Math 340 Lecture 7

Infeasible initial dictionaries. Started discussing this last time, with the example

- Maximize $-2x_1 - x_2$ subject to
  - $-x_1 + x_2 \leq -1$
  - $-x_1 - 2x_2 \leq -2$
  - $x_2 \leq 1$
  - $x_1, x_2 \geq 0$.

Setting up the slack variables gives us an infeasible dictionary, i.e. one where putting in $x_1 = x_2 = 0$ is not a feasible solution:

\[
\begin{align*}
x_3 &= -1 + x_1 - x_2 \\
x_4 &= -2 + x_1 + 2x_2 \\
x_5 &= 1 - x_2 \\
z &= -2x_1 - x_2.
\end{align*}
\]

What we said we should do is introduce an auxiliary LP with the goal of finding us a feasible dictionary. The idea is to add some “wiggle room” in the form of an extra variable $x_0$, to the right-hand side of our initial constraints. Then we get the constraints

- $-x_1 + x_2 \leq -1 + x_0$
- $-x_1 - 2x_2 \leq -2 + x_0$
- $x_2 \leq 1 + x_0$

Here you can see why I’m thinking of it as “wiggle room” - if we make $x_0$ really big, our inequalities become less stringent, and it’s easier to find feasible solutions. So our goal is to start with $x_0$ being really big, and then work our way down to as small as possible. If we can get a feasible solution with $x_0 = 0$, i.e. no wiggle room, then this is feasible for our original problem! So what we want to do is minimize $x_0$ subject to our constraints above, and we’re back to trying to solve an auxiliary linear program.

- Maximize $-x_0$ subject to
  - $-x_1 + x_2 - x_0 \leq -1$
  - $-x_1 - 2x_2 - x_0 \leq -2$
  - $x_2 - x_0 \leq 1$
  - $x_0, x_1, x_2 \geq 0$.

Putting this in standard form just modifies the dictionary we started with, and changes the objective function to the auxiliary one $z' = -x_0$.

\[
\begin{align*}
x_3 &= -1 + x_1 - x_2 + x_0 \\
x_4 &= -2 + x_1 + 2x_2 + x_0 \\
x_5 &= 1 - x_2 + x_0 \\
z' &= -x_0
\end{align*}
\]

So this is an initial dictionary for our auxiliary problem, but it’s still infeasible. So did we really improve anything? Well, the answer is yes, because in this case we can always make a single pivot to feasibility: we make $x_0$ enter, and whichever slack variable is “most infeasible” (i.e. the one with the most negative constant) leave. In this case it’s $x_4$, and we substitute $x_0 = 2 - x_1 - 2x_2 + x_4$ everywhere to get

\[
\begin{align*}
x_0 &= 2 - x_1 - 2x_2 + x_4 \\
x_3 &= 1 - 3x_2 + x_4 \\
x_5 &= 3 - x_1 - 3x_2 + x_4 \\
z' &= -2 + x_1 + 2x_2 - x_4
\end{align*}
\]
This is feasible so we just proceed with the simplex method with it normal. By the standard rule we pick \(x_2\) to enter (it has the largest coefficient in \(z\)) and then pick \(x_3\) to leave (the constraint there only lets \(x_2\) go up to \(1/3\)) and get

\[
\begin{align*}
x_0 &= \frac{4}{3} - x_1 + \frac{2}{3} x_3 + \frac{1}{3} x_4 \\
x_2 &= \frac{3}{4} - \frac{1}{3} x_3 + \frac{1}{3} x_4 \\
x_5 &= \frac{2}{3} - x_1 + x_3 \\
z' &= -\frac{1}{3} + x_1 - \frac{2}{3} x_3 - \frac{1}{3} x_4
\end{align*}
\]

