simplex algorithm:
  $a = a_q$, $\alpha = \alpha_q$, where x_q is entering variable

  Thus to update the inverse we can reuse already computed data!

- Using this update: B^{-1} is not actually represented as a square table, but as follows

- Assume initial basis is B_0 (e.g., unit matrix I).
  Then at the k-th iteration of the simplex algorithm
  the inverse matrix is $B^{-1} = E_k E_{k-1} \cdots E_2 E_1 B_0^{-1}$,
  where E_i is the E matrix of the i-th iteration

- Each E matrix can be stored compactly (vector η + column index p)

- We can represent B^{-1} as the list $E_k E_{k-1} \cdots E_2 E_1$, B_0^{-1}:
  Product Form of the Inverse (PFI)

- When the list is long we reset: the inverse is computed (reinversion)

- Other ways of representing B^{-1}: LU factoritzation

Bounded Variables

- Now we want to revise the simplex algorithm for LP's of the form

$$
\begin{aligned}
&\min c^T x \\
&Ax = b \\
&\ell \le x \le u \qquad (-\infty \le \ell_k \le u_k \le +\infty)
\end{aligned}
$$

- In practice, internally variables are translated so that $\ell, u = -\infty$ or $\ell = 0$ or $u = 0$ to save arithmetic operations

- Variable x_k is lower bounded if $\ell_k > -\infty$
- Variable x_k is upper bounded if $u_k < +\infty$
- Variable x_k is free if $\ell_k = -\infty$ and $u_k = +\infty$

Basic Solution

- For any basis B, recall that
values of basic variables can be expressed in terms of non-basic variables:

$$x_{\mathcal{B}} = B^{-1}b - B^{-1}Rx_{\mathcal{R}}$$

- The values of a non-basic variable x_j can be:

If lower bounded:	ℓ_k	(we say it is at lower bound;	denoted by $x_j \in \mathcal{L}$)
If upper bounded:	u_k	(we say it is at upper bound;	denoted by $x_j \in \mathcal{U}$)
If free:	0	(we say it is at zero level;	denoted by $x_j \in \mathcal{Z}$)

- Basic solution:

$$\beta := B^{-1}b - \sum_{j\in\mathcal{L}} B^{-1}a_j\ell_j - \sum_{j\in\mathcal{U}} B^{-1}a_j u_j$$

Optimality Condition

- Recall the cost function in terms of non-basic variables:

$$c^T x = c_{\mathcal{B}}^T B^{-1} b + d_{\mathcal{R}} x_{\mathcal{R}}$$

  where $d_j = c_j - c_{\mathcal{B}}^T B^{-1} a_j$ for all variable x_j

- If $x_j \in \mathcal{L}$: cannot improve if $d_j \geq 0$
- If $x_j \in \mathcal{U}$: cannot improve if $d_j \leq 0$
- If $x_j \in \mathcal{Z}$: cannot improve if $d_j = 0$
- Optimality condition: no improving non-basic variable

Objective Function

- Recall the cost function in terms of non-basic variables:

$$c^T x = c_{\mathcal{B}}^T B^{-1} b + d_{\mathcal{R}} x_{\mathcal{R}}$$

where $d_j = c_j - c_{\mathcal{B}}^T B^{-1} a_j$ for all variable x_j

- Value z of cost function at current basic solution:

$$z = c_{\mathcal{B}}^T B^{-1} b + \sum_{j \in \mathcal{L}} d_j \ell_j + \sum_{j \in \mathcal{U}} d_j u_j$$

Improving a Non-Optimal Solution

- Let x_q be a non-basic variable
  that can improve objective value by increasing its value.

  This can happen when

  - x_q is lower bounded and $x_q \in \mathcal{L}$; or
  - x_q is free (so $x_q \in \mathcal{Z}$)

- Since increasing x_q can improve objective value: $d_q < 0$
- Let $t \geq 0$ be difference of new value x_q wrt old value

$$\begin{aligned} & x_{\mathcal{B}}(t) \\ = \; & B^{-1}b - B^{-1}Rx_{\mathcal{R}}(t) \\ = \; & B^{-1}b - tB^{-1}a_q - \textstyle\sum_{j\in\mathcal{L}} B^{-1}a_j\ell_j - \textstyle\sum_{j\in\mathcal{U}} B^{-1}a_j u_j \\ = \; & \beta - t\alpha_q \end{aligned}$$

where $\beta = B^{-1}b - \sum_{j\in\mathcal{L}} B^{-1}a_j\ell_j - \sum_{j\in\mathcal{U}} B^{-1}a_j u_j$, $\alpha_q = B^{-1}a_q$

