# Module 1 and 2: Introduction and Solving LP

*CO 250: Introduction to Optimization*

*2019 Spring, David Duan*

# 1 Introduction

## 1.1 Optimization Overview

### 1.1.1 Abstract Optimization Problem

Given $A \subseteq \mathbb{R}^n$ and $f : A \to \mathbb{R}$, we want to find $x \in A$ that minimizes or maximizes $f$.

### 1.1.2 Important Special Cases

- **Linear Programming (LP)** $A$ is implicitly given by *linear* constraints and $f$ is *linear*.
- **Integer Programming (IP)** Same as LP, but we want to max/min over *integers* in $A$.
- **Nonlinear Programming (NLP)** Constraints and $f$ are now non-linear.

### 1.1.3 Typical Workflow

1. **Practical Problem** A description in plain English language with supporting data.
2. **Mathematical Model** Capture problem in mathematics using LP, IP, NLP, etc.
3. **Practical Implementations** Feed the model and data into a solver.

### 1.1.4 Terminology

- **Formulation** A mathematical representation of the optimization problem.
  - **Variable** Various parameters we wish to determine.
  - **Objective Function** Represent the quantity that need to be maximized/minimized.
  - **Mathematical Constraint** Each represent a constraint in the problem.
- **Solution** An assignment of values to each of the variables of the formulation.
  - **Feasible Solution** All constraints in the problem are satisfied.
  - **Optimal Solution** A feasible solution that yields the max/min of the objective fn.

### 1.1.5 Concrete Example

Suppose WaterTech manufactures four products, requiring time on two machines and two types (skilled and unskilled) of labor:

| Product | Machine 1 (h) | Machine 2 (h) | Skilled Labour (h) | Unskilled Labour (h) | Unit Sale Price ($) |
|---|---|---|---|---|---|
| 1 | 11 | 4 | 8 | 7 | 300 |
| 2 | 7 | 6 | 5 | 8 | 260 |
| 3 | 6 | 5 | 5 | 7 | 220 |
| 4 | 5 | 4 | 6 | 4 | 180 |

Each month, 700 hours are available on machine 1 and 500 on machine 2. Each month, the company can purchase up to 600 hours of skilled labour at $8 per hour and up to $600 hours of unskilled labour at $6 per hour. We want to decide how much of each product to produce each month and how much labour of each type to purchase in order to maximize profit.

We introduce the following variables. Let $x_i$ denote the number of units of product $i$ to manufacture, $i \in \{1, 2, 3, 4\}$, $y_u$ and $y_u$ to denote the number of purchased hours of skilled and unskilled labour, respectively.

Our objective function is simple -- revenue minus cost:

$$300x_1 + 260x_2 + 220x_3 + 180x_4 - 8y_s - 6y_u$$

Our constraints have three parts. First, machine time:

$$\begin{aligned} 11x_1 + 7x_2 + 6x_3 + 5x_4 &\leq 700 \\ 4x_1 + 6x_2 + 6x_3 + 4x_4 &\leq 500 \end{aligned}$$

Next, labour time:

$$\begin{aligned} 8x_1 + 5x_2 + 5x_3 + 6x_4 &\leq y_s \\ 7x_1 + 8x_2 + 7x_3 + 4x_4 &\leq y_u \end{aligned}$$

Finally, labour cost:

$$y_s \leq 600, \quad y_u \leq 650$$

Also, there is an implicit constraint: all variables must be non-negative.

Combining everything above, we write the following "hello world" program:

$$\begin{array}{lrl} \max & 300x_1 + 260x_2 + 220x_3 + 180x_4 - 8y_s - 6y_u & \\ s.t. & 11x_1 + 7x_2 + 6x_3 + 5x_4 & \leq 700 \\ & 4x_1 + 6x_2 + 6x_3 + 4x_4 & \leq 500 \\ & 8x_1 + 5x_2 + 5x_3 + 6x_4 & \leq y_s \\ & 7x_1 + 8x_2 + 7x_3 + 4x_4 & \leq y_u \\ & y_s & \leq 600 \\ & y_u & \leq 650 \\ & x_1, x_2, x_3, x_4, y_u, y_s & \geq 0 \end{array}$$

## 1.2 Linear Programs

### 1.2.1 Affine Function

A function $f : \mathbb{R}^n \to \mathbb{R}$ is **affine** if $f(\vec{x}) = \alpha^T \vec{x} + b$, where $\vec{x}, \alpha \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Note that if $b = 0$ , $f$ becomes a *linear* function. Thus, every linear function is affine but the converse is not true.

### 1.2.2 Linear Constraint

A **linear constraint** is a constraint that is one of the following forms (after algebraic manipulations): $f(x) \leq \beta$, or $f(x) \geq \beta$, or $f(x) = \beta$, where $f(x)$ is a linear function and $\beta \in \mathbb{R}$.

### 1.2.3 Linear Program

A **linear program** is the problem of maximizing or minimizing an *affine* function subject to a *finite* number of linear constraints.

### 1.2.4 Example: LP

The following is a linear program:

$$\begin{aligned}
\max \quad & -2x_1 + 7x_3 \\
s.t. \quad & x_1 + 7x_2 \leq 3 \\
& 3x_2 + 4x_3 \leq 2 \\
& x_1, x_2 \geq 0 \quad (\text{or } \vec{x} \geq \vec{0})
\end{aligned}$$

### 1.2.5 Example: Not LP

The following is not a linear program:

$$\begin{aligned}
\max \quad & -1/x_1 - 3 \\
s.t. \quad & 2x_1 + x_3 < 3 \\
& x_1 + \alpha x_2 = 2 \quad \forall \alpha \in \mathbb{R}
\end{aligned}$$

1. The objective function is not affine.
2. The inequalities cannot be strict.
3. The number of constraints must be finite.

### 1.2.6 Multi-period Models

A **multi-period model** is one where time is split into periods, we want to make a decision in each period, and all decisions influence the final outcome.

