CS498: Algorithmic Engineering

Lecture 2: Simplex & Duality

Chandra Chekuri & Elfarouk Harb

University of Illinois Urbana-Champaign

01/22/2026

Outline

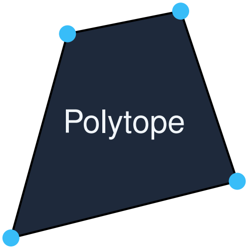
How do Solvers actually work?

Last lecture, we defined the LP:

$$\max c^T x \quad \text{s.t. } Ax \leq b,\ x \geq 0$$

The Fundamental Theorem:

- The optimal solution lies on a **Vertex** (corner).



Naive Algorithm:

- List all vertices. Check objective value. Pick max.

  Problem: A hypercube in n dimensions has 2^n vertices. Too slow.

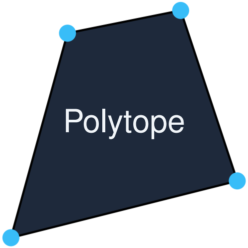
How do Solvers actually work (cont'd)?

Last lecture, we defined the LP:

$$\max c^T x \quad \text{s.t. } Ax \leq b,\ x \geq 0$$

The Fundamental Theorem:

- The optimal solution lies on a **Vertex** (corner).



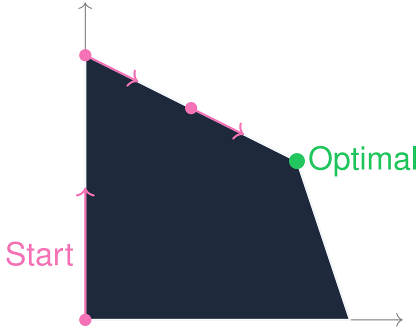
Key Insight:

- Vertices are connected by **edges**.
- We can "walk" from vertex to vertex improving our objective.

The Simplex Intuition (Hill Climbing)

Algorithm (Dantzig, 1947):

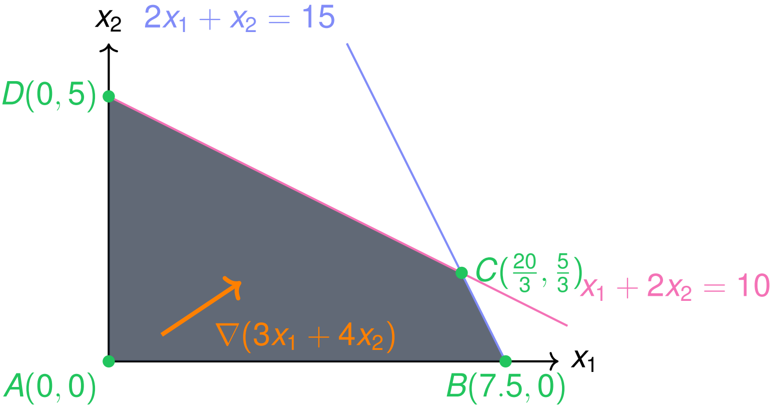
1. **Start** at any vertex (usually Origin).
2. **Look** along edges connected to current vertex.
3. **Is a neighbor better?**
   - **Yes:** Move there (Pivot). Go to 2.
   - **No:** You are done. (Local max = Global max).



"Vertex Hopping"

Geometric View: A 2D Example

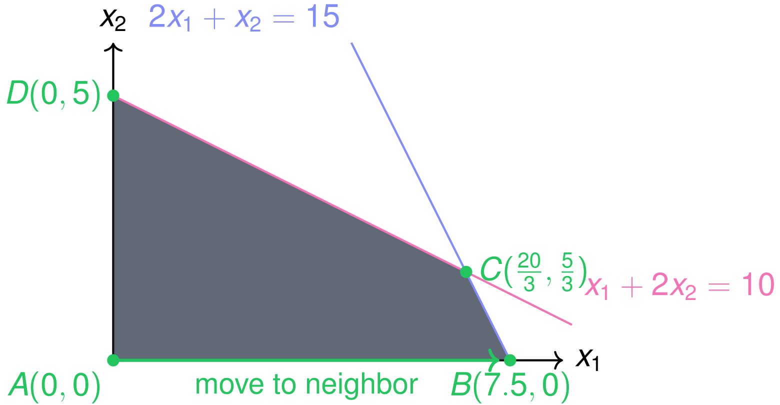
Example: $\max 3x_1 + 4x_2$ subject to: $x_1 + 2x_2 \leq 10, 2x_1 + x_2 \leq 15, x_1, x_2 \geq 0$.



