The Simplex Method

Eklavya Sharma

This document describes the *simplex method* for solving linear programs.

1 Preliminaries

Theorem 1. *Any linear programming problem can be reduced to the following problem (called a* standard form linear program*):*

$$\min_{x \in \mathbb{R}^n} c^T x \text{ where } Ax = b \text{ and } x \geq 0.$$

Here $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$.

We will also assume without loss of generality that $\operatorname{rank}(A) = m$.

Lemma 2. *Let $B = [u_1, u_2, \ldots, u_n]$ be a basis of a vector space V. Let $w = \sum_{i=1}^n \lambda_i u_i$. Then $B' = B - \{u_r\} \cup \{w\}$ is a basis of V iff $\lambda_r \neq 0$.*

1.1 Notation

For any non-negative integer n, let $[n] := \{1, 2, \ldots, n\}$ (or $[n] := [1, 2, \ldots, n]$, depending on the context).

Definition 1. *Let* $\operatorname{stdLP}(A, b, c)$ *denote this LP:*

$$\min_{x \geq 0} c^T x \quad \text{where} \quad Ax = b.$$

2 Bases

Consider this linear program:

$$\min_{x \in \mathbb{R}^n} c^T x \text{ where } Ax = b \text{ and } x \geq 0.$$

Here $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$.

Definition 2 (Basis)**.** *Let J be a sequence of m distinct numbers from $[n]$. Let $B := A[*, J]$. Then J is called a* basis *of the LP iff* $\operatorname{rank}(B) = m$. *$J$ is called a feasible basis iff it is a basis and $B^{-1}b \geq 0$.*

Let $\overline{J}$ be the increasing sequence of values of $[n]$ that are not in J. Define `solve`(J) *as a vector $\widehat{x} \in \mathbb{R}^n$, where $\widehat{x}[J] = B^{-1}b$ and $\widehat{x}[\overline{J}] = 0$.*

The following two results show that to find an optimal BFS of the LP, we can find a feasible basis J that minimizes $c^T \texttt{solve}(J)$, and then return $\texttt{solve}(J)$.

Lemma 3. *Let J be a feasible basis and $\widehat{x} = \texttt{solve}(J)$. Then $\widehat{x}$ is a BFS of the LP.*

Lemma 4. *Let $\widehat{x}$ be a BFS of the LP. Then there exists a feasible basis J such that $\widehat{x} = \texttt{solve}(J)$.*

3 The Simplex Algorithm

See Algorithm 1 for the description of the simplex algorithm. The input to the algorithm is (A, b, c, J), where J is a feasible basis of $Ax = b$. The algorithm initializes a data structure D using J (by calling the subroutine `simplexInit`), and then iteratively updates J and the data structure D (by calling subroutines `simplexMove` and `updateDS`). Specifically, if the `status` output by `simplexMove` is `move`, then it outputs a pair (k, r) of integers, where $k \in [n] - J$ and $r \in [m]$. It then sets the r^{th} element of J to k. We say that $J[r]$ *leaves the basis* and k *enters the basis*.

Algorithm 1 $\texttt{simplex}(A, b, c, J)$: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and J is a feasible basis for $\text{stdLP}(A, b, c)$.

```
// contains some Python assignment syntax
D = simplexInit(A, b, c, J)
while true do
    status, *outs = simplexMove(D, J)
    // status can be optimal, unbounded, or move.
    // outs is a list
    if status == move then
        (k, r, δ) = outs
        J[r] = k
        D = updateDS(D, J, k, r)
    else
        return (status, J, *outs)
    end if
end while
```

There are different variants of the simplex algorithm, depending on what data structure D they maintain. We will look at 3 variants: naive simplex, tableau simplex, and revised simplex. In the *naive simplex method*, we set $D := (A, b, c)$. Hence, `simplexInit` and `updateDS` are trivial for naive simplex. The main advantage of tableau and revised over naive is that they speed up `simplexMove`.

