

CE 205A

Transportation Logistics

Lecture 1

Linear Programming Review - Part I

Lecture Outline

- 1 Geometry of LPs
- 2 The Simplex Algorithm

Geometry of LPs

Geometry of LPs

Canonical and Standard Form

One way of describing LPs is using Matrix notation for which we first write the LP in the following Canonical form:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

To exploit ideas from linear algebra, it is also common to write LPs in the following standard form.

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

How do you convert a general LP with \geq or \leq constraints to standard form?

Geometry of LPs

Canonical and Standard Form

Original LP:

$$\begin{aligned} \min \quad & 2x_1 + 5x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 6 \\ & x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Standard Form:

$$\begin{aligned} \min \quad & 2x_1 + 5x_2 \\ \text{s.t.} \quad & x_1 + x_2 - x_3 = 6 \\ & x_1 + 2x_2 + x_4 = 18 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

The variables x_3 and x_4 are called **surplus** and **slack** variables respectively.

Note: **CPLEX** adds slack/artificial variables even if the original constraint is of the equality form.

Geometry of LPs

Example: Stigler's Diet Problem

Consider the problem of optimizing nutritional requirements from various food types. Find the quantity of each food type to purchase to meet the minimum daily requirements (MDR) from different quantities of food while minimizing the total cost. [\[PDF\]](#)



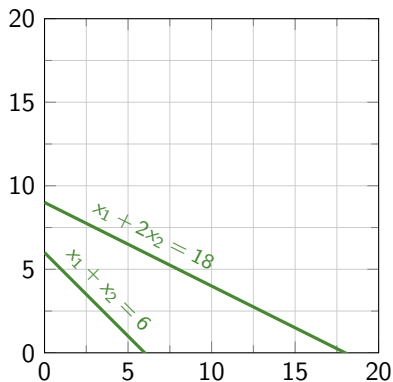
	Calories	Protien	Calcium	Iron	Unit Price
Milk	6.1	310	10.5	18	11
Wheat	44.7	1411	2	365	36
Rice	21.2	460	0.6	41	7.5
Sugar	34.9	0	0	0	51.7
Potatoes	14.3	336	1.8	118	34
Spinach	1.1	106	0	138	8.1
MDR	3000	70	0.8	12	

Geometry of LPs

Feasible Region

What does the feasible region of the following LP look like?

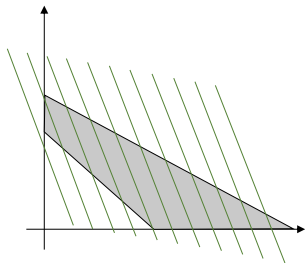
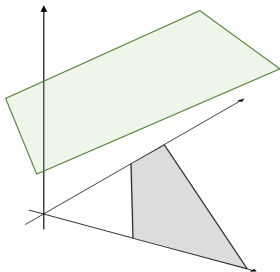
$$\begin{aligned} \min \quad & 2x_1 + 5x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 6 \\ & x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Geometry of LPs

Adding the Objective

In the previous example it is possible to visualize the objective function in the third dimension.



As you can guess by looking at the objective, the optimal value must lie somewhere on the boundary.

Alternately, we can visualize the problem by constructing level sets. Level sets or iso-profit lines are points in the feasible region at which the function is constant.

Geometry of LPs

Solutions

The following cases can arise for solutions to LPs.

- ▶ Unique solution
- ▶ Multiple solutions (infinitely many)
- ▶ Infeasible
- ▶ Unbounded

Can you sketch the feasible region and the objective for each of the above instances?

Geometry of LPs

Column and Null Space

Let us now start looking at general $m \times n$ matrices \mathbf{A} .

Definition (Column Space)

The column space of a matrix \mathbf{A} is defined as the set of vectors obtained by taking the linear combination of all columns of \mathbf{A} and is denoted by $C(\mathbf{A})$. The column space is also called the span of \mathbf{A} .

A system of equations $\mathbf{Ax} = \mathbf{b}$ has a solution only if $\mathbf{b} \in C(\mathbf{A})$. Is the column space a subspace? In fact, it is the smallest subspace containing columns of \mathbf{A} .

Definition (Null Space)

The null space of a matrix \mathbf{A} is denoted by $N(\mathbf{A})$ and is defined as the set of all vectors \mathbf{x} that satisfy $\mathbf{Ax} = \mathbf{0}$

Note that the null space is a subspace of \mathbb{R}^n and the column space is a subspace of \mathbb{R}^m .

