Linear Programming: Canonical Form

- *Canonical form* of linear programming:

$$
\begin{aligned}
\max_{x_1,\ldots,x_n} \sum_{j=1}^{n} c_j x_j & \\
\text{s.t.} \quad \sum_{i} a_{ij} x_j \le b_i, & \quad i = 1, 2, \ldots, m; \\
x_j \ge 0, & \quad j = 1, 2, \ldots, n.
\end{aligned}
$$

Linear Programming: Canonical Form

- *Canonical form* of linear programming:

$$\max_{x_1,\dots,x_n} \sum_{j=1}^{n} c_j x_j$$
$$\text{s.t.} \sum_{i} a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, m;$$
$$x_j \geq 0, \quad j = 1, 2, \dots, n.$$

- Written in matrix form:

$$\max_{\mathbf{x}} \quad \mathbf{c}^{\mathsf{T}} \cdot \mathbf{x}$$
$$\text{s.t. } \mathbf{A}\mathbf{x} \leq \mathbf{b},$$
$$\mathbf{x} \geq 0;$$

where $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m, \mathbf{A} \in \mathbb{R}^{m \times n}$, and $\leq$ means coordinate-wise less than or equal to, and $\geq$ similarly.

The Geometry of LP

- A *(closed) half-space* in $\mathbb{R}^n$ is a set $\{\mathbf{x} : \mathbf{a}^\intercal \cdot \mathbf{x} \leq b\}$.

The Geometry of LP

- A *(closed) half-space* in $\mathbb{R}^n$ is a set $\{\mathbf{x} : \mathbf{a}^\intercal \cdot \mathbf{x} \leq b\}$.
- The intersection of finitely many half-spaces is a *convex polytope* (henceforth just polytope).

The Geometry of LP

- A *(closed) half-space* in $\mathbb{R}^n$ is a set $\{\mathbf{x} : \mathbf{a}^\intercal \cdot \mathbf{x} \leq b\}$.
- The intersection of finitely many half-spaces is a *convex polytope* (henceforth just polytope).
- The feasible region of an LP is a polytope.

The Geometry of LP

- A *(closed) half-space* in $\mathbb{R}^n$ is a set $\{\mathbf{x} : \mathbf{a}^\intercal \cdot \mathbf{x} \leq b\}$.
- The intersection of finitely many half-spaces is a *convex polytope* (henceforth just polytope).
- The feasible region of an LP is a polytope.
- A set $S \subseteq \mathbb{R}^n$ is *convex* if $\forall \mathbf{x}, \mathbf{y} \in S$, $\forall \lambda \in [0,1]$, $\lambda \mathbf{x} + (1-\lambda)\mathbf{y} \in S$.

The Geometry of LP

- A *(closed) half-space* in $\mathbb{R}^n$ is a set $\{\mathbf{x} : \mathbf{a}^\intercal \cdot \mathbf{x} \leq b\}$.
- The intersection of finitely many half-spaces is a *convex polytope* (henceforth just polytope).
- The feasible region of an LP is a polytope.
- A set $S \subseteq \mathbb{R}^n$ is *convex* if $\forall \mathbf{x}, \mathbf{y} \in S$, $\forall \lambda \in [0, 1]$, $\lambda \mathbf{x} + (1 - \lambda)\mathbf{y} \in S$.
- Example: a half-space is convex.

The Geometry of LP

- A *(closed) half-space* in $\mathbb{R}^n$ is a set $\{\mathbf{x} : \mathbf{a}^\intercal \cdot \mathbf{x} \leq b\}$.
- The intersection of finitely many half-spaces is a *convex polytope* (henceforth just polytope).
- The feasible region of an LP is a polytope.
- A set $S \subseteq \mathbb{R}^n$ is *convex* if $\forall \mathbf{x}, \mathbf{y} \in S$, $\forall \lambda \in [0, 1]$, $\lambda \mathbf{x} + (1 - \lambda)\mathbf{y} \in S$.
- Example: a half-space is convex.
- The intersection of convex sets is still convex.