Simplex: pick a basic feasible solution (a vertex), $A(0,0)$ with $z = 3x_1 + 4x_2 = 0$.

Geometric View: Simplex Walk (Step 1)

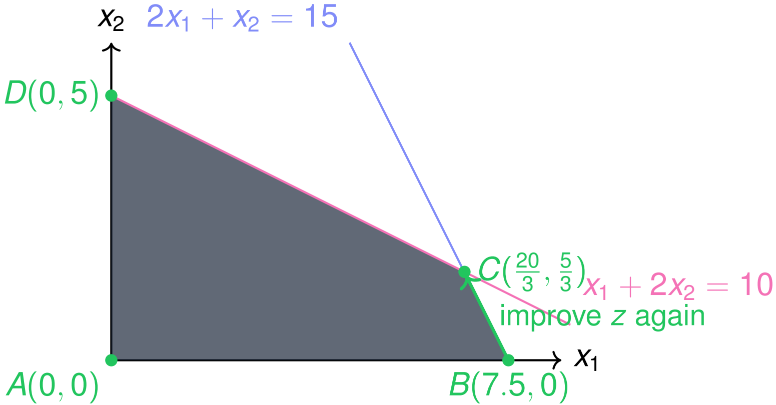
Start at $A = (0, 0)$, $z = 0$. Check neighboring vertices on the polytope.



At $B = (7.5, 0)$: $z = 3 \cdot 7.5 + 4 \cdot 0 = 22.5 > 0$, so simplex pivots from A to B.

Geometric View: Simplex Walk (Step 2)

Now at $B = (7.5, 0)$ with $z = 22.5$. Check its neighbors on the polytope.



At $C = \left(\frac{20}{3}, \frac{5}{3}\right)$, $z = 3x_1 + 4x_2 = 3 \cdot \frac{20}{3} + 4 \cdot \frac{5}{3} = \frac{80}{3} \approx 26.7$

No neighbor improves z further $\Rightarrow$ simplex stops: C is optimal.

Vertices and Basic Feasible Solutions

Setup: Consider the LP in two variables:

$$\begin{aligned} \max \quad & z = 3x_1 + 4x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 \le 10, \\ & 2x_1 + x_2 \le 15, \\ & x_1 \ge 0, \; x_2 \ge 0. \end{aligned}$$

- The constraints define a **feasible region** (a polygon).
- A **vertex** is a “corner” of this region.
- Geometrically: at a vertex, we are **on the boundary** of enough constraints to pin down a single point.

Key idea: In 2D, a vertex is the intersection of **2 boundaries**.

From Geometry to Algebra

Boundaries = constraints tight as equalities.

In our example, boundaries include:

$$x_1 = 0, \quad x_2 = 0, \quad x_1 + 2x_2 = 10, \quad 2x_1 + x_2 = 15.$$

- To get a vertex in 2D, we choose **2** boundaries and solve them as a system of **2 equations**.
- This gives the coordinates of that corner point (**if it is feasible**).

In general:

- In an LP with m variables, a vertex comes from m **tight constraints**.
- Solving those m equalities gives a single point.

Examples of Vertices as Intersections

Vertex $A = (0, 0)$

- Tight: $x_1 = 0$ and $x_2 = 0$.

Vertex $B = (7.5, 0)$

- Tight: $x_2 = 0$ and $2x_1 + x_2 = 15 \implies x_1 = 7.5, x_2 = 0$.

Vertex $D = (0, 5)$

- Tight: $x_1 = 0$ and $x_1 + 2x_2 = 10 \implies x_1 = 0, x_2 = 5$.

