DDA5002 Optimization
Lecture 3 Geometry of LP

Yuang Chen

School of Data Scienc
The Chinese University of Hong Kong, Shenzhen

September 17, 2025

Agenda

- Geometry of LP: September 17
- Simplex Method: September 24, 28
- The simplex method is the most challenging part of this class because most students are unfamiliar with matrix-form equations.
- Do your best to understand the underlying mathematical logic. If it's too difficult for you, practice with many exercises, learn how to solve the problems manually by hand, and then revisit the theory to deepen your understanding.

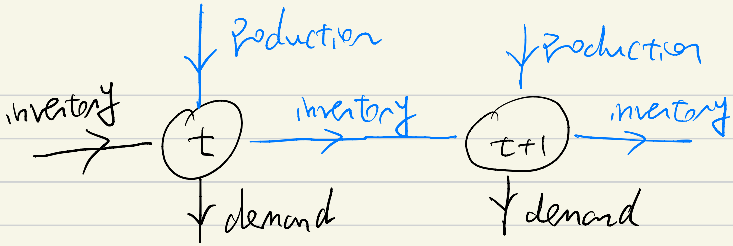
Outline

Outline

Multi-stage Inventory Control

Lot-sizing Problem

- A manufacturing company needs to plan the production and inventory levels for an item for the next T periods.
- The demand in period t is denoted by d_t.
- In each period $t = 1, ..., T$, the company observes the current inventory and demand, then decides the production and inventory levels to meet demand.
- The inventory and production cost is h_t and c_t per item for each period t. The initial inventory is I_0.



Decision variables:

x_t : # items produced at time t

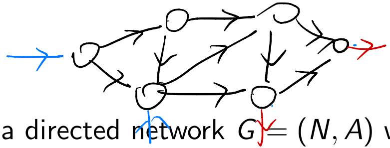
I_t : # items in inventory at the end of time t

$$\min \sum_{t=1}^{T} c_t x_t + h_t I_t$$

$$\text{s.t.} \quad I_{t-1} + x_t = d_t + I_t, \quad \forall t = 1, \dots, T$$

$$x_t \geq 0 \qquad \forall t = 1, \dots, T$$

$$I_t \geq 0$$

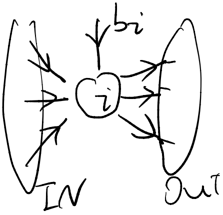


- We are given a directed network $G = (N, A)$ with a set of nodes N and a set of arcs A.
- Each node $i \in N$ has an associated "supply" b_i. If $b_i > 0$ the node is a supply node, if $b_i < 0$ it is a demand node. We will assume that the network is balanced, i.e. $\sum_{i \in N} b_i = 0$.
- Each arc $(i, j) \in A$ has an associated cost c_{ij} and capacity u_{ij}.
- A flow on this network is a set of values on the arcs that obey capacities and satisfy flow conservation at the nodes.
- The goal is to find flows on each arc with minimum cost.

items on (i,j)

MCNF Formulation

x_{ij} : flow on arc (i,j)



$$\min \sum_{(ij)\in A} c_{ij} x_{ij}$$

$$\text{s.t.} \sum_{j:(ij)\in A} \overset{\text{out}}{x_{ij}} - \sum_{j:(ji)\in A} \overset{\text{IN}}{x_{ji}} = b_i \qquad \forall i \in N$$

$$0 \le x_{ij} \le u_{ij} \qquad \forall (ij) \in A$$

L_{ij}

$\cancel{x_{ij} \in \mathbb{Z}}$

MCNF with “Soft” Demands

Manufacturing Plants (Supply)

	min	max
SF	300	500
LA	500	700

Sales Markets (Demand)

	min	max
NY	300	400
CHI	200	400
DAL	250	350

decision!

Production Costs per Unit

Plant	Cost
SF	1
LA	2

Revenue per Unit

Market	Revenue
NY	5
CHI	3
DAL	4

cost

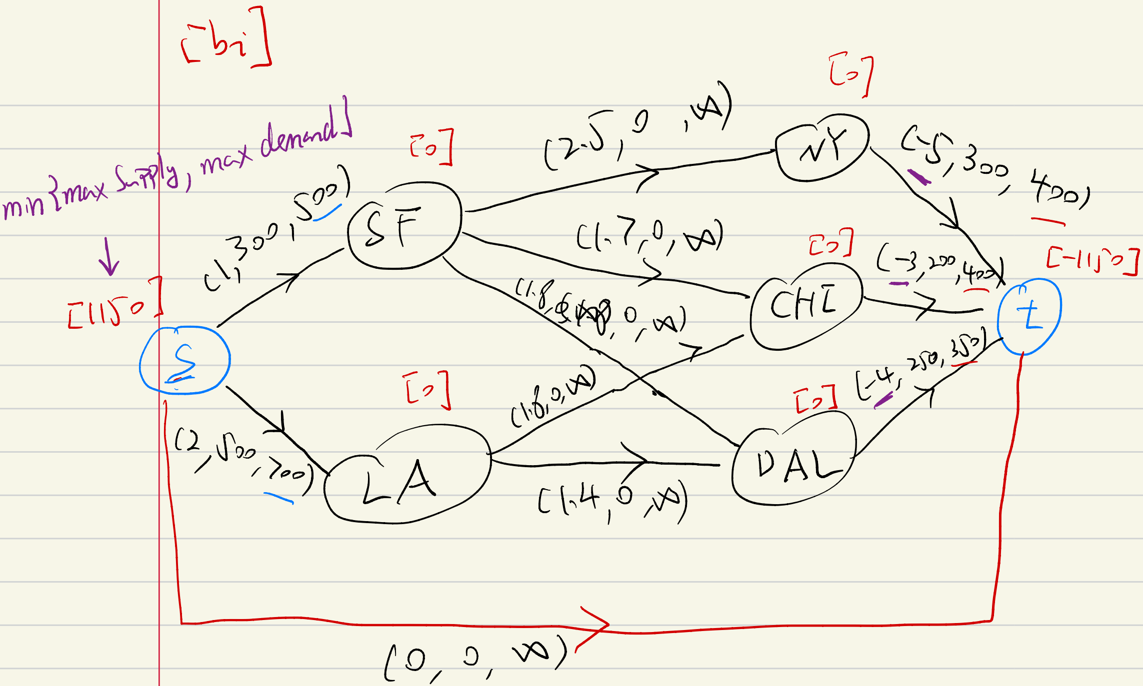
(c_{ij}, l_{ij}, u_{ij})

$i \longrightarrow j$

$l_{ij} \le x_{ij} \le u_{ij}$

Shipping Costs per Unit

	NY	CHI	DAL
SF	2.5	1.7	1.8
LA	–	1.8	1.4



min. total cost

s.t. $IN + b_i = OUT$

$l_{ij} \le flow \le u_{ij}$

$$b_i = \begin{cases} 1150 & i = s \\ -1150 & i = t \\ 0 & i \neq s, t \end{cases}$$

data

Outline

Linear Program Standard Form

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

constraints

variable types

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- $x \in \mathbb{R}^n$, i.e. there are n variables
- $A \in \mathbb{R}^{m \times n}$, i.e. there are m equality constraints
- We always assume all the m equality constraints are linearly independent (non-parallel), otherwise we can remove all redundant linearly dependent constraints.
- Always assume $n > m$, i.e. more variables than constraints (why?).