The Geometry of LP

- A *(closed) half-space* in $\mathbb{R}^n$ is a set $\{\mathbf{x} : \mathbf{a}^\intercal \cdot \mathbf{x} \leq b\}$.
- The intersection of finitely many half-spaces is a *convex polytope* (henceforth just polytope).
- The feasible region of an LP is a polytope.
- A set $S \subseteq \mathbb{R}^n$ is *convex* if $\forall \mathbf{x}, \mathbf{y} \in S$, $\forall \lambda \in [0, 1]$, $\lambda \mathbf{x} + (1 - \lambda)\mathbf{y} \in S$.
- Example: a half-space is convex.
- The intersection of convex sets is still convex.
- Example: a polytope is convex.

Basic convex geometry

Definition

A point $\mathbf{x}$ is a *convex combination* of $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^k$ if there exists $\lambda_1, \cdots \lambda_k \geq 0$, such that $\sum_i \lambda_i = 1$ and $\sum_i \lambda_i \mathbf{y}^i = \mathbf{x}$.

Basic convex geometry

Definition

A point $\mathbf{x}$ is a *convex combination* of $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^k$ if there exists $\lambda_1, \cdots \lambda_k \geq 0$, such that $\sum_i \lambda_i = 1$ and $\sum_i \lambda_i \mathbf{y}^i = \mathbf{x}$.

Definition

The *convex hull* of a set X of points is the set of all convex combinations of the points in X.

Basic convex geometry

Definition

A point $\mathbf{x}$ is a *convex combination* of $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^k$ if there exists $\lambda_1, \cdots \lambda_k \geq 0$, such that $\sum_i \lambda_i = 1$ and $\sum_i \lambda_i \mathbf{y}^i = \mathbf{x}$.

Definition

The *convex hull* of a set X of points is the set of all convex combinations of the points in X.

Definition

A point $\mathbf{x}$ in a convex set S is an *extreme point* of S if there exist no $\mathbf{y}, \mathbf{z} \in S$ and $\lambda \in (0, 1)$ such that $\mathbf{x} = \lambda \mathbf{y} + (1 - \lambda)\mathbf{z}$.

Basic convex geometry

Definition

A point $\mathbf{x}$ is a *convex combination* of $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^k$ if there exists $\lambda_1, \cdots \lambda_k \geq 0$, such that $\sum_i \lambda_i = 1$ and $\sum_i \lambda_i \mathbf{y}^i = \mathbf{x}$.

Definition

The *convex hull* of a set X of points is the set of all convex combinations of the points in X.

Definition

A point $\mathbf{x}$ in a convex set S is an *extreme point* of S if there exist no $\mathbf{y}, \mathbf{z} \in S$ and $\lambda \in (0, 1)$ such that $\mathbf{x} = \lambda \mathbf{y} + (1 - \lambda)\mathbf{z}$.

- The extreme points of a polytope are called *vertices*.

Lemma

A bounded polytope is the convex hull of its vertices.

Basic facts on polytopes

Corollary

On a bounded polytope, the maximum value of a linear function can always be attained on a vertex.

- In a linear programming the optimal solution that is a vertex is called a *basic feasible solution*.

Basic facts on polytopes

Corollary

On a bounded polytope, the maximum value of a linear function can always be attained on a vertex.

- In a linear programming the optimal solution that is a vertex is called a *basic feasible solution*.

Lemma (Characterizing vertices)

A point $\mathbf{v}$ *in a polytope* $\{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ *is a vertex if and only if*

1. $\mathbf{A}\mathbf{v} \leq \mathbf{b}$;
2. *If* $i_1, \ldots, i_k$ *index the constraints that are tight at* $\mathbf{v}$, *then* $\mathbf{A}_{i_1}, \cdots, \mathbf{A}_{i_k}$ *has full rank, where* $\mathbf{A}_i$ *denotes the* i*-th row of* $\mathbf{A}$.

The Simplex Algorithm

- The simplex algorithm (Dantzig, 1940's): Start from a vertex of the polytope; repeat until failure: find a neighboring vertex with strictly larger objective value, and move to that vertex; or find an adjacent edge along which the objective function gets unbounded.

The Simplex Algorithm

- The simplex algorithm (Dantzig, 1940's): Start from a vertex of the polytope; repeat until failure: find a neighboring vertex with strictly larger objective value, and move to that vertex; or find an adjacent edge along which the objective function gets unbounded.
- Moving from a vertex to an adjacent one is called a *pivot operation*.

