CSCI 5654: Initialization Phase for General Form Simplex.

Sriram Sankaranarayanan

We will present the initialization phase algorithm for general form Simplex problems. Roughly, our initialization phase will have four stages:

1. Form the initial dictionary.
2. Add extra set of *auxilliary variables* $\vec{e}$ with bounds to force feasibility.
3. Change the objective to form the initial dictionary for the auxilliary LP.
4. Pivot to optimal solution of the auxilliary.
5. If feasibility concluded then remove non-basic auxilliary variables and change basic aux. variable bounds to 0.
6. Replace original LP objective and start pivoting for the original problem.

1 Form the Initial Dictionary

The overall idea is quite similar to the initialization phase simplex that we studied for standard form problems. Therein, we added a single variable x_0 to the problem and changed the objective to $-x_0$. We will show that adding a single variable does not suffice for general form problems and instead add potentially as many variables as the number of constraints in the problem. We will call these variables $\vec{e} : (e_1, \ldots, e_k)$.

Consider a general form simplex problem:

$$\begin{array}{lccccc} \max & & & \vec{c}^T\ \vec{x} & & \\ \textsf{s.t.} & \vec{a} & \leq & A\vec{x} & \leq & \vec{b} \\ & \vec{l} & \leq & \vec{x} & \leq & \vec{u} \end{array}$$

Assumption: We will assume *for the time being*, that there are no free variables x_i with bounds $-\infty \leq x_i \leq \infty$. In other words, if $l_i = -\infty$ then $u_i < \infty$. Similarly, if $u_i = \infty$ then $l_i > -\infty$ (Similar assumption already holds for $\vec{a}_i, \vec{b}_i$. I.e, if $a_i = -\infty$ and $b_i = \infty$ then the i^{th} row constraint is not necessary and can be removed without changing the problem.).

Initial dictionary can be formed directly from the problem description:

$$\begin{array}{cc|c|cccc} a_1 & b_1 & w_1 & A_{11}x_1 & +A_{12}x_2 & +\cdots & +A_{1n}x_n \\ a_2 & b_2 & w_2 & A_{21}x_1 & +A_{22}x_2 & +\cdots & +A_{2n}x_n \\ \vdots & & \vdots & & & \ddots & \\ a_m & b_m & w_m & A_{m1}x_1 & +A_{m2}x_2 & +\cdots & +A_{mn}x_n \\ \hline & & z & c_1x_1 & +c_2x_2 & +\cdots & +c_mx_m \\ & & & l_1 & l_2 & \cdots & l_m \\ & & & u_1 & u_2 & \cdots & u_m \end{array}$$

Q: We need to decide how to rest the non-basic variables? Should non-basic x_i rest on l_i or on u_i?

For the initial dictionary, we may choose the following strategy:

1. If $l_i = -\infty$ then choose u_i. If $u_i = \infty$ choose l_i. We have already assumed that both $l_i = -\infty$ and $u_i = \infty$ is not possible.

2. If both $-\infty < l_i \leq u_i < \infty$ are finite, we choose that bound which maximizes the objective (but really, we may as well toss a coin and choose. It does not matter!).

As a result, we have a starting dictionary that may or may not be feasible.

$$
\begin{array}{cc|c|cccc}
a_1 & b_1 & w_1 & A_{11}x_1 & +A_{12}x_2 & +\cdots & +A_{1n}x_n \\
a_2 & b_2 & w_2 & A_{21}x_1 & +A_{22}x_2 & +\cdots & +A_{2n}x_n \\
\vdots & & \vdots & & & \ddots & \\
a_m & b_m & w_m & A_{m1}x_1 & +A_{m2}x_2 & +\cdots & +A_{mn}x_n \\
\hline
 & & z & c_1x_1 & +c_2x_2 & +\cdots & +c_mx_m \\
 & & & \boxed{\mathbf{l_1}} & l_2 & \cdots & \boxed{\mathbf{l_m}} \\
 & & & u_1 & \boxed{\mathbf{u_2}} & \cdots & u_m
\end{array}
$$

Based on this, we compute values of $w_1, \ldots, w_m$. If this dictionary is feasible, then nothing more needs to be done. We are in the fortunate case where the problem comes to us with feasible initial dictionary. In most cases, this will not be a feasible dictionary. Our present goal is to first force it to be feasible by adding some auxilliary variables. We will then remove the aux. variables (or make their presence harmless) eventually. We illustrate this process using an example.