$$\begin{cases} n = 1 \\ m = 2 \end{cases} \qquad \begin{matrix} x = 3 \\ 2x = 7 \end{matrix} \rightarrow \text{infeasible}$$

Standard Form LP

$$
\begin{aligned}
\min \quad & c^\intercal x \quad \text{[Minimization]} \\
\text{s.t.} \quad & Ax = b \quad \text{[Only equality constraints]} \\
& x \geq 0 \quad \text{[All variables nonnegative]}
\end{aligned}
$$

$\max \boldsymbol{c}^\top \boldsymbol{x}$	$\Leftrightarrow$	$-\min(-\boldsymbol{c}^\top \boldsymbol{x})$
$\boldsymbol{a}_i^\top \boldsymbol{x} \geq b_i$	$\Leftrightarrow$	$\boldsymbol{a}_i^\top \boldsymbol{x} - s_i = b_i, s_i \geq 0$
$\boldsymbol{a}_i^\top \boldsymbol{x} \leq b_i$	$\Leftrightarrow$	$\boldsymbol{a}_i^\top \boldsymbol{x} + s_i = b_i, s_i \geq 0$
$x_j \leq 0$	$\Leftrightarrow$	$-x_j \geq 0$
x_j free	$\Leftrightarrow$	$x_j = x_j^+ - x_j^-, x_j^+ \geq 0, x_j^- \geq 0$

- $a_i^T x \geq b_i$

  $\hookrightarrow a_i^T x = b_i + s_i, \quad s_i \geq 0$

slack variables

- $a_i^T x \leq b_i$

  $\hookrightarrow a_i^T x + s_i = b_i, \quad s_i \geq 0$

- x_j free

  $\hookrightarrow x_j = x_j^+ - x_j^-, \quad x_j^+ \geq 0, \quad x_j^- \geq 0$

$x_j > 0 \rightarrow \begin{cases} x_j^+ \text{ takes value} \\ x_j^- = 0 \end{cases}$

$x_j < 0 \rightarrow \begin{cases} x_j^- \text{ takes value} \\ x_j^+ = 0 \end{cases}$

$x_j = 0 \rightarrow x_j^+ = x_j^- = 0$

Example I

$$
\begin{array}{lrrl}
\text{maximize} & x_1 & +2x_2 & \\
\text{subject to} & x_1 & & \leq 100 \\
& & 2x_2 & \leq 200 \\
& x_1 & +x_2 & \leq 150 \\
& x_1, & x_2 & \geq 0
\end{array}
$$

Standard form

$$
\begin{array}{lrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +s_1 & & & = 100 \\
& & 2x_2 & & +s_2 & & = 200 \\
& x_1 & +x_2 & & & +s_3 & = 150 \\
& x_1, & x_2, & s_1, & s_2, & s_3 & \geq 0
\end{array}
$$

$$
\begin{aligned}
\text{maximize} \quad & x_1 + 4x_2 + x_3 \\
\text{s.t.} \quad & 2x_1 + 2x_2 + x_3 \leq 4, \\
& x_1 - x_3 \geq 1, \\
& x_1,\ x_2 \geq 0
\end{aligned}
$$

Standard form

x_3^+ $\quad$ x_3^-

s_1

$$
\begin{aligned}
\text{minimize} \quad & -x_1 - 4x_2 - x_4 + x_5 \\
\text{s.t.} \quad & 2x_1 + 2x_2 + x_4 - x_5 + x_6 = 4, \\
& x_1 - x_4 + x_5 - x_7 = 1, \\
& x_1,\ x_2,\ x_4,\ x_5,\ x_6,\ x_7 \geq 0
\end{aligned}
$$

s_2

$$
\begin{aligned}
\text{maximize} \quad & x_1 + 4x_2 + x_3 \\
\text{s.t.} \quad & 2x_1 + 2x_2 + x_3 \le 4, \\
& x_1 - x_3 \ge 1, \\
& \underline{x_1,\ x_2 \ge 0}\ ,\ x_3 \text{ free}
\end{aligned}
$$

$$x_3 = x_3^+ - x_3^-$$

$$
\begin{aligned}
\text{min.} \quad & -x_1 - 4x_2 \overbrace{- x_3^+ + x_3^-}^{-x_3} \\
\text{s.t.} \quad & 2x_1 + 2x_2 + x_3^+ - x_3^- + s_1 = 4 \\
& x_1 - x_3^+ + x_3^- - s_2 = 1 \\
& x_1,\ x_2,\ x_3^+,\ x_3^-,\ s_1,\ s_2 \ge 0
\end{aligned}
$$

Standard Form LP

- Standard form is mainly used for analysis purposes. We don't need to write a problem in standard form unless necessary. Usually just write in a way that is easy to understand.
- However, being able to transform an LP into the standard form is an important skill. It is helpful for analyzing LP problems as well as using some software to solve it.

Outline

Starting Point: Graphical Solutions to LP

It is very helpful to study a small LP from a graphical point of view.

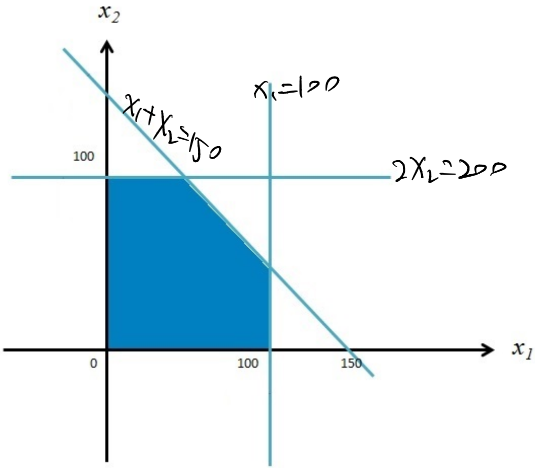
Recall the production problem:

$$\begin{array}{llrl} \text{maximize} & x_1 & +2x_2 & \\ \text{subject to} & x_1 & & \leq 100 \\ & & 2x_2 & \leq 200 \\ & x_1 & +x_2 & \leq 150 \\ & x_1, & x_2 & \geq 0 \end{array}$$

How can we solve this from a graph?

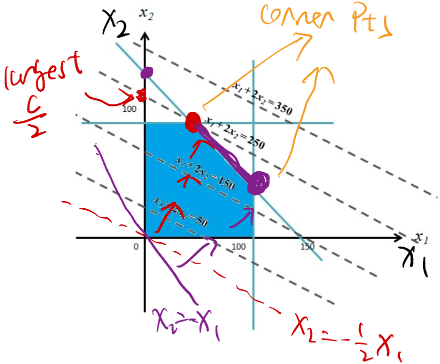
Solve LP from Graph

We first draw the feasible region.



To Maximize $x_1 + 2x_2$...

Then we draw the function $x_1 + 2x_2 = c$ for different values of c.



$x_1 + 2x_2 = c$

$\Rightarrow x_2 = -\frac{1}{2}x_1 + \frac{1}{2}c$

- The optimal solution is the highest one among these lines that touch the feasible region
- The coordinates: $(50, 100)$. Objective value: 250
- What if the objective changes to $\max x_1 + x_2$?

$x_1 + x_2 = c$

$x_2 = -x_1 + c$

Some Observations

active: take "=" sign

- The feasible region of LP is a polygon (polyhedron).
- The optimal solution tends to be at a corner of the feasible region
- Some constraints are *active* at the optimal solution ($x_2 \leq 100$, $x_1 + x_2 \leq 150$), some are not ($x_1 < 100$).