Vertex C

- Tight: $x_1 + 2x_2 = 10$ and $2x_1 + x_2 = 15 \implies x_1 = \frac{20}{3}, x_2 = \frac{5}{3}$.

Infeasible Pair 1

- Tight: $x_1 = 0$ and $2x_1 + x_2 = 15 \implies x_1 = 0, x_2 = 15$.

Infeasible Pair 2

- Tight: $x_2 = 0$ and $x_1 + 2x_2 = 10 \implies x_1 = 10, x_2 = 0$.

Basic Feasible Solutions (Algebraic View)

Basic Feasible Solution

A point is a **basic feasible solution (BFS)** if:

- It is **feasible** (satisfies all constraints).
- It is a **vertex** of the feasible region:
  - in m dimensions: it lies at the intersection of m **tight constraints** (including $x_j \geq 0$), and those m equalities have a unique solution.

Key facts:

- Every vertex $\iff$ a BFS.
- Simplex method moves from one BFS (vertex) to another, improving the objective each time.

Pivoting from a Vertex: Setup at D

Current vertex:

$$D = (0, 5), \quad z(D) = 3 \cdot 0 + 4 \cdot 5 = 20.$$

Tight constraints at D:

$$x_1 = 0, \qquad x_1 + 2x_2 = 10.$$

These two equalities define D.

- To pivot, we relax *one* of them (this gives an edge),
- move along that edge,
- and stop when a *new* constraint becomes tight.

Goal: choose the edge that **increases** z.

Step 1: Which Edge Improves the Objective?

Edge 1: keep $x_1 = 0$, relax $x_1 + 2x_2 = 10$ to $x_1 + 2x_2 \leq 10 \implies x_2 \leq 5$.
On this edge:

$$x_1 = 0, \quad x_2 = 5 - t, \quad t \geq 0 \quad \text{(moving down from } D\text{)}.$$

Objective:

$$z(t) = 3 \cdot 0 + 4(5 - t) = 20 - 4t.$$

Slope: z **decreases** as t increases.

Edge 2: keep $x_1 + 2x_2 = 10$, relax $x_1 = 0$ to $x_1 \geq 0$.
On this edge:

$$x_1 = t, \quad x_2 = \frac{10 - t}{2}, \quad t \geq 0 \quad \text{(moving right from } D\text{)}.$$

Objective: $z(t) = 3t + 4 \cdot \frac{10-t}{2} = 3t + 20 - 2t = 20 + t$. Slope: z **increases** as $t \uparrow$

So we pivot along **Edge 2.** (Entering variable: x_1.)

Step 2: Which Constraint Becomes Tight First? (Ratio Test)

We move along Edge 2:

$$x_1 = t, \qquad x_2 = \frac{10 - t}{2}, \qquad t \geq 0.$$

Plug this into the other constraints and see when they hit equality.

Constraint $2x_1 + x_2 \leq 15$:

$$2t + \frac{10 - t}{2} = 15 \;\Rightarrow\; \frac{4t + 10 - t}{2} = 15 \;\Rightarrow\; 3t + 10 = 30 \;\Rightarrow\; t = \frac{10}{3}.$$

Constraint $x_2 \geq 0$: $\frac{10-t}{2} = 0 \;\Rightarrow\; t = 10.$
Constraint $x_1 \geq 0$ is fine for all $t \geq 0$.

Smallest nonnegative t is $\dfrac{10}{3}$. So the next constraint to become tight is:

$$2x_1 + x_2 = 15.$$

Step 3: New Vertex = Solve a 2×2 System

At the new vertex, the tight constraints are:

$$x_1 + 2x_2 = 10,$$
$$2x_1 + x_2 = 15.$$

Solve:

$$x_1 = \frac{20}{3}, \quad x_2 = \frac{5}{3}.$$

New BFS (after the pivot):

$$C = \left(\frac{20}{3}, \frac{5}{3}\right).$$

A pivot is literally: change one equation, solve a tiny linear system.

