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 $\operatorname{rank}(A) = m$.

Read the following concepts at TheoremDep (`https://sharmaeklavya2.github.io/theoremdep/`):

- Basic feasible solution (BFS)
- Extreme point of a convex set
- Extreme point iff BFS
- LP in orthant is optimized at BFS

Due to the last point above, we will focus on finding an optimal solution that is also a BFS.

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 $\lambda_r \neq 0$.*

Proof. (See `https://sharmaeklavya2.github.io/theoremdep/nodes/linear-algebra/vector-spaces/basis/replace-vector.html`.) □

Lemma 3. *For any matrix A, we have $\operatorname{rank}(A) = \operatorname{rank}(A^T)$.*

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).

Let $v \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$. Let $i \in [m]$ and $j \in [n]$. Then the j^{th} element of v is denoted as v_j or $v[j]$. The element of A in the i^{th} row and j^{th} column of A is denoted as $A_{i,j}$ or $A[i, j]$. $A[*, j]$ denotes the j^{th} column of A and $A[i, *]$ denotes the i^{th} row of A.

Let $J = [j_1, j_2, \dots, j_r]$ be a sequence of r integers in $[n]$. $v[J]$ is defined as the vector $[v[j_1], v[j_2], \dots, v[j_n]]$. $A[*, J]$ is defined as the matrix whose k^{th} column is $A[*, j_k]$. Let $K = [k_1, k_2, \dots, k_q]$ be a sequence of q integers in $[m]$. Then $A[K, *]$ is defined as the matrix whose i^{th} column is $A[k_i, *]$.

For matrices $A \in \mathbb{R}^{m \times n_1}$ and $B \in \mathbb{R}^{m \times n_2}$, let $C = [A, B]$ denote the matrix in $\mathbb{R}^{m \times (n_1 + n_2)}$ where the first n_1 columns in C are the columns of A and the last n_2 columns in C are the columns of B. We call C the *horizontal concatenation* of A and B. We can similarly define horizontal concatenation of more than two matrices. We can similarly define vertical concatenation of A and B, which we denote as $\begin{bmatrix} A \\ B \end{bmatrix}$.

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

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

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 $\texttt{solve}(J)$ *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 4. *Let J be a feasible basis and $\widehat{x} = \texttt{solve}(J)$. Then $\widehat{x}$ is a BFS of the LP.*

Proof. It's trivial to see that $\widehat{x} \geq 0$. Let $B = A[*, J]$ and $N = A[*, \overline{J}]$. Then

$$A\widehat{x} = B\widehat{x}[J] + N\widehat{x}[\overline{J}] = B(B^{-1} b) = b.$$

Hence, $\widehat{x}$ is feasible for the LP.

Because we can rearrange variables and constraints, we can assume without loss of generality that $J = [m]$. The equality constraints are tight, and their coefficient matrix is $A = [B, N]$. The non-negativity constraints $\{x_j \geq 0 : j \in \overline{J}\}$ are tight, and their coefficient matrix is $I_n[\overline{J}, *] = [0, I_{n-m}]$, where I_k denotes the k-by-k identity matrix. Hence, the rank of the coefficient matrix of tight constraints at $\widehat{x}$ is

$$\operatorname{rank}\left(\begin{bmatrix} B & N \\ 0 & I_{n-m} \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} B & 0 \\ 0 & I_{n-m} \end{bmatrix}\right) = \operatorname{rank}(B) + (n - m) = n.$$

The first equation follows from the fact that rank is unaffected by row operations. The third equation follows from the fact that J is a basis. Since the coefficient matrix of tight constraints of $\widehat{x}$ has rank n, $\widehat{x}$ is a BFS of the LP. □

Lemma 5. *Let $\widehat{x}$ be a BFS of the LP. Then there exists a feasible basis J such that $\widehat{x} = \texttt{solve}(J)$.*

Proof. Since $\widehat{x}$ is a BFS, there exist n linearly independent constraints that are tight at $\widehat{x}$. m of these are the equality constraints, whose coefficient matrix is A. The rest are inequality constraints. Let $\overline{J}$ be the indices of these $n - m$ inequality constraints. This implies $\widehat{x}[\overline{J}] = 0$. Since we can rearrange variables, assume without loss of generality that $\overline{J} = [m+1, m+2, \ldots, n]$. The coefficient matrix of these constraints is $I_n[\overline{J}, *] = [0, I_{n-m}]$.