Definition 3. *Let $J := [j_1, \ldots, j_m]$ be a basis of* $\text{stdLP}(A, b, c)$, *where $A \in \mathbb{R}^{m\times n}$, and let $k \in [n] - J$. Let $B := A[*, J]$ and $Y := B^{-1}A$. Then define* $\text{direction}(J, k) \in \mathbb{R}^n$ *as the vector y where*

$$y_t = \begin{cases} -Y[i, k] & \textit{if } t = j_i \\ 1 & \textit{if } t = k \\ 0 & \textit{otherwise} \end{cases}.$$

The core of the simplex algorithm is `simplexMove`, which tells us how to move from one basis to another. `simplexMove` is described in Algorithm 2. Specifically, when `simplexMove`(D, J) outputs $(\texttt{move}, k, r, \delta)$, it moves from `solve`$(J)$ to `solve`$(J) + \delta\,\mathrm{direction}(J, k)$ (we will prove this soon).

Algorithm 2 `simplexMove`(D, J): J is a feasible basis of $\mathrm{stdLP}(A, b, c)$.

Let $B := A[*, J]$, $Y := B^{-1}A$, $\overline{b} := B^{-1}b$, and $z = Y^T c[J]$.
// We will lazily compute B, Y, $\overline{b}$, and z using D.
if $c - z \geq 0$ **then**
    return $(\texttt{optimal}, \texttt{solve}(J), c[J]^T\overline{b})$
end if
Find $k \in [n]$ such that $c_k - z_k < 0$.
if $Y[*, k] \leq 0$ **then**
    return $(\texttt{unbounded}, \texttt{solve}(J), \mathrm{direction}(J, k), k)$
end if
$r = \underset{i \in [m] : Y[i,k] > 0}{\operatorname{argmin}} \dfrac{\overline{b}_i}{Y[i, k]}$
$\delta = \overline{b}_r / Y[r, k]$
return $(\texttt{move}, k, r, \delta)$.

Since `simplexMove` requires J to be a feasible basis of $\mathrm{stdLP}(A, b, c)$, and we're changing J in line 9, we need to prove that after this change, J continues to be a feasible basis of $\mathrm{stdLP}(A, b, c)$.

Theorem 5. *If* `simplex` *outputs* $(\texttt{optimal}, J, \widehat{x}, \beta)$, *then $\widehat{x}$ is a BFS of the LP and an optimal solution to the LP. Furthermore, $\widehat{x} =$* `solve`(J) *and $\beta = c^T\widehat{x}$.*

Proof sketch. For any feasible x, we can show that $c^T x = c[J]^T\overline{b} + (c - z)[\overline{J}]^T x[\overline{J}]$. Since $c[J]^T\overline{b} = c^T\widehat{x}$, $x[\overline{J}] \geq 0$, and $c - z \geq 0$, we get $c^T x \geq c^T\widehat{x}$. $\square$

Lemma 6. *$z[J] = c[J]$.*

Proof. $z[J]^T = c[J]^T(B^{-1}A)[*, J] = c[J]^T B^{-1}A[*, J] = c[J]^T$. $\square$

Lemma 6 implies that $k \notin J$, since $c_k - z_k < 0$.

Lemma 7. *$Y[*, J] = I$. Let $J = [j_1, j_2, \ldots, j_m]$. Then $Y[i, j_p] = \begin{cases} 1 & \text{if } p = i \\ 0 & \text{if } p \neq i \end{cases}$.*

Lemma 8. *Let $y = \mathrm{direction}(J, k)$. Then $Yy = Ay = 0$.*

Lemma 9. *Let $y := \mathrm{direction}(J, k)$. Then $c^T y = c_k - z_k$.*

Theorem 10. *If* `simplex` *outputs $(unbounded, J, \widehat{x}, y, k)$, then the LP's cost reduces along the ray $\{\widehat{x} + \lambda y : \lambda \geq 0\}$ and the ray is feasible, which implies that the LP is unbounded. Furthermore, $y \geq 0$, $\widehat{x} =$* `solve`(J), *and $y = \mathrm{direction}(J, k)$.*