Geometry of LPs

Linear Independence

Definition (Span)

A collection of vectors $\mathbf{x}^1, \dots, \mathbf{x}^k \in \mathbb{R}^n$ is said to span \mathbb{R}^n if any vector $\mathbf{b} \in \mathbb{R}^n$ can be expressed as a linear combination of $(\mathbf{x}^1, \dots, \mathbf{x}^k)$.

Definition (Basis)

A collection of vectors $\mathbf{x}^1, \dots, \mathbf{x}^k \in \mathbb{R}^n$ is said to form a basis if it spans \mathbb{R}^n and removing one vector results in a collection that does not span \mathbb{R}^n .

A collection of vectors $\mathbf{x}^1, \dots, \mathbf{x}^k \in \mathbb{R}^n$ forms a basis of \mathbb{R}^n iff $k = n$ and the vectors are linearly independent.

Definition (Basic Solution)

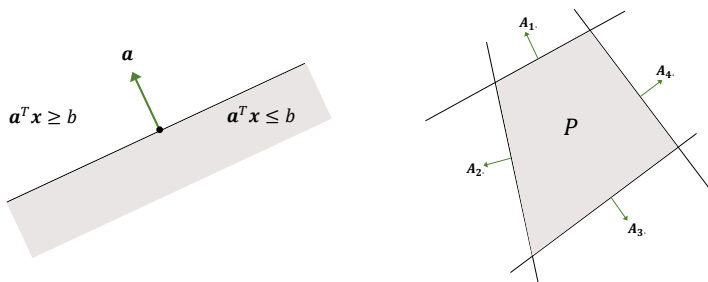
A basic solution of $\mathbf{Ax} = \mathbf{b}$ is a solution that only uses linearly independent columns of \mathbf{A} .

Geometry of LPs

Polyhedra

Definition (Halfspace)

Sets of the form $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^T \mathbf{x} \geq b\}$, where $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{a} \neq \mathbf{0}$, $b \in \mathbb{R}$ are called halfspaces.



Definition (Polyhedron)

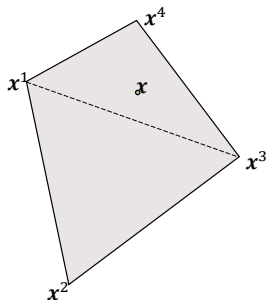
A polyhedron is a set of the form $P = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$ where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$.

Geometry of LPs

Convex Spaces

Definition (Convex Combinations)

Let $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^k \in \mathbb{R}^n$. Suppose $\lambda_1, \lambda_2, \dots, \lambda_k \geq 0$ such that $\sum_{i=1}^k \lambda_i = 1$. Then, $\mathbf{x} = \sum_{i=1}^k \lambda_i \mathbf{x}^i$ is called a convex combination of $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^k$.

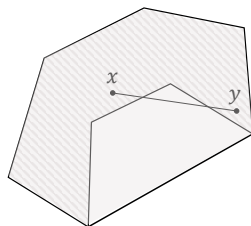


Geometry of LPs

Convex Spaces

Definition (Convex Hull)

Convex hull of a set X is the intersection of all convex sets containing X . It is denoted as $Conv(X)$.



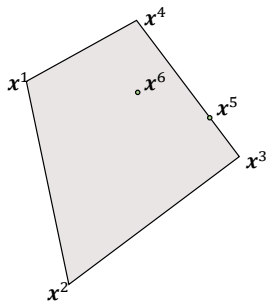
Convex hull of X is the smallest convex set that contains X . It can be defined for spaces containing a discrete number of points as well.

Geometry of LPs

Extreme Points

Definition (Extreme Points)

An element $\mathbf{x} \in X$ is an extreme point of a convex set X if \nexists distinct $\mathbf{x}^1, \mathbf{x}^2 \in X$ and $\lambda \in (0, 1)$ such that $\mathbf{x} = \lambda \mathbf{x}^1 + (1 - \lambda) \mathbf{x}^2$



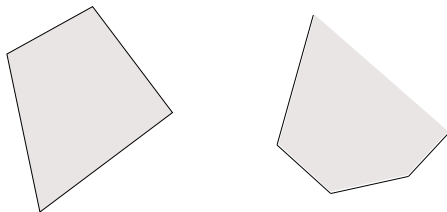
In other words, extreme points are the ones which cannot be expressed as a strict convex combination of any two distinct points of a convex set.