Repeat the Same Three Steps at the New Vertex

At C there are again two tight constraints (two equations). To pivot again, we repeat the same pattern:

1. **Choose an edge:** Relax one tight constraint, keep the other. Parametrize the edge, compute $z(t)$, pick the edge with increasing z.
2. **Ratio test:** Plug the parametrization into all constraints, solve for t, and find which constraint hits equality first.
3. **New vertex:** Replace the old constraint with this new tight constraint, solve the resulting 2×2 system.

In m dimensions, it's the same idea: solve m equations, relax one, ratio test, solve a new $m \times m$ system.

Degeneracy: When More Than Two Constraints Meet

In 2 variables, a vertex normally comes from **two** tight constraints.

Sometimes, **more than two** constraints happen to be tight at the same point:
$x_1 + x_2 \leq 1, x_1 \leq 0.5, x_2 \leq 0.5$.

At the point $(0.5, 0.5)$ all three become equalities. **What does this mean for simplex?**

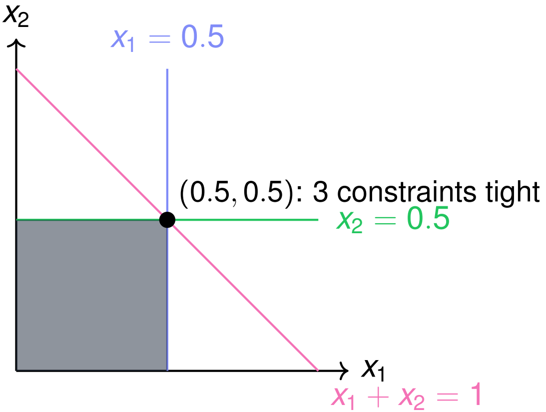
- A vertex must be described using exactly 2 tight equations.
- But here we have **3 different pairs** that all solve to the *same point*.
- These different pairs lead to **different parameterizations** of the same vertex.

This situation is called **degeneracy**: the vertex is the unique solution of more equations than needed.

Degeneracy: Visualizing the Point $(0.5, 0.5)$

Constraints:

$$x_1 + x_2 \leq 1, \quad x_1 \leq 0.5, \quad x_2 \leq 0.5, \quad x_1 \geq 0, \quad x_2 \geq 0$$



At $(0.5, 0.5)$, **all three** constraints are tight, but it is still just *one* vertex $\Rightarrow$ **degeneracy**.

Why Degeneracy Matters for Pivoting

At a degenerate vertex (e.g. $(0.5, 0.5)$), when simplex tries to pivot:

Step 1: Relax one tight constraint to form an edge. But several tight constraints remain; depending on which two you pick, you may be describing the *same* point.

Step 2: Ratio test. Because the point already satisfies more than two equations, the "new" constraint might hit equality immediately:

$$t = 0.$$

Step 3: Solve the new linear system. You get the *same* point again:

$$(x_1, x_2) = (0.5, 0.5).$$

This is a **degenerate pivot**: simplex changes which two equations define the vertex but does *not move* in space.

If this happens repeatedly, simplex can "stall" or even cycle (cycling is fixed in practice by Bland's rule and similar tie-breakers).

Why Simplex Terminates

Convergence Argument:

1. The feasible region is a **polytope** (bounded polyhedron)
2. A polytope has **finitely many vertices** and **edges**
3. Each pivot moves to an adjacent vertex with strictly better objective (in non-degenerate case)
4. We never revisit a vertex (objective strictly improves)
5. Must reach optimal vertex in finite steps

Upper Bound on Iterations

With m constraints and n variables:

- Maximum $\binom{m}{n}$ possible bases (assuming all constraints are equalities)
- In practice: much fewer iterations needed
- Typical: $O(m)$ to $O(m \log n)$ iterations

Implementation: Not How We Did It by Hand

The way we computed pivots manually (parametrizing edges, solving small systems) is **not** how simplex is implemented in practice (although you can certainly implement it that way!).

Real implementations use a compact **table of numbers** (called a tableau) that lets the algorithm:

- test entering and leaving variables instantly,
- and update everything with fast row operations.