The Simplex Algorithm

- The simplex algorithm (Dantzig, 1940's): Start from a vertex of the polytope; repeat until failure: find a neighboring vertex with strictly larger objective value, and move to that vertex; or find an adjacent edge along which the objective function gets unbounded.
- Moving from a vertex to an adjacent one is called a *pivot operation*.

Theorem

The Simplex algorithm terminates in a finite number of steps. When it does, it finds an optimal solution to the linear program.

The Simplex Algorithm

- The simplex algorithm (Dantzig, 1940's): Start from a vertex of the polytope; repeat until failure: find a neighboring vertex with strictly larger objective value, and move to that vertex; or find an adjacent edge along which the objective function gets unbounded.
- Moving from a vertex to an adjacent one is called a *pivot operation*.

Theorem

The Simplex algorithm terminates in a finite number of steps. When it does, it finds an optimal solution to the linear program.

- The running time of the Simplex: exponential time in the worst case, and close to linear time "in practice".

The Simplex Algorithm

- The simplex algorithm (Dantzig, 1940's): Start from a vertex of the polytope; repeat until failure: find a neighboring vertex with strictly larger objective value, and move to that vertex; or find an adjacent edge along which the objective function gets unbounded.
- Moving from a vertex to an adjacent one is called a *pivot operation*.

Theorem

The Simplex algorithm terminates in a finite number of steps. When it does, it finds an optimal solution to the linear program.

- The running time of the Simplex: exponential time in the worst case, and close to linear time "in practice".
- Polynomial time algorithms are known for linear programs:
  - Ellipsoid method (Khachiyan, first polytime algorithm);
  - Interior point method (e.g. Karmarkar's algorithm, runtime $\tilde{O}(n^{3.5}L)$.

Duality

Given a linear program,

$$\max_{\mathbf{x}} \quad \mathbf{c}^T \cdot \mathbf{x}$$
$$\text{s.t. } \mathbf{A}\mathbf{x} \leq \mathbf{b},$$
$$\mathbf{x} \geq 0;$$

is there a way to obtain upper bounds on the objective?

Duality

Given a linear program,

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{c}^{\mathsf{T}} \cdot \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq 0; \end{aligned}$$

is there a way to obtain upper bounds on the objective?

The dual program of the linear program above is

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{b}^{\mathsf{T}} \cdot \mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}^{\mathsf{T}}\mathbf{y} \geq \mathbf{c}, \\ & \mathbf{y} \geq 0. \end{aligned}$$

Duality

Given a linear program,

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{c}^{\mathsf{T}} \cdot \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq 0; \end{aligned}$$

is there a way to obtain upper bounds on the objective?
The dual program of the linear program above is

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{b}^{\mathsf{T}} \cdot \mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}^{\mathsf{T}}\mathbf{y} \geq \mathbf{c}, \\ & \mathbf{y} \geq 0. \end{aligned}$$

Theorem (Strong Duality Theorem)

If both the primal and the dual programs are feasible, the two have the same optimal value.

Example: `Kőnig-Egerváry Theorem`: In a bipartite graph, the size of a maximum matching is equal to the size of a minimum vertex cover.

Example: `Kőnig-Egerváry Theorem`: In a bipartite graph, the size of a maximum matching is equal to the size of a minimum vertex cover.
Integer Programming is NP-hard!

Example: `Kőnig-Egerváry Theorem`: In a bipartite graph, the size of a maximum matching is equal to the size of a minimum vertex cover.
Integer Programming is NP-hard!

Corollary (Complementary Slackness)

Let $\mathbf{y}^*$ *be an optimal solution to the dual program. Then*

- *If* $y_i > 0$*, then the i-th constraint is tight in any optimal solution to the primal program.*
- *If there is an optimal solution to the primal program for which the i-th constraint is not tight, then the* $y_i = 0$*.*

Example: `Kőnig-Egerváry Theorem`: In a bipartite graph, the size of a maximum matching is equal to the size of a minimum vertex cover.

Integer Programming is NP-hard!

