Primal-Dual Algorithm III

Math 482, Lecture 31

Misha Lavrov

April 22, 2020

Starting the primal-dual algorithm

So far, we know how to do iterations of the primal-dual algorithm: given a dual-feasible point $\mathbf{u}$, improve it to a better point $\mathbf{u}$. By repeating this, we can solve the LP.

Starting the primal-dual algorithm

So far, we know how to do iterations of the primal-dual algorithm: given a dual-feasible point $\mathbf{u}$, improve it to a better point $\mathbf{u}$. By repeating this, we can solve the LP.

But how do we start the first iteration? How do we get the starting $\mathbf{u}$?

Starting the primal-dual algorithm

So far, we know how to do iterations of the primal-dual algorithm: given a dual-feasible point $\mathbf{u}$, improve it to a better point $\mathbf{u}$. By repeating this, we can solve the LP.

But how do we start the first iteration? How do we get the starting $\mathbf{u}$?

There are several possible answers:

- Sometimes, a simple point like $\mathbf{u} = \mathbf{0}$ is obviously feasible.

Starting the primal-dual algorithm

So far, we know how to do iterations of the primal-dual algorithm: given a dual-feasible point $\mathbf{u}$, improve it to a better point $\mathbf{u}$. By repeating this, we can solve the LP.

But how do we start the first iteration? How do we get the starting $\mathbf{u}$?

There are several possible answers:

- Sometimes, a simple point like $\mathbf{u} = \mathbf{0}$ is obviously feasible.
- The only fully general answer is a two-phase method. If we do this, we might as well not use the primal-dual algorithm.

Starting the primal-dual algorithm

So far, we know how to do iterations of the primal-dual algorithm: given a dual-feasible point $\mathbf{u}$, improve it to a better point $\mathbf{u}$. By repeating this, we can solve the LP.

But how do we start the first iteration? How do we get the starting $\mathbf{u}$?

There are several possible answers:

- Sometimes, a simple point like $\mathbf{u} = \mathbf{0}$ is obviously feasible.
- The only fully general answer is a two-phase method. If we do this, we might as well not use the primal-dual algorithm.
- In some cases, there is a trick we can do to create a dual feasible solution.

The trick

Consider the following primal-dual pair of linear programs:

$$
(\mathbf{P}) \begin{cases} \text{minimize} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1, x_2, x_3 \geq 0 \end{cases}
$$

$$
(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 \\ \text{subject to} & u_1 + u_2 \leq 2 \\ & 2u_1 - u_2 \leq -1 \\ & -3u_1 + u_2 \leq 4 \end{cases}
$$

The trick

Consider the following primal-dual pair of linear programs:

$$(\mathbf{P})\begin{cases}\text{minimize} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1, x_2, x_3 \geq 0\end{cases}$$

$$(\mathbf{D})\begin{cases}\text{maximize} & 2u_1 + 3u_2 \\ \text{subject to} & u_1 + u_2 \leq 2 \\ & 2u_1 - u_2 \leq -1 \\ & -3u_1 + u_2 \leq 4\end{cases}$$

The trick relies on making a guess: that the optimal solution to (**P**) has $x_1 + x_2 + x_3 \leq 100$.

The trick

Consider the following primal-dual pair of linear programs:

$$
(\mathbf{P}) \begin{cases} \text{minimize} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1 + x_2 + x_3 \leq 100 \\ & x_1, x_2, x_3 \geq 0 \end{cases}
$$

$$
(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 \\ \text{subject to} & u_1 + u_2 \leq 2 \\ & 2u_1 - u_2 \leq -1 \\ & -3u_1 + u_2 \leq 4 \end{cases}
$$

The trick relies on making a guess: that the optimal solution to (**P**) has $x_1 + x_2 + x_3 \leq 100$.

The trick

Consider the following primal-dual pair of linear programs:

$$(\mathbf{P}) \begin{cases} \text{minimize} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1 + x_2 + x_3 + x_4 = 100 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 \\ \text{subject to} & u_1 + u_2 \leq 2 \\ & 2u_1 - u_2 \leq -1 \\ & -3u_1 + u_2 \leq 4 \end{cases}$$

The trick relies on making a guess: that the optimal solution to (**P**) has $x_1 + x_2 + x_3 \leq 100$.

