**Coursera – Discrete Optimization**

**LP 2: Algebraic View, Naïve Algorithm (**[**link**](https://www.coursera.org/learn/discrete-optimization/lecture/ODOcV/lp-2-algebraic-view-naive-algorithm)**)**

This lecture covers how to move from solving a linear program geometrically to algebraically. As a refreshed on how to solve the LP geometrically:

* Enumerate all the vertices
* Select the one with the smallest objective value

Intuitively, completing these two steps is simple, but how we do that programmatically is through the *simplex algorithm*, which is a more intelligent way to explore the vertices that connects the geometric and algebraic views of the LP.

The simplex algorithm was invented by George Dantzig, and is one of the most interesting algorithms out there because of the following reason: It works incredibly well in practice, solving problems really quickly. There are certain problems that would cause the algorithm incredibly slow, but these problems almost never occur in practice; these problems almost always are constructed for the sole purpose of showing the weakness of the algorithm.

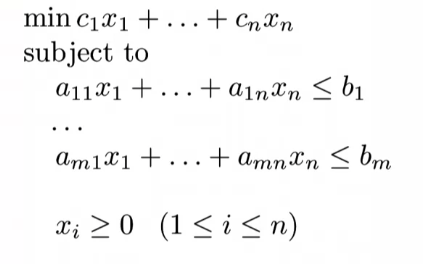
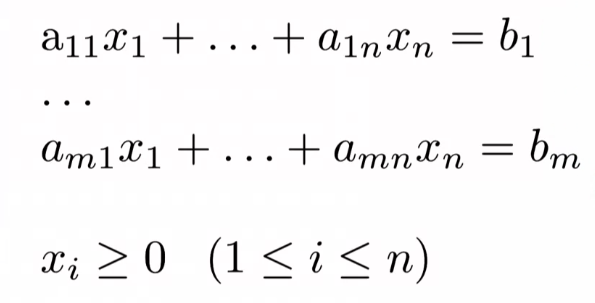
To describe the intuition behind the simplex algorithm, we can use the metaphor of having a goal to be on top of the world. In order to find the point that is at the top of the world, we can make these five conclusions:

1. The top of the world is at the top of a mountain
2. The top of a mountain is a Beautiful Fantastic Spot (BFS)
3. You can move from one BFS to a neighboring BFS
4. If you are at a BFS which is at the top of the world, you know that you are at the top of the world
5. From any BFS, you can move to another BFS

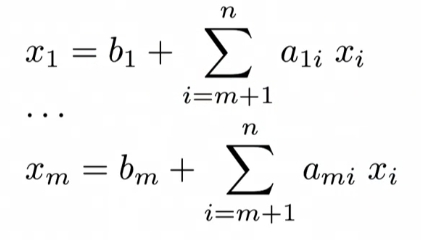
Moving from the metaphor of trying to reach the top of the world to wanting to solve a linear program, we can make these five conclusions:

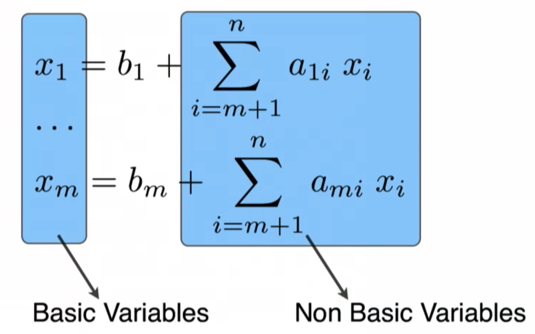
1. An optimal solution is located at a vertex
2. A vertex is a Basic Feasible Solution (BFS)
3. You can move from one BFS to a neighboring BFS
4. You can detect whether a BFS is optimal
5. From any BFS, you can move to a BFS with a better cost (smallest cost if the problem is minimization, largest cost if the problem is maximization)

The remaining part of the lecture focuses on how to express each of these 5 facts algebraically. As mentioned previously, a linear program is a set of linear equations – one of which is called the objective function, others called the constraints.