Let $J = [m]$. Let $B = A[*, J]$ and $N = A[*, J]$. Then $A = [B, N]$. Since $\widehat{x}$ is a BFS, we get

$$n = \operatorname{rank}\left(\begin{bmatrix} B & N \\ 0 & I_{n-m} \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} B & 0 \\ 0 & I_{n-m} \end{bmatrix}\right) = \operatorname{rank}(B) + (n - m).$$

This implies that $\operatorname{rank}(B) = m$, which shows that J is a basis of the LP.

Furthermore, since $\widehat{x}$ is feasible for the LP, we get that $b = A\widehat{x} = B\widehat{x}[J] + N\widehat{x}[\overline{J}] = B\widehat{x}[J]$. Hence, $\widehat{x}[J] = B^{-1}b$. Since $\widehat{x}$ is feasible for the LP, we get $\widehat{x} \geq 0 \implies \widehat{x}[J] \geq 0 \implies B^{-1}b \geq 0$. Hence, J is a feasible basis and $\texttt{solve}(J) = \widehat{x}$. □

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 `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*

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

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`$(J) + \delta\,\text{direction}(J, k)$ (we will prove this soon).

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

1: Let $B := A[*, J]$, $Y := B^{-1}A$, $\overline{b} := B^{-1}b$, and $z = Y^Tc[J]$.
2: // *We will lazily compute B, Y, $\overline{b}$, and z using D.*
3: **if** $c - z \ge 0$ **then**
4:     **return** $(\texttt{optimal}, \texttt{solve}(J), c[J]^T\overline{b})$
5: **end if**
6: Find $k \in [n]$ such that $c_k - z_k < 0$.
7: **if** $Y[*, k] \le 0$ **then**
8:     **return** $(\texttt{unbounded}, \texttt{solve}(J), \text{direction}(J, k), k)$
9: **end if**
10: $r = \operatorname*{argmin}_{i\in[m]:Y[i,k]>0} \dfrac{\overline{b}_i}{Y[i,k]}$
11: $\delta = \overline{b}_r / Y[r, k]$
12: **return** $(\texttt{move}, k, r, \delta)$.

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

Theorem 6. *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 $c^Tx = c[J]^T\overline{b} + (c - z)[\overline{J}]^Tx[\overline{J}]$. Since $c[J]^T\overline{b} = c^T\widehat{x}$, $x[\overline{J}] \ge 0$, and $c - z \ge 0$, we get $c^Tx \ge c^T\widehat{x}$. □

Proof. By line 4 of `simplexMove`, $\widehat{x} =$ `solve`(J) and $\beta = c[J]^T\overline{b}$. Hence, $\widehat{x}$ is a BFS by Lemma 4 and $c^T\widehat{x} = \beta$. Now we just need to prove that $\widehat{x}$ is optimal.

Let $\overline{J} = [n] - J$. Let $N = A[*, \overline{J}]$. Let $x_B = x[J]$ and $x_N = x[\overline{J}]$. Then

$$Ax = b \iff Bx_B + Nx_N = b \iff x_B = \overline{b} - B^{-1}Nx_N.$$

Note that since the constraint $x_B = \overline{b} - B^{-1}Nx_N$ is equivalent to $Ax = b$, we can replace $Ax = b$ by $x_B = \overline{b} - B^{-1}Nx_N$ in the LP without affecting the set of feasible solutions.