The trick

Consider the following primal-dual pair of linear programs:

$$(\mathbf{P}) \begin{cases} \text{minimize} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1 + x_2 + x_3 + x_4 = 100 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}$$

The trick relies on making a guess: that the optimal solution to (**P**) has $x_1 + x_2 + x_3 \leq 100$.

The trick, continued

The new **(D)** always has a feasible solution!

$$
(\mathbf{D}) \begin{cases}
\underset{\mathbf{u}\in\mathbb{R}^3}{\text{maximize}} & 2u_1 + 3u_2 + 100u_3 \\
\text{subject to} & u_1 + u_2 + u_3 \leq 2 \\
& 2u_1 - u_2 + u_3 \leq -1 \\
& -3u_1 + u_2 + u_3 \leq 4 \\
& u_3 \leq 0
\end{cases}
$$

The trick, continued

The new (**D**) always has a feasible solution!

$$(\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^3}{\text{maximize}} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}$$

- Set $u_1 = u_2 = 0$. (In general, set all variables to 0 except the extra one, u_{m+1}.)
- The inequalities simplify to $u_3 \leq 2$, $u_3 \leq -1$, $u_3 \leq 4$, $u_3 \leq 0$. (In general, to many upper bounds on u_{m+1}.)
- Set $u_3 = -1$. (In general, set u_{m+1} to the least upper bound.)

An example of the primal-dual method

Let's solve this example to see the primal-dual algorithm in action.

An example of the primal-dual method

Let's solve this example to see the primal-dual algorithm in action.

$$
(\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^3}{\text{maximize}} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}
$$

An example of the primal-dual method

Let's solve this example to see the primal-dual algorithm in action.

$$
(\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^3}{\text{maximize}} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}
$$

At $\mathbf{u} = (0, 0, -1)$, only the second constraint is tight.

An example of the primal-dual method

Let's solve this example to see the primal-dual algorithm in action.

$$(\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^3}{\text{maximize}} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}$$

At $\mathbf{u} = (0, 0, -1)$, only the second constraint is tight.

In $(\mathbf{RP})$, all variables except x_2 will be frozen.

An example of the primal-dual method

Let's solve this example to see the primal-dual algorithm in action.

$$(\mathbf{D}) \begin{cases} \underset{\mathbf{u}\in\mathbb{R}^3}{\text{maximize}} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}$$

At $\mathbf{u} = (0, 0, -1)$, only the second constraint is tight.

In (**RP**), all variables except x_2 will be frozen.

We will start (**RP**) with the basic feasible solution it always has: where the **y**-variables are all basic.

Writing down (**RP**)'s tableau

We look at (**P**) to write a starting tableau for (**RP**).

$$
(\mathbf{P}) \begin{cases} \text{minimize} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1 + x_2 + x_3 + x_4 = 100 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{cases}
$$

Although only x_2 will be present in (**RP**), we'll include all columns, and "freeze" the ones we don't want.

Writing down (RP)'s tableau

We look at (**P**) to write a starting tableau for (**RP**).

$$(\mathbf{P}) \begin{cases} \text{minimize} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1 + x_2 + x_3 + x_4 = 100 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

Although only x_2 will be present in (**RP**), we'll include all columns, and "freeze" the ones we don't want.

	$\mathbf{x_1}$	x_2	$\mathbf{x_3}$	$\mathbf{x_4}$	y_1	y_2	y_3	
y_1	**1**	2	**−3**	**0**	1	0	0	2
y_2	**1**	−1	**1**	**0**	0	1	0	3
y_3	**1**	1	**1**	**1**	0	0	1	100
$-z_{rp}$	**0**	0	**0**	**0**	1	1	1	0

Writing down (RP)'s tableau

We look at (**P**) to write a starting tableau for (**RP**).

$$(\mathbf{P}) \begin{cases} \text{minimize} & 2x_1 - x_2 + 4x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 = 2 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1 + x_2 + x_3 + x_4 = 100 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

Although only x_2 will be present in (**RP**), we'll include all columns, and "freeze" the ones we don't want.

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
y_1	1	2	−3	0	1	0	0	2
y_2	1	−1	1	0	0	1	0	3
y_3	1	1	1	1	0	0	1	100
$-z_{rp}$	−3	−2	1	−1	0	0	0	−105

The first iteration: pivoting in (RP)

In this tableau, there's only one pivoting step we can do: bring in x_2, remove y_1.