Geometry of LPs

Polytopes

Definition (Polytope)

A polyhedron $P \subset \mathbb{R}^n$ is bounded, also called a polytope, if there exists a constant $C > 0$ such that $|x_i| \leq C \forall i = 1, \dots, n$



All points inside a polytope can be expressed as a convex combination of its extreme points. Mathematically, let $X = \{\mathbf{x} : \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ be a polytope.

$$X = \left\{ \mathbf{x} : \mathbf{x} = \sum_{i=1}^k \lambda_i \mathbf{x}^i, \boldsymbol{\lambda} \geq \mathbf{0}, \sum_{i=1}^k \lambda_i = 1 \right\}$$

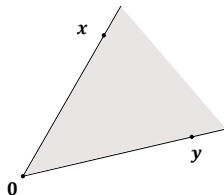
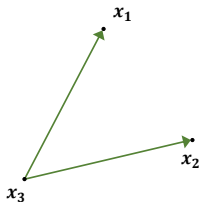
Geometry of LPs

Cones

In order to get a mathematical hold on unbounded polyhedra, it is important to understand the geometry of cones.

Definition (Cone)

A set C is called a cone if for every $\mathbf{x} \in C$ and $\lambda \geq 0$, $\lambda\mathbf{x} \in C$.



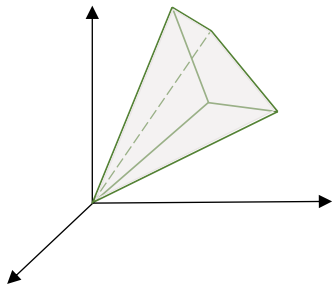
Definition (Convex Cone)

A set C is called a convex cone if it is convex and a cone, i.e., $\forall \mathbf{x}, \mathbf{y} \in C$ and $\lambda_1, \lambda_2 \geq 0$, $\lambda_1\mathbf{x} + \lambda_2\mathbf{y} \in C$.

Geometry of LPs

Polyhedral Cones

Polyhedral cones are special types of cones which are both polyhedra and cones. They can be expressed as non-negative linear combinations of a **finite number of extreme directions**.



Extreme directions are those that cannot be expressed as non-negative combinations of two distinct directions.

Note that the vertex of all cones must be the origin. Also, the linear combination must involve non-negative scalars according to the definition of a cone.

Polyhedral cones have some neat connections with restricted versions of the column spaces of the constraint matrix (Minkowsky-Weyl theorems).

Geometry of LPs

Unbounded Polyhedra

It turns out that unbounded polyhedra can be viewed as a combination of a polytope (with a finite number of extreme points) and a polyhedral cone (with a finite number of extreme directions). For example, consider this feasible region:

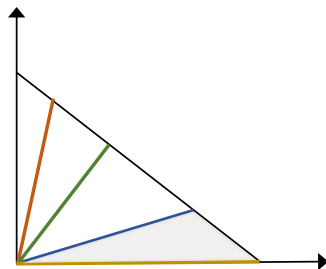
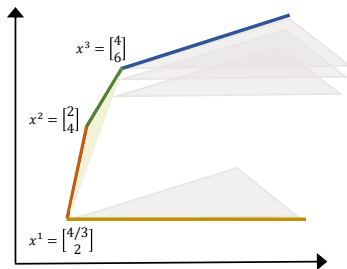
$$-3x_1 + x_2 \leq -2$$

$$-x_1 + x_2 \leq 2$$

$$-x_1 + 2x_2 \leq 8$$

$$-x_2 \leq -2$$

$$x_1, x_2 \geq 0$$



Geometry of LPs

Resolution and Separation Theorem

Theorem (Goldman (1956))

A non-empty polyhedron X can be expressed as the sum of a polytope P and a polyhedral cone C . That is,

$$X = P + C = \left\{ \mathbf{y} : \mathbf{y} = \mathbf{x} + \mathbf{d}, \mathbf{x} \in P, \mathbf{d} \in C \right\}$$

P can be expressed as $\text{Conv}(\{\mathbf{x}^1, \dots, \mathbf{x}^k\})$ and C can be expressed as $C_{\geq}(\{\mathbf{d}^1, \dots, \mathbf{d}^l\})$ (non-negative combination of extreme directions). Alternately, we can write it as,

$$\mathbf{x} = \sum_{i=1}^k \lambda_i \mathbf{x}^i + \sum_{i=1}^l \mu_i \mathbf{d}^i$$

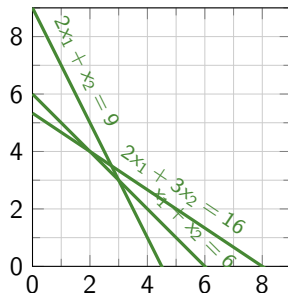
for some $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ satisfying $\sum_{i=1}^k \lambda_i = 1, \boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu} \geq \mathbf{0}$

Geometry of LPs

Row View

Consider the following example. Since we care only about the corner points, let us pick two constraints at a time and find the solution to their equality versions and evaluate the objective at these points. How many such solutions are possible?

$$\begin{aligned} \max \quad & 6x_1 + 4x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 6 \\ & 2x_1 + x_2 \leq 9 \\ & 2x_1 + 3x_2 \leq 16 \\ & x_1, x_2 \geq 0 \end{aligned}$$



The constraints which create the optimal solution are called *active* or *binding* constraints.

