

L06

Integer Programming and Matching

CS 169 / CS 268
Introduction to Optimization

Ioannis Panageas

Fall 2026

Integer decisions

An LP allows fractional decisions. Some applications require **whole units** or **yes/no choices**.

$$x_i \in \mathbb{Z}_{\geq 0} \quad \text{or} \quad x_i \in \{0, 1\}$$

Examples: how many machines to buy; which projects to fund; which worker to assign to a job.

The objective and constraints can still be linear.

Integer linear programming

$$\begin{array}{ll}\text{maximize} & c^T x \\ \text{subject to} & Ax \leq b, \\ & x \geq 0, \quad x \in \mathbb{Z}^n.\end{array}$$

The integrality requirement changes the feasible set: the line segment between two feasible integer points may leave the set.

$$(0,0), (1,0) \in \mathbb{Z}^2, \quad (1/2,0) \notin \mathbb{Z}^2.$$

A binary model

Three projects have values 6, 4, 5 and costs 4, 3, 2. The budget is 5.

$$y_i = \begin{cases} 1, & \text{fund project } i, \\ 0, & \text{otherwise.} \end{cases}$$

$$\begin{aligned} \max \quad & 6y_1 + 4y_2 + 5y_3 \\ \text{s.t.} \quad & 4y_1 + 3y_2 + 2y_3 \leq 5, \\ & y_1, y_2, y_3 \in \{0, 1\}. \end{aligned}$$

Each feasible binary vector describes a subset of projects.

The LP relaxation

Drop the integer restriction and keep the linear constraints.

$$\text{binary:} \quad y_i \in \{0, 1\} \longrightarrow 0 \leq y_i \leq 1$$

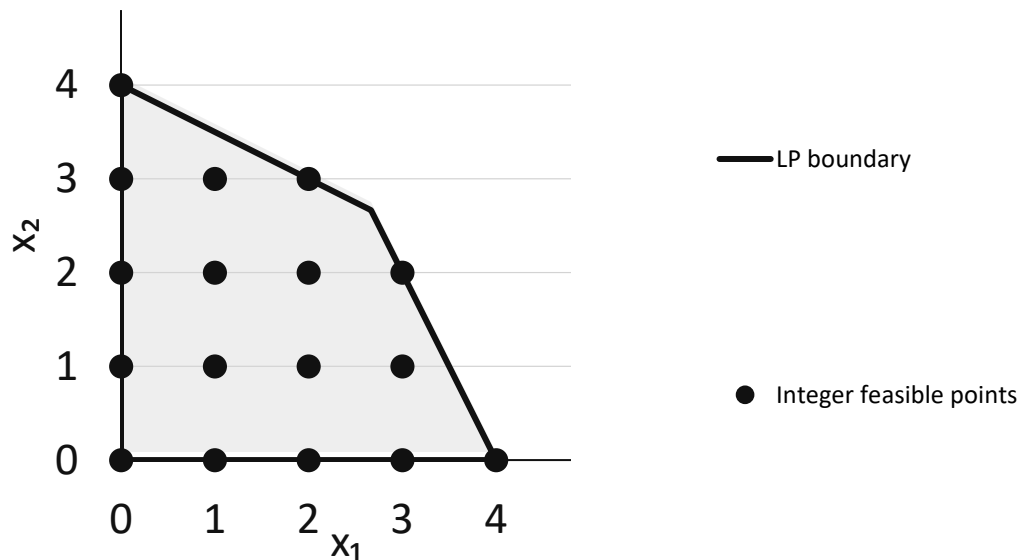
$$F_{\text{IP}} \subseteq F_{\text{LP}}$$

$$\text{maximization:} \quad z_{\text{IP}} \leq z_{\text{LP}}$$

An integer feasible solution gives a lower bound. The LP relaxation gives an upper bound.