We can use these new constraints to express the objective value as a function of x_N.

$$\begin{aligned} c^T x &= c[J]^T x_B + c[\overline{J}]^T x_N \\ &= c[J]^T \left(\overline{b} - B^{-1}Nx_N\right) + c[\overline{J}]^T x_N \\ &= c[J]^T\overline{b} + (c[\overline{J}]^T - c[J]^T B^{-1}N)x_N \end{aligned}$$

$$z[\overline{J}]^T = (c[J]^T Y)[\overline{J}] = c[J]^T B^{-1} A[*, \overline{J}] = c[J]^T B^{-1} N.$$
$$\implies c^T x = c[J]^T \overline{b} + (c - z)[\overline{J}]^T x_N.$$

From the non-negativity constraints, we know that $x_N \geq 0$. We also know that $c - z \geq 0$, since `simplexMove`'s output status is `optimal`. Hence, for every feasible x, we have $c^T x = c[J]^T\overline{b} + (c - z)[\overline{J}]^T x_N \geq c[J]^T\overline{b} = c^T\widehat{x}$. Hence, $\widehat{x}$ is an optimal solution to the LP. □

Lemma 7. $z[J] = c[J]$.

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

Lemma 7 implies that $k \notin J$, since $c_k - z_k < 0$.

Lemma 8. *$Y[*, J] = I$. Let $J = [j_1, j_2, \ldots, j_m]$. Then* $Y[i, j_p] = \begin{cases} 1 & \text{if } p = i \\ 0 & \text{if } p \neq i \end{cases}.$

Proof.

$$Y[*, J] = (B^{-1}A)[*, J] = B^{-1}A[*, J] = B^{-1}B = I.$$

$$Y[i, j_p] = Y[*, J][i, p] = I[i, p] = \begin{cases} 1 & \text{if } p = i \\ 0 & \text{if } p \neq i \end{cases}.$$ □

Lemma 9. *Let $y = \text{direction}(J, k)$. Then $Yy = Ay = 0$.*

Proof.

$$\begin{aligned} (Yy)_i &= \sum_{j=1}^{n} Y[i, j]y_j = \sum_{p=1}^{m} Y[i, j_p]y_{j_p} + Y[i, k]y_k \\ &= y_{j_i} + Y[i, k]y_k = -Y[i, k] + Y[i, k] = 0. \end{aligned}$$

$$Ay = B^{-1}Yy = B^{-1}0 = 0.$$ □

Lemma 10. *Let $y := \text{direction}(J, k)$. Then $c^T y = c_k - z_k$.*

Proof.

$$\begin{aligned} c^T y &= \sum_{j=1}^{n} c_j y_j = c_k y_k + \sum_{p=1}^{m} c_{j_p} y_{j_p} = c_k - \sum_{p=1}^{m} c_{j_p} Y[p, k] \\ &= c_k - \sum_{p=1}^{m} Y^T[k, p] c[J]_p = c_k - (Y^T c[J])_k = c_k - z_k < 0. \end{aligned}$$ □

Theorem 11. *If* `simplex` *outputs* $(\text{unbounded}, J, \widehat{x}, y, k)$*, 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,* $y \geq 0$*,* $\widehat{x} = \texttt{solve}(J)$*, and* $y = \operatorname{direction}(J, k)$*.*

Proof. By line Line 8 of `simplexMove`, we know that $\widehat{x} = \texttt{solve}(J)$ and $y = \operatorname{direction}(J, k)$.

By Lemma 9, we know that $Ay = 0$. Hence, $A(\widehat{x} + \lambda y) = A\widehat{x} = b$. Since `simplexMove` returned $(\text{unbounded}, \widehat{x}, y, k)$, we get that $Y[*, k] \leq 0$ (by Line 7). Hence, $y \geq 0$ and so $\widehat{x} + \lambda y \geq \widehat{x} \geq 0$. Hence, $\widehat{x} + \lambda y$ is feasible for the LP for all $\lambda \geq 0$.