Geometry of LPs

Column View

Notice that we can work with the original/canonical LP for the row view. However, for the column view, we will need to convert it into the standard form. The Simplex algorithm follows the column view.

Like before, we can visualize the solutions to $\mathbf{Ax} = \mathbf{b}$ as selecting different amounts of the columns of \mathbf{A} to produce \mathbf{b} . In other words, \mathbf{b} must belong to the polyhedral cone $C_{\geq}(\mathbf{A}_{\cdot 1}, \dots, \mathbf{A}_{\cdot n})$.

Geometry of LPs

Column View

Converting the original LP into standard form:

$$\max 6x_1 + 4x_2$$

$$\text{s.t. } x_1 + x_2 \leq 6$$

$$2x_1 + x_2 \leq 9$$

$$2x_1 + 3x_2 \leq 16$$

$$x_1, x_2 \geq 0$$

$$\max 6x_1 + 4x_2$$

$$\text{s.t. } x_1 + x_2 + x_3 = 6$$

$$2x_1 + x_2 + x_4 = 9$$

$$2x_1 + 3x_2 + x_5 = 16$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

We want $\mathbf{b} = (6, 9, 16)$ to lie in the polyhedral cone created by

$$\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

\mathbf{b} can be constructed by scaling these vectors in multiple ways. Each one of them will produce a certain objective and we need to pick the one with the maximum objective. Note that the solution now lies in a 5D space and we have only 3 equations.

Geometry of LPs

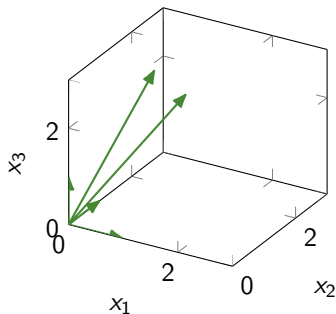
Column View

The trick in the column view is to select three of the five columns at a time and try to produce \mathbf{b} while setting the amounts of the other two to 0. In other words, we are trying to find *basic* solutions.

This reduces the problem to m equations in m unknowns! But we have ignored the non-negativity constraints. So we check if the resulting solution satisfies the left-out constraints. If yes, we call this a *basic feasible solution*.

In this example, can you check if we have a basic feasible solution when we select columns 1, 2, and 3? How about 2, 4, and 5?

$$\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

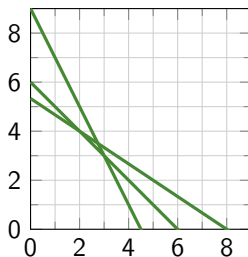


Geometry of LPs

Summary

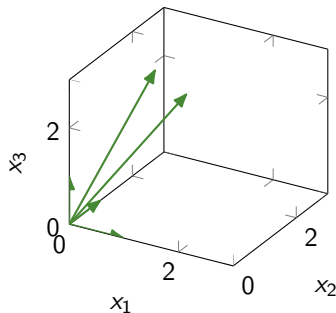
Row View

$$\begin{aligned} \max \quad & 6x_1 + 4x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 6 \\ & 2x_1 + x_2 \leq 9 \\ & 2x_1 + 3x_2 \leq 16 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Column View

$$\begin{aligned} \max \quad & 6x_1 + 4x_2 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 = 6 \\ & 2x_1 + x_2 + x_4 = 9 \\ & 2x_1 + 3x_2 + x_5 = 16 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$



The Simplex Algorithm

The Simplex Algorithm

Motivating Questions

Before we try to design an algorithm that finds the optimal solution to an LP, let us try to characterize it by understanding its properties.

Specifically, we will ask the following questions:

- 1 Do the optima have any special property that we could exploit?
- 2 Does the problem have a minimum or is it unbounded?
- 3 Is there certificate of optimality that we can use to claim that we have reached an optimal solution?

We've seen that answers to Q1 has something to do with extreme points and Q2 is related to extreme directions. We will formalize these now, along with Q3, which helps us terminate the algorithm with an optima.

The Simplex Algorithm

Question 1: What does the Optima Look Like?