Next we will formalize these observations and study the following questions:

- How to characterize corner points?
- How to find all corner points?
- How to identify the corner point corresponding to the optimal solution?

Outline

Geometric Objects and Algebraic Forms

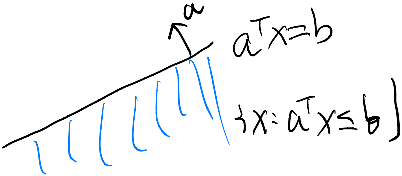
- Point: $x \in \mathbb{R}^n$, e.g., $a = (1, 1)$, $b = (3, 2)$

- Direction: $d \in \mathbb{R}^n$ e.g., $d = b - a = (2, 1)$
- Ray: $x = a + \theta d$ for $\theta \geq 0$

- Line (parametric expression): $x = a + \theta d$ for $\theta \in \mathbb{R}$
- Line (linear equation): $a_1x_1 + a_2x_2 + ... + a_nx_n = b$ or $a^\intercal x = b$

affine

- Hyperplane: $a_1x_1 + a_2x_2 + ... + a_nx_n = b$ or $a^\intercal x = b$
- Normal direction: for hyperplane $a^\intercal x = b$, the normal direction is a
- Halfspace: $\{x \in \mathbb{R}^n | a^\intercal x \leq b\}$

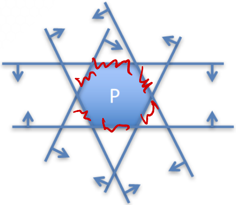


Halfspace Representation of Polyhedron

- The intersection of many halfspaces is called a **polyhedron**.
- Matrix form of polyhedron: $P = \{x \in \mathbb{R}^n : Ax \leq b\}$
- Halfspace representation of polyhedron:

$$
\begin{aligned}
P = \{x \in \mathbb{R}^n : a_i^\intercal x &= b_i \quad i \in M_1 \\
a_i^\intercal x &\leq b_i \quad i \in M_2 \\
a_i^\intercal x &\geq b_i \quad i \in M_3\}
\end{aligned}
$$

- Polyhedron is exactly the feasible region of an LP.



Example - Standard form LP

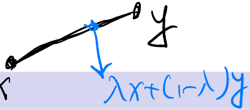
$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Is the feasible region of the standard form LP a polyhedron?

YES

$$\{Ax \geq b,\ Ax \leq b,\ Ix \geq 0\}$$

Outline

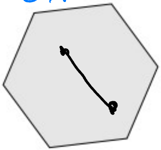


Definition

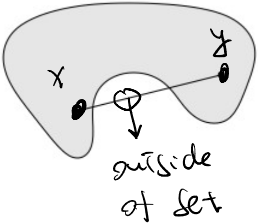
A set $\mathcal{X}$ is **convex** if for any $x, y \in \mathcal{X}$, and any $\lambda \in [0, 1]$, $\lambda x + (1 - \lambda)y \in \mathcal{X}$.

- Convex set contains line segment between any two points in the set
- Example:

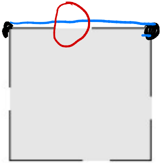
CONVEX

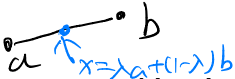


NOT CONVEX

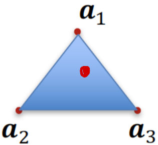


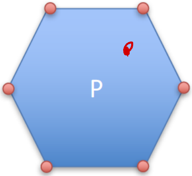
NOT CONVEX





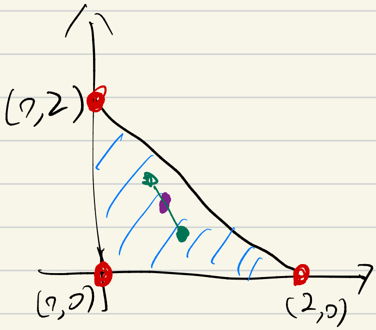
- Given two points $a, b \in \mathbb{R}^n$, a **convex combination** of them is $x = \lambda a + (1-\lambda)b$ for some $\lambda \in [0,1]$. Geometrically, the convex combination of two points is a point on the line segment between the two points.
- For any $\overbrace{x_1, \ldots, x_n}^{\text{vectors}}$ and $\overbrace{\lambda_1, \ldots, \lambda_n}^{\text{scalars}} \geq 0$ satisfying $\lambda_1 + \cdots + \lambda_n = 1$, we call $\sum_{i=1}^n \lambda_i x_i$ a **convex combination** of $x_1, \ldots, x_n$.



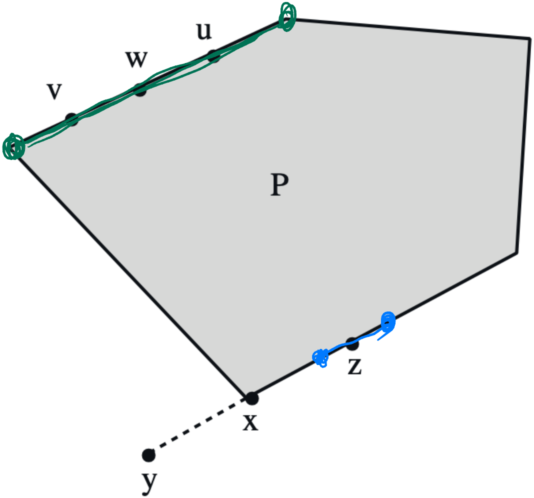


Extreme Point

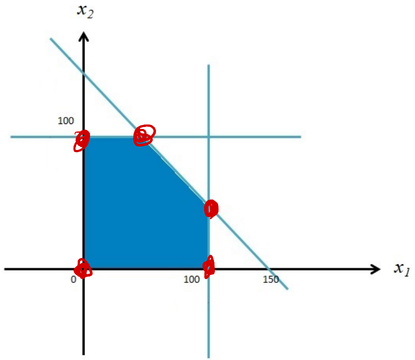
- A point x in the polyhedron P is an **extreme point** of P if and only if x is not a convex combination of other two different points in P, i.e., there does not exist $y, z \in P$ $(x \neq y, x \neq z)$ and $\lambda \in [0, 1]$ so that $x = \lambda y + (1 - \lambda)z$.
- Example: the polyhedron (triangle) by three halfspaces $x_1 + x_2 \leq 2$, $x_1 \geq 0$, $x_2 \geq 0$ has three extreme (corner) points $a = (2, 0)$, $b = (0, 2)$, $c = (0, 0)$. Any convex combination of these three corner points is a point in the triangle, and vice versa, that is, any point in the triangle can be written as a convex combination of these three extreme points.
- We sometimes call the extreme point the vertex, or corner point of the polyhedron.


(0,2)
(0,0)
(2,0)

Extreme Point Example



Example



How many extreme points are there in this feasible region?

- Answer: 5

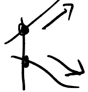
Extreme Point Representation of Polyhedron*

Not Required!

- A **convex hull** of m points a_1, a_2,...,a_m is the set of all convex combinations of these m points, denoted as conv$\{a_1, ..., a_m\}$.
- conv$\{a_1, ..., a_m\} = \{x | \lambda_1 a_1 + ... + \lambda_m a_m, \lambda_1 + ... + \lambda_m = 1, \lambda_i \geq 0, \forall i\}$
- A **Polytope** is a nonempty and bounded polyhedron. A bounded polyhedron is a polyhedron that does not extend to infinity in any direction.
- A polytope is the convex hull of a finite number of extreme points.
- In other words, given a non-empty and bounded polyhedron P, we can always find a finite set of extreme points $x_1, ..., x_m$ such that $P =$ conv $\{x_1, ..., x_m\}$. ex Pts
- In low dimension, you may consider the extreme points are corner points.