Example 1.1. *Consider the following problem:*

$$
\begin{array}{lccccccccc}
\mathsf{max} & & & 2x_1 & -x_2 & +8x_3 & -2x_4 & & \\
\mathsf{s.t.} & 10 & \leq & x_1 & +2x_2 & & +x_4 & \leq & 30 \\
 & 0 & \leq & x_1 & -x_2 & & -3x_4 & \leq & \infty \\
 & 6 & \leq & & -x_2 & +x_3 & & \leq & 8 \\
 & -10 & \leq & & & x_3 & -x_4 & \leq & 10 \\
\hline
 & & & & & & & & \\
 & 0 & \leq & x_1 & & & & \leq & 20 \\
 & -\infty & \leq & & x_2 & & & \leq & 10 \\
 & 0 & \leq & & & x_3 & & \leq & \infty \\
 & -10 & \leq & & & & x_4 & \leq & 0
\end{array}
$$

The initial dictionary is

$$
\begin{array}{cc|c|cccc|c}
10 & 30 & w_1 & x_1 & +2x_2 & & +x_4 & = 40 \\
0 & \infty & w_2 & x_1 & -x_2 & & -3x_4 & = 10 \\
6 & 8 & w_3 & & -x_2 & +x_3 & & = -10 \\
-10 & 10 & w_4 & & & x_3 & -x_4 & = 0 \\
\hline
 & & z & 2x_1 & -x_2 & +8x_3 & -2x_4 & \\
 & & & 0 & -\infty & \boxed{\mathbf{0}} & -10 & \\
 & & & \boxed{\mathbf{20}} & \boxed{\mathbf{10}} & \infty & \boxed{\mathbf{0}} &
\end{array}
$$

Notice that $w_1 = 40$ and $w_3 = -10$ are not in their respective bounds. Therefore the dictionary above is infeasible. To restore feasibility, we will first consider adding extra variables e_1 and e_2 to restore the value of w_1, w_3 in bounds. We will ensure that $e_1, e_2 \geq 0$. Therefore:

1. *e_1 must be subtracted from w_1 as its current value is over the maximum allowed value of* 30.

2. e_2 must be added as its current value is below the current bounds.

$$\begin{array}{cc|c|cccccc|l}
10 & 30 & w_1 & x_1 & +2x_2 & & +x_4 & -e_1 & & = 40 \\
0 & \infty & w_2 & x_1 & -x_2 & & -3x_4 & & & = 10 \\
6 & 8 & w_3 & & -x_2 & +x_3 & & & +e_2 & = -10 \\
-10 & 10 & w_4 & & & x_3 & -x_4 & & & = 0 \\ \hline
 & & z & 2x_1 & -x_2 & +8x_3 & -2x_4 & & & \\
 & & & 0 & -\infty & \boxed{\mathbf{0}} & -10 & & & \\
 & & & \boxed{\mathbf{20}} & \boxed{\mathbf{10}} & \infty & \boxed{\mathbf{0}} & & &
\end{array}$$

How much should e_1, e_2 be. Ideally, they can be as large as we want. But we note that setting $e_1 = 10$ and $e_2 = 16$ will minimally suffice to restore feasibility.

Therefore, we set the bounds as

$$0 \leq e_1 \leq 10, \ \textit{and } 0 \leq e_2 \leq 16$$

Next, the original objective is removed and the new objective $-e_1 - e_2$ is set to be maximized (or $e_1 + e_2$ is to be minimized). We obtain the dictionary:

$$\begin{array}{cc|c|cccccc|l}
10 & 30 & w_1 & x_1 & +2x_2 & & +x_4 & -e_1 & & = 40 - 10 = 30 \\
0 & \infty & w_2 & x_1 & -x_2 & & -3x_4 & & & = 10 \\
6 & 8 & w_3 & & -x_2 & +x_3 & & & +e_2 & = -10 + 16 = 6 \\
-10 & 10 & w_4 & & & x_3 & -x_4 & & & = 0 \\ \hline
 & & z & \boldsymbol{0}x_1 & +\boldsymbol{0}x_2 & +\boldsymbol{0}x_3 & +\boldsymbol{0}x_4 & -\mathbf{1}e_1 & -\mathbf{1}e_2 & \\
 & & & 0 & -\infty & \boxed{\mathbf{0}} & -10 & 0 & 0 & \\
 & & & \boxed{\mathbf{20}} & \boxed{\mathbf{10}} & \infty & \boxed{\mathbf{0}} & \boxed{\mathbf{10}} & \boxed{\mathbf{16}} &
\end{array}$$

We can start Simplex algorithm from this dictionary. If the optimal value of this auxilliary LP is 0 then we conclude that the problem is feasible. Or else, we conclude that the problem is infeasible.