We can derive a feasible solution by setting the free variables to 0 and finding a basic solution and checking if it satisfies the non-negative constraints.

Theorem

Suppose $X = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. \mathbf{x}_B is a basic feasible solution of $X \Leftrightarrow \mathbf{x}_B$ is an extreme point of X .

These basic feasible solutions are the same as the extreme points of the feasible region, i.e., they can't be written as a convex combination of two distinct points in X .

The Simplex Algorithm

Question 1: What does the Optima Look Like?

Intuitively, we see why the optimal solution must occur at the extreme point. But why is $\mathbf{c}^T \mathbf{x}$ minimized at such a point? The answer's in Goldman's separation and resolution theorem. To make things simple, let's imagine that the feasible region is a polytope.

$$\begin{aligned} \min_{\lambda} \quad & \sum_{i=1}^k \lambda_i \mathbf{c}^T \mathbf{x}^i \\ \text{s.t.} \quad & \sum_{i=1}^k \lambda_i = 1 \\ & \lambda \geq \mathbf{0} \end{aligned}$$

What are the decision variables in the above model? Can you construct an argument based on contradiction?

The Simplex Algorithm

Question 2: When is an LP Unbounded?

Characterizing unboundedness is easy. Let's use Goldman's separation and resolution theorem again.

$$\begin{aligned} \min_{\lambda, \mu} \quad & \sum_{i=1}^k \lambda_i \mathbf{c}^T \mathbf{x}^i + \sum_{i=1}^l \mu_i \mathbf{c}^T \mathbf{d}^i \\ \text{s.t.} \quad & \sum_{i=1}^k \lambda_i = 1 \\ & \boldsymbol{\lambda} \geq \mathbf{0} \\ & \boldsymbol{\mu} \geq \mathbf{0} \end{aligned}$$

What are the decision variables in the above model? When does the above LP become unbounded?

The Simplex Algorithm

Question 2: When is an LP Unbounded

Theorem

If the feasible region has an extreme direction \mathbf{d} for which $\mathbf{c}^T \mathbf{d} < 0$, then the LP is unbounded.

Answers to Q1 and Q2 indicate that if we can devise a method that finds extreme points and extreme directions, we can check if the problem is unbounded or keep track of the objective for different extreme points and find the optimum.

But how do we know when to stop? Enumerate all corner points? Can we do better?

The Simplex Algorithm

Question 3: Certificate of Optimality

Before we begin, we need to partition the data into those corresponding to the free and pivot columns. We will call these *basic* and *non-basic* variables. The original LP is recast as

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

$$\begin{aligned} \min \quad & \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \\ \text{s.t.} \quad & [\mathbf{B} \quad \mathbf{N}] \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \mathbf{b} \\ & \mathbf{x}_B, \mathbf{x}_N \geq \mathbf{0} \end{aligned}$$

We then set the non-basic variables to 0 and hence $\begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{-1} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$

The Simplex Algorithm

Question 3: Certificate of Optimality

Expanding the constraints we can write $\mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N = \mathbf{b}$. Hence, $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N$. Writing the objective function in terms of the non-basic variables

$$\begin{aligned}z &= \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \\&= \mathbf{c}_B^T (\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N) + \mathbf{c}_N^T \mathbf{x}_N \\&= \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N})\mathbf{x}_N\end{aligned}$$

Theorem

Suppose \mathbf{x}^ is a basic feasible solution and $\mathbf{c}^T - \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{A} \geq 0^T$, then \mathbf{x}^* is optimal*

The Simplex Algorithm

Question 3: Certificate of Optimality

The term $\mathbf{c}^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A} \geq \mathbf{0}^T$ is also called the *reduced cost vector*. Let us denote it using $\bar{\mathbf{c}}^T$. What are the reduced costs of the basic variables?

$$\text{Reduced cost of variable } x_j = c_j - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A}_{.j}$$

We care only about the reduced cost of the non-basic variables $\bar{\mathbf{c}}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$

Theorem (Optimality Condition)

Suppose \mathbf{x}^ is a basic feasible solution and $\bar{\mathbf{c}}_N \geq \mathbf{0}$, then \mathbf{x}^* is optimal*

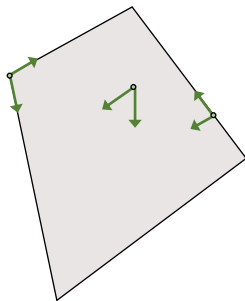
This is an important result in linear programming. We'll soon see that it not only tells us when to stop but also shows us the direction in which we should move if we haven't reached optimality.