Summary of Polyhedron Representations*

- Halfspace representation: A polyhedron is the intersection of a finite number of halfspaces. This representation applies to all polyhedra, bounded or unbounded
- Extreme-point representation: A bounded polyhedron is the convex hull of all its extreme points. Convex hull can only generate a bounded set.
- Question: How to extend extreme-point representation to an unbounded polyhedron? Not covered in this undergraduate class.
    - Extreme ray is an analogy to the definition of an extreme point.
    - Conic hull is an analogy to the definition of convex hull.
    - Weyl-Caratheodory Representation Theorem: Any point x in a polyhedron P can be written as a sum of two vectors, $x = x' + d$, where x' is in the convex hull of its extreme points and d is in the conic hull of its extreme rays.



Outline

Linearly Independence

$x_i \neq c_1 x_1 + \cdots + c_k x_k \quad \forall i$

- Linear independent: A set of vectors $\{x_1, x_2, ... x_k\}$ is said to be **linearly independent** if no vector can be represented as a linear combination of the remaining vectors. Conversely, if one vector belonging to the set can be represented as a linear combination of the remaining vectors, then the vectors are said to be **linearly dependent**.
- Mathematically, a set of vectors $\{x_1, x_2, ..., x_k\}$ is said to be linearly independent if the relation $c_1x_1 + c_2x_2 + ... + c_kx_k = 0$ implies that $c_1 = c_2 = ... = c_k = 0$. Otherwise, $\{x_1, x_2, ..., x_k\}$ is said to be linearly dependent.

Linearly Independent Constraints

$$\begin{cases} x_1 + x_2 \le 5 \\ 2x_1 + 2x_2 \le 4 \end{cases}$$

'un-Parallel'

- A set of constraints are **linearly independent** if the normal directions (vectors) of constraints are linearly independent.
- Example: $x_1 + x_2 \le 2$ and $x_2 \ge 0$ with normal directions $(1, 1)^\intercal$ and $(0, 1)^\intercal$.
- When two or more linearly independent constraints are active/binding at a certain point (take the equal "=" sign), then we call them **linearly independent constraints active/binding at this point**.
- Example: $x_1 + x_2 \le 2$ and $x_2 \ge 0$ are linearly independent constraints active/binding at $(2, 0)$

Outline

Basic solution

at least m constraints $m > n$

Consider a polyhedron defined by linear equalities and linear inequalities, and let x be an element of $\mathbb{R}^n$. The vector x is a **basic solution** if:

- There are n linearly independent constraints active at x;
- All equality constraints are active at x.

Remark: n linearly independent constraints may include some (all) equality constraints.

Quiz: Is a basic solution always feasible?

No!

Basic Feasible Solutions

Definition

If a basic solution x satisfies all constraints, then we call it a **basic feasible solution** (BFS).

To find a BFS

- First find a basic solution x
- Check if x satisfies all constraints

Theorem (B&T Theorem 2.3)

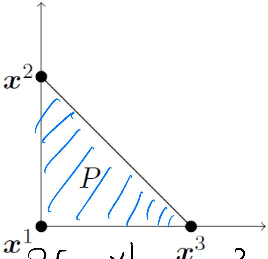
Let P be a nonempty polyhedron and let $x^* \in P$. Then, the following are equivalent:

1. *x^* is an extreme point;*
2. *x^* is a basic feasible solution.*

Exercise

$n=2$

$$\begin{cases} x_1 + x_2 \le 2 \\ x_1 \ge 0 \\ x_2 \ge 0 \end{cases}$$

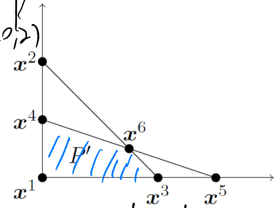


BS: x^1, x^2, x^3

BFS: x^1, x^2, x^3

$$x_1 + x_2 \le 2$$
$$x_1 \ge 0$$
$$x_2 \ge 0$$
$$x_1 + 3x_2 \le 3$$

(0,2)



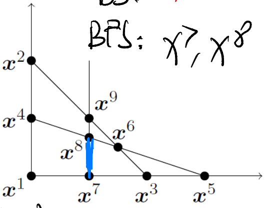
BS: $x^1 - x^6$

BFS: x^1, x^3, x^4, x^6

$$x_1 + x_2 \le 2$$
$$x_1 \ge 0$$
$$x_2 \ge 0$$
$$x_1 + 3x_2 \le 3$$
$$x_1 = 1$$

BS: x^7, x^8, x^9

BFS: x^7, x^8

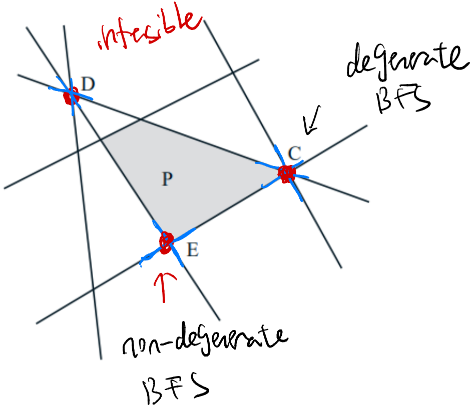


(1,0)

Degeneracy

Degeneracy

A basic feasible solution x^* is called degenerate if there are more than n active constraints at x^*.



Existence of BFS

Definitions: A polyhedron contains a line if $\exists x \in P$ and $d \in \mathbb{R}^n$, such that

$$x + \theta d \in P \quad \forall \theta$$

Theorem (B&T Theorem 2.6)

A polyhedron P has at least one extreme point (BFS) if and only if P does not contain a line.

$\Downarrow$

Corollary (B&T Corollary 2.2)

- Nonempty polyhedron in standard form $P = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$ always has a BFS.

  vecter

- Bounded polyhedron always has a BFS.

Optimality of BFS

Consider an LP over a polyhedron P

$$\begin{aligned} \min \quad & c^{\mathsf{T}} x \\ \text{s.t.} \quad & x \in P = \{x | Ax \geq b\} \end{aligned}$$

Theorem (B&T Theorem 2.8)

If P is nonempty and has at least one extreme point (BFS), then the LP is either unbounded or there exists an extreme point (BFS) which is optimal.

- In order to find an optimal solution, we only need to look among basic feasible solutions.
- We need to find a algebraic way to manually find BFS of a polyhedron - try standard form.

Outline

$x+y=2$
$2x+2y=4$

I need $\underline{1}$ LI equations to determine 1 unknown.

$\underline{2}$ LI equations to determine 2 unknowns.

$\vdots$

$\underline{n}$ LI equations to determine n unknowns

Key Idea

- An extreme point (BFS) is a corner point where multiple constraints intersect, i.e. multiple constraints are active at an extreme point.
- Considering an LP with n variables, an ~~extreme point~~ BS is the intersection of n linearly independent active constraints. Since n linearly independent linear equations have a unique solution, an extreme point is the unique solution of the set of n constraints when all of them take equal signs.
- In other words, an extreme point (BFS) $x \in \mathbb{R}^n$ is the solution of a set of n non-parallel linear equations, which come from n linearly independent constraints active at x.