A two-variable example

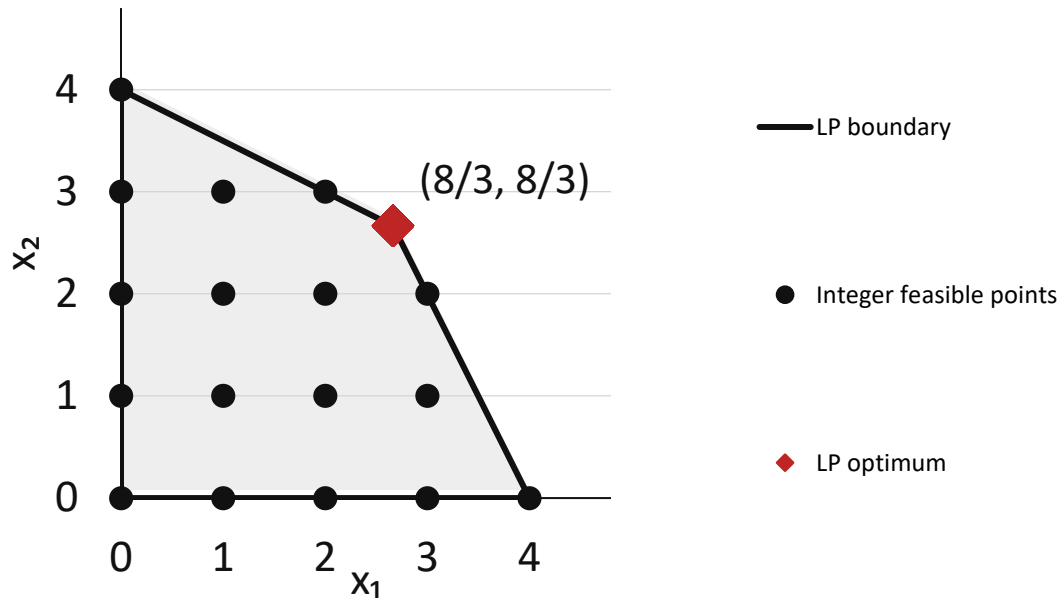
$$\begin{aligned} \max \quad & 4x_1 + 3x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 \leq 8, \quad x_1 + 2x_2 \leq 8, \\ & x_1, x_2 \in \mathbb{Z}_{\geq 0} \end{aligned}$$



The shaded polygon is continuous; only its integer points are feasible for the integer program.

Solve the relaxation

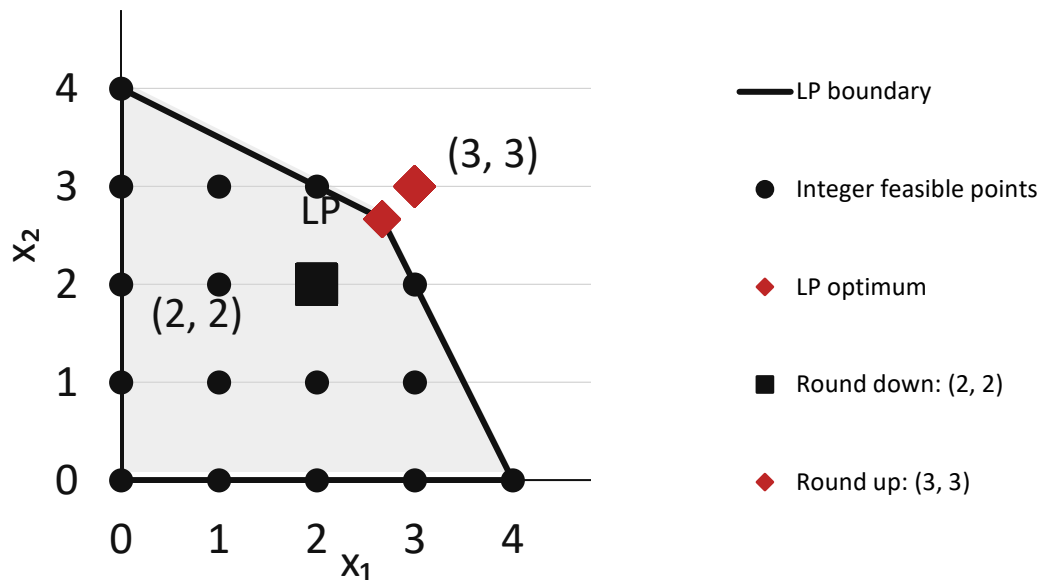
$$x^{\text{LP}} = \left(\frac{8}{3}, \frac{8}{3} \right), \quad z_{\text{LP}} = \frac{56}{3}.$$



The optimal LP vertex is **not an integer point**.