Lemma 11. *Suppose* `simplexMove`(D, J) *outputs* $(\texttt{move}, k, r, \delta)$. *Let $\widetilde{J}$ be the new sequence obtained by changing $J[r]$ to k (at line 9 of* `simplex`*). Then $\widetilde{J}$ is a basis of the LP.*

Proof. Let $J = [j_1, j_2, \ldots, j_m]$. The set of values in $\widetilde{J}$ is $J - \{j_r\} \cup \{k\}$. Since $k \notin J$, $\widetilde{J}$ has distinct values.

Let a_j be the j^{th} column of A. Let $B = A[*, J]$. Let $\widetilde{B} = A[*, \widetilde{J}]$. Let $S = \{a_{j_1}, a_{j_2}, \ldots, a_{j_m}\}$ be the set of columns of B and let $\widetilde{S} = S - \{a_{j_r}\} \cup \{a_k\}$ be the set of columns of $\widetilde{B}$. Since J is a basis, $\operatorname{rank}(B) = m$, so S is a set of linearly independent vectors. Since $|S| = m$, we get that S is a basis of $\mathbb{R}^m$. Hence, $a_k \in \operatorname{span}(S)$.

Let $a_k = \sum_{i=1}^{m} \lambda_i a_{j_i}$. Let $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_m]$. Then $B\lambda = \sum_{i=1}^{m} \lambda_i a_{j_i} = a_k$. Hence, $\lambda = B^{-1}a_k = Y[*, k]$. Since $Y[r, k] > 0$, we get that $\lambda_r > 0$. Hence, by Lemma 2, we get that $\widetilde{S}$ is also a basis of $\mathbb{R}^m$. Hence, $\operatorname{rank}(\widetilde{B}) = m$, so $\widetilde{J}$ is a basis. □

Lemma 12. *Suppose* `simplexMove`(D, J) *outputs* $(\texttt{move}, k, r, \delta)$. *Let* $\widetilde{J}$ *be the new sequence obtained by changing* $J[r]$ *to* k *(at line 9 of* `simplex`*). Then* $\widetilde{J}$ *is a feasible basis of the LP. Furthermore, let* $y = \operatorname{direction}(J, k)$, $\widehat{x} = \texttt{solve}(J)$, *and* $\widetilde{x} = \widehat{x} + \delta y$. *Then* $\widetilde{x} = \texttt{solve}(\widetilde{J})$ *and* $c^T\widetilde{x} \leq c^T\widehat{x}$.

Proof sketch. We can show that $A\widetilde{x} = b$, $\widetilde{x} \geq 0$, and $\widetilde{x}_j = 0$ when $j \notin \widetilde{J}$. Let $\widetilde{B} := A[*, \widetilde{J}]$. Then $b = A\widetilde{x} = A[*, \widetilde{J}]\widetilde{x}[\widetilde{J}] = \widetilde{B}\widetilde{x}[\widetilde{J}]$. So, $\widetilde{x}[\widetilde{J}] = \widetilde{B}^{-1}b$, which implies $\widetilde{x} = \texttt{solve}(\widetilde{J})$. Also, $c^T(\widetilde{x} - \widehat{x}) = \delta(c^T y) = \delta(c_k - z_k) \leq 0$ by Lemma 9. □

This completes the correctness of `simplex`.

4 Implementations of Simplex

The naive simplex method has a large running time of $O(m^2(m+n))$ per iteration, since we compute B^{-1}, Y, $\bar{b}$ and z afresh in each iteration. We will now see how the tableau method and the revised simplex method improve the running time per iteration.