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_2	$1/2$	1	$-3/2$	0	$1/2$	0	0	1
y_2	$3/2$	0	$-1/2$	0	$1/2$	1	0	4
y_3	$1/2$	0	$5/2$	1	$-1/2$	0	1	99
$-z_{rp}$	-2	0	-2	-1	1	0	0	-103

The first iteration: pivoting in (**RP**)

In this tableau, there's only one pivoting step we can do: bring in x_2, remove y_1.

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_2	$1/2$	1	$-3/2$	**0**	$1/2$	0	0	1
y_2	$3/2$	0	$-1/2$	**0**	$1/2$	1	0	4
y_3	$1/2$	0	$5/2$	**1**	$-1/2$	0	1	99
$-z_{rp}$	**−2**	0	**−2**	**−1**	1	0	0	−103

The optimal solution to (**DRP**) has

$$\mathbf{v} = \mathbf{1} - \mathbf{r}_{\mathcal{Y}} = (1, 1, 1) - (0, 0, 0) = (0, 1, 1).$$

The first iteration: pivoting in (**RP**)

In this tableau, there's only one pivoting step we can do: bring in x_2, remove y_1.

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_2	$1/2$	1	$-3/2$	0	$1/2$	0	0	1
y_2	$3/2$	0	$-1/2$	0	$1/2$	1	0	4
y_3	$1/2$	0	$5/2$	1	$-1/2$	0	1	99
$-z_{rp}$	-2	0	-2	-1	1	0	0	-103

The optimal solution to (**DRP**) has

$$\mathbf{v} = \mathbf{1} - \mathbf{r}_{\mathcal{Y}} = (1,1,1) - (0,0,0) = (0,1,1).$$

Next, we will augment $\mathbf{u} = (0,0,-1)$ by adding a multiple of $\mathbf{v} = (0,1,1)$ to it, while maintaining dual feasibility.

The first iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$
(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}
$$

We are going from $\mathbf{u} = (0, 0, -1)$ to $\mathbf{u} + t\mathbf{v} = (0, t, t-1)$.

The first iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}$$

We are going from $\mathbf{u} = (0, 0, -1)$ to $\mathbf{u} + t\mathbf{v} = (0, t, t - 1)$.

- $u_1 + u_2 + u_3 \leq 2$ says $2t - 1 \leq 2$ or $t \leq \frac{3}{2}$.

The first iteration: augmenting u

Here are the dual constraints:

$$
(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}
$$

We are going from $\mathbf{u} = (0, 0, -1)$ to $\mathbf{u} + t\mathbf{v} = (0, t, t-1)$.

- $u_1 + u_2 + u_3 \leq 2$ says $2t - 1 \leq 2$ or $t \leq \frac{3}{2}$.
- $2u_1 - u_2 + u_3 \leq -1$ says $-1 \leq -1$. (It will remain tight but never be violated.)

The first iteration: augmenting u

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}$$

We are going from $\mathbf{u} = (0, 0, -1)$ to $\mathbf{u} + t\mathbf{v} = (0, t, t-1)$.

- $u_1 + u_2 + u_3 \le 2$ says $2t - 1 \le 2$ or $t \le \frac{3}{2}$.
- $2u_1 - u_2 + u_3 \le -1$ says $-1 \le -1$. (It will remain tight but never be violated.)
- $-3u_1 + u_2 + u_3 \le 4$ says $2t - 1 \le 4$ or $t \le \frac{5}{2}$.

The first iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}$$

We are going from $\mathbf{u} = (0, 0, -1)$ to $\mathbf{u} + t\mathbf{v} = (0, t, t-1)$.

- $u_1 + u_2 + u_3 \le 2$ says $2t - 1 \le 2$ or $t \le \frac{3}{2}$.
- $2u_1 - u_2 + u_3 \le -1$ says $-1 \le -1$. (It will remain tight but never be violated.)
- $-3u_1 + u_2 + u_3 \le 4$ says $2t - 1 \le 4$ or $t \le \frac{5}{2}$.
- $u_3 \le 0$ says $t \le 1$. (It becomes tight when $t = 1$.)

Preparing the second iteration

Out of $t \le \frac{3}{2}$, $t \le \frac{5}{2}$, $t \le 1$, the limit $t = 1$ is the strictest, so we go to the new point $\mathbf{u} + 1\mathbf{v} = (0, 1, 0)$.

