Converting a Linear Program to Standard Form

In this tutorial, we briefly explain what standard form is, and how to convert a linear program to standard form

Cleaver, an MIT Beaver

Linear Programs in Standard Form

We say that a linear program is *in standard form* if the following are all true:

1. **Non-negativity constraints for all variables.**
2. **All remaining constraints are expressed as equality constraints.**
3. **The right hand side vector, *b*, is non-negative.**

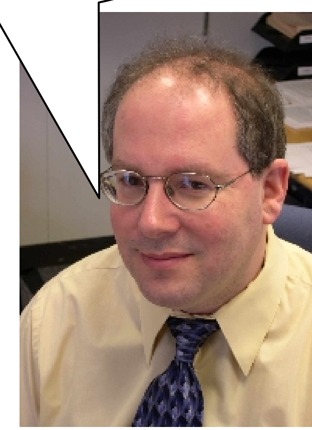
An LP not in Standard Form

maximize	$z = 3x_1 + 2x_2 - x_3 + x_4$	
	$x_1 + 2x_2 + x_3 - x_4 \leq 5$;	**not equality**
	$-2x_1 - 4x_2 + x_3 + x_4 \leq -1$;	**not equality, and negative RHS**
	$x_1 \geq 0, x_2 \geq 0$	**x_3 and x_4 may be negative**

And …

Isn’t this exactly the same as was done in Lecture 4 of 15.053?

Yes. I’m repeating some of the material from Lecture 4.
But, I’m also adding some new material.



Tim, the turkey

Converting Inequalities into Equalities Plus Non-negatives

Before

$x_1 + 2x_2 + x_3 - x_4 \leq 5$

To convert a "≤" constraint to an equality, add a slack variable.

After

$x_1 + 2x_2 + x_3 - x_4 + s_1 = 5$

$s_1 \geq 0$

s_1 is called a *slack variable*, which measures the amount of "unused resource."

Note that $s_1 = 5 - x_1 - 2x_2 - x_3 + x_4$.

Before

$x_1 + 2x_2 + x_3 \geq 7$

To convert a "≥" constraint to an equality, add a surplus variable.

After

$x_1 + 2x_2 + x_3 - s_2 = 7$

$s_2 \geq 0$

s_2 is called a *surplus variable*, which measures the amount of excess.

Note that $s_1 = 7 - x_1 - 2x_2 - x_3$.

Transforming Max to Min

How can one convert a minimization problem to a maximization problem?

Example: Minimize $z = 3x_1 + 2x_2$

subject to "constraints"

Has the same optimum solution(s) as

Maximize $v = -3x_1 - 2x_2$

subject to "constraints"

And …

So, when is there going to be any new material?

Please be patient. The transformations on the next slides were not presented in class.

Tim

Getting Rid of Negative Variables

Before

Max $z = 3x_1 + 4x_2 + 5x_3$

s.t. $2x_1 - 5x_2 + 2x_3 = 7$

other constraints

$x_1 \leq 0,\ x_2 \geq 0,\ x_3 \geq 0$

After

max $z = -3y_1 + 4x_2 + 5x_3$

$-2y_1 - 5x_2 + 2x_3 = 7$

other constraints

$y_1 \geq 0,\ x_2 \geq 0\ x_3 \geq 0$

Transformation: replace x_1 by $y_1 = -x_1$; and require that $y_1 \geq 0$.

If y_1, x_2, x_3 is an optimal solution for the transformed problem, then $-y_1, x_2, x_3$ is an optimal solution for the original problem.

Other constraints?

Why did we writing "other constraints on the previous slide? It doesn't seem to add anything.

"Other constraints" is added to emphasize that the transformation works even if there are lots of constraints. And the transformation $y_1 = -x_1$ is used regardless of the other constraints.

Tim

Ollie,
the computationally
wise owl.

Transforming variables that may take on negative values.

Before

max $z = 3x_1 + 4x_2 + 5x_3$

$2x_1 - 5x_2 + 2x_3 = 7$

other constraints

$x_1 \geq 0$, x_2 is unconstrained in sign, $x_3 \geq 0$

The transformation here is a little more complex. We use the fact that a real number can be expressed as the difference of two non-negative numbers (in an infinite number of ways). For example, -5 can be written as 0 – 5 or as 1-6 or as 2-7 etc.

We replace x_2 by $x_2 = y_3 - y_2$; and require that $y_2 \geq 0$, $y_3 \geq 0$.

After

max $z = 3x_1 + 4(y_3 - y_2) + 5x_3$

$2x_1 - 5y_3 + 5y_2 + 2x_3 = 7$

other constraints

$x_1, y_2, y_3, x_3 \geq 0$

If x_1, y_2, y_3, x_3 is an optimal solution for the transformed problem, then $x_1, y_3 - y_2, x_3$ is an optimal solution for the original problem.

Last Slide

Remember that the major reason we do this is because the simplex method starts with a linear program in standard form. But it turns out that this type of transformation is useful for other types of algorithms too. Perhaps we shall see their usefulness some time later in this course.

Well, that concludes this tutorial on transforming a linear program into standard form. I hope to see you again soon.