### 1.2.7 Concrete Example

Suppose a company has the following demand of oil for each month where the prices are as given:

| Month | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Demand (litre) | 5000 | 8000 | 9000 | 6000 |
| Price ($/litre) | 0.75 | 0.72 | 0.92 | 0.90 |

The company has a storage tank on its facility which can hold up to 4000 litres of oil. At the beginning of month 1, it contains 2000 liters already. We want to calculate how much oil it should purchase at the beginning of each of the four months such that it satisfies the demand at the minimum possible total cost. Note that, oil that is left over at the end of the month goes into storage.

Let $p_i$, $i \in \{1, 2, 3, 4\}$ denote the number of litres of oil purchased at the beginning of each month, and $t_i$, $i \in \{1, 2, 3, 4\}$ denote the number of litres in the storage tank at the beginning of each month (we are given that $t_1 = 2000$). Then the objective function is

$$\min \quad 0.75p_1 + 0.72p_2 + 0.92p_3 + 0.90p_4.$$

For constraints, we first recognize the relationship between $t_i$ and $p_i$ -- the amount of available oil at the beginning of month $i$ is comprised of two parts, namely the $p_i$ litres of oil (newly) purchased in month $i$, and $t_i$ litres of left over from month $i-1$. For example, we could express $t_2$ as $p_1 + t_1 - 5000$, since we purchased $p_1$ litres in addition to the original $t_1$ litres, then used 5000 during the first month. Rearranging, we get the following equations:

$$\begin{aligned} p_1 + t_1 &= 5000 + t_2 \\ p_2 + t_2 &= 8000 + t_3 \\ p_3 + t_3 &= 9000 + t_4 \end{aligned}$$

Finally, in order to satisfy the demand in month 4, we have $p_4 + t_4 \geq 6000$.

Notice $t_i$ for $i \in \{2, 3, 4\}$ appears in two of the four constraints above. The constraints are therefore linked by $t_i$'s. You will often see such linkage in multi-period models. These constraints are sometimes called *balance constraints* as they *balance* demand and inventory between periods.

In addition to these, we know that $t_1 = 2000$, $t_i \leq 4000$ for $i \in \{2, 3, 4\}$, and all variables are non-negative. Combining everything above, we get the following model for this multi-period problem:

$$\begin{aligned} \min \quad & 0.75p_1 + 0.72p_2 + 0.92p_3 + 0.90p_4 \\ s.t. \quad & \\ & p_1 + t_1 = 5000 + t_2 \\ & p_2 + t_2 = 8000 + t_3 \\ & p_3 + t_3 = 9000 + t_4 \\ & p_4 + t_4 \geq 6000 \\ & t_1 = 2000 \\ & t_i \leq 4000, \quad i \in \{2, 3, 4\} \\ & t_i, p_i \geq 0, \quad i \in \{1, 2, 3, 4\} \end{aligned}$$

## 1.3 Integer Programs

### 1.3.1 Integer Program

An **integer program** is obtained by taking a linear program and adding the condition that a non-empty subset of variables be required to take integer values. When all variables are required to take integer value, the integer program is called a **pure integer program**; otherwise, it is called a **mixed integer program**.

### 1.3.2 Example: IP

The following is a (mixed) integer program:

$$\begin{array}{lrl} \max & x_1 + x_2 + 2x_4 & \\ s.t. & x_1 + x_2 & \leq 1 \\ & -x_2 - x_3 & \geq -1 \\ & x_1 + x_3 & = 1 \\ & x_1, x_2, x_3 & \geq 0 \\ & x_1, x_3 & \in \mathbb{N} \end{array}$$

### 1.3.3 Efficiency

An algorithm is efficient if its running time, $f(n)$, is a polynomial of $n$. LPs can be solved efficiently, but IPs are very unlikely to have efficient algorithms.

### 1.3.4 Binary Variables

Variables that can take only a value of 0 or 1 are called binary; they are useful for expressing **logical conditions**.

### 1.3.5 Assignment Problem

Let $i \in I$ be an employee, $j \in J$ be a job, then $\sum_{j \in J} x_{ij}$ is the number of jobs employee $i$ is assigned to and $\sum_{i \in I} x_{ij}$ is the number of employees job $j$ is assigned to. We want both quantity to be one, so the following should hold:

$$\sum_{j \in J} x_{ij} = 1 \quad (i \in I), \quad \sum_{i \in I} x_{ij} = 1 \quad (j \in J), \quad x_{ij} \in \{0, 1\} \quad (i \in I, j \in J).$$

Finally, let $c_{ij}$ denote the resource (e.g., time) needed for employee $i$ to finish job $j$, then our objective function is

$$\min \quad \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}.$$

### 1.3.6 Knapsack Problem

Let $x_i$ denote the number of type $i$ selected. Suppose we want to express the condition "at least one of the following two conditions has to be satisfied":

- A total of at least four crates of type 1 or type 2 is selected, or
- A total of at least four crates of type 5 or type 6 is selected.

We introduce a binary variable $y$. If $y = 1$, then we want the first condition to be true, and if $y = 0$ we want the second condition to be true. This can be achieved by adding the constraints $x_1 + x_2 \geq 4y$ and $x_5 + x_6 \geq 4(1 - y)$.

## 1.4 Optimization Problems on Graphs

### 1.4.1 Graph Overview (See Introduction to Combinatorics)

A graph $G$ is an ordered pair $(V, E)$, where $V$ is a finite set and $E$ is a set of (unordered) pairs of elements of $V$. Elements of $V$ are called **vertices** and elements of $E$ **edges**. For $u, v \in V(G)$, $u$ and $v$ are adjacent if there exists an edge $uv \in E(G)$; we call vertices $u$ and $v$ the **endpoints** of edge $uv$, and edge $uv$ is **incident** to vertices $u$ and $v$.

For vertices $s, t \in V(G)$, an $(s, t)$ path $P$ is a sequence of edges $v_1v_2, v_2v_3, \ldots, v_{k-2}v_{k-1}, v_{k-1}v_k$ such that $v_1 = s$, $v_k = e$, and $v_i \neq v_j$ for all $i \neq j$. The length/cost/weight $c(P)$ of an $(s, t)$ path is defined as the sum of the lengths/costs/weights of the edges of $P$, i.e., $c(P) = \sum_{e \in P} c_e$.