Preparing the second iteration

Out of $t \leq \frac{3}{2}$, $t \leq \frac{5}{2}$, $t \leq 1$, the limit $t = 1$ is the strictest, so we go to the new point $\mathbf{u} + 1\mathbf{v} = (0, 1, 0)$.

We saw that the second constraint of $(\mathbf{D})$ remains tight, and at $t = 1$, the fourth constraint becomes tight.

Preparing the second iteration

Out of $t \leq \frac{3}{2}$, $t \leq \frac{5}{2}$, $t \leq 1$, the limit $t = 1$ is the strictest, so we go to the new point $\mathbf{u} + 1\mathbf{v} = (0, 1, 0)$.

We saw that the second constraint of (**D**) remains tight, and at $t = 1$, the fourth constraint becomes tight.

In our tableau for (**RP**), we unfreeze x_4:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_2	$1/2$	1	$-3/2$	0	$1/2$	0	0	1
y_2	$3/2$	0	$-1/2$	0	$1/2$	1	0	4
y_3	$1/2$	0	$5/2$	1	$-1/2$	0	1	99
$-z_{rp}$	-2	0	-2	-1	1	0	0	-103

The second iteration: pivoting in (**RP**)

In this tableau, once we pivot to bring in x_4 and remove y_3, we're optimal again:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_2	$1/2$	1	$-3/2$	0	$1/2$	0	0	1
y_2	$3/2$	0	$-1/2$	0	$1/2$	1	0	4
x_4	$1/2$	0	$5/2$	1	$-1/2$	0	1	99
$-z_{rp}$	$-3/2$	0	$1/2$	0	$1/2$	0	1	-4

The second iteration: pivoting in (RP)

In this tableau, once we pivot to bring in x_4 and remove y_3, we're optimal again:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_2	$1/2$	1	$-3/2$	0	$1/2$	0	0	1
y_2	$3/2$	0	$-1/2$	0	$1/2$	1	0	4
x_4	$1/2$	0	$5/2$	1	$-1/2$	0	1	99
$-z_{rp}$	$-3/2$	0	$1/2$	0	$1/2$	0	1	-4

The optimal solution to (**DRP**) has

$$\mathbf{v} = \mathbf{1} - \mathbf{r}_{\mathcal{Y}} = (1,1,1) - (\tfrac{1}{2},0,1) = (\tfrac{1}{2},1,0).$$

The second iteration: pivoting in (RP)

In this tableau, once we pivot to bring in x_4 and remove y_3, we're optimal again:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_2	$1/2$	1	$-3/2$	0	$1/2$	0	0	1
y_2	$3/2$	0	$-1/2$	0	$1/2$	1	0	4
x_4	$1/2$	0	$5/2$	1	$-1/2$	0	1	99
$-z_{rp}$	$-3/2$	0	$1/2$	0	$1/2$	0	1	-4

The optimal solution to (**DRP**) has

$$\mathbf{v} = \mathbf{1} - \mathbf{r}_{\mathcal{Y}} = (1,1,1) - (\tfrac{1}{2},0,1) = (\tfrac{1}{2},1,0).$$

Next, we will augment $\mathbf{u} = (0,1,0)$ by adding a multiple of $\mathbf{v} = (\frac{1}{2},1,0)$ to it, while maintaining dual feasibility.

The second iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$
(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}
$$

We are going from $\mathbf{u} = (0, 1, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{2}t, 1 + t, 0)$.

The second iteration: augmenting u

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}$$

We are going from $\mathbf{u} = (0, 1, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{2}t, 1 + t, 0)$.

- $u_1 + u_2 + u_3 \leq 2$ says $1 + \frac{3}{2}t \leq 2$ or $t \leq \frac{2}{3}$.

The second iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$
(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}
$$

We are going from $\mathbf{u} = (0, 1, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{2}t, 1 + t, 0)$.

- $u_1 + u_2 + u_3 \le 2$ says $1 + \frac{3}{2}t \le 2$ or $t \le \frac{2}{3}$.
- $2u_1 - u_2 + u_3 \le -1$ says $-1 \le -1$. (It will remain tight but never be violated.)

The second iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}$$

We are going from $\mathbf{u} = (0, 1, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{2}t, 1 + t, 0)$.