Standard Form LP

In the following, we consider LP in its standard form:

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

Constraints

- $x \in \mathbb{R}^n$, i.e. there are n variables
- $A \in \mathbb{R}^{m \times n}$, i.e. there are m equality constraints
- Assumption I: all the m equality constraints are linearly independent (or equivalently A has linearly independent rows or A has full row rank m). If not, we can remove all redundant linearly dependent constraints.
- Assumption II: $n > m$, i.e. more variables than constraints.

$$A = \begin{bmatrix} — a_1^T — \\ — a_2^T — \\ \vdots \\ — a_m^T — \end{bmatrix}$$

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \rightarrow m \text{ equations} \\ & x \geq 0 \rightarrow \text{need } n-m \text{ equations} \end{aligned}$$

need n equations in total,
have m equations from $Ax=b$,
Still $n-m$ equations from $x \geq 0$.

Basic Solution of Standard Form LP

- A basic solution is the unique solution to n linearly independent constraints taking equal signs.
- For a standard form LP, we already have m linearly independent constraints.
- Need $n - m$ additional linearly independent constraints. Where to find them? From nonnegative constraints: $x \geq 0$. But which x_i to choose?

Basic Variables and Non-basic Variables

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

- $Ax = b$ gives m ~~constraints~~ equations.
- Since $m < n$, need additional $n - m$ constraints from $x \geq 0$.
- Select $n - m$ variables from x and put in a vector x_N.
- $x_N \in \mathbb{R}^{n-m}$ are **non-basic variables**.
- Let $N = \{N(1), \dots, N(n-m)\}$ denote the indices in x_N called **non-basic indices**.
- Put the remaining m variables in a vector x_B.
- $x_B \in \mathbb{R}^m$ are **basic variables**.
- Let $B = \{B(1), \dots, B(m)\}$ denote the indices in x_B called **basic indices** or **basis**.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \qquad n = 6 \qquad m = 4$$

$$X_N = \begin{bmatrix} x_2 \\ x_6 \end{bmatrix} \qquad X_B = \begin{bmatrix} x_1 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

Select m basic variables

$\Updownarrow$

Select $n-m$ nonbasic variables

Two Key Findings

- For a linear system of equations $Ax = b$, i-th column of A corresponds to specific variable x_i.
- We can switch the columns of A as long as they correspond to the order sequence of the variables in x.
- Example:

$$\begin{aligned} 2x_1 + 3x_2 + x_3 + 4x_4 + x_5 &= 12, \\ x_1 + 5x_2 + 2x_3 + 3x_4 + 4x_5 &= 18, \\ 4x_1 + x_2 + 3x_3 + 2x_4 + 5x_5 &= 20 \end{aligned}$$

- What if x_2 and x_4 are zeros?

$$X_B = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix} \qquad X_N = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}$$

$$
\begin{aligned}
2x_1 + 3x_2 + x_3 + 4x_4 + x_5 &= 12,\\
1x_1 + 5x_2 + 2x_3 + 3x_4 + 4x_5 &= 18,\\
4x_1 + x_2 + 3x_3 + 2x_4 + 5x_5 &= 20
\end{aligned}
\quad \Rightarrow Ax=b
$$

$$
\underbrace{\begin{array}{c}
\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array}\\
\begin{bmatrix} 2 & 3 & 1 & 4 & 1 \\ 1 & 5 & 2 & 3 & 4 \\ 4 & 1 & 3 & 2 & 5 \end{bmatrix}
\end{array}}_{A}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 12 \\ 18 \\ 20 \end{bmatrix}
$$

row $\leftrightarrow$ constraint

column $\leftrightarrow$ variable

$$
\begin{array}{c}
\begin{array}{ccccc} x_4 & x_5 & x_2 & x_3 & x_1 \end{array}\\
\begin{bmatrix} 4 & 1 & 3 & 1 & 2 \\ 3 & 4 & 5 & 2 & 1 \\ 2 & 5 & 1 & 3 & 4 \end{bmatrix}
\end{array}
\begin{bmatrix} x_4 \\ x_5 \\ x_2 \\ x_3 \\ x_1 \end{bmatrix} = \begin{bmatrix} 12 \\ 18 \\ 20 \end{bmatrix}
$$

$$
\begin{cases}
4x_4 + x_5 + 3x_2 + x_3 + 2x_1 = 12\\
3x_4 + 4x_5 + 5x_2 + 2x_3 + x_1 = 18\\
2x_4 + 5x_5 + x_2 + 3x_3 + 4x_1 = 20
\end{cases}
$$

$\rightarrow x_2 = x_4 = 0$

$$
\begin{aligned}
2x_1 + \cancel{3x_2} + x_3 + \cancel{4x_4} + x_5 &= 12, \\
x_1 + \cancel{5x_2} + 2x_3 + \cancel{3x_4} + 4x_5 &= 18, \\
4x_1 + \cancel{x_2} + 3x_3 + \cancel{2x_4} + 5x_5 &= 20
\end{aligned}
$$

$$
\begin{cases}
2x_1 + x_3 + x_5 = 12 \\
x_1 + 2x_3 + 4x_5 = 18 \\
4x_1 + 3x_3 + 5x_5 = 20
\end{cases}
$$

$$A_B x_B = b$$

$$\underbrace{\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 4 \\ 4 & 3 & 5 \end{bmatrix}}_{A_B} \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} 12 \\ 18 \\ 20 \end{bmatrix}$$

$$x_B = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix} = A_B^{-1} b = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 4 \\ 4 & 3 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 12 \\ 18 \\ 20 \end{bmatrix}$$

Basis Matrix

m $n-m$

After rearranging the variables, we have

- $x = [x_B, x_N]$ where $x_B \in \mathbb{R}^m$ are **basic variables** and $x_N \in \mathbb{R}^{n-m}$ are **non-basic variables**.
- Select the columns of A corresponding to basic variables and put them in a matrix A_B, called **basis matrix**.
- $$A_B = \begin{bmatrix} | & | & & | \\ A_{B(1)} & A_{B(2)} & \cdots & A_{B(m)} \\ | & | & & | \end{bmatrix}$$
- The remaining columns of A corresponding to non-basic vairables are **the non-basic columns**. Put them in a matrix A_N, called **non-basis matirx**.
- After rearranging columns of A, we have $A = [A_B \quad A_N]$.

$$Ax = b \rightarrow m \text{ equations}$$

$$\begin{bmatrix} A_B & A_N \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b$$

$$\begin{cases} A_B x_B + A_N x_N = b & m \text{ equations} \\ x_N = 0 \rightarrow & n-m \text{ equations} \end{cases}$$

Logic to Find a Basic Solution

- We can write the n equations as

$$\begin{bmatrix} A_B & A_N \\ 0 & I \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

$A_B x_B + A_N x_N = b$
$x_N = 0$

- Since A_B is an invertible matrix, and I is the identity matrix, the whole matrix is invertible; therefore, the n equations are linearly independent.
- Thus, there is only one solution, which is a basic solution.
- The solution can be computed:

$$A_B x_B = b \Rightarrow x_B = A_B^{-1} b$$

$$x_N = 0$$

Finding a Basic Solution in Standard Form LP

Procedures to find a basic solution:

1. Choose any m linearly independent columns of A: $A_{B(1)}, ..., A_{B(m)}$ and form the basis matrix $A_B = [A_{B(1)}, ..., A_{B(m)}]$. Denote the rest of A as matrix A_N.
2. Let $x_i = 0$ for all $i \neq B(1), ..., B(m)$.
3. Solve the equation $Ax = b$ for the remaining $x_{B(1)}, ..., x_{B(m)}$.