### 1.4.2 Shortest Path Problem

Given a graph $G = (V, E)$ with non-negative weights $c_e$ for every edge $e \in E(G)$ and distinct vertices $s$ and $t$, we wish to find the one of minimum cost among all possible $(s, t)$ paths:

$$\begin{aligned} \min \quad & c(P) \\ s.t. \quad & P \text{ is an } (s,t) \text{ path} \end{aligned}$$

### 1.4.3 Bipartite Graphs (See Introduction to Combinatorics)

A graph $G = (V, E)$ is **bipartite** if we can partition the vertices into two sets, say $I$ and $J$, such that every edge has one endpoints in $I$ and the other one in $J$. Given a graph, a subset of edges $M$ is called a **matching** if no vertex is incident to more than one edge of $M$. A matching is called **perfect** if every vertex is incident to exactly one edge in the matching. The **weight** $c(M)$ of a matching $M$ is defined as the sum of the weights of all edges in $M$, i.e., $c(M) = \sum_{e \in M} c_e$.

### 1.4.4 Minimum Cost Perfect Matching Problem

Given a bipartite graph $G = (V, E)$ with non-negative edge weights $c_e$ for all $e \in E(G)$, we wish to solve

$$\begin{aligned} \min \quad & c(M) \\ s.t. \quad & M \text{ is a perfect matching} \end{aligned}$$

## 1.5 Integer Programs Continued

### 1.5.1 Minimum Cost Perfect Matching

Let $G = (V, E)$ be a graph and let $v \in V$. We denote by $\delta(v)$ the set of edges that have $v$ as one of the endpoints, i.e., $\delta(v)$ is the set of edges incident to $v$.

Given a graph $G = (V, E)$ and edge weight $c_e$ for every edge $e$, we want to determine the variables, the objective function, and the constraints.

We introduce a binary variable $x_e$ for every edge $e$, where $x_e = 1$ indicates that edge $e$ is selected to be part of our perfect matching and $x_e = 0$ indicates that edge $e$ is not selected. We can thus express the edges in the matching by $\{e \in E : x_e = 1\}$.

Let $v$ be a vertex. The number of edges incident to $v$ that are selected is $\sum(x_e : e \in \delta(v))$. Since we want a perfect matching, we need that number to be equal to 1, so $\sum(x_e : e \in \delta(v)) = 1$.

To minimize the total weight of the edges in the matching $M$ is selected, the objective function should return the weight of $M$. If $e$ is an edge of $M$, we will have $x_e = 1$ and we should contribute $c_e$ to the objective function, otherwise $x_e = 0$ and there is no contribution. This can be modelled by the term $c_e x_e$. Thus, the objective function is $\sum(c_e x_e : e \in E)$.

$$
\begin{array}{lrll}
\min & \sum(c_e x_e : e \in E) & & \\
s.t. & & & \\
& \sum(x_e : e \in \delta(v)) & = 1 & (v \in V) \\
& x_e & \geq 0 & (e \in E) \\
& x_e & \in \mathbb{Z} &
\end{array}
$$

Note that we could also use an adjacency matrix for constraints.

### 1.5.2 Shortest Path Problem

Let $G = (V, E)$ be a graph and let $U \subseteq V$ be a subset of the vertices. Generalizing from the previous section, let $\delta(U)$ denote the set of edges that have exactly one endpoint in $U$, i.e.,

$$\delta(U) = \{uv \in E : u \in U, v \notin U\}.$$

Consider a graph $G = (V, E)$ with distinct vertices $s$ and $t$ and let $P$ be an $st$-path of $G$. Let $\delta(U)$ be an arbitrary $st$-cut of $G = (V, E)$. Follow the path $P$, starting from $s$ and denote by $u$ the last vertex of $P$ in $U$, and denote by $u'$ the vertex that follows $u$ in $P$. Since $s \in U$ and $t \notin U$, by definition, $uu'$ is an edge that is in $\delta(u)$.

# Lecture 7. Possible Outcomes

## 7.1 Feasibility and Optimality

An assignment of values to each of the variables is a **feasible solution** if all of the constraints are satisfied. An optimization is **feasible** if it has at least one feasible solution. Otherwise, it is **infeasible**.

For a *maximization* problem, an **optimal solution** is a feasible solution that *maximize* the object function. For a *minimization* problem, an **optimal solution** is a feasible solution that *minimize* the object function. Note that an optimization problem can have several optimal solutions.

## 7.2 Fundamental Theorem of Linear Programming

**Theorem 7.2.1** For any linear program, *exactly one* of the following holds:

1. It has an optimal solution.
2. It is infeasible.
3. It is unbounded.

*Remark.* The statement only holds for LPs; there are optimization problems where none of the above hold.

**Example 7.2.2**

- An LP with an optimal solution: $\max x_1$ with $x_1 \leq 1$.
- An infeasible LP: $\max x_1$ with $x_1 \leq 0$ and $x_1 \geq 1$.
- An unbounded LP: $\max x_1$ with $x_1 \geq 1$.

**Definition 7.2.3** We now describe what we mean by *solving a linear program*:

1. If the LP has an optimal solution, return an optimal solution with a proof.
2. If the LP is infeasible, return a proof of infeasibility.
3. If the LP is unbounded, return a proof of unboundedness.

We always need to justify our solution!

# Lecture 8. Certificates

## 8.1 Proving Infeasibility

**Example 8.1.1** Consider the following LP. Does the LP have a feasible solution?

$$\begin{aligned} \max \quad & x_1 + x_2 + x_3 \\ s.t. \quad & x_1 + x_2 + x_3 = 6 \\ & 2x_1 + 3x_2 + x_3 = 8 \\ & 2x_1 + x_2 + 3x_3 = 0 \end{aligned}$$

We can rewrite the set of constraints and use linear algebra.