Why rounding can fail

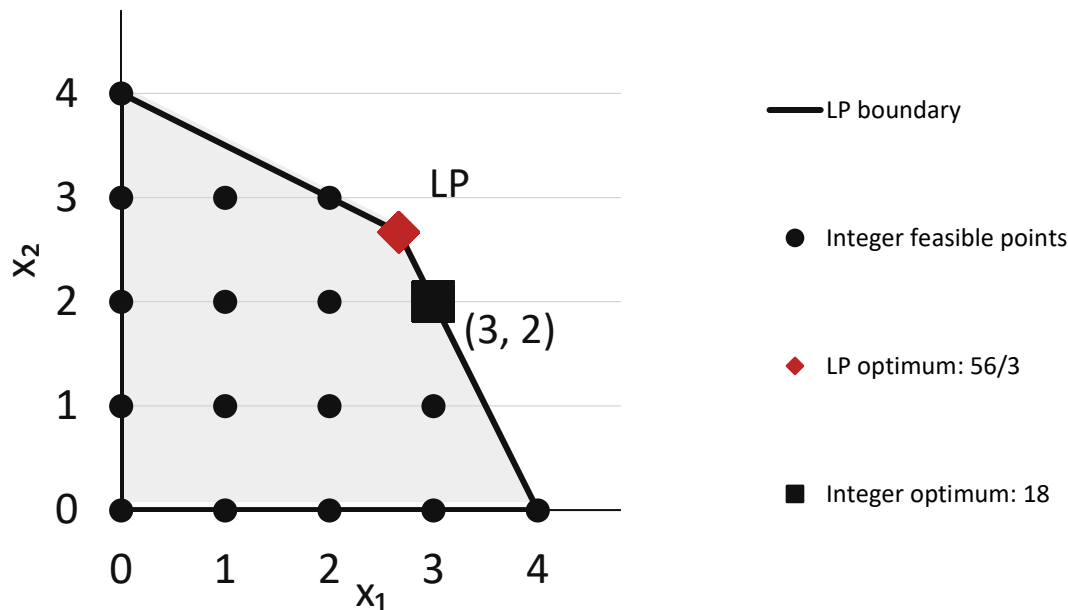
$$\left(\frac{8}{3}, \frac{8}{3}\right) \longrightarrow (3, 3) \quad \text{or} \quad (2, 2)$$



$(3, 3)$ violates both constraints. $(2, 2)$ is feasible, but its value is only 14.

An integer solution and bounds

$$x = (3, 2), \quad 4x_1 + 3x_2 = 18$$



$$\underbrace{18}_{\text{feasible value}} \leq z_{\text{IP}} \leq \underbrace{56/3}_{\text{LP bound}}$$

Branch on a fractional variable

At the LP optimum, $x_1 = 8/3$. Every integer value satisfies one of two alternatives:

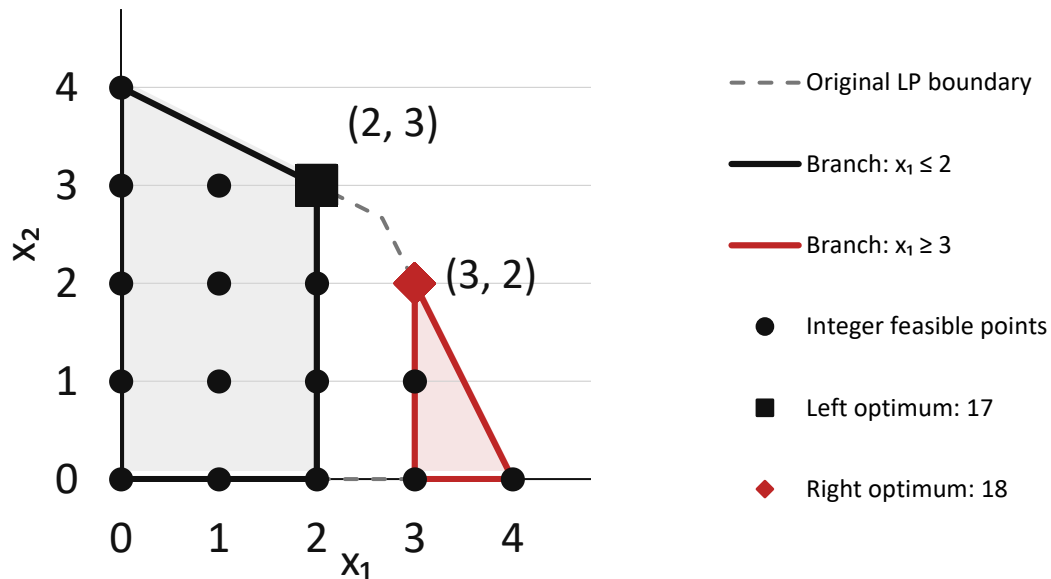
$$x_1 \leq 2 \quad \text{or} \quad x_1 \geq 3.$$

Solve an LP in each branch, adding its extra constraint.

The fractional optimum is removed, but **no integer feasible point is lost**.

The two branches

$$\begin{array}{ll} x_1 \leq 2 & x_1 \geq 3 \\ (2, 3), z = 17 & (3, 2), z = 18 \end{array}$$



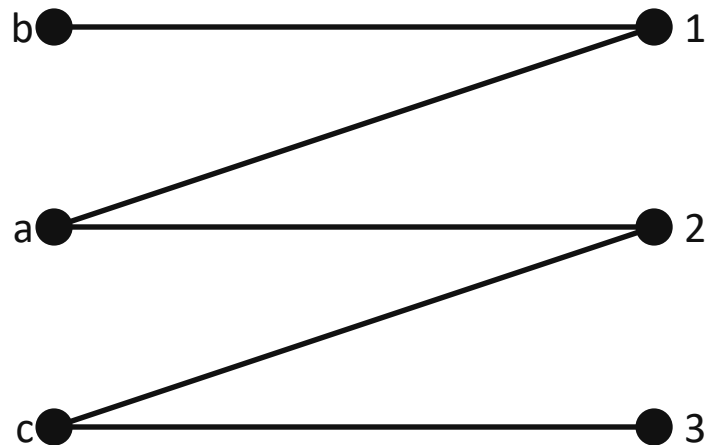
Both branch optima are integer. Therefore the original integer optimum is 18.

Branch-and-bound: the idea

1. Solve the LP relaxation of the current subproblem.
2. If its solution is integer, record the candidate. Otherwise, branch on a fractional coordinate.
3. Discard an infeasible branch, or one whose upper bound cannot beat the best integer value found.
4. Continue until every branch has been resolved or discarded.

A matching problem

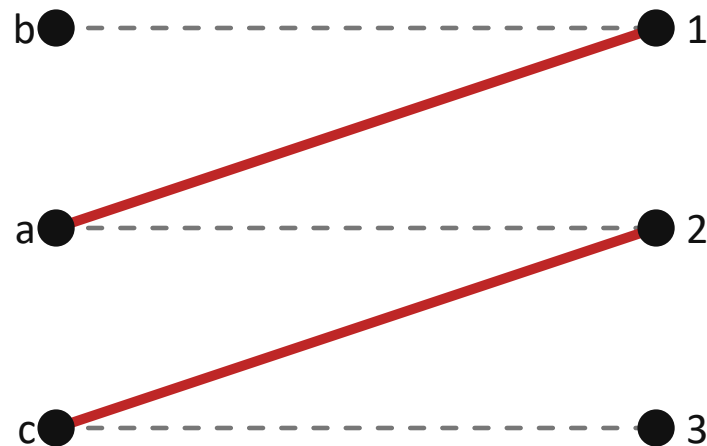
Three workers and three jobs. An edge means that a worker can do that job.



Choose as many assignments as possible. Each worker and each job can be used at most once.