By Lemma 10, we know that $c^T y = c_k - z_k < 0$, Hence, moving along the ray will reduce cost indefinitely. This implies that the LP is unbounded. □

Lemma 12. *Suppose* $\texttt{simplexMove}(D, J)$ *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 $k \not\in J$, $\widetilde{J}$ 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 13. *Suppose* $\texttt{simplexMove}(D, J)$ *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 $j \not\in \widetilde{J}$. 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 10. □

Proof. By Lemma 9, we get that $Ay = 0$. Hence, $A\widetilde{x} = A\widehat{x} + \delta(Ay) = A\widehat{x} = b$.

If $i \not\in J$ or $Y[i, k] \leq 0$, then $y_i \geq 0$, and hence $\widetilde{x}_i = \widehat{x}_i + \delta y_i \geq \widehat{x}_i \geq 0$. Now let $i \in J$ and $Y[i, k] > 0$. Let $J = [j_1, j_2, \ldots, j_m]$. Then

$$\delta = \frac{\overline{b}_r}{Y[r, k]} \leq \frac{\overline{b}_i}{Y[i, k]}.$$

$$\implies \widetilde{x}_{j_i} = \widehat{x}_{j_i} + \delta y_{j_i} = \overline{b}_i - \delta Y[i, k] \geq 0.$$

Hence, $\widetilde{x} \geq 0$. Therefore, $\widetilde{x}$ is feasible for the LP.

Let $i \in [n] - \widetilde{J}$. If $i = j_r$, then

$$\widetilde{x}_i = \widehat{x}_{j_r} + \delta y_{j_r} = \overline{b}_r - \delta Y[r, k] = 0.$$

If $i \in [n] - J - \{k\}$, then $\widetilde{x}_i = \widehat{x}_i + \delta y_i = 0 + \delta 0 = 0$. Hence, $\widetilde{x}_i = 0$ when $i \notin \widetilde{J}$. Let $\widetilde{B} := A[*, \widetilde{J}]$. Then

$$b = A\widetilde{x} = A[*, \widetilde{J}]\widetilde{x}[\widetilde{J}] = \widetilde{B}\widetilde{x}[\widetilde{J}].$$

By Lemma 12, $\widetilde{J}$ is a basis, so $\widetilde{B}$ is invertible. Hence, $\widetilde{x}[\widetilde{J}] = \widetilde{B}^{-1}b$. Furthermore, $\widetilde{x}[[n] - \widetilde{J}] = 0$, so $\widetilde{x} = \texttt{solve}(\widetilde{J})$. Since $\widetilde{x} \geq 0$, we get that $\widetilde{B}^{-1}b \geq 0$. Hence, $\widetilde{J}$ is a feasible basis.

Also, $c^T(\widetilde{x} - \widehat{x}) = \delta(c^T y) = \delta(c_k - z_k) \leq 0$ by Lemma 10. Hence, $c^T\widetilde{x} \leq c^T\widehat{x}$. □

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, $\overline{b}$ 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

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

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

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

where the rows are numbered from 0 instead of 1. It is easy to see that we can quickly compute Y, $\overline{b}$, 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 13, $\widetilde{J}$ 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

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

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.

$$R[i,j] = \begin{cases} \dfrac{I[r,j]}{Y[r,k]} & \text{if } i = r \\ I[i,j] - \dfrac{Y[i,k]}{Y[r,k]} I[r,j] & \text{if } i \neq r \end{cases}$$

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

Lemma 14. *$\widetilde{B}^{-1} = RB^{-1}$ and $\widetilde{Y} = RY$ and $\bar{\bar{b}} = R\bar{b}$.*

Proof. Let $J = [j_1, j_2, \ldots, j_m]$. $\widetilde{J} = J - \{j_r\} \cup \{k\}$. By Lemma 8, we get that $Y[*,J] = \widetilde{Y}[*,\widetilde{J}] = I$. We will try to show that $\widehat{Y}[*,\widetilde{J}] = I$.