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 1 \\ 2 & 1 & 3 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 6 \\ 8 \\ 0 \end{pmatrix}.$$

The problem now becomes, does there exist an $\vec{x} \in \mathbb{R}^3$ such that $A\vec{x} = \vec{b}$? Using Gaussian elimination, we find a vector $\vec{y} = (4, -1, -1)^T$ such that $\vec{y}^T A\vec{x} = 0$ and $\vec{y}^T \vec{b} = 16 \neq 0$. This vector $\vec{y}$ is a **certificate of infeasibility**.

**Theorem 8.1.2 [Gauss]** Let $A \in \mathbb{R}^{m,n}$, $\vec{b} \in \mathbb{R}^m$. Then exactly one of the following holds:

1. (Has Solution) There exists $\vec{x} \in \mathbb{R}^n$ such that $A\vec{x} = \vec{b}$, or
2. (No Solution) There exists $\vec{y} \in \mathbb{R}^m$ such that $\vec{y}^T A = \vec{0}$ and $\vec{y}^T \vec{b} \neq \vec{0}$.

In general, we would like to be able to perform such tests for the inequality $A\vec{x} \leq \vec{b}$.

**Theorem 8.1.3 [Farkas]** Let $A \in \mathbb{R}^{m,n}$, $b \in \mathbb{R}^m$. Then exactly one of the following holds:

1. (Feasible) There exists $x \in \mathbb{R}^n$ such that $Ax = b$ and $x \geq 0$, or
2. (Infeasible) There exists $y \in \mathbb{R}^m$ such that $y^T A \leq 0$ and $y^T b > 0$.

## 8.2 Proving Optimality

**Example 8.2.1** Suppose we are given the following LP.

$$\begin{aligned} \max \quad & z(\vec{x}) := \begin{pmatrix} -1 & -4 & 0 & 0 \end{pmatrix} \vec{x} + 4 \\ s.t. \quad & \begin{pmatrix} 1 & 3 & 1 & 0 \\ -2 & 6 & 0 & 1 \end{pmatrix} \vec{x} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}, \\ & \vec{x} \geq \mathbf{0}. \end{aligned}$$

One of the optimal solution is $\vec{x} = (0, 0, 4, 5)^T$, which yields a value of 4.

To show optimality of the solution, we can use the following argument. Let $x'$ be an arbitrary feasible solution. Then $z(\vec{x}) = \begin{pmatrix} -1 & -4 & 0 & 0 \end{pmatrix} \vec{x} + 4 \leq 4$ as $\begin{pmatrix} -1 & -4 & 0 & 0 \end{pmatrix} \vec{x} \leq 0$. The result follows.

**Theorem 8.2.2** To show optimality of $\bar{x}$ given an objective vector $\boldsymbol{c} \leq \mathbf{0}$: Let $x'$ be an arbitrary feasible solution. Since $\boldsymbol{c} \leq \mathbf{0}$ and $x \geq \mathbf{0}$, we have $z(x) = \boldsymbol{c}^T x + m \leq m$ as $\boldsymbol{c}^T x \leq 0$. The result follows.

## 8.3 Proving Unboundedness

**Example 8.3.1** Suppose we are given the following LP.

$$\begin{array}{ll} \max & z(\vec{x}) := \begin{pmatrix} -1 & 0 & 0 & 1 \end{pmatrix} \vec{x} \\ s.t. & \begin{pmatrix} -1 & -1 & 1 & 0 \\ -2 & 1 & 0 & 1 \end{pmatrix} \vec{x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \\ & \vec{x} \geq \mathbf{0}. \end{array}$$

Observe (1) $x(t) = (0,0,2,1)^T + t(1,0,1,2)^T$ is feasible for all $t \geq 0$ and (2) $t \to \infty \implies z \to \infty$.

*Proof for (1):* Define $\bar{x} := (0,0,2,1)^T$ and $\vec{r} := (1,0,1,2)^T$. Observe $x(t) = \bar{x} + t\vec{r} \geq 0$ for all $t \geq 0$ as $\bar{x}, \vec{r} \geq 0$. Then $Ax(t) = A[\bar{x} + t\vec{r}] = A\bar{x} + tA\vec{r} = \vec{b} + \vec{0} = \vec{b}$.

*Proof for (2):* Define $\vec{c} = (-1,0,0,1)^T$. Then $z = \vec{c}^T x(t) = \vec{c}^T [\bar{x} + t\vec{r}] = \vec{c}^T \bar{x} + tc^T \vec{r}$ where $\vec{c}^T \vec{r} = 1 > 0$.

We now generalize the result to form the following theorem.

**Theorem 8.3.2** The LP $\max\{\vec{c}^T \vec{x} : A\vec{x} = \vec{b}, \vec{x} \geq \mathbf{0}\}$ is unbounded if we can find $\bar{x}$ and $\vec{r}$ such that

$$\bar{x} \geq \mathbf{0}, r \geq \mathbf{0}, A\bar{x} = \vec{b}, A\vec{r} = \mathbf{0}, \text{ and } \vec{c}^T \vec{r} > \vec{0}.$$

# Lecture 9. Standard Equality Forms

## 9.1 Standard Equality Form

**Definition 9.1.1** A LP is in **Standard Equality Form** if

1. It is a *maximization* problem.
2. For every variable $x_j$ we have a constraint $x_j \geq 0$, and
3. All other constraints are *equality* constraints.

**Example 9.1.2** Is the following LP in SEF?

$$\begin{aligned} \max \quad & x_1 + x_2 + 17 \\ s.t. \quad & x_1 - x_2 = 0 \\ & x_1 \geq 0 \end{aligned}$$

No, as there is no constraint $x_2 \geq 0$. We say $x_2$ is *free*. Note that $x_2 \geq 0$ is implied by the constraints, but it is still free as the constraint $x_2 \geq 0$ is not given *explicitly*.

**Remark 9.1.3** We introduce this SEF as we will develop an algorithm called the **Simplex** that can solve any LP as long as it is in SEF.

**Remark 9.1.4** What do we do if the LP is not in SEF?

1. Find an equivalent LP in SEF.
2. Solve the equivalent LP using Simplex.
3. Use the solution of the equivalent LP to get the solution of the original LP.