Corollary (Complementary Slackness)

Let $\mathbf{y}^*$ be an optimal solution to the dual program. Then

- *If $y_i > 0$, then the i-th constraint is tight in any optimal solution to the primal program.*
- *If there is an optimal solution to the primal program for which the i-th constraint is not tight, then the $y_i = 0$.*

The dual of the dual program is the primal program.

Example: Kőnig-Egerváry Theorem: In a bipartite graph, the size of a maximum matching is equal to the size of a minimum vertex cover.
Integer Programming is NP-hard!

Corollary (Complementary Slackness)

Let $\mathbf{y}^*$ *be an optimal solution to the dual program. Then*

- *If* $y_i > 0$*, then the i-th constraint is tight in any optimal solution to the primal program.*
- *If there is an optimal solution to the primal program for which the i-th constraint is not tight, then the* $y_i = 0$.

The dual of the dual program is the primal program.

Corollary

If $\mathbf{x}$ *and* $\mathbf{y}$ *are feasible solutions to a linear program and its dual, respectively, and if* $\mathbf{x}$ *and* $\mathbf{y}$ *satisfy the two complementary slackness conditions to each other, then both are optimal solutions.*

Basic Game Theory

Definition

A two-player normal form game is specified by a pair of $m \times n$ *payoff matrices*, $(\mathbf{R}, \mathbf{C}$, where m is the number of actions available to the *row* player and n the number of actions available to the *column* player. That is, $R_{i,j}$ and $C_{i,j}$ are the payoffs for the row player and the column player, resptively, if the row player plays action i and the column player plays action j.

Definition

A two-player normal form game is specified by a pair of $m \times n$ *payoff matrices*, $(\mathbf{R}, \mathbf{C}$, where m is the number of actions available to the *row* player and n the number of actions available to the *column* player. That is, $R_{i,j}$ and $C_{i,j}$ are the payoffs for the row player and the column player, resptively, if the row player plays action i and the column player plays action j.

Definition

A *pure Nash equilibrium* is an action pair (i, j) so that both players are best responding to each other, i.e., $R_{i,j} = \max_{i'} R_{i',j}$ and $C_{i,j} = \max_{j'} C_{i,j'}$.

Basic Game Theory

Definition

A two-player normal form game is specified by a pair of $m \times n$ *payoff matrices*, $(\mathbf{R}, \mathbf{C}$, where m is the number of actions available to the *row* player and n the number of actions available to the *column* player. That is, $R_{i,j}$ and $C_{i,j}$ are the payoffs for the row player and the column player, resptively, if the row player plays action i and the column player plays action j.

Definition

A *pure Nash equilibrium* is an action pair (i, j) so that both players are best responding to each other, i.e., $R_{i,j} = \max_{i'} R_{i',j}$ and $C_{i,j} = \max_{j'} C_{i,j'}$.

Example

Prisonner's Dilemma.

Basic Game Theory (continued)

Example

Battle of the Sexes.

Basic Game Theory (continued)

Example

Battle of the Sexes.

Example

Matching pennies.

Basic Game Theory (continued)

Example

Battle of the Sexes.

Example

Matching pennies.

- We need to allow players to play *mixed strategies*, i.e., probability distributions over the actions. For the row player, the set of mixed strategies is represented by $\Delta_m := \{\mathbf{x} \in \mathbb{R}^m_{\geq 0} \,|\, \sum_i x_i = 1\}$. Similarly, Δ_n is the set of mixed strategies for the column player.

Basic Game Theory (continued)

Example

Battle of the Sexes.

Example

Matching pennies.

- We need to allow players to play *mixed strategies*, i.e., probability distributions over the actions. For the row player, the set of mixed strategies is represented by $\Delta_m := \{\mathbf{x} \in \mathbb{R}^m_{\geq 0} | \sum_i x_i = 1\}$. Similarly, Δ_n is the set of mixed strategies for the column player.
- The *expected* payoff for the row player when she plays $\mathbf{x} \in \Delta_m$ and her opponent plays $\mathbf{y} \in \Delta_n$ is $\mathbf{x}^\mathsf{T}\mathbf{R}\mathbf{y}$. Similarly, the column player's expected payoff is $\mathbf{x}^\mathsf{T}\mathbf{C}\mathbf{y}$.