Same exact ideas, but more efficient.

We Won't Implement Simplex Ourselves

Although tableau methods are standard, **we won't code simplex by hand**.

In this course we use **Gurobi**, which already includes:

- simplex methods,
- interior-point methods,
- and other state-of-the-art solvers.

Our focus is on the modelling. Gurobi handles the actual solver implementation.

Motivation: Finding Upper Bounds

Example LP:

$$\max 4x_1 + x_2 + 3x_3 \quad \text{s.t.} \quad x_1 + 4x_2 \leq 1, \quad 3x_1 - x_2 + x_3 \leq 3, \quad x_1, x_2, x_3 \geq 0$$

Finding Lower Bounds (Easy):

- Try $(x_1, x_2, x_3) = (1, 0, 0)$: objective $= 4$. So $Z \geq 4$.
- Try $(x_1, x_2, x_3) = (0, 0, 3)$: objective $= 9$. So $Z \geq 9$.

Finding Upper Bounds (Harder): Let's multiply constraint 1 by 2 and constraint 2 by 3:

$$\begin{aligned} 2(x_1 + 4x_2) &\leq 2 \cdot 1 \\ + \quad 3(3x_1 - x_2 + x_3) &\leq 3 \cdot 3 \end{aligned}$$

Sum them: $11x_1 + 5x_2 + 3x_3 \leq 11$. Since $x_1, x_2, x_3 \geq 0$:

$$4x_1 + x_2 + 3x_3 \leq 11x_1 + 5x_2 + 3x_3 \leq 11$$

So $Z \leq 11$. We've bounded the optimum: $9 \leq Z \leq 11$.

Getting the Tightest Upper Bound

Question: Can we find *better* multipliers? Let $y_1, y_2 \geq 0$ be multipliers for constraints 1 and 2:

$$\begin{aligned} & y_1(x_1 + 4x_2) \leq y_1 \cdot 1 \\ + \quad & y_2(3x_1 - x_2 + x_3) \leq y_2 \cdot 3 \end{aligned}$$

Sum: $(y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2x_3 \leq y_1 + 3y_2$

For this to upper bound $4x_1 + x_2 + 3x_3$, we need:

$$\begin{aligned} 4 &\leq y_1 + 3y_2 && \text{(coefficient of } x_1\text{)} \\ 1 &\leq 4y_1 - y_2 && \text{(coefficient of } x_2\text{)} \\ 3 &\leq y_2 && \text{(coefficient of } x_3\text{)} \end{aligned}$$

Then: $4x_1 + x_2 + 3x_3 \leq (y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2x_3 \leq y_1 + 3y_2$

Goal: Minimize $y_1 + 3y_2$ subject to those constraints on y!

The Dual Problem Emerges

We naturally arrived at:

Primal (P)

$$\begin{aligned} \max \quad & 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Dual (D)

$$\begin{aligned} \min \quad & y_1 + 3y_2 \\ \text{s.t.} \quad & y_1 + 3y_2 \geq 4 \\ & 4y_1 - y_2 \geq 1 \\ & y_2 \geq 3 \\ & y_1, y_2 \geq 0 \end{aligned}$$

Key Insight:

- Every feasible y gives an upper bound on the primal optimum
- The dual finds the *best* (tightest) upper bound
- This is **duality theory**!

General Duality: Matrix Form

Primal (P)	Dual (D)
$\max c^T x$	$\min b^T y$
$Ax \leq b$	$A^T y \geq c$
$x \geq 0$	$y \geq 0$

Conversion Rules:

- Each primal **constraint** $\leftrightarrow$ one dual **variable**
- Each primal **variable** $\leftrightarrow$ one dual **constraint**
- Primal $\max \leftrightarrow$ Dual $\min$
- Constraint matrix transposes: $A \rightarrow A^T$

Duality Rules: The Full Picture

Primal (max $c^\top x$)	**Dual (min $b^\top y$)**
$\sum_j a_{ij} x_j \leq b_i$	$y_i \geq 0$
$\sum_j a_{ij} x_j \geq b_i$	$y_i \leq 0$
$\sum_j a_{ij} x_j = b_i$	y_i free
$x_j \geq 0$	$\sum_i y_i a_{ij} \geq c_j$
$x_j \leq 0$	$\sum_i y_i a_{ij} \leq c_j$
x_j free	$\sum_i y_i a_{ij} = c_j$

Key symmetry: the dual of the dual is your original primal.