▲

In the general case, consider the dictionary D that is infeasible:

$$\begin{array}{cc|c|cccc}
a_1 & b_1 & w_1 & A_{11}x_1 & +A_{12}x_2 & +\cdots & +A_{1n}x_n \\
a_2 & b_2 & w_2 & A_{21}x_1 & +A_{22}x_2 & +\cdots & +A_{2n}x_n \\
\vdots & & \vdots & & & \ddots & \\
a_m & b_m & w_m & A_{m1}x_1 & +A_{m2}x_2 & +\cdots & +A_{mn}x_n \\ \hline
 & & z & c_1x_1 & +c_2x_2 & +\cdots & +c_mx_m \\
 & & & \boxed{\mathbf{l_1}} & l_2 & \cdots & \boxed{\mathbf{l_m}} \\
 & & & u_1 & \boxed{\mathbf{u_2}} & \cdots & u_m
\end{array}$$

Let $s(x_i; D)$ denote the solution for x_i in dictionary D. Similarly, we will denote $s(w_j; D)$ to denote solution to slack variable w_j in D. The dictionary is infeasible if for some $j \in [1, m]$, we have

$$\underbrace{s(w_j; D) < a_j}_{w_j \text{ is too small in } D} \quad \text{OR} \quad \underbrace{s(w_j; D) > b_j}_{w_j \text{ is too large in } D} .$$

In general, more than one basic variable $w_{j_1}, \ldots, w_{j_k}$ will be infeasible in the dictionary. We introduce variables $e_1, \ldots, e_k$ that will force these infeasible variables back into their appropriate ranges. Suppose variable w_k has an out of range value in dictionary, then introduce fresh e_k variable.

1. If w_k is too large, then SUBTRACT e_k from the corresponding row.
2. If w_k is too small, then ADD e_k to the corresponding row.

The bounds for e_k corresponding to w_k are

$$\begin{array}{ll}
0 \leq e_k \leq l_k - s(w_k; D), & \text{If } w_k \text{ too small} \\
0 \leq e_k \leq s(w_k; D) - u_k, & \text{If } w_k \text{ too large}
\end{array}$$

As a result, by setting each e_i to its upper bound, we can ensure that the resulting dictionary with objective changed to $-\sum_k e_k$ is zero.

$$\begin{array}{cc|c|cccccccc}
a_1 & b_1 & w_1 & A_{11}x_1 & +A_{12}x_2 & +\cdots & +A_{1n}x_n & \pm e_1 & & & \\
a_2 & b_2 & w_2 & A_{21}x_1 & +A_{22}x_2 & +\cdots & +A_{2n}x_n & & \pm e_2 & & \\
\vdots & & \vdots & & & \ddots & & & & \ddots & \\
a_m & b_m & w_m & A_{m1}x_1 & +A_{m2}x_2 & +\cdots & +A_{mn}x_n & & & & \pm e_m \\ \hline
 & & z & 0x_1 & +0x_2 & +\cdots & +0x_m & -1e_1 & -1e_2 & \cdots & -1e_m \\
 & & & \boxed{\mathbf{l_1}} & l_2 & \cdots & \boxed{\mathbf{l_m}} & 0 & 0 & \cdots & 0 \\
 & & & u_1 & \boxed{\mathbf{u_2}} & \cdots & u_m & \boxed{\mathbf{f_1}} & \boxed{\mathbf{f_2}} & \cdots & \boxed{\mathbf{f_m}}
\end{array}$$

As a result of the auxilliary variables $\vec{e}$, this dictionary is feasible.

Lemma 1.1. *The optimal value of the auxilliary LP is zero iff the original LP is feasible.*

Proof. Let us assume that the optimal value is zero. The auxilliary variables $e_1, \ldots, e_m$ are constrained to be non-negative. Therefore, $\vec{e}$ satisfies the following conditions

$$\begin{array}{ll}
e_1 + \ldots + e_m = 0, & \because \text{ the optimal value is assumed } 0 \\
e_1 \geq 0, \ldots, e_m \geq 0 & \text{original setup constraints}
\end{array}$$

Therefore the solution $\vec{x}, \vec{w}$ of the auxilliary LP satisfies the original system of constraints.