In the Tableau method, the data structure D is

$$\begin{bmatrix} c - z & -c[J]^T\bar{b} \\ Y & \bar{b} \end{bmatrix},$$

where the rows are numbered from 0 instead of 1. In the Revised simplex method, the data structure D is given by the pair (D_1, D_2), where $D_1 := (A, b, c)$ and

$$D_2 := \begin{bmatrix} -c[J]^T B^{-1} & -c[J]^T\bar{b} \\ B^{-1} & \bar{b} \end{bmatrix},$$

where the rows are numbered from 0 instead of 1. It is easy to see that we can quickly compute Y, $\bar{b}$, and $c - z$ in `simplexMove` in both methods. `simplexInit` is implemented in the obvious straightforward way. We will now see how to implement `updateDS` using elementary row operations.

Definition 4 (pivoting)**.** *Let* $A \in \mathbb{R}^{m\times n}$ *be a matrix,* $i \in [m]$, *and* $j \in [n]$ *such that* $A[i, j] \neq 0$. *Then pivoting is the operation of applying elementary row operations to* A *to get a new matrix* $\widehat{A} \in \mathbb{R}^{m\times n}$ *such that* $\widehat{A}[i, j] = 1$ *and* $\widehat{A}[i', j] = 0$ *for all* $i' \in [m] - \{i\}$.

In the tableau method, $\texttt{updateDS}(D, J, k, r)$ is performed by pivoting D at (r, k). In the revised simplex method, $\texttt{updateDS}(D, J, k, r)$ is performed by horizontally concatenating the column $\begin{bmatrix} c_k - z_k \\ Y[*, k] \end{bmatrix}$ to D_2, (which becomes the $(m+2)^{\text{th}}$ column), pivoting at $(r, m + 2)$, and then discarding the $(m + 2)^{\text{th}}$ column.

Let J be a feasible basis of the LP. Let $B := A[*, J]$, $Y := B^{-1}A$, $\overline{b} := B^{-1}b$ and $z := Y^T c[J]$. Based on how k and r are chosen, we know that $c_k - z_k < 0$, $Y[r, k] > 0$, and $r \in \operatorname{argmin}_{i \in [m]: Y[i,k] > 0} \frac{\overline{b}_i}{Y[i,k]}$. Let $\widetilde{J}$ be the sequence obtained by changing the r^{th} element of J to k. By Lemma 12, $\widetilde{J}$ is a feasible basis. Let $\widetilde{B} := A[*, \widetilde{J}]$, $\widetilde{Y} := \widetilde{B}^{-1}A$, $\overline{\overline{b}} := \widetilde{B}^{-1}b$ and $\widetilde{z} := \widetilde{Y}^T c[\widetilde{J}]$. We will now see how to compute $\widetilde{Y}$, $\widetilde{z}$ and $\overline{\overline{b}}$ from Y, z and $\overline{b}$.

Define the matrix $\widehat{Y}$ as

$$\widehat{Y}[i, j] = \begin{cases} \dfrac{Y[r, j]}{Y[r, k]} & \text{if } i = r \\[2ex] Y[i, j] - \dfrac{Y[i, k]}{Y[r, k]} Y[r, j] & \text{if } i \neq r \end{cases}.$$

Note that $\widehat{Y}$ is obtained from Y by pivoting on (r, k). Let R be the matrix of these row operations. Then $\widehat{Y} = RY$. We can find R by applying these row operations to the m-by-m identity matrix.

$$\begin{aligned} R[i, j] &= \begin{cases} \dfrac{I[r, j]}{Y[r, k]} & \text{if } i = r \\[2ex] I[i, j] - \dfrac{Y[i, k]}{Y[r, k]} I[r, j] & \text{if } i \neq r \end{cases} \\ &= \begin{cases} \dfrac{1}{Y[r, k]} & \text{if } i = r = j \\[2ex] -\dfrac{Y[i, k]}{Y[r, k]} & \text{if } i \neq r \wedge j = r \\ 1 & \text{if } i \neq r \wedge j = i \\ 0 & \text{if } j \notin \{i, r\} \end{cases}. \end{aligned}$$