- The basic solution is $x = [x_B, x_N]$, where the basic variables are $x_B = A_B^{-1} b$ and the nonbasic variables are $x_N = 0$.
- Since $A_{B(1)}, ..., A_{B(m)}$ are linearly independent, the last step must produce a unique solution.
- Basic solution of an LP only depends on its constraints, it has nothing to do with the objective function.

Definitions for Standard Form LP

Now we study the definitions for the standard form LP.

Definition

We call $\boldsymbol{x}$ a **basic solution** of the LP if and only if

1. $A\boldsymbol{x} = \boldsymbol{b}$
2. There exist indices $B(1), ..., B(m)$ such that the columns $A_{B(1)}, ..., A_{B(m)}$ are linearly independent, and $x_i = 0$ for $i \neq B(1), ..., B(m)$

Definition

If a basic solution $\boldsymbol{x}$ also satisfies that $\boldsymbol{x} \geq 0$, then we call it a **basic feasible solution** (BFS).

To find a BFS

- First find a basic solution $\boldsymbol{x}$
- Check if $\boldsymbol{x} \geq 0$

Theorems for Standard Form LP

Theorem (LP fundamental theorem)

Given a linear optimization problem in standard form where A has full row rank m

1. *If there is a feasible solution, there is a basic feasible solution;*
2. *If there is an optimal solution, there is an optimal solution that is a basic feasible solution.*

Corollary

If an LP with m constraints (in the standard form) has an optimal solution, then there must be an optimal solution such that there is no more than m positive entries.

Example

$$
\begin{array}{lrrl}
\text{maximize} & x_1 & +2x_2 & \\
\text{subject to} & x_1 & & \leq 100 \\
 & & 2x_2 & \leq 200 \\
 & x_1 & +x_2 & \leq 150 \\
 & x_1, & x_2 & \geq 0
\end{array}
$$

The standard form:

$$
\begin{array}{lrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +s_1 & & & = 100 \\
 & & 2x_2 & & +s_2 & & = 200 \\
 & x_1 & +x_2 & & & +s_3 & = 150 \\
 & x_1, & x_2, & s_1, & s_2, & s_3 & \geq 0
\end{array}
$$

Example

We can write the feasible set by $\{x : Ax = b, x \geq 0\}$. where

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

Choose three independent columns of A, e.g., the first three, we get the corresponding basic solution is

$$x_B = \begin{bmatrix} x_1 \\ x_2 \\ s_1 \end{bmatrix} = A_B^{-1} b = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix} = \begin{bmatrix} 50 \\ 100 \\ 50 \end{bmatrix}$$

That is $x_1 = 50, x_2 = 100, s_1 = 50$. Since all basic variables are nonnegative, therefore $(50, 100, 50, 0, 0)$ is a basic feasible solution. One can find other basic feasible solutions by choosing other sets of columns.

Example

Select columns 2, 3, and 4 of A as the basis matrix:

$$A_B = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_N = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$x_B = \begin{bmatrix} x_2 \\ x_3 \\ s_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix} = \begin{bmatrix} 150 \\ 100 \\ -100 \end{bmatrix}$$

$$\mathbf{x}_N = \begin{bmatrix} x_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Since $x_4 = -50 < 0$, the basis solution is not feasible. This is a basic solution but not a basic feasible solution (BFS).

Example

Select columns 3, 4, and 5 of A as the basis matrix:

$$A_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_N = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}$$

$$x_B = \begin{bmatrix} x_3 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \\ 150 \end{bmatrix}$$

$$\mathbf{x}_N = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Since $x_B \geq 0$, the basic solution $(0, 0, 100, 200, 150)$ is a basic feasible solution (BFS).

Example Continued

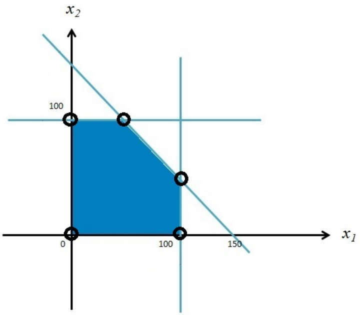
We can list the basic (feasible) solutions

Indices	$\{1, 2, 3\}$	$\{1, 2, 4\}$	$\{1, 2, 5\}$	$\{1, 3, 4\}$
Solution	(50, 100, 50, 0, 0)	(100, 50, 0, 100, 0)	(100, 100, 0, 0, -50)	(150, 0, -50, 200, 0)
Status	BFS	BFS	Basic but not feasible	Basic but not feasible
Indices	$\{1, 4, 5\}$	$\{2, 3, 4\}$	$\{2, 3, 5\}$	$\{3, 4, 5\}$
Solution	(100, 0, 0, 200, 50)	(0, 150, 100, -100, 0)	(0, 100, 100, 0, 50)	(0, 0, 100, 200, 150)
Status	BFS	Basic but not feasible	BFS	BFS

The other two choices $\{1, 3, 5\}$ and $\{2, 4, 5\}$ lead to dependent basic columns (therefore no basic solutions can be obtained)

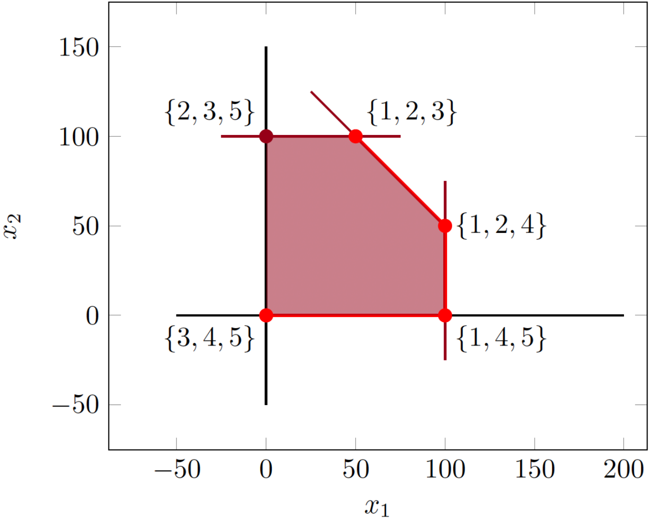
Verify

They indeed correspond to all the corners of the feasible sets.



Verify

Each BFS correspond to a selection of columns (basis matrix).



Quiz

How many non-zeros could one have in a basic solution (assuming there are m constraints)?

- No more than m
- Could be anything between 0 to m, but typically it is m

How many basic solutions can one have for a linear program with m constraints and n variables?

- At most $C(n, m) = \frac{n!}{m!(n-m)!}$ (Combination number)
- Therefore for a finite number of linear constraints, there can only be a finite number of basic solutions

Search Among BFS

Now we know that we only need to search among basic feasible solutions for the optimal solution.

How to search among the basic feasible solutions?

- One may suggest to list all the basic feasible solutions and compare their objective values. However, there are too many of them.
- For a linear optimization with m constraints and n variables, how many basic feasible solutions it may have?
- $C(n, m)$.. If $n = 1000$, $m = 100$, then the value is 10^{143}..

Simplex Method

Therefore we need a smarter way to find the optimal solution.