Improving a Non-Optimal Solution

- How does the objective value change as a function of t?

$$
\begin{aligned}
&z(t) = \\
&c^T x(t) = \\
&c_{\mathcal{B}}^T x_{\mathcal{B}}(t) + c_{\mathcal{R}}^T x_{\mathcal{R}}(t) = \\
&c_{\mathcal{B}}^T x_{\mathcal{B}}(t) + tc_q + \sum_{j \in \mathcal{L}} c_j \ell_j + \sum_{j \in \mathcal{U}} c_j u_j = \\
&c_{\mathcal{B}}^T B^{-1} b + \sum_{j \in \mathcal{L}} (c_j - c_{\mathcal{B}}^T B^{-1} a_j) \ell_j + \sum_{j \in \mathcal{U}} (c_j - c_{\mathcal{B}}^T B^{-1} a_j) u_j + tc_q - tc_{\mathcal{B}}^T \alpha_q = \\
&c_{\mathcal{B}}^T B^{-1} b + \sum_{j \in \mathcal{L}} d_j \ell_j + \sum_{j \in \mathcal{U}} d_j u_j + tc_q - tc_{\mathcal{B}}^T \alpha_q = \\
&z + tc_q - tc_{\mathcal{B}}^T \alpha_q = \\
&z + tc_q - tc_{\mathcal{B}}^T B^{-1} a_q = \\
&z + td_q
\end{aligned}
$$

- Hence the improvement in cost is $\Delta z = z(t) - z = td_q$

Improving a Non-Optimal Solution

- Basic variables have indices $\mathcal{B} = (k_1, ..., k_m)$
- Let $i \in \{1, ..., m\}$. The i-th basic variable is x_{k_i}
- Value of x_{k_i} as a function of t is the i-th component of $x_{\mathcal{B}}(t)$: $\beta_i - t\alpha_q^i$
- Let $\lambda_i := \ell_{k_i}$, $\mu_i := u_{k_i}$

  We need $\lambda_i \leq \beta_i - t\alpha_q^i \leq \mu_i$

  - If $\alpha_q^i > 0$: $\beta_i - t\alpha_q^i \geq \lambda_i \quad \Rightarrow \quad \frac{\beta_i - \lambda_i}{\alpha_q^i} \geq t$
  - If $\alpha_q^i < 0$: $\beta_i - t\alpha_q^i \leq \mu_i \quad \Rightarrow \quad \frac{\beta_i - \mu_i}{\alpha_q^i} \geq t$

- But we need $x_q(t) \leq u_q$ too!
- Best improvement achieved with

$$\theta := \min(\quad u_q - \ell_q, \quad \min\{\tfrac{\beta_i - \lambda_i}{\alpha_q^i} \mid \alpha_q^i > 0\}, \quad \min\{\tfrac{\beta_i - \mu_i}{\alpha_q^i} \mid \alpha_q^i < 0\})$$

- If $\theta = +\infty$ we have unboundedness
- Else if $\theta = u_q - \ell_q$ we have a bound flip: no pivot needed!

Improving a Non-Optimal Solution

$$\theta := \min(\quad u_q - \ell_q, \quad \min\{\tfrac{\beta_i - \lambda_i}{\alpha_q^i} \mid \alpha_q^i > 0\}, \quad \min\{\tfrac{\beta_i - \mu_i}{\alpha_q^i} \mid \alpha_q^i < 0\})$$

- Assume $\theta \neq +\infty$, $\theta \neq u_q - \ell_q$.
- Thus variable x_q enters the basis and variable x_{k_p} leaves
- If $\theta = \min\{\frac{\beta_i - \lambda_i}{\alpha_q^i} \mid \alpha_q^i > 0\}$ then x_{k_p} leaves the basis at lower bound
- If $\theta = \min\{\frac{\beta_i - \mu_i}{\alpha_q^i} \mid \alpha_q^i < 0\}$ then x_{k_p} leaves the basis at upper bound

Improving a Non-Optimal Solution

- Let x_q be a non-basic variable
  that can improve objective value by decreasing its value.