Let $p, q \in [m] - \{r\}$.

$\widehat{Y}[*,\widetilde{J}][r,r] = \widehat{Y}[r,\widetilde{J}[r]] = \widehat{Y}[r,k] = 1$.

$$\widehat{Y}[*,\widetilde{J}][r,q] = \widehat{Y}[r,\widetilde{J}[q]] = \widehat{Y}[r,j_q] = \frac{Y[r,j_q]}{Y[r,k]} = 0. \qquad \text{(by Lemma 8)}$$

$$\widehat{Y}[*,\widetilde{J}][p,r] = \widehat{Y}[p,\widetilde{J}[r]] = \widehat{Y}[p,k] = Y[p,k] - \frac{Y[p,k]}{Y[r,k]} Y[r,k] = 0.$$

$$\widehat{Y}[*,\widetilde{J}][p,q] = \widehat{Y}[p,\widetilde{J}[q]] = \widehat{Y}[p,j_q] = Y[p,j_q] - \frac{Y[p,k]}{Y[r,k]} Y[r,j_q] = Y[p,j_q]$$

$$= \begin{cases} 1 & \text{if } p = q \\ 0 & \text{otherwise} \end{cases}. \qquad \text{(by Lemma 8)}$$

Hence, $\widehat{Y}[*,\widetilde{J}] = I$.

$$I = \widehat{Y}[*,\widetilde{J}] = (RB^{-1}A)[*,\widetilde{J}] = RB^{-1}A[*,\widetilde{J}] = RB^{-1}\widetilde{B}.$$

Hence, $\widetilde{B}^{-1} = RB^{-1}$.

$\widetilde{Y} = \widetilde{B}^{-1}A = RB^{-1}A = RY$.

$\bar{\bar{b}} = \widetilde{B}^{-1}b = RB^{-1}b = R\bar{b}$. □

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

Proof. Let $J = [j_1, j_2, \ldots, j_m]$. Then $\widetilde{J} = J - \{j_r\} \cup \{k\}$. Let $i \in [m] - \{r\}$. Then

$$\widehat{z}[\widetilde{J}]_i = \widehat{z}_{j_i} = z_{j_i} + \frac{c_k - z_k}{Y[r,k]} Y[r, j_i] = z_{j_i}.$$

By Lemma 8, we get $Y[r, j_i] = 0$. By Lemma 7, we get $z_{j_i} = c_{j_i}$. Hence, $\widehat{z}[\widetilde{J}]_i = c_{j_i} = c[\widetilde{J}]_i$.

$$\widehat{z}[\widetilde{J}]_r = \widehat{z}_k = z_k + \frac{c_k - z_k}{Y[r,k]} Y[r,k] = c_k = c[\widetilde{J}]_r.$$

Hence, $\widehat{z}[\widetilde{J}] = c[\widetilde{J}]$.

$$Y[r, *] = (B^{-1}A)[r, *] = B^{-1}[r, *]A.$$

$$\overline{b}_r = (B^{-1}b)_r = B^{-1}[r, *]b.$$

Let $\alpha = (c_k - z_k)/Y[r,k]$. Then

$$\widehat{z}^T = z^T + \alpha Y[r, *] = c[J]^T B^{-1} A + \alpha B^{-1}[r, *]A.$$

$$\eta = c[J]^T \overline{b} + \alpha \overline{b}_r = c[J]^T B^{-1} b + \alpha B^{-1}[r, *]b.$$

Let $u^T = c[J]^T B^{-1} + \alpha B^{-1}[r, *]$. Then $\widehat{z}^T = u^T A$ and $\eta = u^T b$.