Let us assume that the original LP is feasible. Then the optimal solution to the auxilliary is at least 0. This is because, we may take any feasible solution $(\vec{x}, \vec{w})$ to the original LP and set $(\vec{x}, \vec{w}, \vec{e} = 0)$ to satisfy the aux. LP constraints. But we already know that since $e_k >= 0$, the objective value $-(\sum_k e_k) \leq 0$. Thus, we have shown that if the original LP is feasible then the optimal solution to the auxiliary is given by augmenting the solution to the original LP by $\vec{e} = 0$ and the optimal value itself is 0.

□

Example 1.2. *Continuing with the running example, we left off with the dictionary at the start of the initialization phase:*

$$\begin{array}{cc|c|cccccc|l}
10 & 30 & w_1 & x_1 & +2x_2 & & +x_4 & -e_1 & & = \textcolor{red}{40} - 10 = 30 \\
0 & \infty & w_2 & x_1 & -x_2 & & -3x_4 & & & = 10 \\
6 & 8 & w_3 & & -x_2 & +x_3 & & & +e_2 & = \textcolor{red}{-10} + 16 = 6 \\
-10 & 10 & w_4 & & & x_3 & -x_4 & & & = 0 \\ \hline
 & & z & \boldsymbol{0}x_1 & +\boldsymbol{0}x_2 & +\boldsymbol{0}x_3 & +\boldsymbol{0}x_4 & \mathbf{-1}e_1 & \mathbf{-1}e_2 & \\
 & & & 0 & -\infty & \boxed{\mathbf{0}} & -10 & 0 & 0 & \\
 & & & \boxed{\mathbf{20}} & \boxed{\mathbf{10}} & \infty & \boxed{\mathbf{0}} & \boxed{\mathbf{10}} & \boxed{\mathbf{16}} &
\end{array}$$

After solving simplex on this dictionary, we obtain the following dictionary that has the following form (we need not bother with the contents for now):

0	20	x_1	$\frac{4}{7}w_1$	$\frac{3}{7}w_2$	$\frac{5}{7}w_3$	$-\frac{5}{7}w_4$	$-\frac{4}{7}e_1$	$+\frac{5}{7}e_2$
$-\infty$	10	x_2	$\frac{1}{7}w_1$	$-\frac{1}{7}w_2$	$-\frac{4}{7}w_3$	$+\frac{4}{7}w_4$	$-\frac{1}{7}e_1$	$-\frac{4}{7}e_2$
0	∞	x_3	$\frac{1}{7}w_1$	$-\frac{1}{7}w_2$	$+\frac{3}{7}w_3$	$+\frac{4}{7}w_4$	$-\frac{1}{7}e_1$	$+\frac{3}{7}e_2$
-10	0	x_4	$\frac{1}{7}w_1$	$-\frac{1}{7}w_2$	$+\frac{3}{7}w_3$	$-\frac{3}{7}w_4$	$-\frac{1}{7}e_1$	$+\frac{3}{7}e_2$
		z	$\boldsymbol{0}w_1$	$+\boldsymbol{0}w_2$	$+\boldsymbol{0}w_3$	$+\boldsymbol{0}w_4$	$\boldsymbol{-1}e_1$	$\boldsymbol{-1}e_2$
			$\boxed{\mathbf{10}}$	$\boxed{\mathbf{0}}$	$\boxed{\mathbf{6}}$	-10	$\boxed{\mathbf{0}}$	$\boxed{\mathbf{0}}$
			30	∞	8	$\boxed{\mathbf{10}}$	10	16

We note that e_1, e_2 are non-basic in this final dictionary (it need not always be the case), and they are at 0. So we can remove them and reintroduce the original objective in terms of $w_1, \ldots, w_4$.