- Simplex method

The simplex method proceeds from one BFS (a corner point of the feasible region) to a neighboring one, in such a way as to continuously improve the value of the objective function until reaching optimality.

- We need to define what it means by *adjacent* or *neighboring* solution
- We need to design an efficient way to find (and move to) the neighboring BFS (e.g., we should try to avoid taking matrix inversions every time)
- We need to design a valid stopping criterion

Outline

Basic Structure of an Optimization Algorithm

At each iteration k,

1. Start from a feasible solution $\mathbf{x}^k$.
2. Find a "good" direction $\mathbf{d}$ that (a) points inside the feasible region and (b) decreases the objective value.
3. Find a "good" step length θ along $\mathbf{d}$ to move to next iteration point: $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \theta \mathbf{d}$.
4. If no good direction or step length can be found, terminate. Otherwise $k \leftarrow k+1$ and go back to step 1.

Simplex Method Framework

At each iteration k,

1. Start from a *basic feasible solution* $\mathbf{x}^k$.
2. Find a direction $\mathbf{d}$ that (a) points to an *adjacent BFS* and (b) decreases the objective value.
3. Find a step length θ so that the next iteration point, $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \theta\mathbf{d}$, is a better adjacent BFS.
4. If no such direction or step length can be found, terminate. Otherwise $k \leftarrow k + 1$ and go back to step 1.

Simplex Method

The simplex method proceeds from one BFS (a corner point of the feasible region) to a neighboring one, in such a way as to continuously improve the value of the objective function until reaching optimality.

- We need to define what it means by *adjacent* or *neighboring* solution
- We need to design an efficient way to find (and move to) the neighboring BFS (e.g., we should try to avoid taking matrix inversions every time)
- We need to design a valid stopping criterion

Outline

Simplex Method Starting Point

First, we assume that we have somehow found a BFS whose basis is

$$B = \{B(1), \ldots, B(m)\}.$$

Define:

$$A_B = \begin{bmatrix} | & | & & | \\ A_{B(1)} & A_{B(2)} & \cdots & A_{B(m)} \\ | & | & & | \end{bmatrix}$$

and let A_N be the matrix consisting of the non-basic columns of A.

In the sequel, N will denote the non-basic index set.

Rearranging the variables, we can write

$$A = [A_B, \, A_N], \qquad x = \begin{bmatrix} x_B \\ x_N \end{bmatrix},$$

where x_B are the basic variables and x_N are the non-basic variables.

By definition, we have:

$$x_B = A_B^{-1} b, \qquad x_N = 0.$$

Current BFS and Update Step

Let the current basic feasible solution be:

$$\mathbf{x} = \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \\ x_1 \\ x_4 \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \end{bmatrix}, \quad \mathbf{x}_N = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix}$$

Next iterate:

$$\hat{\mathbf{x}} = \mathbf{x} + \theta \mathbf{d} = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \\ x_1 \\ x_4 \end{bmatrix} + \theta \begin{bmatrix} d_2 \\ d_3 \\ d_5 \\ d_1 \\ d_4 \end{bmatrix}$$

Direction Vector Partition

Direction vector $\mathbf{d}$ is partitioned as:

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_B \\ \mathbf{d}_N \end{bmatrix} = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \\ d_1 \\ d_4 \end{bmatrix}, \quad \mathbf{d}_B = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \end{bmatrix}, \quad \mathbf{d}_N = \begin{bmatrix} d_1 \\ d_4 \end{bmatrix}$$

Here, $\mathbf{d}_B$ corresponds to the basic variables (x_2, x_3, x_5), and $\mathbf{d}_N$ corresponds to the nonbasic variables (x_1, x_4).

Feasible Direction - Maintain Feasibility

Starting from a basic feasible solution x, the simplex method considers a feasible direction d to move away from the BFS x to $\hat{x} := x + \theta d$. The new point $x + \theta d$ needs to be (a) a feasible point and (b) an adjacent BFS. For (a), we need

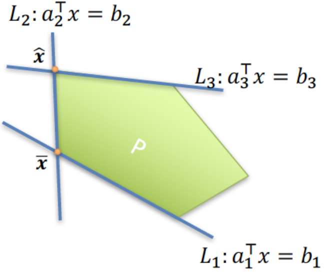
$$\begin{aligned} & A(x+\theta d) = b \\ \Rightarrow \quad & Ax + \theta Ad = b \\ \Rightarrow \quad & Ad = 0 \end{aligned}$$

$$Ad = [A_B \;\; A_N] \begin{bmatrix} d_B \\ d_N \end{bmatrix} = A_B d_B + A_N d_N = 0$$

Outline

Neighboring/Adjacent BFS in Standard Form LP

- Definition: Two basic feasible solutions x and $\hat{x}$ of a polyhedron P are called adjacent if they share the same $n-1$ linearly independent active constraints.



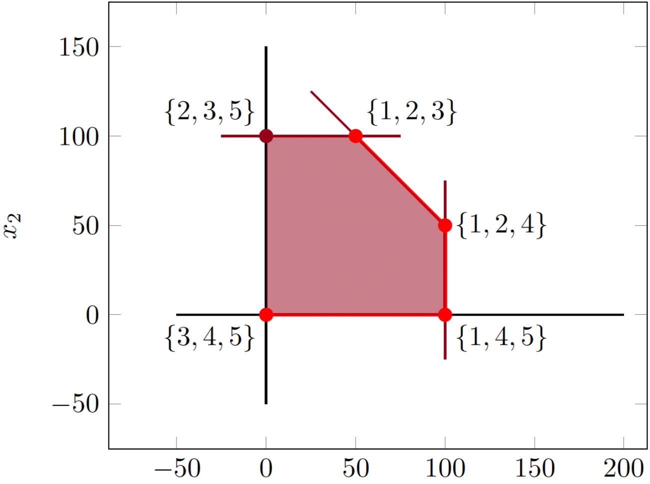
Feasible region P:

$$
\begin{aligned}
a_1^\top x &\le b_1\\
a_2^\top x &\le b_2\\
a_3^\top x &\le b_3\\
a_4^\top x &\le b_4\\
a_5^\top x &\le b_5
\end{aligned}
$$

At $\overline{\boldsymbol{x}}$, L_1 and L_2 are L.I.A.C.
At $\widehat{\boldsymbol{x}}$, L_2 and L_3 are L.I.A.C.
$\overline{\boldsymbol{x}}$ and $\widehat{\boldsymbol{x}}$ are adjacent BFS's
and they share 1 L.I.A.C. (L_2)

Adjacent BFS in Standard Form LP

Two basic solutions are neighboring / adjacent if they differ by exactly one basic (or non-basic) index.



Adjacent BFS in Standard Form LP

- Standard form LP

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

- Two adjacent BFS differ the basis matrix A_B in exactly one column.
- Two adjacent BFS differ by exactly one basic (or non-basic) variable.
- In a standard form LP, two BFS x and $\hat{x}$ are adjacent if they have the same $n - m - 1$ nonbasic variables, and differ in one nonbasic variable.
- Because $n - m - 1$ nonbasic variables of x need to remain nonbasic, i.e. at zero value, d_N must have n-m-1 components at zero value; and because one nonbasic variable of x needs to become basic, i.e. to increase from zero value to some positive value, then the corresponding component of d_N has to be a positive number.