Basically, two LPs are equivalent if they behave in the same way. We now give the formal definition.

## 9.2 Equivalent LPs

**Definition 9.2.1** LPs $P$ and $Q$ are **equivalent** if

- $P$ infeasible $\iff$ $Q$ infeasible.
- $P$ unbounded $\iff$ $Q$ unbounded.
- Given an optimal solution of one, we could construct an optimal solution of the other.

**Theorem 9.2.2** Every LP has an equivalent LP that is in SEF.

**Example 9.2.3** We now provide a series of examples to turn an LP into its SEF equivalent.

1. Replace minimization by maximization: $\min f(x) \implies \max -f(x)$.
2. Replacing inequalities by equalities: $x \leq a \implies x + s = a$, $s \geq 0$; $x \geq a \implies x - s = a$, $s \geq 0$.
3. Free variables: Suppose we are given

$$\begin{aligned} \max \quad & z = (1,2,3)(x_1,x_2,x_3)^T \\ s.t. \quad & \begin{pmatrix} 1 & 5 & 3 \\ 2 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \end{pmatrix} \\ & x_1, x_2 \geq 0, x_3 \text{ free} \end{aligned}$$

Since any number is the difference between two *non-negative* numbers, we set $x_3 := a - b$ where $a, b \geq 0$. We then rewrite the objective function and constraints by carrying out the arithmetic (omitted) and arrive at the following:

$$\begin{aligned} \max \quad & z = (1,2,3,-3)(x_1,x_2,a,b)^T \\ s.t. \quad & \begin{pmatrix} 1 & 5 & 3 & -3 \\ 2 & -1 & 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ a \\ b \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \end{pmatrix} \\ & x_1, x_2, a, b \geq 0. \end{aligned}$$

Note this new LP is in SEF.

# Lecture 11. Basis and Basic Solutions

## 11.1 Basis

**Remark 11.1.1** Let $A \in \mathbb{Z}^{m,n}$ be an integer matrix and $B$ be a subset of column indices. Then $A_B$ is a column sub-matrix of $A$ indexed by set $B$; $A_j$ denotes column $j$ of $A$.

**Example 11.1.2** Consider the following.

$$A = \begin{pmatrix} 1 & 2 & -1 & 1 & -1 \\ 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix},$$

$$B = \{1,2,3\} \implies A_B = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \{5\} \implies A_B = A_5 = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}.$$

**Definition 11.1.3** Let $A \in \mathbb{Z}^{m,n}$ and $B$ be a subset of column indices. Then $B$ is a **basis** if

1. $A_B$ is a *square matrix*,
2. $A_B$ is *non-singular*, i.e., columns are *independent*.

**Example 11.1.4** Consider the following.

$$A = \begin{pmatrix} 1 & 2 & -1 & 1 & -1 \\ 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix},$$

1. $B = \{1,5\}$ is not a basis, because $A_B$ is not a square matrix.
2. $B = \{2,3,4\}$ is not a basis, because the columns are independent $(A_2 + A_3 = A_4)$.

**Remark 11.1.5** Does every matrix have a basis? No. Recall from linear algebra, the max number of independent columns equals the max number of independent rows. Thus, if not all rows are independent, there will not be a basis for the matrix. On the other hand, if all rows of $A$ are independent, then $B$ is a basis if and only if $B$ is a maximal set of independent columns of $A$.

## 11.2 Basic Solutions I

**Definition 11.2.1** Let $B$ be a basis for $A$. Then $x_j$ is a **basic variable** if and only if $j \in B$.

**Example 11.2.2** Given basis $B = \{1,2,4\}$, $x_1, x_2, x_4$ are basic variables and $x_3, x_5$ are not.

**Definition 11.2.3** We say $x$ is a **basic solution** for basis $B$ if $Ax = b$ and $x_j = 0$ for any $j \notin B$
.

**Example 11.2.4** Consider the following.

$$A = \begin{pmatrix} 1 & 2 & -1 & 1 & -1 \\ 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix}.$$

For basis $B = \{1,3,4\}$, $x = (1,0,0,1,0)$ is a basic solution as $Ax = b$ and $x_2 = x_5 = 0$.

## 11.3 Basic Solutions II

**Example 11.3.1** To find a basic solution for $Ax = b$ given a basis $B$, we rewrite the equation in the form of $\sum_j x_j A_j = b$, set $x_j = 0$ for all $j \notin B$, then solve for rest of the entries.

**Theorem 11.3.2** Given $Ax = b$ and a basis $B$ of $A$, there exists a unique basic solution $x$ for $B$.

*Proof.* Observe $b = Ax = \sum_j A_j x_j = \sum_{j \in B} A_j x_j + \sum_{j \notin B} A_j x_j = \sum_{j \in B} A_j x_j + 0 = A_B x_B$. Since $B$ is a basis, $A_B$ is non-singular, so $A_B^{-1}$ exists. Hence, $x_B = A_B^{-1} b$. □

**Definition 11.3.3** We say $x$ is a **basic solution** if it is a basic solution for some basis $B$.

**Remark 11.3.4** To show that $x$ is a basic solution, find an appropriate basis $B$; to show that $x$ is not a basic solution, prove by contradiction: suppose $x$ is a basic solution for basis $B$ and reach the conclusion that $A_B$ is singular or not square.

**Remark 11.3.5** A basic solution can be the basic solution for more than one basis. Note we used "the" here as each basis has only one corresponding basic solution.

**Remark 11.3.6** How is this related to LPs? Recall LPs in SEF:
$(P) := \max\{c^T x : Ax = b, x \geq \mathbf{0}\}$. If the rows of $A$ are dependent, then either there is no solution to $Ax = b$, so $(P)$ is *infeasible*, or a constraint of $Ax = b$ is redundant and can be removed without changing the solutions. Thus, we may always assume that, when trying to solve the LP, the rows of $A$ are independent.