Lemma 13. *$\widetilde{B}^{-1} = RB^{-1}$ and $\widetilde{Y} = RY$ and $\overline{\overline{b}} = R\overline{b}$.*

Define $\widehat{z} \in \mathbb{R}^n$ and η as

$$\widehat{z}_j = z_j + \frac{c_k - z_k}{Y[r, k]} Y[r, j] \qquad\qquad \eta = c[J]^T \overline{b} + \frac{c_k - z_k}{Y[r, k]} \overline{b}_r.$$

Lemma 14. *$\widehat{z} = \widetilde{z}$ and $\eta = c[\widetilde{J}]^T \overline{\overline{b}}$.*

In the revised simplex method, we can obtain further speedup in $\texttt{simplexMove}$. Compute $c[J]^T B^{-1}$ by multiplying $c[J]^T$ and B^{-1}. Then we iterate over $j \in [n] - \widetilde{J}$, and compute $z_j = (c[J]^T B^{-1}) A[*, j]$. We stop iterating when we find a suitable $k \in [n] - \widetilde{J}$ such that $c_k - z_k < 0$, or if $c_j - z_j \geq 0$ for all $j \in [n] - \widetilde{J}$. Next, we compute $u = B^{-1}A[*, k]$ and $\overline{b} = B^{-1}b$. At the end of the iteration, we can update B^{-1} using row operations as per Lemma 13. This is possible since R is defined by u.

The time taken is $O(m(t+m))$, where t is the number of variables that need to be considered till we find k. Note that $t \leq n - m$. The space complexity of revised simplex (in addition to storing the input) is $O(m^2)$.

5 Duality

Definition 5 (Dual LP)**.** *The dual LP of* stdLP(A, b, c) *is defined to be the following LP:*

$$\max_w b^T w \quad \textit{where} \quad A^T w \leq c.$$

We denote this LP as stdDLP(A, b, c)*.*

Definition 6 (dual feasible basis)**.** *Let J be a basis of* stdLP(A, b, c)*. J is called* dual feasible *if $c - z \geq 0$, where $B := A[*, J]$ and $z^T := c[J]^T B^{-1} A$. Define* `dualSolve`(J) *as* $(c[J]^T B^{-1})^T$*. (Note that $z = A^T$* `dualSolve`(J)*).*

Lemma 15. *Let J be a dual feasible basis and $\widehat{w} :=$* `dualSolve`(J)*. Then $\widehat{w}$ is a BFS of* stdDLP(A, b, c)*.*

Lemma 16. *Let $\widehat{w}$ be a BFS of* stdDLP(A, b, c)*. Then there exists a dual feasible basis J of* stdLP(A, b, c) *such that $\widehat{w} =$* `dualSolve`(J)*.*

Lemma 17. *Let J be a basis of* stdLP(A, b, c)*. Let $\widehat{x} :=$* `solve`(J) *and $\widehat{w} :=$* `dualSolve`(J)*. Then $c^T \widehat{x} = b^T \widehat{w} = c[J]^T \overline{b}$. Furthermore, if J is both feasible and dual feasible, then $\widehat{x}$ and $\widehat{w}$ are optimal solutions to* stdLP(A, b, c) *and* stdDLP(A, b, c)*, respectively.*

6 Properties of Solutions

Definition 7 (degeneracy)**.** *Let $A \in \mathbb{R}^{m \times n}$. Let J be a basis of* stdLP(A, b, c)*. Let $B := A[*, J]$ and $z^T := c[J]^T B^{-1} b$.*