  This can happen when

  - x_q is upper bounded and $x_q \in \mathcal{U}$; or
  - x_q is free (so $x_q \in \mathcal{Z}$)

- Since decreasing x_q can improve objective value: $d_q > 0$
- Let $t \leq 0$ be the difference of new value x_q wrt old value
- Again $x_{\mathcal{B}}(t) = \beta - t\alpha_q$
- Again the improvement in cost is $\Delta z = z(t) - z = td_q$

Improving a Non-Optimal Solution

- Basic variables have indices $\mathcal{B} = (k_1, \ldots, k_m)$
- Let $i \in \{1, \ldots, m\}$. The i-th basic variable is x_{k_i}
- Value of x_{k_i} as a function of t is the i-th component of $x_{\mathcal{B}}(t)$: $\beta_i - t\alpha_q^i$
- Let $\lambda_i := \ell_{k_i}$, $\mu_i := u_{k_i}$

  We need $\lambda_i \leq \beta_i - t\alpha_q^i \leq \mu_i$

  - If $\alpha_q^i > 0$: $\beta_i - t\alpha_q^i \leq \mu_i \quad \Rightarrow \quad \frac{\beta_i - \mu_i}{\alpha_q^i} \leq t$
  - If $\alpha_q^i < 0$: $\beta_i - t\alpha_q^i \geq \lambda_i \quad \Rightarrow \quad \frac{\beta_i - \lambda_i}{\alpha_q^i} \leq t$

- But we need $\ell_q \leq x_q(t)$ too!
- Best improvement achieved with

$$\theta := \max\left(\quad \ell_q - u_q, \quad \max\{\tfrac{\beta_i - \lambda_i}{\alpha_q^i} \mid \alpha_q^i < 0\}, \quad \max\{\tfrac{\beta_i - \mu_i}{\alpha_q^i} \mid \alpha_q^i > 0\}\right)$$

- If $\theta = -\infty$ we have unboundedness
- Else if $\theta = \ell_q - u_q$ we have a bound flip: no pivot needed!

Improving a Non-Optimal Solution

$$\theta := \max\left(\quad \ell_q - u_q, \quad \max\{\tfrac{\beta_i - \lambda_i}{\alpha_q^i} \mid \alpha_q^i < 0\}, \quad \max\{\tfrac{\beta_i - \mu_i}{\alpha_q^i} \mid \alpha_q^i > 0\}\right)$$

- Assume $\theta \neq -\infty$, $\theta \neq \ell_q - u_q$.
- Thus variable x_q enters basis and variable x_{k_p} leaves
- If $\theta = \max\{\frac{\beta_i - \lambda_i}{\alpha_q^i} \mid \alpha_q^i < 0\}$, x_{k_p} leaves basis at lower bound
- If $\theta = \max\{\frac{\beta_i - \mu_i}{\alpha_q^i} \mid \alpha_q^i > 0\}$, x_{k_p} leaves basis at upper bound

Update

- New objective value: $\bar{z} = z + \theta d_q$
- If bound flip
  - Flip status of x_q (i.e., $\bar{x}_q \in \mathcal{L} \Leftrightarrow x_q \in \mathcal{U}$)
  - New basic solution: $\bar{\beta} = \beta - \theta \alpha_q$
- Else
  - New basic indices: $\bar{\mathcal{B}} = (k_1, \ldots, k_{p-1}, q, k_{p+1} \ldots, k_m)$
  - New basic solution: $\bar{\beta}_p = x_q + \theta$, $\bar{\beta}_i = \beta_i - \theta \alpha_q^i$ if $i \neq p$
  - New basis inverse: $\bar{B}^{-1} = EB^{-1}$
  - If entering variable comes from lower bound
    $\hat{\mathcal{L}} = \mathcal{L} - \{x_q\}$ else $\hat{\mathcal{U}} = \mathcal{U} - \{x_q\}$
  - If leaving variable leaves to lower bound
    $\bar{\mathcal{L}} = \hat{\mathcal{L}} \cup \{x_{k_p}\}$ else $\bar{\mathcal{U}} = \hat{\mathcal{U}} \cup \{x_{k_p}\}$

Tableau vs. Revised Simplex

- Time is saved:
  - ✗ **Tableau:** all d_k, all α_k are computed
  - ✓ **Revised:** no. of non-basic variables x_k for which d_k, α_k are computed can be adjusted
- Space is saved:
  - ✗ **Tableau:** even if A sparse, tableau tends to get filled
  - ✓ **Revised:** sparsity of A can be exploited for storage, and pivots can be chosen to represent B^{-1} compactly
- Better numerical behaviour:
  - ✗ **Tableau:** errors due to floating-point arithmetic accumulate at each pivoting step
  - ✓ **Revised:** reinversion (PFI representation of B^{-1}) or refactorization (LU representation of B^{-1}) can be used for resetting

Original vs. Bounds Simplex

- Time is saved:
  - ✗ **Original:** no special treatment of bounds
  - ✓ **Bounds:** bound flips are much cheaper than pivoting steps in simplex iterations (basis does not change)

- Space is saved:
  - ✗ **Original:** each bound constraint becomes a row
  - ✓ **Bounds:** bounds are stored cheaply in arrays