0	20	x_1	$\frac{4}{7}w_1$	$\frac{3}{7}w_2$	$\frac{5}{7}w_3$	$-\frac{5}{7}w_4$
$-\infty$	10	x_2	$\frac{1}{7}w_1$	$-\frac{1}{7}w_2$	$-\frac{4}{7}w_3$	$+\frac{4}{7}w_4$
0	∞	x_3	$\frac{1}{7}w_1$	$-\frac{1}{7}w_2$	$+\frac{3}{7}w_3$	$+\frac{4}{7}w_4$
-10	0	x_4	$\frac{1}{7}w_1$	$-\frac{1}{7}w_2$	$+\frac{3}{7}w_3$	$-\frac{3}{7}w_4$
		z	$\frac{55}{7}w_1$	$\frac{-33}{7}w_2$	$+\frac{19}{7}w_3$	$+\frac{33}{7}w_4$
			$\boxed{\mathbf{10}}$	$\boxed{\mathbf{0}}$	$\boxed{\mathbf{6}}$	-10
			30	∞	8	$\boxed{\mathbf{10}}$

A little bit of calculation is enough to convince that this dictionary is actually feasible. We can start our general form simplex from here. ▲

We note the two prominent differences from the algorithm we presented for the standard form problem:

1. We add more than one initial variables and set their bounds appropriately, to ensure feasibility.
2. We do not have any special pivoting rules for the first pivot or leaving variable selection rules (x_0 gets to leave preferentially).

If the optimal objective value of the auxilliary LP is 0 then original problem is deduced to be feasible, else it is not. We now do three changes to prepare to solve the original LP.

1. If any of the e_i variables is non-basic (independent), then it must rest on its 0 bound. So remove it from the dictionary and the corresponding column. This will not affect feasibility.
2. If any of the e_i variables are basic, then it cannot be removed from the dictionary. But we note that its solution in the dictionary must be 0. So we set both its bounds to 0 in the new dictionary.
3. We change the objective to the original problem objective and rewrite it in terms of the non-basic (independent) variables.

But this does not quite give us the original problem. We may still have some e_j variables in the basis. But both the upper and lower bounds of such variables are 0. We can do one of two things:

1. Force e_j variables out of the basis using a special pivoting step upfront (not recommended and not presented here).
2. Keep the e_j variables around. Whenever they leave the basis as a result of a pivot, we remove them from further consideration.

In fact, keeping e_js around and removing them selectively should not introduce much of a overhead. We simply lose them when they leave the basis.

So that sums up the basic idea behind initialization phase for general form simplex. We just have one more thing to discuss.

Handling free variables: What if we had a variable $-\infty \leq x_i \leq \infty$ that was free. I.e, both its column bounds are infinite. We cannot form a dictionary unless we allow a non-basic variable to rest on an infinite bound. There are two solutions:

1. Split x_i into two parts $x_i = x_i^+ - x_i-$ and add the constraints $x_i^+ \geq 0$, $x_i^- \geq 0$.

2. Alternatively, we force a pivoting move that forces x_i to enter and some variable w_j that has at least one non-infinite bound to leave. The condition is that the expression for w_j must involve x_i with a non-zero coefficient.

   If no such w_j can be found, then it means that x_i is not involved in any rows in the problem. If it is involved in the objective with a non-zero coefficient the problem is trivially unbounded or else, x_i variable can be safely removed without affecting the problem.

Why not add a single variable? In the initialization phase simplex, a single variable x_0 was added and this sufficed to force the problem to be feasible by setting x_0 to be the value of the least constant. For general form simplex, we insist on adding several variables. Why? Modifying running example slightly, consider the dictionary:

10	15	w_1	x_1	$+2x_2$		$+x_4$	$= 40$
0	∞	w_2	x_1	$-x_2$		$-3x_4$	$= 10$
6	8	w_3		$-x_2$	$+x_3$		$= -10$
-10	10	w_4			x_3	$-x_4$	$= 0$
		z	$2x_1$	$-x_2$	$+8x_3$	$-2x_4$	
			0	$-\infty$	$\boxed{\mathbf{0}}$	-10	
			$\boxed{\mathbf{20}}$	$\boxed{\mathbf{10}}$	∞	$\boxed{\mathbf{0}}$	

Clearly this is infeasible. To restore feasibility let us say we introduce a single variable $x_0 \geq 0$. We will have to subtract x_0 from the first constraint corr. to w_1 but add x_0 to the constraint corr.to w_3. How much should x_0 be to guarantee feasibility? Well for w_1, x_0 must be at least 25 (to reach the upper bound of 15), but for w_3, x_0 must be at least 16 but at most 18 (or else it will whiz past the upper bound on w_3). Thus, we have conflicting requirements on the single variable x_0. So a single variable will not suffice, in general.

That is all folks!!