Example 1: Building the Dual Step by Step

Primal:

$$\begin{aligned} \max\ & 5x_1 + 3x_2 \\ \text{s.t. } & 2x_1 + x_2 \leq 8 \\ & x_1 + 3x_2 \leq 9 \\ & x_1, x_2 \geq 0 \end{aligned}$$

We now construct the dual using the duality rules.

Example 1: Step 1 — Dual Variables

Primal constraint types $\Rightarrow$ dual variable signs

$$2x_1 + x_2 \le 8 \Rightarrow y_1 \ge 0,$$

$$x_1 + 3x_2 \le 9 \Rightarrow y_2 \ge 0.$$

Primal (for reference)

$$\begin{aligned} \max\ & 5x_1 + 3x_2 \\ \text{s.t. } & 2x_1 + x_2 \le 8 \\ & x_1 + 3x_2 \le 9 \\ & x_1, x_2 \ge 0 \end{aligned}$$

Matrix summary

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix},\ b = \begin{pmatrix} 8 \\ 9 \end{pmatrix},\ c = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

Duality rules

Primal	Dual
$\max c^\top x$	$\min b^\top y$
$a_i^\top x \le b_i$	$y_i \ge 0$
$a_i^\top x \ge b_i$	$y_i \le 0$
$a_i^\top x = b_i$	y_i free

Example 1: Step 2 — Dual Objective

Objective direction:

$$\max \implies \min .$$

Dual objective uses the RHS values:

$$b = (8,\ 9).$$

So the dual objective is

$$\min\ 8y_1 + 9y_2.$$

Primal (for reference)

$$\begin{aligned} \max\ & 5x_1 + 3x_2 \\ \text{s.t. } & 2x_1 + x_2 \le 8 \\ & x_1 + 3x_2 \le 9 \\ & x_1, x_2 \ge 0 \end{aligned}$$

Matrix summary

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix},\ b = \begin{pmatrix} 8 \\ 9 \end{pmatrix},\ c = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

Duality rules

Primal	Dual
$\max c^\top x$	$\min b^\top y$
$a_i^\top x \le b_i$	$y_i \ge 0$
$a_i^\top x \ge b_i$	$y_i \le 0$
$a_i^\top x = b_i$	y_i free

Example 1: Step 3 — Dual Constraints

Primal variable signs $\Rightarrow$ dual constraints

$x_1 \geq 0$:

$$2y_1 + y_2 \geq 5$$

$x_2 \geq 0$:

$$y_1 + 3y_2 \geq 3$$

Primal (for reference)

$$\begin{aligned} \max\ & 5x_1 + 3x_2 \\ \text{s.t. } & 2x_1 + x_2 \leq 8 \\ & x_1 + 3x_2 \leq 9 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Matrix summary

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix},\ b = \begin{pmatrix} 8 \\ 9 \end{pmatrix},\ c = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

Duality rules

Primal	Dual
$\max c^\top x$	$\min b^\top y$
$x_j \geq 0$	$(A^\top y)_j \geq c_j$
$x_j \leq 0$	$(A^\top y)_j \leq c_j$
x_j free	$(A^\top y)_j = c_j$

Example 1: Final Dual

$$\boxed{\begin{aligned} \min\ & 8y_1 + 9y_2 \\ \text{s.t. } & 2y_1 + y_2 \geq 5 \\ & y_1 + 3y_2 \geq 3 \\ & y_1, y_2 \geq 0 \end{aligned}}$$

Every coefficient comes directly from the primal via the duality rules.