Bipartite graphs and matchings

A graph is **bipartite** if its vertices split into two sets and every edge joins the two sets.



A **matching** is a set of edges with no shared endpoint. Here $M = \{a1, c2\}$ has size 2.

Matching as an integer program

$$x_e = \begin{cases} 1, & e \text{ is selected,} \\ 0, & e \text{ is not selected.} \end{cases}$$

$$\begin{array}{ll} \max & \sum_{e \in E} x_e \\ \text{s.t.} & \sum_{e \in \delta(v)} x_e \leq 1 \quad \text{for every vertex } v, \\ & x_e \in \{0, 1\} \quad \text{for every edge } e. \end{array}$$

$\delta(v)$ denotes the edges incident to vertex v .

Constraints in our example

$$\max \quad x_{a1} + x_{a2} + x_{b1} + x_{c2} + x_{c3}$$

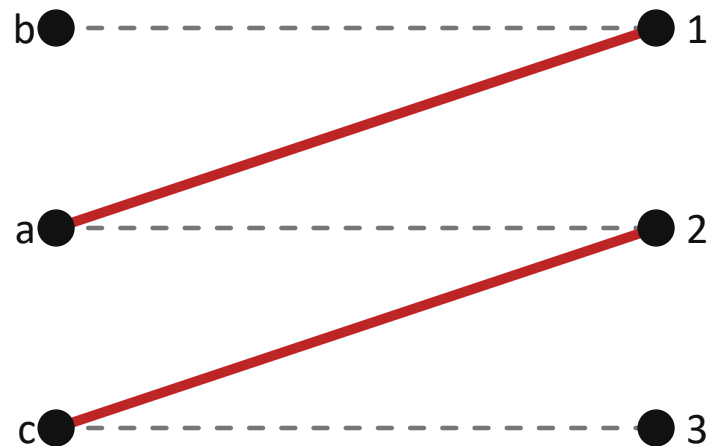
$x_{a1} + x_{a2}$	$\leq$	1	$x_{a1} + x_{b1}$	$\leq$	1
x_{b1}	$\leq$	1	$x_{a2} + x_{c2}$	$\leq$	1
$x_{c2} + x_{c3}$	$\leq$	1	x_{c3}	$\leq$	1

Left: one inequality per worker. Right: one per job.

$$x_{a1}, x_{a2}, x_{b1}, x_{c2}, x_{c3} \in \{0, 1\}.$$

Maximal is not maximum

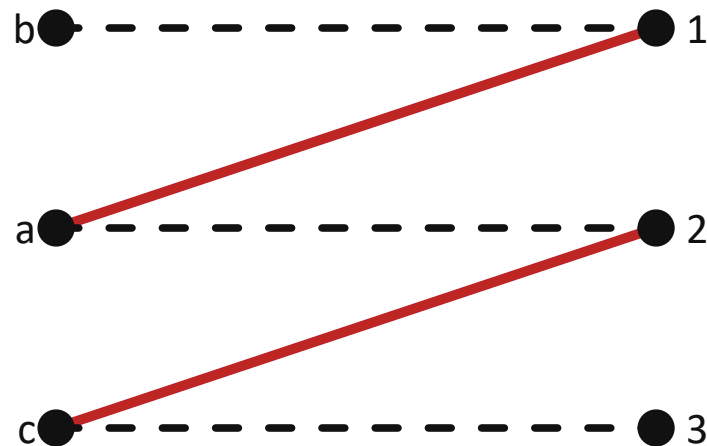
Maximal: no unused edge can be added. **Maximum:** no matching has larger size.



This matching is maximal, but we can improve it by **changing several assignments**.