Non-basic Direction

- To find a neighbor, we want to select a nonbasic variable x_j, $j \in I_N$ (remember initially $x_j = 0$) to enter the basis: increase x_j to a positive number while keeping other nonbasic variables at zero.
- We consider moving x (the current BFS) to a neighboring one y by $y = x + \theta d$.
- For nonbasic variables, we need $d_j = 1$ for some $j \in N$ and $d_{j'} = 0$ for all other non-basic indices $j' \neq j$.
- $d_N = e_j^\intercal = [0, ..., 0, 1, 0, ...0]^\intercal$ for some $d_j = 1, j \in N$.

Basic Direction

To maintain feasible, we need

$$
\begin{aligned}
& A_B d_B + A_N d_N = 0 \\
\Rightarrow \quad & A_B d_B + A_j = 0 \\
\Rightarrow \quad & d_B = -A_B^{-1} A_j
\end{aligned}
$$

Put together, we have $d = \begin{bmatrix} -A_B^{-1} A_j \\ e_j \end{bmatrix} = [-A_B^{-1} A_j; 0; \ldots; 1; 0; \ldots; 0]$. The direction d is uniquely determined once j is chosen. We refer to this direction as the **j-th basic direction**.

Example

$$\begin{array}{rrrrrrl} \text{minimize} & -x_1 & -2x_2 & & & & \\ \text{subject to} & x_1 & & +x_3 & & & = 100 \\ & & 2x_2 & & +x_4 & & = 200 \\ & x_1 & +x_2 & & & +x_5 & = 150 \\ & x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0 \end{array}$$

Current basis: $\{x_2, x_3, x_5\}$, corresponding to columns 2, 3, and 5 of A.

$$A_B = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

The current basic feasible solution is

$$\mathbf{x}_B = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \\ 50 \end{bmatrix}, \quad \mathbf{x}_N = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Example

- The nonbasic variables $x_1 = x_4 = 0$, then its adjacent BFS must share one of these two nonbasic variables, i.e., $x_1 = x_2 = 0$ may be nonbasic variables in an adjacent BFS. Let's select nonbasic variable x_4 to enter the basis.
- This means d_N contains 1 zero and 1 one component:

$$d_N = \begin{bmatrix} d_1 \\ d_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Then $\mathbf{x} + \theta \mathbf{d}$ will make x_4 positive, i.e., increasing from zero.
- Compute the direction for basic variables:

$$d_B = \begin{bmatrix} d_2 \\ d_3 \\ d_5 \end{bmatrix} = -A_B^{-1} A_4 = -\begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

Outline

Objective Function Partition

The linear objective is:

$$\mathbf{c}^T\mathbf{x} = \begin{bmatrix} c_2 & c_3 & c_5 & c_1 & c_4 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_5 \\ x_1 \\ x_4 \end{bmatrix}$$

We partition $\mathbf{c}$ and $\mathbf{x}$ into basic and nonbasic parts:

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix} = \begin{bmatrix} c_2 \\ c_3 \\ c_5 \\ c_1 \\ c_4 \end{bmatrix}, \quad \mathbf{c}_B = \begin{bmatrix} c_2 \\ c_3 \\ c_5 \end{bmatrix}, \quad \mathbf{c}_N = \begin{bmatrix} c_1 \\ c_4 \end{bmatrix}$$

So the objective becomes:

$$\mathbf{c}^T\mathbf{x} = \begin{bmatrix} \mathbf{c}_B & \mathbf{c}_N \end{bmatrix} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N$$

Choosing a Direction and Pivoting

- At the current BFS, each nonbasic variable x_j provides a direction:

$$d_N = e_j \quad \text{and} \quad d_B = -A_B^{-1} A_j,$$

  pointing to an adjacent BFS.
- Which direction should the algorithm pick?
- The algorithm should pick a direction to reduce objective cost.
- How does the objective value change along a direction?
  - $c^\top(x + \theta d) - c^\top x = \theta c^\top d$ is the change of objective value
  - $c^\top d$ is the change of objective value for a unit stepsize
  - $\theta c^\top d$ is the total change of objective value after moving θd
- The algorithm should pick a d such that $c^\top d < 0$
- Selecting a $j \in N$ such that $c^\top d < 0$ is called **pivoting**: make x_j enter the basis

Cost Change

$$
\begin{aligned}
c^\intercal(x+\theta d) - c^\intercal x &= \theta c^\top d \\
&= \theta \begin{bmatrix} c_B^\intercal & c_N^\intercal \end{bmatrix} \begin{bmatrix} d_B \\ d_N \end{bmatrix} \\
&= \theta(c_B^\intercal d_B + c_N^\intercal d_N) \\
&= \theta(\underbrace{-c_B^\intercal A_B^{-1} A_j + c_j}_{\text{reduced cost}})
\end{aligned}
$$

Reduced Cost

For each $j \in N$, we define the **reduced cost** $\bar{c}_j$ of the variable x_j to be $\bar{c}_j = c_j - c_B^\intercal A_B^{-1} A_j$.

- We want to select a nonbasic variable x_j such that the reduced cost $\bar{c}_j < 0$, which means the objective value will decrease.

Quiz

What is the reduced cost of a basic variable?

Optimality Conditions in Simplex

Optimality Conditions in Simplex

Consider a basic feasible solution x associated with basis B, and let $\bar{c}$ be the corresponding vector of reduced costs.

- If $\bar{c}_j \geq 0$ for all $j \in N$, then x is optimal.
- If x is optimal and nondegenerate, then $\bar{c}_j \geq 0$ for all $j \in I_N$

- Thus, we want to pick $j \in N$ such that the reduced cost $\bar{c}_j < 0$.
- This theorem gives a stopping criterion to the simplex algorithm: We stop when all the reduced costs are non-negative.
- It also means that if one could not find a neighbor solution that is better, then one must have already achieved optimal solution.

Proof of First Part

Let x be a BFS with $\bar{c} \geq 0$. Let z with $Az = b$ and $z \geq 0$ be an arbitrary feasible point.
Define $u = z - x$. We have $Au = 0$. That is

$$0 = [A_B \;\; A_N] \begin{bmatrix} u_B \\ u_N \end{bmatrix} = A_B u_B + \sum_{j \in N} A_j u_j.$$

This gives

$$u_B = -\sum_{j \in N} A_B^{-1} A_j u_j.$$

Let us now compute

$$c^T u = [c_B^T \;\; c_N^T] \begin{bmatrix} u_B \\ u_N \end{bmatrix} = c_B^T u_B + \sum_{j \in N} c_j u_j = \sum_{j \in N} (c_j - c_B^T A_B^{-1} A_j) u_j = \sum_{j \in N} \bar{c}_j u_j.$$

Notice that $\bar{c}_j \geq 0$, $\forall j$ and $u_j = z_j - x_j = z_j \geq 0$, $\forall j \in N$. We have

$$c^T u = c^T z - c^T x \geq 0$$

for any arbitrary feasible z.

Example

$$
\begin{array}{rrrrrrl}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +x_3 & & & = 100 \\
& & 2x_2 & & +x_4 & & = 200 \\
& x_1 & +x_2 & & & +x_5 & = 150 \\
& x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0
\end{array}
$$

If we are at basis $\{2, 3, 5\}$, then the reduced costs are:

$$
\bar{c}_1 = -1 - \begin{bmatrix} -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = -1
$$

$$
\bar{c}_4 = 0 - \begin{bmatrix} -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 1
$$

Therefore only including x_1 in the basis in the next iteration will reduce the objective value.