- $u_1 + u_2 + u_3 \le 2$ says $1 + \frac{3}{2}t \le 2$ or $t \le \frac{2}{3}$.
- $2u_1 - u_2 + u_3 \le -1$ says $-1 \le -1$. (It will remain tight but never be violated.)
- $-3u_1 + u_2 + u_3 \le 4$ says $1 - \frac{1}{2}t \le 4$ or $t \ge -6$. (Not relevant.)

The second iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$
(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}
$$

We are going from $\mathbf{u} = (0, 1, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{2}t, 1 + t, 0)$.

- $u_1 + u_2 + u_3 \leq 2$ says $1 + \frac{3}{2}t \leq 2$ or $t \leq \frac{2}{3}$.
- $2u_1 - u_2 + u_3 \leq -1$ says $-1 \leq -1$. (It will remain tight but never be violated.)
- $-3u_1 + u_2 + u_3 \leq 4$ says $1 - \frac{1}{2}t \leq 4$ or $t \geq -6$. (Not relevant.)
- $u_3 \leq 0$ says $0 \leq 0$. (It will remain tight but never be violated.)

Preparing the third iteration

Our only limit on t is $t \leq \frac{2}{3}$, so we go to the new point $\mathbf{u} + \frac{2}{3}\mathbf{v} = (\frac{1}{3}, \frac{5}{3}, 0)$.

Preparing the third iteration

Our only limit on t is $t \leq \frac{2}{3}$, so we go to the new point $\mathbf{u} + \frac{2}{3}\mathbf{v} = (\frac{1}{3}, \frac{5}{3}, 0)$.

The second and fourth constraint of $(\mathbf{D})$ remain tight; at $t = \frac{2}{3}$, the first constraint also becomes tight.

Preparing the third iteration

Our only limit on t is $t \leq \frac{2}{3}$, so we go to the new point $\mathbf{u} + \frac{2}{3}\mathbf{v} = (\frac{1}{3}, \frac{5}{3}, 0)$.

The second and fourth constraint of (**D**) remain tight; at $t = \frac{2}{3}$, the first constraint also becomes tight.

In our tableau for (**RP**), we unfreeze x_1:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_2	$1/2$	1	$-3/2$	0	$1/2$	0	0	1
y_2	$3/2$	0	$-1/2$	0	$1/2$	1	0	4
x_4	$1/2$	0	$5/2$	1	$-1/2$	0	1	99
$-z_{rp}$	$-3/2$	0	$1/2$	0	$1/2$	0	1	-4

The third iteration: pivoting in (**RP**)

In this tableau, we can pivot on x_1, and it will replace x_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	2	-3	0	1	0	0	2
y_2	0	-3	4	0	-1	1	0	1
x_4	0	-1	4	1	-1	0	1	98
$-z_{rp}$	0	3	-4	0	2	0	1	-1

The third iteration: pivoting in (**RP**)

In this tableau, we can pivot on x_1, and it will replace x_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	2	**−3**	0	1	0	0	2
y_2	0	−3	**4**	0	−1	1	0	1
x_4	0	−1	**4**	1	−1	0	1	98
$-z_{rp}$	0	3	**−4**	0	2	0	1	−1

The optimal solution to (**DRP**) has

$$\mathbf{v} = \mathbf{1} - \mathbf{r}_{\mathcal{Y}} = (1,1,1) - (2,0,1) = (-1,1,0).$$

The third iteration: pivoting in (RP)

In this tableau, we can pivot on x_1, and it will replace x_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	2	−3	0	1	0	0	2
y_2	0	−3	4	0	−1	1	0	1
x_4	0	−1	4	1	−1	0	1	98
$-z_{rp}$	0	3	−4	0	2	0	1	−1

The optimal solution to (**DRP**) has

$$\mathbf{v} = \mathbf{1} - \mathbf{r}_{\mathcal{Y}} = (1, 1, 1) - (2, 0, 1) = (-1, 1, 0).$$

Next, we will augment $\mathbf{u} = (\frac{1}{3}, \frac{5}{3}, 0)$ by adding a multiple of $\mathbf{v} = (-1, 1, 0)$ to it, while maintaining dual feasibility.

The third iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$
(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}
$$

We are going from $\mathbf{u} = (\frac{1}{3}, \frac{5}{3}, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{3} - t, \frac{5}{3} + t, 0)$.

The third iteration: augmenting u

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}$$

We are going from $\mathbf{u} = (\frac{1}{3}, \frac{5}{3}, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{3} - t, \frac{5}{3} + t, 0)$.

