Math 482: Linear Programming[1] Mikhail Lavrov

Lecture 31: Wrapping Up the Primal-Dual Algorithm

April 22, 2020 *University of Illinois at Urbana-Champaign*

1 Starting the algorithm

Right now, we know how to iterate the primal-dual algorithm: starting from a dual feasible vector $\mathbf{u}$, we can improve it to find a better $\mathbf{u}$, until it is optimal.

But how do we pick a starting point $\mathbf{u}$?

There are two boring answers:

- We might be able to guess an initial value of $\mathbf{u}$. For example, if the vector $\mathbf{u} = \mathbf{0}$ is feasible for $(\mathbf{D})$, it is a reasonable starting point.
- As always, we can use a two-phase method for finding an initial value. If we have to resort to this, then maybe the primal-dual algorithm is not a great choice.

There is one sneaky strategy that doesn't always work, but might *sometimes* work, and is better than using a two-phase method when it does.

Consider the primal-dual pair

$$(\mathbf{P}) \begin{cases} \underset{\mathbf{x}\in\mathbb{R}^3}{\text{minimize}} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1, x_2, x_3 \ge 0 \end{cases} \qquad (\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^2}{\text{maximize}} & 2u_1 + 3u_2 \\ \text{subject to} & u_1 + u_2 \le 2 \\ & 2u_1 - u_2 \le -1 \\ & -3u_1 + u_2 \le 4 \end{cases}$$

We can make a guess that is technically not rigorously justified but looks overwhelmingly likely: that the optimal solution to $(\mathbf{P})$ satisfies $x_1 + x_2 + x_3 \le 100$. We can rewrite this guess as the inequality $x_1 + x_2 + x_3 + x_4 = 100$ for some $x_4 \ge 0$, and add it to $(\mathbf{P})$. This gives us a new primal-dual pair:

$$(\mathbf{P}) \begin{cases} \underset{\mathbf{x}\in\mathbb{R}^4}{\text{minimize}} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1 + x_2 + x_3 + x_4 = 100 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{cases} \qquad (\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^3}{\text{maximize}} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}$$

The reason to do this is that we can always find a feasible solution $\mathbf{u}$ to the new dual program. Just set $u_1 = u_2 = 0$, then choose u_3 to be whatever makes the inequalities work. (In general, set

[1] This document comes from the Math 482 course webpage: https://faculty.math.illinois.edu/~mlavrov/courses/482-spring-2020.html

all but the last, new dual variable to 0.) Here, we have $u_3 \leq 2$, $u_3 \leq -1$, $u_3 \leq -4$, and $u_3 \leq 0$, so $\mathbf{u} = (0, 0, -1)$ is a feasible solution to (**D**).

We expect that the constraint we added to (**P**) was already true at the optimal solution, and so we haven't changed anything. There are two ways to verify this assumption when we're done:

check $u_3 = 0$.

$u_3 \leq 0$ tight i.e. $u_3 = 0$

- From the point of view of (**P**), it should be the case that in the optimal solution $\mathbf{x}$, we have $x_4 > 0$, corresponding to the inequality $x_1 + x_2 + x_3 \leq 100$ being slack. If that's true, then the inequality wasn't doing any work, so we can drop it and get an equally valid optimal answer.

  If $x_4 = 0$ in the optimal solution, then we are on the boundary of $x_1 + x_2 + x_3 \leq 100$. It is very likely in such a case that the real optimal solution has $x_1 + x_2 + x_3 > 100$, and so 100 was too small a value. (Of course it's also possible that (**P**) was unbounded, and no value would have been large enough.)

- From the point of (**D**), it should be the case that in the optimal solution, $u_3 = 0$: when we set $u_3 = 0$, the new (**D**) simplifies to the old (**D**). if $u_3 < 0$ in the optimal solution, then things have gone wrong.

  Actually, as soon as we hit a point where $u_3 = 0$, we can switch back to the old version of (**D**), because at that point, we've found a feasible solution for it.

It shouldn't matter how large 100 is beyond "large enough". But if there's no clear choice of a number to pick here such that the inequality we guess is very likely to be true, then this is not a good strategy.

2 The revised simplex method

The "frozen variables" method is like training wheels for the primal-dual algorithm. By the way, it is not official terminology: you will not find your textbook, or other sources, talking about frozen variables.

Instead, a common strategy is to use the primal-dual algorithm in conjunction with the revised simplex method. Recall that with this method, we don't maintain the simplex tableau: we will just keep around a set $\mathcal{B}$ of basic variables, the inverse matrix $A_{\mathcal{B}}^{-1}$, and maybe the vector $A_{\mathcal{B}}^{-1}\mathbf{b}$. Everything else, we can compute on the fly.

With the revised simplex method, we of course don't keep any "frozen variable" columns around in the tableau, because we don't keep *any* columns around in the tableau. Instead, to use the revised simplex method to solve the restricted primal (**RP**), we just:

- Proceed as usual, making sure that when we pick an entering variable, we are only picking among variables that are supposed to be part of (**RP**) at the current iteration. (That is, don't pivot on any frozen variables.)
- When going from one iteration of (**RP**) to the next, the inverse matrix $A_{\mathcal{B}}^{-1}$ and the vector $A_{\mathcal{B}}^{-1}\mathbf{b}$ remain valid. In other words, we don't need to make any changes to continue using the revised simplex method for the new iteration.

3 The primal-dual algorithm as a pivoting rule

This section of the notes is just some philosophical questioning. From the point of view of (**RP**), what is going on when we apply the primal-dual algorithm?

Well, we are constantly working with the same tableau. That tableau is actually similar to a phase-one tableau for the ordinary simplex method for solving (**P**). We have added artificial variables $y_1, y_2, \dots, y_m$ to the equations, and we are trying to minimize $y_1 + \dots + y_m$.

There are two differences between the primal-dual algorithm and the phase-one simplex method. First of all, we have a restriction on which variables can be brought into the basis. From time to time, we do some calculation that makes no sense from (**RP**)'s point of view (the augmenting step) that modifies this restriction.

But ultimately, what's happening is that we have an extra pivoting rule in play to help us choose entering variables. It is telling us that the entering variable must come from the set of unfrozen variables, whatever those are at any given time.

The second difference is that in the phase-one problem, when we bring $y_1 + \dots + y_m$ to 0, our work has just begun: this has given us a feasible solution to the original problem, but now we must optimize it.

With the primal-dual algorithm, as soon as this happens, we don't just have a feasible solution to (**P**), but an optimal one. The pivoting rule "don't pivot on frozen variables" guides us to eliminating $y_1, \dots, y_m$ in the best possible way.