The Simplex Algorithm

Descent Direction

In general, we can move from one solution \mathbf{x} to another by taking a step in direction \mathbf{d} . That is, $\hat{\mathbf{x}} = \mathbf{x} + \eta\mathbf{d}$, where η is some positive scalar.

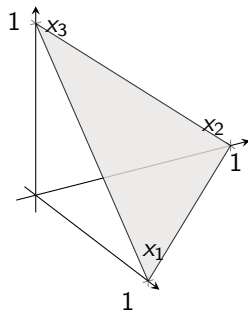
We however don't want to move in any direction, but one that improves the objective. That is, an ideal move should satisfy $\mathbf{c}^T\hat{\mathbf{x}} < \mathbf{c}^T\mathbf{x} \Rightarrow \mathbf{c}^T\mathbf{d} < 0$.



Directions which satisfy $\mathbf{c}^T\mathbf{d} < 0$ are called *descent directions*. Did we see this condition before? How is this different? If $\mathbf{c}^T\mathbf{d} < 0$ and \mathbf{d} is an extreme direction, then the problem is unbounded.

The Simplex Algorithm

Descent Direction



- ▶ How can you move from one basic feasible solution to another?
- ▶ Does moving along the edges produce an extreme direction? Check if $\mathbf{A}\mathbf{d} = \mathbf{0}$, $\mathbf{d} \geq \mathbf{0}$

If $\mathbf{c}^T \mathbf{d} < 0$ and \mathbf{d} is an extreme direction, then the problem is unbounded.

The Simplex Algorithm

Reverse Engineering

In the earlier example, we could see the extreme points and hence we're able to easily construct direction vectors to move from one corner to the other.

How can we do this in the general case? Imagine the column view. Moving from one corner to another is equivalent to moving from one basic feasible solution to another.

What if we move between basic feasible solutions by changing the non-basic/free variable status of one column? Let's increase a non-basic variable x_j by moving in its direction

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{d}_N \end{bmatrix} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{e}_j \end{bmatrix}$$

The Simplex Algorithm

Reverse Engineering

Can you compute $\mathbf{c}^T \mathbf{d}$ when we move in the direction that increases the non-basic variable x_j ?

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{d}_N \end{bmatrix} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{e}_j \end{bmatrix}$$

$$\mathbf{c}^T \mathbf{d} = \begin{bmatrix} \mathbf{c}_B^T & \mathbf{c}_N^T \end{bmatrix} \begin{bmatrix} \mathbf{d}_B \\ \mathbf{e}_j \end{bmatrix} = \mathbf{c}_B^T \mathbf{d}_B + \mathbf{c}_N^T \mathbf{e}_j = \mathbf{c}_B^T \mathbf{d}_B + c_j$$

Recall, that the reduced cost of non-basic variable x_j is $c_j - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A}_{.j}$ and if we are not at an optimal solution, we may find a j such that $\bar{c}_j < 0^*$

Hence, we pick a direction vector

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{d}_N \end{bmatrix} = \begin{bmatrix} -\mathbf{B}^{-1} \mathbf{A}_{.j} \\ \mathbf{e}_j \end{bmatrix}$$

The Simplex Algorithm

Reverse Engineering

Theorem

If a variable x_j has negative reduced cost $\bar{c}_j < 0$, then the direction

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{d}_N \end{bmatrix} = \begin{bmatrix} -\mathbf{B}^{-1}\mathbf{A}_{.j} \\ \mathbf{e}_j \end{bmatrix}$$

is a descent direction, i.e., $\mathbf{c}^T\mathbf{d} < 0$.

The Simplex Algorithm

Step Size

We have a basic feasible solution/corner point and a direction. The immediate question is how far should we move along this direction.

Too small a step size and we will be close to where we started. Too big and we may overshoot the feasible region. We want the right step size to help us land at the extreme point.

Let's find the largest step size which will keep the new point $\hat{\mathbf{x}}$ feasible. Hence, the following conditions must be true

$$\mathbf{A}(\mathbf{x} + \eta\mathbf{d}) = \mathbf{b} \quad (\text{Condition 1})$$

$$\mathbf{x} + \eta\mathbf{d} \geq \mathbf{0} \quad (\text{Condition 2})$$

The Simplex Algorithm

Step Size

For what values of η is Condition 1 true?

$$\begin{aligned}\mathbf{A}(\mathbf{x} + \eta\mathbf{d}) &= \mathbf{b} + \eta\mathbf{A}\mathbf{d} = \mathbf{b} + \eta \begin{bmatrix} \mathbf{B} & \mathbf{N} \end{bmatrix} \begin{bmatrix} -\mathbf{B}^{-1}\mathbf{A}_{\cdot j} \\ \mathbf{e}_j \end{bmatrix} \\ &= \mathbf{b} - \eta\mathbf{B}\mathbf{B}^{-1}\mathbf{A}_{\cdot j} + \eta\mathbf{N}\mathbf{e}_j \\ &= \mathbf{b} - \eta\mathbf{A}_{\cdot j} + \eta\mathbf{A}_{\cdot j} \\ &= \mathbf{b}\end{aligned}$$

This aligns with our observation in the earlier example. Any step along the edges of the feasible region will always keep us on $x_1 + x_2 + x_3 = 1$.