$$c[\widetilde{J}]^T = \widehat{z}[\widetilde{J}]^T = (u^T A)[\widetilde{J}] = u^T A[*, \widetilde{J}] = u^T \widetilde{B}.$$

Hence, $u^T = c[\widetilde{J}]^T \widetilde{B}^{-1}$. So, $\widehat{z} = c[\widetilde{J}]^T \widetilde{B}^{-1} A = c[\widetilde{J}]^T \widetilde{Y} = \widetilde{z}$ and $\eta = c[\widetilde{J}]^T \widetilde{B}^{-1} b = c[\widetilde{J}]^T \overline{\widetilde{b}}$. □

In the revised simplex method, we can obtain further speedup in `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 14. 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:*

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

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 $c - z \ge 0$, 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 16. *Let J be a dual feasible basis and $\widehat{w} :=$* `dualSolve`(J)*. Then $\widehat{w}$ is a BFS of* stdDLP(A, b, c).

Proof. $A^T[J, *]\widehat{w} = B^T(c[J]^T B^{-1})^T = c[J]$. Hence, m constraints in $A^T w \le c$ are tight. Furthermore, $\text{rank}(A^T[J, *]) = \text{rank}(B) = m$, so the tight constraints have rank(m). Hence, $\widehat{w}$ is a BFS of stdDLP(A, b, c). □

Lemma 17. *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).

Proof. Since $\widehat{w}$ is a BFS, it has m linearly independent tight constraints in stdDLP(A, b, c). Let J be the indices of those constraints. Then $\text{rank}(A[*, J]) = m$, so J is a basis. Furthermore, $c[J] = A^T[J, *]\widehat{w}$, so $\widehat{w}^T = B^{-1}c[J]^T$, where $B := A[*, J]$ Hence, $\widehat{w} =$ `dualSolve`(J). J is also dual feasible, since $c - z = c - A^T\widehat{w} \ge 0$. □

Lemma 18. *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.*

Proof. Optimality of $\widehat{x}$ and $\widehat{w}$ follows from the weak duality theorem for LPs. □

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* $|\,\text{support}(\widehat{x})| < m$.
- *$\widehat{w} \in \mathbb{R}^m$ is called* degenerate *for* stdDLP(A, b, c) *if* $|\,\text{support}(c - A^T w)| < n - m$.
- *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 19. *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 20. *Let J_1 and J_2 be two bases of* $\text{stdLP}(A, b, c)$ *such that* $\texttt{sorted}(J_1) \neq \texttt{sorted}(J_2)$ *and* $\widehat{x} := \texttt{solve}(J_1) = \texttt{solve}(J_2)$. *Then $\widehat{x}$ is degenerate for* $\text{stdLP}(A, b, c)$.

Lemma 21. *Let J_1 and J_2 be two bases of* $\text{stdLP}(A, b, c)$ *such that* $\texttt{sorted}(J_1) \neq \texttt{sorted}(J_2)$ *and* $\widehat{w} := \texttt{dualSolve}(J_1) = \texttt{dualSolve}(J_2)$. *Then $\widehat{w}$ is degenerate for* $\text{stdDLP}(A, b, c)$.

The converse of Lemmas 20 and 21 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* $\text{stdLP}(A, b, c)$. *However, both* $\texttt{solve}(J) = [0, 0, 0]$ *and* $\texttt{dualSolve}(J) = [0, 0]$ *are degenerate.*

6.2 Degeneracy and Optimality

Lemma 22 (dual non-degen $\implies$ unique primal opt)**.** *Let J be a dual feasible and dual non-degenerate basis of* $\text{stdLP}(A, b, c)$. *Let* $\widehat{x} := \texttt{solve}(J)$. *Let P be the set of feasible solutions to* $\text{stdLP}(A, b, c)$. *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* $\text{stdLP}(A, b, c)$.*)*