An augmenting path

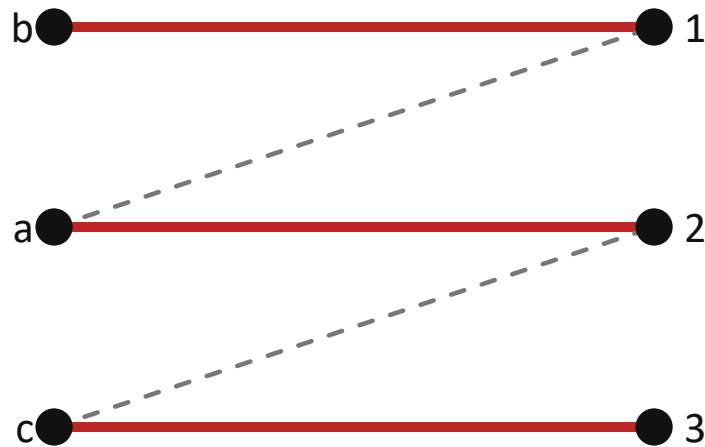
$b \longrightarrow 1 \longrightarrow a \longrightarrow 2 \longrightarrow c \longrightarrow 3$



It starts and ends at unmatched vertices, and alternates between unused and selected edges.

Flip along the path

$$\{a1, c2\} \longrightarrow \{b1, a2, c3\}$$



The matching size increases from 2 to 3. It is maximum: there are only three workers.

The augmenting-path algorithm

1. Start with the empty matching.
2. Find an augmenting path, if one exists. Flip selected and unused edges along it.
3. Repeat until there is no augmenting path.

Theorem: A matching is maximum if and only if it has no augmenting path.

The matching LP relaxation

$$\begin{aligned} \max \quad & \sum_{e \in E} x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} x_e \leq 1 \quad \text{for every vertex } v, \\ & x_e \geq 0 \quad \text{for every edge } e. \end{aligned}$$

Each edge variable is automatically at most 1 because it appears in an endpoint constraint.

Can the LP choose fractions and obtain a better objective?

Bipartite matching is integral

Theorem: For a bipartite graph, every vertex of the matching LP feasible region is integral.

An integral optimal solution exists, $z_{LP} = z_{IP}$.

Because the variables lie between 0 and 1, an integral vertex describes a matching.

The graph structure makes this relaxation exact.

Fractional points can still exist

With two workers, two jobs, and all four possible edges, consider the assignment matrix

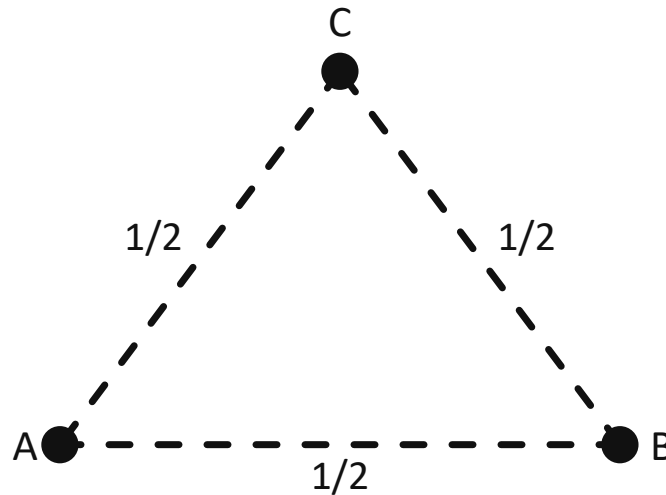
$$\begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Every row and column sums to 1. Its objective value is 2, just like either perfect matching.

This fractional optimum is **not a vertex**: it lies between two different feasible points.

What fails on a triangle?

Assign $x_e = 1/2$ to each edge. Every vertex sees a total of 1.



$$z_{\text{LP}} = \frac{3}{2} > z_{\text{IP}} = 1.$$

Reading

Chong, Lu, and Zak, 5th edition

Chapter 19: Integer Linear Programming. Focus on Sections 19.1 and 19.2; Section 19.3 is optional further reading on cutting planes.

Supplement: Michel Goemans, *Lecture notes on bipartite matching*, MIT. Read the opening matching and augmenting-path discussion.

Our slides give the branch-and-bound example and the matching LP examples.

Conceptual questions

1. If an LP solver returns a fractional optimal solution, does that alone prove that the integer optimum is worse? Why?
2. Why can a matching that cannot be extended by adding one edge still fail to be maximum? What kind of change is needed?