The above calculation also has a hidden result. $\mathbf{A}\mathbf{d} = \mathbf{0}$, or \mathbf{d} is always in the null space of \mathbf{A} . So if $\mathbf{d} \geq \mathbf{0}$, then it is an extreme direction.

Theorem (Infeasibility Condition)

If $\bar{c}_j < 0$ and $\mathbf{d}_B \geq \mathbf{0}$, then the LP is unbounded

The Simplex Algorithm

Step Size

For what values of η is Condition 2 true?

$$\mathbf{x} + \eta \mathbf{d} \geq \mathbf{0} \Rightarrow \begin{bmatrix} \mathbf{x}_B \\ \mathbf{0} \end{bmatrix} + \eta \begin{bmatrix} \mathbf{d}_B \\ \mathbf{e}_j \end{bmatrix} \geq \mathbf{0}$$

We can ignore the rows associated with the non-basic variables. We are left with m inequalities $\mathbf{x}_B + \eta \mathbf{d}_B \geq \mathbf{0}$

Writing this component-wise, $\eta(d_B)_k \geq -(x_B)_k$. What happens when $(d_B)_k \geq 0$ and < 0 ?

In other words,

$$\eta = \min \left\{ -\frac{(x_B)_k}{(d_B)_k} : (d_B)_k < 0 \right\}$$

Why take the minimum? This result is also called the **min-ratio rule**.

The Simplex Algorithm

Pseudocode

We now have all the ingredients to construct an algorithm for finding the optimal solutions of an LP.

In summary, our method will

- ▶ Find a basic feasible solution \mathbf{x}_B and compute the reduced costs and check for its optimality.
- ▶ Determine a descent direction and check if it leads to unboundedness.
- ▶ If not, go as far as possible along that direction while maintaining feasibility.
- ▶ End up at a new corner point and repeat the above steps.

The Simplex Algorithm

Pseudocode

- 1 Start with an initial basic feasible solution $\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$
- 2 Compute the reduced cost vector $\bar{\mathbf{c}}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$
- 3 If $\bar{\mathbf{c}}_N \geq 0$, then \mathbf{x} is optimal and terminate, else go to Step 4
- 4 Pick $j^* : \bar{c}_{j^*} < 0$ and compute descent direction $\mathbf{d} = \begin{bmatrix} -\mathbf{B}^{-1} \mathbf{A}_{\cdot j^*} \\ \mathbf{e}_{j^*} \end{bmatrix}$
- 5 If $\mathbf{d}_B \geq 0$, then the LP is unbounded, else go to Step 6
- 6 Set $k^* \in \arg \min \left\{ -\frac{(x_B)_k}{(d_B)_k} : (d_B)_k < 0 \right\}$
- 7 Modify the basis by swapping $\mathbf{B}_{\cdot k^*}$ and $\mathbf{A}_{\cdot j^*}$, set $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$, and go to Step 2.

The Simplex Algorithm

Example

Consider the problem

$$\begin{aligned} \max \quad & 6x_1 + 4x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 6 \\ & 2x_1 + x_2 \leq 9 \\ & 2x_1 + 3x_2 \leq 16 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Converting it into standard form

$$\begin{aligned} \min \quad & -6x_1 - 4x_2 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 = 6 \\ & 2x_1 + x_2 + x_4 = 9 \\ & 2x_1 + 3x_2 + x_5 = 16 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

The Simplex Algorithm

Example

In matrix notation, the standard form is

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \text{s.t. } \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 2 & 3 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 6 \\ 9 \\ 16 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} -6 \\ -4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The Simplex Algorithm

Iteration 1

Starting with the initial basic feasible solution by setting the slacks to the right hand sides, we have