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 23 (primal non-degen $\implies$ unique dual opt)**.** *Let J be a primal feasible and primal non-degenerate basis of* $\text{stdLP}(A, b, c)$. *Let* $\widehat{w} := \texttt{dualSolve}(J)$ *and* $\widehat{x} := \texttt{solve}(J)$. *Let Q be the set of feasible solutions to* $\text{stdDLP}(A, b, c)$. *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* $\text{stdDLP}(A, b, c)$.*)*

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

Proof. Since J is dual degenerate, $\exists k \notin J$ such that $c_k - z_k = 0$. Let $d := \text{direction}(J, k)$. Then $Ad = 0$ by Lemma 9 and $c^T d = c_k - z_k = 0$ by Lemma 10. Since J is primal non-degenerate, $\overline{b} > 0$.

Pick $\epsilon > 0$ such that $\overline{b}_i \geq \epsilon Y[i, k]$. Let $\widetilde{x} := \widehat{x} + \epsilon d$. Then $A\widetilde{x} = b$ and $c^T \widetilde{x} = c^T \widehat{x}$. For $j \in \overline{J} - \{k\}$, $\widetilde{x}_j = \widehat{x}_j \geq 0$. $\widetilde{x}_k = \widehat{x}_k + \epsilon > 0$. Let $J := [j_1, \ldots, j_m]$. Then $\widetilde{x}[j_i] = \overline{b}_i - \epsilon Y[i, k] \geq 0$. Hence, $\widetilde{x} \geq 0$. Hence, $\widetilde{x}$ is feasible for $\text{stdLP}(A, b, c)$. □

Lemma 25 (primal degen and dual non-degen $\implies$ non-unique dual opt)**.** *Let J be a dual feasible basis of* stdLP(A, b, c) *that is primal degenerate and dual non-degenerate. Let* $\widehat{x} :=$ `solve`(J) *and* $\widehat{w} :=$ `solve`(J)*. Then* $\exists$ *a dual feasible solution* $\widetilde{w}$ *to* 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. ☐

Proof. Since J is primal degenerate, $\exists r$ such that $\overline{b}_r = 0$. Pick $\epsilon > 0$ such that $(c - z)[\overline{J}]^T + \epsilon Y[r, \overline{J}] \geq 0$. This is possible since $(c - z)[\overline{J}] > 0$, since J is dual feasible and dual non-degenerate. Let $v^T := B^{-1}[r, *]$. Let $\widetilde{w} := \widehat{w} - \epsilon v$. $v^T b = B^{-1}[r, *]b = \overline{b}_r = 0$. Hence, $\widetilde{w}^T b = \widehat{w}^T b$.

$v^T A = B^{-1}[r, *]A = (B^{-1}A)[r, *] = Y[r, *]$. $c^T - \widetilde{w}^T A = c^T - \widehat{w}^T A + \epsilon v^T A = (c - z)^T + \epsilon Y[r, *]$. Let $J := [j_1, \ldots, j_m]$. Then $(c^T - \widetilde{w}^T A)[j_i] = (c - z)[j_i] + \epsilon Y[r, j_i]$. By Lemma 7, $(c - z)[j_i] = 0$. By Lemma 8, $Y[r, j_i] \geq 0$. Hence, $(c^T - \widetilde{w}^T A)[J] \geq 0$. Given how we chose ϵ, we get $(c^T - \widetilde{w}^T A)[\overline{J}] \geq 0$. Hence, $A^T\widetilde{w} \leq c$. Hence, $\widetilde{w}$ is feasible for stdDLP(A, b, c). ☐

Example 2. *Let $b = 0$, $c = (0, 0)$. Let J be any basis of* stdLP(A, b, c) *($|J| = 1$). Let* $\widehat{x} :=$ `solve`(J) *and* $\widehat{w} :=$ `dualSolve`(J)*. $\overline{b} = B^{-1}b = 0$, so $\widehat{x} = (0, 0)$, which is feasible for* 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* 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