We pivot once more, making \(x_1\) leave the basis (it has the only positive term in the objective function) and \(x_0\) leave (it imposes the strictest constraint on how far \(x_1\) can rise). Thus we get

\[
\begin{align*}
x_1 &= \frac{4}{3} - x_0 + \frac{2}{3} x_3 + \frac{1}{3} x_4 \\
x_2 &= \frac{3}{4} - \frac{1}{3} x_3 + \frac{1}{3} x_4 \\
x_5 &= \frac{2}{3} + x_0 + \frac{1}{3} x_3 - \frac{1}{3} x_4 \\
z' &= -x_0
\end{align*}
\]

And now we’re done with the auxiliary problem! We’ve come up with a feasible solution with \(x_0\) as a nonbasic variable again, i.e. a feasible solution where we naturally take \(x_0 = 0\). But \(x_0\) is our “wiggle room” variable, so making it zero is exactly what we want to get a feasible solution to our original problem without \(x_0\). So we can just kick \(x_0\) out of our problem entirely now and see that if we take \(x_3\) and \(x_4\) to be nonbasic variables and \(x_1, x_2, x_5\) to be basic, this gives us a feasible dictionary for our original problem:

\[
\begin{align*}
x_1 &= \frac{4}{3} + \frac{2}{3} x_2 + \frac{1}{3} x_4 \\
x_2 &= \frac{3}{4} - \frac{1}{3} x_3 + \frac{1}{3} x_4 \\
x_5 &= \frac{2}{3} + \frac{1}{3} x_3 - \frac{1}{3} x_4 \\
z &= -3 - x_3 - x_4
\end{align*}
\]

(Here I had to take our original formula \(z = -2x_1 - x_2\) and substitute in the equations for \(x_1\) and \(x_2\) to get it in terms of \(x_3\) and \(x_4\)). In this case we can see this dictionary is already optimal - the solution to the original LP is that it has a maximum of \(z = 3\) when \(x_1 = 4/3\) and \(x_2 = 1/3\) (and slack variables \(x_3 = x_4 = 0\) and \(x_5 = 2/3\)). If this dictionary wasn’t optimal, we’d just continue on with the simplex method as normal (the “second phase” of the two-phase simplex method, with the “first phase” being the auxiliary problem we just did).

**Summary of the simplex method.** So now we can give a pretty full summary of how the simplex method should work.

- Insert slack variables, and get a starting dictionary.
  - If the starting dictionary is infeasible, set up the auxiliary LP to solve (add in the “wiggle room” variable \(x_0\) and try to maximize \(-x_0\)).
  - The starting dictionary for the auxiliary LP is infeasible, but we can make one pivot to feasibility for it.
  - Iterate the pivoting process for the auxiliary LP until we’ve solved it. If the maximum of \(-x_0\) is less than 0 then we’re done and conclude the initial LP is infeasible. If the maximum of \(-x_0\) is exactly 0, we’ve found a feasible solution with \(x_0 = 0\) and this corresponds to a feasible solution to the original problem, so continue.
- Start with a feasible dictionary for the original LP (either the starting LP if it’s feasible, or one obtained by the above process if not).
- Start pivoting; with each iteration pick an entering variable and an exiting variable according to some rule (the standard rule for us) - this should always preserve feasibility, and never decrease the constant in the objective function.
• Keep doing this until you can’t. If there’s no choice for entering variable this means we’ve found an optimum solution for the LP. If there’s no choice for exiting variable this means we’ve proven the LP is unbounded.

Degeneracy. So the one problem we’re let with is: how do we know that this process actually terminates, and we don’t get stuck in an infinite loop? Well, this is kind of a subtle question. But one way we can get stuck in an infinite loop is if the simplex method cycles: we return to the same dictionary two times. If we keep following the same rule for choosing entering/exiting variables then this will actually just circle around the same list of dictionaries forever!

The first thing we can see is this is the only thing that can keep the simplex method going forever.

• Fact: The only way the simplex method can fail to terminate is if it cycles.

This is simple to prove: a dictionary is determined by its choice of how we divide up the basic and nonbasic variables. There’s only finitely many choices, so if we go on long enough we must hit the same one twice.