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 2 & 3 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 6 \\ 9 \\ 16 \end{bmatrix} \mathbf{b} = \begin{bmatrix} 6 \\ 9 \\ 18 \end{bmatrix} \mathbf{c} = \begin{bmatrix} -6 \\ -4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Computing the reduced cost vector using $\bar{\mathbf{c}}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$ and the descent direction using $\mathbf{d}_B = -\mathbf{B}^{-1} \mathbf{A}_{.j^*}$

$$\bar{\mathbf{c}} = \begin{bmatrix} -6 \\ -4 \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathbf{d} = \begin{bmatrix} 1 \\ 0 \\ -1 \\ -2 \\ -2 \end{bmatrix}$$

The Simplex Algorithm

Iteration 2

Swapping the first and fourth columns and setting $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$,

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 2 & 3 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 9/2 \\ 0 \\ 3/2 \\ 0 \\ 9 \end{bmatrix} \mathbf{b} = \begin{bmatrix} 6 \\ 9 \\ 18 \end{bmatrix} \mathbf{c} = \begin{bmatrix} -6 \\ -4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Computing the reduced cost vector using $\bar{\mathbf{c}}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$ and the descent direction using $\mathbf{d}_B = -\mathbf{B}^{-1} \mathbf{A}_{.j^*}$

$$\bar{\mathbf{c}} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 3 \\ 0 \end{bmatrix} \mathbf{d} = \begin{bmatrix} -1/2 \\ 1 \\ -1/2 \\ 0 \\ -2 \end{bmatrix}$$

The Simplex Algorithm

Iteration 3

Swapping the second and third columns and setting $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$,

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 2 & 3 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{b} = \begin{bmatrix} 6 \\ 9 \\ 18 \end{bmatrix} \mathbf{c} = \begin{bmatrix} -6 \\ -4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

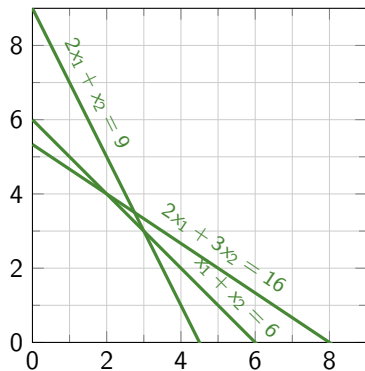
Computing the reduced cost vector using $\bar{\mathbf{c}}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$, we find that all the reduced costs of the non-basic variables are positive. Hence, the current solution is optimal.

$$\bar{\mathbf{c}} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 2 \\ 0 \end{bmatrix}$$

The Simplex Algorithm

Graphical Solution

Can you trace the path of the Simplex iterations on the original feasible region?



The Simplex Algorithm

Why Does it Work?

The series of steps in the Simplex algorithm involved finding inverses and performing matrix multiplications to obtain the reduced cost vector and the descent direction.

$$\begin{aligned} \min z &= \mathbf{c}^T \mathbf{x} \\ \text{s.t. } \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

These steps can be compactly written in a table like format using row operations and Gauss-Jordan style elimination.

	\mathbf{x}_B	\mathbf{x}_{NB}	RHS
RC	$\mathbf{0}$	$\mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$	$-\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b}$
\mathbf{x}_B	\mathbf{I}	$\mathbf{B}^{-1} \mathbf{N}$	$\mathbf{B}^{-1} \mathbf{b}$

The reduced costs follow from row operations that create $\mathbf{0}$ above \mathbf{I} . Each column in the Tableau is $\mathbf{B}^{-1} \mathbf{A}_j$. Hence, the min-cost rule and the unbounded condition in the tableau method involves a sign reversal.

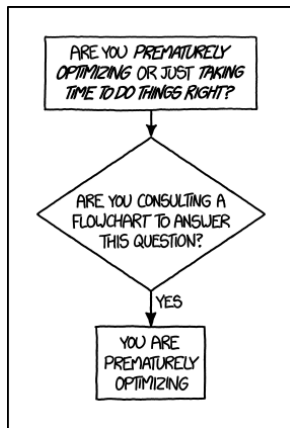
The Simplex Algorithm

Food for Thought

- ▶ How do we find the \mathbf{B}^{-1} in any iteration? This is similar to finding the inverse by augmenting a matrix with the identity matrix $[\mathbf{A} \quad \mathbf{I}]$ and using Gauss-Jordan elimination.
- ▶ Can a variable leave and re-enter the table? Yes, think about binding constraints.
- ▶ Can we have a negative RHS? No, because the solution should be basic feasible. The min-cost rule always ensures we don't create negative RHS during row operations. Recall Condition 2 that gave us the min-cost rule.

Caution: Books on optimization may use different sign conventions in the Simplex Tableau, but they all do similar operations.

Your Moment of Zen



Source: xkcd