**Definition 11.3.7** A basic solution $x$ of $Ax = b$ is **feasible** if $x \geq 0$, i.e., if it is feasible for $(P)$.

# Lecture 12. Canonical Forms

## 12.1 Canonical Form

**Definition 12.1.1** Let $B$ be a basis of $A \in \mathbb{Z}^{m,n}$. Then (P) is in **canonical form** for $B$ if

1. $A_B = I$, and
2. $c_j = 0$ for all $j \in B$.

**Example 12.1.2** The following IP is in canonical form for $B = \{2, 3\}$:

$$\begin{aligned} \max \quad & (-2, 0, 0, 6)^T x + 2 \\ s.t. \quad & \\ & \begin{pmatrix} -1 & 1 & 0 & 3 \\ 1 & 0 & 1 & -1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

## 12.2 Rewriting (P) into Canonical Form

**Proposition 12.2.1** For any basis $B$, there exists (P') in canonical form for $B$ such that

1. (P) and (P') have the same feasible region, and
2. Feasible solutions have the same objective values for (P) and (P').

**Proposition 12.2.2** To rewrite (P) in canonical form for basis $B$:

1. Replace $Ax = b$ by $A'x = b'$ with $A' * B = I$. This is done by multiplying $A_B^{-1}$ to both sides: $Ax = b \implies \underbrace{A_B^{-1} A}_{A'} x = \underbrace{A_B^{-1} b}_{b'}$
2. Replace $c^T x$ by $\bar{c}^T x + \bar{z}$ where $\bar{c}_j = 0$ for $j \in B$ and $\bar{z}$ is a constant.
3. Construct a new objective function by multiplying constraint $i$ by $y_i$, $i \in 1, \ldots, |B|$ and adding the resulting constraints to the objective function.

$$\begin{array}{l} 0 = -y^T Ax + y^T b \\ z = c^T x \\ \hline z = [c^T - y^T A]x + y^T b \end{array}$$

4. Choose $y_i$, $i \in 1, \ldots, |B|$ to get $\bar{c}_j = 0$ for $j \in B$.

$$\begin{aligned}
\bar{c}_B^T = 0^T &= c_B^T - y^T A_B \\
y^T A_B &= c_B^T \\
A_B^T y &= c_B \\
y = (A_B^T)^{-1} c_B &= A_B^{-T} c_B \qquad\qquad\qquad *
\end{aligned}$$

(Note that for any non-singular matrix $M$, $(M^T)^{-1} = (M^{-1})^T =: M^{-T}$.)

**Example 12.2.3** Consider the following example, where we rewrite its constraints.

$$\begin{aligned}
\begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} x &= \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} x &= \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
\begin{pmatrix} -1 & 1 & 0 & 3 \\ 1 & 0 & 1 & -1 \end{pmatrix} x &= \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\end{aligned}$$

Observe we multiplied the inverse of $A_{\{2,3\}}$ to both sides, which guaranteed to turn $A_{\{2,3\}}$ to $I_2$.

**Example 12.2.4** Consider the following example, where we rewrite the objective function.

$$\begin{aligned}
\max \quad & z = (0, 0, 2, 4)^T x \\
s.t. \quad & \\
& \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} x \quad = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
& x_1, x_2, x_3, x_4 \geq 0
\end{aligned}$$

First, construct a new objective function:

$$\begin{aligned}
\begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} x &= \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
\begin{pmatrix} y_1 & y_2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} x &= \begin{pmatrix} y_1 & y_2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
0 &= - \begin{pmatrix} y_1 & y_2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} x + \begin{pmatrix} y_1 & y_2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
z &= \begin{pmatrix} 0 & 0 & 2 & 4 \end{pmatrix} x \\
z &= \underbrace{\left[ \begin{pmatrix} 0 & 0 & 2 & 4 \end{pmatrix} - \begin{pmatrix} y_1 & y_2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} \right]}_{\bar{c}^T} x + \underbrace{\begin{pmatrix} y_1 & y_2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix}}_{\bar{z}}
\end{aligned}$$

As a remark, line 4 is the objective function; line 5 is the sum of line 3 and line 4. Observe for any choice of $y_1$, $y_2$ and any feasible solution $x$, the objective value of $x$ for the old objective function is equal to that of the new objective function.

Now choose $y_1$ and $y_2$ to make $\bar{c}_2 = \bar{c}_3 = 0$:

$$\bar{c}_B^T = \begin{pmatrix} 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2 \end{pmatrix} - \begin{pmatrix} y_1 & y_2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} y_1 & y_2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^T \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$z = \left[ \begin{pmatrix} 0 & 0 & 2 & 4 \end{pmatrix} - \begin{pmatrix} 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} \right] x + \begin{pmatrix} 2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$z = \begin{pmatrix} -2 & 0 & 0 & 6 \end{pmatrix} x + 2$$

We get the rewritten LP:

$$\begin{aligned} \max \quad & z = \begin{pmatrix} -2 & 0 & 0 & 6 \end{pmatrix} x + 2 \\ s.t. \quad & \\ & \begin{pmatrix} -1 & 1 & 0 & 3 \\ 1 & 0 & 1 & -1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

# Lecture 13. Formalizing the Simplex

## 13.1 Simplex

**Remark 13.1.1** Given an LP (P) in canonical form for some basis $B$ and a feasible solution $\bar{x}$ (w.r.t $B$), how do we find a better feasible solution? One obvious approach is to pick $k \notin B$ where $c_k > 0$, and set $x_k = t \geq 0$ as large as possible while keeping all other non-basic variables at $0$. At each iteration, there will be one index entering the basis and another one leaving it. We stop when we reach the upper bound of the objective function (see previous lectures on proving optimality), or found out the LP is unbounded (see previous lectures on proving unboundedness).