- $u_1 + u_2 + u_3 \leq 2$ says $2 \leq 2$. (It will remain tight but never be violated.)

The third iteration: augmenting $\mathbf{u}$

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \le 2 \\ & 2u_1 - u_2 + u_3 \le -1 \\ & -3u_1 + u_2 + u_3 \le 4 \\ & u_3 \le 0 \end{cases}$$

We are going from $\mathbf{u} = (\frac{1}{3}, \frac{5}{3}, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{3} - t, \frac{5}{3} + t, 0)$.

- $u_1 + u_2 + u_3 \le 2$ says $2 \le 2$. (It will remain tight but never be violated.)
- $2u_1 - u_2 + u_3 \le -1$ says $-1 - t \le -1$. (For $t > 0$, it will become slack.)

The third iteration: augmenting **u**

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}$$

We are going from $\mathbf{u} = (\frac{1}{3}, \frac{5}{3}, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{3} - t, \frac{5}{3} + t, 0)$.

- $u_1 + u_2 + u_3 \leq 2$ says $2 \leq 2$. (It will remain tight but never be violated.)
- $2u_1 - u_2 + u_3 \leq -1$ says $-1 - t \leq -1$. (For $t > 0$, it will become slack.)
- $-3u_1 + u_2 + u_3 \leq 4$ says $\frac{2}{3} + 4t \leq 4$ or $t \leq \frac{5}{6}$.

The third iteration: augmenting **u**

Here are the dual constraints:

$$(\mathbf{D}) \begin{cases} \text{maximize} & 2u_1 + 3u_2 + 100u_3 \\ \text{subject to} & u_1 + u_2 + u_3 \leq 2 \\ & 2u_1 - u_2 + u_3 \leq -1 \\ & -3u_1 + u_2 + u_3 \leq 4 \\ & u_3 \leq 0 \end{cases}$$

We are going from $\mathbf{u} = (\frac{1}{3}, \frac{5}{3}, 0)$ to $\mathbf{u} + t\mathbf{v} = (\frac{1}{3} - t, \frac{5}{3} + t, 0)$.

- $u_1 + u_2 + u_3 \leq 2$ says $2 \leq 2$. (It will remain tight but never be violated.)
- $2u_1 - u_2 + u_3 \leq -1$ says $-1 - t \leq -1$. (For $t > 0$, it will become slack.)
- $-3u_1 + u_2 + u_3 \leq 4$ says $\frac{2}{3} + 4t \leq 4$ or $t \leq \frac{5}{6}$.
- $u_3 \leq 0$ says $0 \leq 0$. (It will remain tight but never be violated.)

Preparing the fourth iteration

Our only limit on t is $t \leq \frac{5}{6}$, so we go to the new point $\mathbf{u} + \frac{5}{6}\mathbf{v} = (-\frac{1}{2}, \frac{5}{2}, 0)$.

Preparing the fourth iteration

Our only limit on t is $t \leq \frac{5}{6}$, so we go to the new point $\mathbf{u} + \frac{5}{6}\mathbf{v} = (-\frac{1}{2}, \frac{5}{2}, 0)$.

The first and fourth constraint of $(\mathbf{D})$ remain tight; at $t = \frac{5}{6}$, the third constraint also becomes tight. However, the second constraint becomes slack.

Preparing the fourth iteration

Our only limit on t is $t \le \frac{5}{6}$, so we go to the new point $\mathbf{u} + \frac{5}{6}\mathbf{v} = (-\frac{1}{2}, \frac{5}{2}, 0)$.

The first and fourth constraint of $(\mathbf{D})$ remain tight; at $t = \frac{5}{6}$, the third constraint also becomes tight. However, the second constraint becomes slack.

In our tableau for $(\mathbf{RP})$, we unfreeze x_3 but freeze x_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	2	−3	0	1	0	0	2
y_2	0	−3	4	0	−1	1	0	1
x_4	0	−1	4	1	−1	0	1	98
$-z_{rp}$	0	3	−4	0	2	0	1	−1

The fourth iteration: pivoting in (**RP**)

In this tableau, we can pivot on x_3, and it will replace y_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	$-1/4$	0	0	$1/4$	$3/4$	0	$11/4$
x_3	0	$-3/4$	1	0	$-1/4$	$1/4$	0	$1/4$
x_4	0	2	0	1	0	-1	1	97
$-z_{rp}$	0	0	0	0	1	1	1	0

The fourth iteration: pivoting in (RP)

In this tableau, we can pivot on x_3, and it will replace y_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	$-1/4$	0	0	$1/4$	$3/4$	0	$11/4$
x_3	0	$-3/4$	1	0	$-1/4$	$1/4$	0	$1/4$
x_4	0	2	0	1	0	-1	1	97
$-z_{rp}$	0	0	0	0	1	1	1	0

Because $z_{rp} = 0$ and because $\mathbf{v} = (0, 0, 0)$, we know we're done.