Example 2: Mixed Constraints — Dual Variables

Constraint types $\Rightarrow$ dual variable signs

$$x_1 + x_2 + x_3 = 10 \Rightarrow y_1 \text{ free}$$

$$2x_1 + x_2 \geq 5 \Rightarrow y_2 \leq 0$$

Dual objective:

$$\min \; 10y_1 + 5y_2.$$

Primal:

$$\begin{aligned} \max\ & 4x_1 + 2x_2 + x_3 \\ \text{s.t. } & x_1 + x_2 + x_3 = 10 \\ & 2x_1 + x_2 \geq 5 \\ & x_1, x_2 \geq 0, \;\; x_3 \text{ free} \end{aligned}$$

Primal summary

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \end{pmatrix}, \; b = \begin{pmatrix} 10 \\ 5 \end{pmatrix}, \; c = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix}$$

Duality rules

Primal	Dual
$\max c^\top x$	$\min b^\top y$
$a_i^\top x \leq b_i$	$y_i \geq 0$
$a_i^\top x \geq b_i$	$y_i \leq 0$
$a_i^\top x = b_i$	y_i free

Example 2: Mixed Constraints — Dual Constraints

Variable types $\Rightarrow$ dual constraints

$x_1 \geq 0$:

$$1 \cdot y_1 + 2 \cdot y_2 \geq 4$$

$x_2 \geq 0$:

$$1 \cdot y_1 + 1 \cdot y_2 \geq 2$$

x_3 free:

$$1 \cdot y_1 + 0 \cdot y_2 = 1$$

Primal:

$$\begin{aligned} \max\ & 4x_1 + 2x_2 + x_3 \\ \text{s.t. } & x_1 + x_2 + x_3 = 10 \\ & 2x_1 + x_2 \geq 5 \\ & x_1, x_2 \geq 0,\ \ x_3 \text{ free} \end{aligned}$$

Primal summary

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \end{pmatrix},\ b = \begin{pmatrix} 10 \\ 5 \end{pmatrix},\ c = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix}$$

Duality rules

Primal	Dual
$\max c^\top x$	$\min b^\top y$
$x_j \geq 0$	$(A^\top y)_j \geq c_j$
$x_j \leq 0$	$(A^\top y)_j \leq c_j$
x_j free	$(A^\top y)_j = c_j$

Example 2: Final Dual

Final dual:

$$\boxed{\begin{aligned} \min 10y_1 + 5y_2 \\ \text{s.t } y_1 + 2y_2 \geq 4, \\ y_1 + y_2 \geq 2, \\ y_1 = 1, \\ y_2 \leq 0 \end{aligned}}$$

Primal:

$$\begin{aligned} \max\ & 4x_1 + 2x_2 + x_3 \\ \text{s.t. } & x_1 + x_2 + x_3 = 10 \\ & 2x_1 + x_2 \geq 5 \\ & x_1, x_2 \geq 0,\ x_3 \text{ free} \end{aligned}$$

Primal summary

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \end{pmatrix},\ b = \begin{pmatrix} 10 \\ 5 \end{pmatrix},\ c = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix}$$

Duality rules

Primal	Dual
$\max c^\top x$	$\min b^\top y$
$x_j \geq 0$	$(A^\top y)_j \geq c_j$
$x_j \leq 0$	$(A^\top y)_j \leq c_j$
x_j free	$(A^\top y)_j = c_j$

Theorems of Duality

1. Weak Duality Theorem

For any feasible primal x and any feasible dual y:

$$c^T x \leq b^T y$$

Primal objective $\leq$ *Dual objective*

Proof: If $Ax \leq b$ and $A^T y \geq c$ with $x, y \geq 0$:

$$c^T x \leq (A^T y)^T x = y^T (Ax) \leq y^T b = b^T y \quad \square$$

Theorems of Duality

2. Strong Duality Theorem

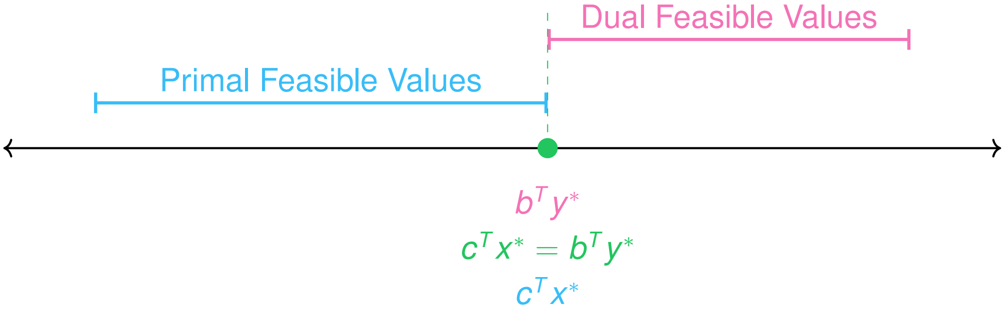
If the Primal has an optimal solution x^*, then the Dual has an optimal solution y^*, and:

$$c^T x^* = b^T y^*$$

*At optimality, the objectives are **equal**—no gap!*

Note: Strong duality proof requires more machinery (Farkas' lemma), but the result is powerful.

Visualizing Weak & Strong Duality



Key Insight:

- Any primal feasible $\leq$ any dual feasible (weak duality)
- At optimum, they meet exactly (strong duality)

Accessing Duals in Gurobi

We can use Gurobi to perform sensitivity analysis automatically.

```python
# ... (Model definition) ...
m.optimize()

print("Optimal Primal (Production):")
for v in m.getVars():
    print(f"{v.VarName}: {v.X}")

print("\nOptimal Dual:")
for c in m.getConstrs():
    # .Pi is the attribute for the Dual Variable (Price)
    print(f"{c.ConstrName}: {c.Pi}")
```

Example: Solving the Primal in Gurobi

Primal (Example 1):

$$\begin{aligned} \max\ & 5x_1 + 3x_2 \\ \text{s.t. } & 2x_1 + x_2 \leq 8 \\ & x_1 + 3x_2 \leq 9 \\ & x_1, x_2 \geq 0 \end{aligned}$$

```python
m = gp.Model()
x1 = m.addVar(lb=0, name="x1")
x2 = m.addVar(lb=0, name="x2")
c1 = m.addConstr(2*x1 + x2 <= 8, name="c1")
c2 = m.addConstr(x1 + 3*x2 <= 9, name="c2")

m.setObjective(5*x1 + 3*x2, gp.GRB.MAXIMIZE)
m.optimize()

print("Optimal primal value:", m.ObjVal)
```

Example: Dual Values and Strong Duality

Dual of Example 1:

$$\begin{aligned} \min\ & 8y_1 + 9y_2 \\ \text{s.t. } & 2y_1 + y_2 \geq 5 \\ & y_1 + 3y_2 \geq 3 \\ & y_1, y_2 \geq 0 \end{aligned}$$

Gurobi gives the dual values as `constraint.Pi`:

```
print("Dual values (shadow prices):")
print("y1 =", c1.Pi)
print("y2 =", c2.Pi)

dual_obj = 8*c1.Pi + 9*c2.Pi
print("Dual objective:", dual_obj)
```

Strong Duality Check

If you run the code, Gurobi returns:

$$x^* = (3,\ 2)$$
$$z_P^* = 21$$

Dual values (from `.Pi`):

$$y_1^* = 2.4, \qquad y_2^* = 0.2$$

Dual objective:

$$8(2.4) + 9(0.2) = 21$$

Primal optimal = Dual optimal.
Strong duality verified!

Summary: What We Learned

The Simplex Algorithm:

- Geometrically: walks from vertex to vertex along edges
- Algebraically: Basic Feasible Solutions (BFS) via pivoting
- Converges because finite vertices, non-revisiting path
- Implemented efficiently via Tableau method

Duality Theory:

- Every LP has a dual that provides upper bounds
- Weak duality: primal $\leq$ dual always
- Strong duality: they meet at optimum (no gap!)
- Conversion rules for mixed constraint types