**Theorem 13.1.2** We now present the simplex algorithm. Given an LP with a feasible basis $B$, we want to output an optimal solution or report that the LP is unbounded. First, we rewrite the LP in canonical form for the basis $B$:

$$\begin{array}{ll} \max & c^T x \\ s.t. & Ax = b \\ & x \geq \mathbf{0} \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \max & z = c_N^T x_N + \bar{z} \\ s.t. & x_B + A_N x_N = b \\ & x \geq \mathbf{0} \end{array}$$

At this stage, $B$ is a feasible basis, $N = \{j \notin B\}$; the LP is in canonical form for $B$, and $\bar{x}$ is a basic solution (Remark: $A_B = I$). Next, we find a better basis $B'$ or get required outcome:

1. If $c_N \leq \mathbf{0}$, STOP. The basic solution $\bar{x}$ is optimal. (Proof A)
2. Pick $k \notin B$ such that $c_k > 0$ and set $x_k = t$.
3. Pick $x_B = b - tA_k$.
4. If $A_k \leq 0$, STOP. The LP is unbounded. (Proof B)
5. Choose $t = \min\{b_i/A_{ik} \mid \forall i : A_{ik} \geq 0\}$.
6. Let $x_r$ be a basis variable forced to 0.
7. Obtain the new basis by having $k$ enter and $r$ leave.

*Proof A. If $c_N \leq \mathbf{0}$, STOP. The basic solution $\bar{x}$ is optimal.*

Given $\bar{x}_N = \mathbf{0}$, $\bar{x}_B = b$, and $\bar{x}$ has value $z = c_N^T \bar{x}_N + \bar{z} = \bar{z}$, let $x$ be a feasible solution. Then

$$z = \underbrace{c_N^T}_{\leq 0} \underbrace{x_N}_{\geq 0} + \bar{z} \leq \bar{z}. \quad \blacksquare$$

*Proof B. If $A_k \leq 0$, STOP. The LP is unbounded.*

First, given $x_k = t \geq 0$ and all other non-basic variables have value zero, $x_B = \underbrace{b}_{\geq 0} - \underbrace{t}_{\geq 0}\underbrace{A_k}_{\leq 0} \geq \mathbf{0}$. Next, as $t \to \infty$, $z = \sum_{j \in N} c_j x_j + \bar{z} = c_k x_k + \bar{z} = c_k t + \bar{x} \to \infty$ as $c_k > 0$. ■

**Remark 13.1.3** Indeed, Simplex tells us the truth: if it claims the LP is unbounded, it is unbounded; if it claims a solution is optimal, it is optimal. However, the Simplex algorithm may not terminate! In other words, starting with a feasible basis $B_1$, we may enter a cycle and reach $B_1$ again. To solve this, we introduce Bland's rule.

**Definition 13.1.4** **Bland's rule** states that, if we have a choice for element entering or leaving the basis, always pick the smallest one.

**Theorem 13.1.5** If we use Bland's rule, then the Simplex algorithm always terminates.

**Remark 13.1.6** So far, we have seen a formal description of the Simplex algorithm; we showed that if the algorithm terminates, then it is correct; we defined Bland's rule and asserted that the Simplex terminates as long as we are following Bland's rule. But to get started with Simplex, we need to have a feasible basis. Therefore, the next step in our journey is to define a procedure to find a feasible basis.

# Lecture 15. Half-Spaces and Convexity

## 15.1 Geometry of Half-Spaces

### Definition 15.1.1

- The **feasible region** for an optimization problem is the set of all feasible solutions.
- $P \subseteq \mathbb{R}^n$ is a **polyhedron** if there exists a matrix $A$ and a vector $b$ such that $P = \{x : Ax \leq b\}$.

**Remark 15.1.2** The feasible region of an LP is a polyhedron.

**Definition 15.1.3** Let $a \neq \mathbf{0}$ be a vector and $\beta \in \mathbb{R}$.

1. $\{x : a^T x = \beta\}$ is a **hyperplane**, which is the set of solutions to a single **linear equation**.
2. $\{x : a^T x \leq \beta\}$ is a **half-space**, which is the set of solutions to a single **linear inequality**.

**Remark 15.1.4** A polyhedron is the intersection of a *finite* set of half-spaces. Thus, understanding the geometry of a polyhedra is satisfied by understanding the geometry of half-spaces.

### Remark 15.1.5

- Let $a, b$ be vectors. Then $a^T b = \|a\|\|b\| \cos(\theta)$ where $\| \cdot \|$ is the Euclidean norm and $\theta$ the angle between $a$ and $b$. Recall $a \perp b \iff a^T b = 0$; $\theta < 90^\circ \iff a^T b > 0$; $\theta > 90^\circ \iff a^T b < 0$.
- Given vector $a \neq 0$ and $\beta = 0$, we have
  - The set of vectors orthogonal to $a$: $H = \{x : a^T x = \beta\}$.
  - The set of vectors on the side of $H$ opposite to $a$: $F = \{x : a^T x \leq \beta\}$.

**Definition 15.1.6** Let $S, S' \subseteq \mathbb{R}^n$. The $S'$ is a **translate** of $S$ if there exists $p \in \mathbb{R}^n$ and $S' = \{s + p : s \in S\}$.

### Proposition 15.1.7

- Let $a \neq \mathbf{0}$ be a vector and $\beta \in \mathbb{R}$ and let $H := \{x : a^T x = \beta\}$ and $H_0 := \{x : a^T x = 0\}$. Then $H$ is a translate of $H_0$.
- Let $a \neq \mathbf{0}$ be a vector and $\beta \in \mathbb{R}$ and let $F := \{x : a^T x \leq \beta\}$ and $F_0 := \{x : a^T x \leq 0\}$. Then $F$ is a translate of $F_0$.

*Proof.* Choose $p \in H$. We want to show $x \in H_0 \iff x + p \in H$. Observe $x \in H_0 \iff a^T x = 0 \iff a^T x + a^T p = 0 + a^T p \iff a^T(x+p) = \beta \iff x + p \in H$. The second part can be proved using an identical argument. $\square$

**Remark 15.1.8** Let $a \in \mathbb{R}^n$, $a \neq \mathbf{0}$ and let $\beta \in \mathbb{R}$. Define $H := \{x : a^T x = \beta\}$ and $H_0 := \{x : a^T x = 0\}$. We define the dimension of $H$ to be the dimension of $H_0$, which is a vector space whose dimension can be computed as $\dim(H_0) = n - \operatorname{rank}(a) = n - 1$. In short, the dimension of a hyperplane in $\mathbb{R}^n$ is $n - 1$.