The fourth iteration: pivoting in (RP)

In this tableau, we can pivot on x_3, and it will replace y_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	$-1/4$	0	0	$1/4$	$3/4$	0	$11/4$
x_3	0	$-3/4$	1	0	$-1/4$	$1/4$	0	$1/4$
x_4	0	2	0	1	0	-1	1	97
$-z_{rp}$	0	0	0	0	1	1	1	0

Because $z_{rp} = 0$ and because $\mathbf{v} = (0, 0, 0)$, we know we're done.

- Our current $\mathbf{u} = (-\frac{1}{2}, \frac{5}{2}, 0)$ is the optimal solution to (**D**).

The fourth iteration: pivoting in (RP)

In this tableau, we can pivot on x_3, and it will replace y_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	$-1/4$	0	0	$1/4$	$3/4$	0	$11/4$
x_3	0	$-3/4$	1	0	$-1/4$	$1/4$	0	$1/4$
x_4	0	**2**	0	1	0	-1	1	97
$-z_{rp}$	0	**0**	0	0	1	1	1	0

Because $z_{rp} = 0$ and because $\mathbf{v} = (0, 0, 0)$, we know we're done.

- Our current $\mathbf{u} = (-\frac{1}{2}, \frac{5}{2}, 0)$ is the optimal solution to (**D**).
- From (**RP**), we read off $\mathbf{x} = (\frac{11}{4}, 0, \frac{1}{4}, 97)$, the optimal solution to (**P**).

The fourth iteration: pivoting in (RP)

In this tableau, we can pivot on x_3, and it will replace y_2:

	x_1	x_2	x_3	x_4	y_1	y_2	y_3	
x_1	1	$-1/4$	0	0	$1/4$	$3/4$	0	$11/4$
x_3	0	$-3/4$	1	0	$-1/4$	$1/4$	0	$1/4$
x_4	0	2	0	1	0	-1	1	97
$-z_{rp}$	0	0	0	0	1	1	1	0

Because $z_{rp} = 0$ and because $\mathbf{v} = (0, 0, 0)$, we know we're done.

- Our current $\mathbf{u} = (-\frac{1}{2}, \frac{5}{2}, 0)$ is the optimal solution to (**D**).
- From (**RP**), we read off $\mathbf{x} = (\frac{11}{4}, 0, \frac{1}{4}, 97)$, the optimal solution to (**P**).

(Ignoring u_3 and x_4, $\mathbf{u} = (-\frac{1}{2}, \frac{5}{2})$ and $\mathbf{x} = (\frac{11}{4}, 0, \frac{1}{4})$ are optimal for the original (**D**) and (**P**).)

Comments on this method

- From the point of view of (**RP**), we've been solving one simplex tableau the whole time.

Comments on this method

- From the point of view of (**RP**), we've been solving one simplex tableau the whole time.
- The augmenting steps give us "hints" about which variables not to pivot on, in the form of frozen variables.

Comments on this method

- From the point of view of (**RP**), we've been solving one simplex tableau the whole time.

- The augmenting steps give us "hints" about which variables not to pivot on, in the form of frozen variables.

  (In a perfect world, there is always only one variable to pivot on: the unfrozen variables are the ones that were already basic, and the one whose dual constraint just became tight. But sometimes this doesn't work out.)

Comments on this method

- From the point of view of (**RP**), we've been solving one simplex tableau the whole time.

- The augmenting steps give us "hints" about which variables not to pivot on, in the form of frozen variables.

  (In a perfect world, there is always only one variable to pivot on: the unfrozen variables are the ones that were already basic, and the one whose dual constraint just became tight. But sometimes this doesn't work out.)

- This algorithm is well-suited for the revised simplex method.

  If we use it, we don't have to keep around the frozen columns: we just compute columns of the tableau as we need them.