- *A solution $\widehat{x}$ to $Ax = b$ is called* degenerate *for* stdLP(A, b, c) *if* $|\operatorname{support}(\widehat{x})| < m$*.*
- *$\widehat{w} \in \mathbb{R}^m$ is called* degenerate *for* stdDLP(A, b, c) *if* $|\operatorname{support}(c - A^T w)| < n - m$*.*
- *J is called* primal degenerate *if $(B^{-1} b)_i = 0$ for some $i \in [m]$.*
- *J is called* dual degenerate *if $(c - z)_j = 0$ for some $j \in [n] - J$.*

Lemma 18. *Let J be a basis of* stdLP(A, b, c)*. Then* `solve`(J) *is degenerate iff J is primal degenerate, and* `dualSolve`(J) *is degenerate iff J is dual degenerate.*

6.1 Multiple Bases for Same Point

Lemma 19. *Let J_1 and J_2 be two bases of* stdLP(A, b, c) *such that* `sorted`$(J_1) \neq$ `sorted`(J_2) *and $\widehat{x} :=$* `solve`$(J_1) =$ `solve`(J_2)*. Then $\widehat{x}$ is degenerate for* stdLP(A, b, c)*.*

Lemma 20. *Let J_1 and J_2 be two bases of* stdLP(A, b, c) *such that* `sorted`$(J_1) \neq$ `sorted`(J_2) *and $\widehat{w} :=$* `dualSolve`$(J_1) =$ `dualSolve`(J_2)*. Then $\widehat{w}$ is degenerate for* stdDLP(A, b, c)*.*

The converse of Lemmas 19 and 20 is not true.

Example 1. *Let* $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, $b = [0,0]^T$, *and* $c = [0,0,0]^T$. *Then* $J = [0,1]$ *is the unique basis (up to permutation) of* $\mathrm{stdLP}(A,b,c)$. *However, both* $\texttt{solve}(J) = [0,0,0]$ *and* $\texttt{dualSolve}(J) = [0,0]$ *are degenerate.*

6.2 Degeneracy and Optimality

Lemma 21 (dual non-degen $\implies$ unique primal opt)**.** *Let* J *be a dual feasible and dual non-degenerate basis of* $\mathrm{stdLP}(A,b,c)$. *Let* $\widehat{x} := \texttt{solve}(J)$. *Let* P *be the set of feasible solutions to* $\mathrm{stdLP}(A,b,c)$. *Then* $c^T\widehat{x} < \min_{x \in P - \{\widehat{x}\}} c^T x$. *(Hence, if* J *is feasible, then* $\widehat{x}$ *is a unique optimum of* $\mathrm{stdLP}(A,b,c)$.*)*

Proof sketch. For any $x \in P$, we can show that $c^T x = c[J]^T\overline{b} + (c-z)[\overline{J}]^T x[\overline{J}]$. Since $c[J]^T\overline{b} = c^T\widehat{x}$, $x[\overline{J}] \geq 0$, $x[\overline{J}] \neq 0$ (since $x \neq \widehat{x}$), and $(c-z)[\overline{J}] > 0$ (by dual feasibility and dual non-degeneracy of J), we get $c^T x > c^T\widehat{x}$. □

Lemma 22 (primal non-degen $\implies$ unique dual opt)**.** *Let* J *be a primal feasible and primal non-degenerate basis of* $\mathrm{stdLP}(A,b,c)$. *Let* $\widehat{w} := \texttt{dualSolve}(J)$ *and* $\widehat{x} := \texttt{solve}(J)$. *Let* Q *be the set of feasible solutions to* $\mathrm{stdDLP}(A,b,c)$. *Then* $b^T\widehat{w} > \max_{w \in Q - \{\widehat{w}\}} b^T w$. *(Hence, if* J *is dual feasible, then* $\widehat{w}$ *is a unique optimum of* $\mathrm{stdDLP}(A,b,c)$.*)*

Proof. Let $w \in Q - \{\widehat{w}\}$. So, $c^T - w^T A \geq 0$. Suppose $(c^T - w^T A)[J] = 0$. Then $w^T = B^{-1}c[J] = \widehat{w}$, which is not possible. Hence, $\exists j \in J$ such that $c_j - (w^T A)_j > 0$.