## 15.2 Convexity

**Definition 15.2.1** Let $x, y \in \mathbb{R}^n$.

- The **line through $x$ and $y$** is defined as $L = \{\lambda x + (1-\lambda)y : \lambda \in \mathbb{R}\}$.
- The **line segment** between $x$ and $y$ is defined as $S = \{\lambda x + (1-\lambda)y : \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1\}$.
- A set $S \subseteq \mathbb{R}^n$ is **convex** if the line segment between $x$ and $y$ is in $S$ for any $x, y \in S$.

**Remark 15.2.2** For $A, B \in \mathbb{R}^n$, if $A$ and $B$ are both convex then $A \cap B$ must be convex.

**Proposition 15.2.3** Let $H = \{x : a^T x \leq \beta\}$ be a half-space. Then $H$ is convex.

*Proof.* Pick arbitrary points $x, y \in H$ and an arbitrary point $\bar{x}$ in the line segment between $x$ and $y$. We want to show that $\bar{x} \in H$:

$$a^T \bar{x} = a^T\big(\lambda x + (1-\lambda)y\big) = \underbrace{\lambda}_{\geq 0}\,\underbrace{a^T x}_{\leq \beta} + \underbrace{(1-\lambda)}_{\geq 0}\underbrace{a^T y}_{\leq \beta} \leq \lambda\beta + (1-\lambda)\beta = \beta \implies \bar{x} \in H. \quad \square$$

**Corollary 15.2.4** If $P$ is a polyhedron, then $P$ is convex.

*Proof.* As a polyhedron, $P$ is the intersection of half-spaces; each half-space is convex and the intersection of convex set is convex. Thus, $P$ is convex. $\square$

**Remark 15.2.5** This tells us that *the feasible region of an LP is always a convex set!*

# Lecture 16. Extreme Points

**Definition** Point $x \in \mathbb{R}^n$ is **properly contained** in the line segment $L$ if $x \in L$ and is distinct from the endpoints of $L$.

**Definition** Let $S$ be a convex set and $\bar{x} \in S$. Then $\bar{x}$ is NOT an **extreme point** if there exists a line segment $L \subseteq S$ where $L$ properly contains $\bar{x}$.

**Remark**

- A convex set may have an *infinite* number of extreme points, e.g., a circle.
- A convex set may have *no* extreme points at all, e.g., a half-space.

**Definition** Let $P = \{x : Ax \leq b\}$ be a polyhedron and let $x \in P$. A constraint is **tight** for $x$ if it is satisfied with equality; we denote the set of all tight constraints by $\bar{A}x \leq \bar{b}$.

**Remark** Let $a, b, c \in \mathbb{R}$, and $\lambda$ where $0 < \lambda < 1$. Suppose $a = \lambda b + (1-\lambda)c$ and $b \leq a, c \leq a$. Then $a = b = c$. To see this, observe $a = \lambda b + (1-\lambda)c \leq \lambda a + (1-\lambda)a = a$. The result follows.

**Theorem** Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ be a polyhedron and let $\bar{x} \in P$.

- If $\mathbf{rank}(\bar{A}) = n$, then $\bar{x}$ is an extreme point.
- If $\mathbf{rank}(\bar{A}) < n$, then $\bar{x}$ is NOT an extreme point.

*Proof.*

(A) Suppose $\bar{x}$ is not an extreme point. Then $\bar{x}$ is properly contained in a line segment with end points $x, y \in P$. That is, $\bar{x} \neq x, y \in P$ and for some $0 < \lambda < 1$, $\bar{x} = \lambda x + (1-\lambda)y$. Observe

$$\bar{b} = \bar{A}\bar{x} = \bar{A}\big(\lambda x + (1-\lambda)y\big) = \lambda \bar{A}x + (1-\lambda)\bar{A}y$$

where $\bar{A}x \leq \bar{b}$ and $\bar{A}y \leq \bar{b}$. By our remark, $\bar{b} = \bar{A}x = \bar{A}y$. However, since $\mathbf{rank}(\bar{A})$, there is a unique solution to $\bar{A}x = b$, which implies $x = y$. Contradiction.

(B) Since $\mathbf{rank}(\bar{A}) < n$, there exists a vector $d \neq \mathbf{0}$ such that $\bar{A}d = \mathbf{0}$. Pick small $\varepsilon > 0$ and define $x = \bar{x} + \varepsilon d$ and $y = \bar{x} - \varepsilon d$. It suffices to prove the following:

1. $\bar{x}$ is properly contained in the line segment between $x$ and $y$.
   Indeed, observe

$$\frac{1}{2}x + \frac{1}{2}y = \frac{1}{2}(\bar{x} + \varepsilon d) + \frac{1}{2}(\bar{x} - \varepsilon d) = \bar{x}.$$

2. $x, y \in P$.
   We have two types of constraints:

a. Tight ones $\bar{A}x \leq \bar{b}$: observe $\bar{A}x = \bar{A}(\bar{x} + \varepsilon d) = \bar{A}\bar{x} + \mathbf{0} = \bar{b}$, so this type is satisfied.

b. Non-tight ones $a^T x \leq \beta$: $a^T x = a^T(\bar{x} + \varepsilon d) = a^T \bar{x} + \varepsilon a^T d < \beta + X^* \leq \beta$.

i. Since $a^T x \leq \beta$ is non-tight, it is satisfied with a strict inequality, i.e., $a^T \bar{x} < \beta$.

ii. We don't know $\varepsilon a^T d$, but we get to choose $\varepsilon$ to make the result small enough.

A mirror argument can be used to show $y \in P$. $\square$

**Remark** We must look at the rank of $\bar{A}$, not just how many rows it has (may be linearly dependent)!

**Theorem** Let $P = \{x \geq \mathbf{0} : Ax = b\}$ where rows of $A$ are independent. TFAE:

- $\bar{x}$ is an extreme point of $P$.
- $\bar{x}$ is a basic feasible solution to $P$.

**Remark** The Simplex algorithm moves from extreme points to extreme points.