Basic Game Theory (continued)

Definition

A pair of strategy $(\mathbf{x}, \mathbf{y}) \in \Delta_m \times \Delta_n$ is a *Nash equilibrium* if

$$\mathbf{xRy} \geq \mathbf{x'Ry}, \quad \forall \mathbf{x'} \in \Delta_m;$$
$$\mathbf{xCy} \geq \mathbf{xCy'}, \quad \forall \mathbf{y'} \in \Delta_n.$$

Basic Game Theory (continued)

Definition

A pair of strategy $(\mathbf{x}, \mathbf{y}) \in \Delta_m \times \Delta_n$ is a *Nash equilibrium* if

$$\mathbf{x}\mathbf{R}\mathbf{y} \geq \mathbf{x}'\mathbf{R}\mathbf{y}, \quad \forall \mathbf{x}' \in \Delta_m;$$
$$\mathbf{x}\mathbf{C}\mathbf{y} \geq \mathbf{x}\mathbf{C}\mathbf{y}', \quad \forall \mathbf{y}' \in \Delta_n.$$

Definition

A two player game is *zero sum* if $\mathbf{R} + \mathbf{C} = 0$.

Basic Game Theory (continued)

Definition

A pair of strategy $(\mathbf{x}, \mathbf{y}) \in \Delta_m \times \Delta_n$ is a *Nash equilibrium* if

$$\mathbf{xRy} \geq \mathbf{x'Ry}, \quad \forall \mathbf{x'} \in \Delta_m;$$
$$\mathbf{xCy} \geq \mathbf{xCy'}, \quad \forall \mathbf{y'} \in \Delta_n.$$

Definition

A two player game is *zero sum* if $\mathbf{R} + \mathbf{C} = 0$.

Theorem (von Neumann)

Any two-player zero sum game has a Nash equilibrium.

Basic Game Theory (continued)

Definition

A pair of strategy $(\mathbf{x}, \mathbf{y}) \in \Delta_m \times \Delta_n$ is a *Nash equilibrium* if

$$\mathbf{xRy} \geq \mathbf{x'Ry}, \quad \forall \mathbf{x'} \in \Delta_m;$$
$$\mathbf{xCy} \geq \mathbf{xCy'}, \quad \forall \mathbf{y'} \in \Delta_n.$$

Definition

A two player game is *zero sum* if $\mathbf{R} + \mathbf{C} = 0$.

Theorem (von Neumann)

Any two-player zero sum game has a Nash equilibrium.

Theorem (Nash)

Any n-player finite game has a Nash equilibrium.

Proof of von Neumann's min-max theorem

A linear program that computes a lower bound on the row player's payoff:

$$\begin{aligned} &\max\ P \\ \text{s.t. } &\sum_j R_{ij} x_j \geq P, \quad i = 1, 2, \ldots, n; \\ &\sum_j x_j = 1; \\ &x_j \geq 0, j = 1, 2, \ldots, m. \end{aligned}$$

Proof of von Neumann's min-max theorem (continued)

Another linear program gives an upper bound on the row player's payoff (by a similar argument on the column player):

$$
\begin{aligned}
& \min \; Q \\
\text{s.t.} \; & \sum_i R_{ij} y_i \leq Q, \quad j = 1, 2, \ldots, m; \\
& \sum_i y_i = 1; \\
& y_i \geq 0, i = 1, 2, \ldots, n.
\end{aligned}
$$

Proof of von Neumann's min-max theorem (continued)

Another linear program gives an upper bound on the row player's payoff (by a similar argument on the column player):

$$\begin{aligned} & \min \ Q \\ \text{s.t.} \ & \sum_i R_{ij} y_i \leq Q, \quad j = 1, 2, \ldots, m; \\ & \sum_i y_i = 1; \\ & y_i \geq 0, i = 1, 2, \ldots, n. \end{aligned}$$

The two programs are dual for each other. By Strong Duality Theorem, the upper bound and the lower bound are equal, and therefore the optimal solutions $\mathbf{x}^*$ and $\mathbf{y}^*$ are best responses to each other, i.e., they constitute a Nash equilibrium.