In order to solve the set of inequality constraints, we can first eliminate the inequality and instead set the constraints .

To solve this system of linear equations, we can use Gaussian elimination. A good description of Gaussian elimination can be found [here](https://math.dartmouth.edu/archive/m23s06/public_html/handouts/row_reduction_examples.pdf). Because the set of constraints all use the same set of variables, we can combine these constraints together in order to simplify the linear equations included in the set of constraints.

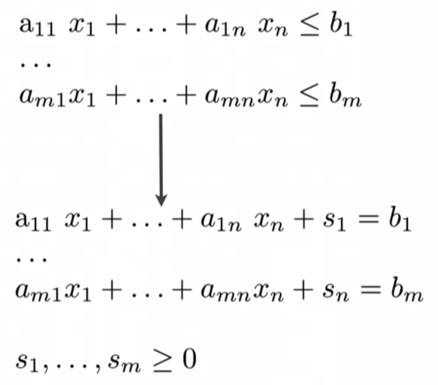
To use Gaussian elimination in the context of the LP, we can express some of the variables in terms of the other ones.

What we did here is to express all the in terms of the others, which is known as a *Basic Solution*. The terms on the left-hand side are called the *basic variables*, while the summations on the right-hand side of the equation are known as the *non-basic variables*.

In order to find a basic solution in this set of linear solutions, we can set all the non-basic variables to 0 which results in assigning all the for each equation in our linear system. Once you have a basic solution for one of the variables, you can then solve for the other in your system of linear equations.

A caveat with the simplex algorithm and how we go from a basic solution to a basic *feasible* solution – all variables (and subsequently, the variables set equal to ) must be non-negative.

*Adding slack variables*

Remember that we had ignored the fact that the constraints in an LP were inequalities, and not equations which allowed us to perform Gaussian elimination. In order to account for the inequalities, we can add *slack variables* to transform the inequalities to equations:

By adding the slack variable term, to each expression, we can transform inequalities into equations. Similar to the being nonnegative, to be a basic feasible solution, all of the have to be nonnegative.

In summary, in order to find a basic feasible solution, these are the steps that must be completed:

* Re-express the constraints as equations by adding slack variables
* Select variables to be the basic variables
* Re-express the basic variables in terms of the non-basic variables via Gaussian elimination
* If all the variables are non-negative, we have a basic feasible solution

*Naïve algorithm to solve LP*

Knowing everything we have covered thus far, we can construct a naïve algorithm to find the optimal BFS since all the vertices of the polytope constructed by the LP constraints is a BFS.

We could generate all basic feasible solutions and select the BFS with the best cost as specified by the objective function.

The issue with this is that there can be a large number of basic feasible solutions, specifically:

A good description of basic and non-basic variables can be found in [this video](https://www.youtube.com/watch?v=KqCrhDVLdOI).

To standardize the constraints of the LP, we add slack variables to make the inequalities into equations. These slack variables, as well as the other *decision* variables that the constraints consist of, must be greater than zero.

By doing this we have a certain number of decision variables, and often times have a *fewer* number of equations representing the constraints (7 decision variables, with 3 equations in the video).

To solve a system of equations with 7 variables and 3 equations, we can set 4 variables equal to 0, and solve the system for the remaining unknown variables with the 3 equations at hand.

The 4 variables we choose to set to 0 in each system of equations are known as the *non-basic variables.*

The 3 variables that we solve for using the equations that are a combination of the constraint inequalities and a slack variable are the *basic variables.*

In this example with 7 decision variables and 3 equations, there are many different ways we can choose 3 variables to solve each system, specifically:

The key to LP becomes how to find the optimal solution out of the 35 different options available. This is simple enough because we can just plug in the values into the objective function, but there may be considerations related to how long it would take your algorithm to solve the LP.