We have $b^T w = w^T A\widehat{x} = (w^T A)[J]\overline{b}$ and $b^T\widehat{w} = c[J]^T\overline{b}$. Since J is feasible and primal non-degenerate, $\overline{b} > 0$. Hence, $b^T\widehat{w} - b^T w = (c[J] - w^T A)[J]\overline{b} \geq (c_j - (w^T A)_j)\overline{b}_j > 0$. □

Lemma 23 (primal non-degen and dual degen $\implies$ non-unique primal opt)**.** *Let* J *be a feasible basis of* $\mathrm{stdLP}(A,b,c)$ *that is primal non-degenerate and dual degenerate. Let* $\widehat{x} := \texttt{solve}(J)$. *Then* $\exists$ *a feasible solution* $\widetilde{x}$ *to* $\mathrm{stdLP}(A,b,c)$ *such that* $\widetilde{x} \neq \widehat{x}$ *and* $c^T\widetilde{x} = c^t\widehat{x}$.

Proof sketch. Find k such that $c_k - z_k = 0$ and then try to pivot. □

Lemma 24 (primal degen and dual non-degen $\implies$ non-unique dual opt)**.** *Let* J *be a dual feasible basis of* $\mathrm{stdLP}(A,b,c)$ *that is primal degenerate and dual non-degenerate. Let* $\widehat{x} := \texttt{solve}(J)$ *and* $\widehat{w} := \texttt{solve}(J)$. *Then* $\exists$ *a dual feasible solution* $\widetilde{w}$ *to* $\mathrm{stdDLP}(A,b,c)$ *such that* $\widetilde{w} \neq \widehat{w}$ *and* $b^T\widetilde{w} = b^t\widehat{w}$.

Proof sketch. Find r such that $\overline{b}_r = 0$ and then try to pivot. □

Example 2. *Let* $b = 0$, $c = (0,0)$. *Let* J *be any basis of* $\mathrm{stdLP}(A,b,c)$ $(|J| = 1)$. *Let* $\widehat{x} := \texttt{solve}(J)$ *and* $\widehat{w} := \texttt{dualSolve}(J)$. $\overline{b} = B^{-1}b = 0$, *so* $\widehat{x} = (0,0)$, *which is feasible for* $\mathrm{stdLP}(A,b,c)$. $\widehat{w}^T = c[J]^T B^{-1} = 0$, *so* $\widehat{w} = 0$. $c - A^T\widehat{w} = (0,0)$, *so* $\widehat{w}$ *is feasible for* $\mathrm{stdDLP}(A,b,c)$. *Hence,* J *is primal feasible and dual feasible. Since* $\overline{b} = 0$, J *is primal degenerate. Since* $(c - A^T\widehat{w})[J] = 0$, J *is dual degenerate.*

Let P *and* Q *be the set of feasible solutions to the primal and dual LPs, respectively. Since the objective function is 0 for both LPs, unique primal optimal solution exists iff* $P = \{(0,0)\}$, *and unique dual optimal solution exists iff* $Q = \{0\}$.

- *If* $A = [1, 1]$, *then* $P = \{(0, 0)\}$ *and* $Q = (-\infty, 0]$.
- *If* $A = [1, -1]$, *then* $P = \{(x, x) : x \geq 0\}$ *and* $Q = \{0\}$.
- *If* $A = [1, 0]$, *then* $P = \{(0, y) : y \geq 0\}$ *and* $Q = (-\infty, 0]$.

Table 1: Unique primal optimum?

	dual degen	dual non-degen
primal degen	depends	yes
primal non-degen	no	yes

Table 2: Unique dual optimum?

	dual degen	dual non-degen
primal degen	depends	no
primal non-degen	yes	yes