Linear Programming

CSE 421 23Winter
Lecture 25

Today

Everything we can cover about a topic in one day.

Linear programming!

Tomorrow we'll use them for an approximation algorithm.

Goal today is just understand what they are and why they might be interesting.

Linear Programming

Used WIDELY in business and operations research.

Excel has a linear program solver.

A very **expressive** language for problem-solving

Can represent a wide-variety of problems, including some we've already seen.

Deep, beautiful theory…that we do not have time to cover.

Outline of LPs

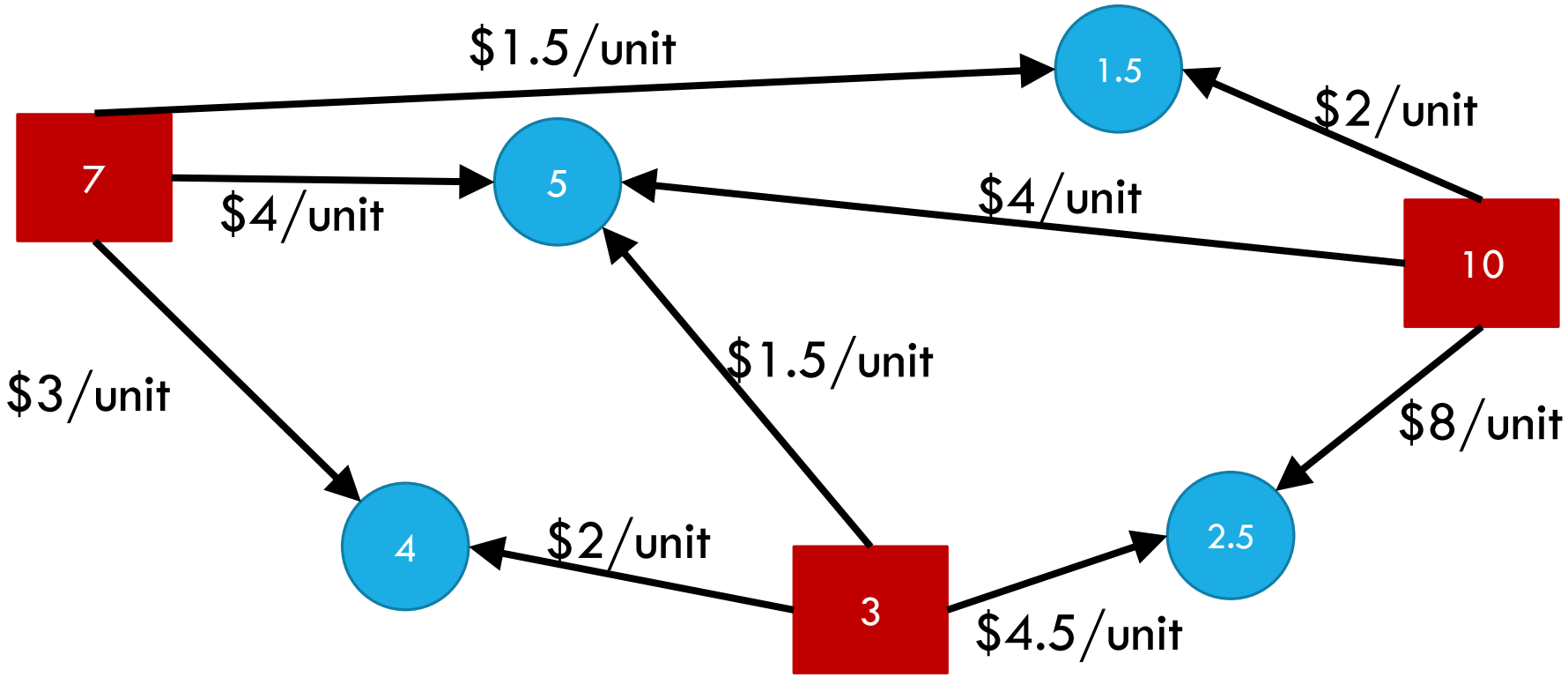
What is a linear program?

A simple example LP

Computational Issues

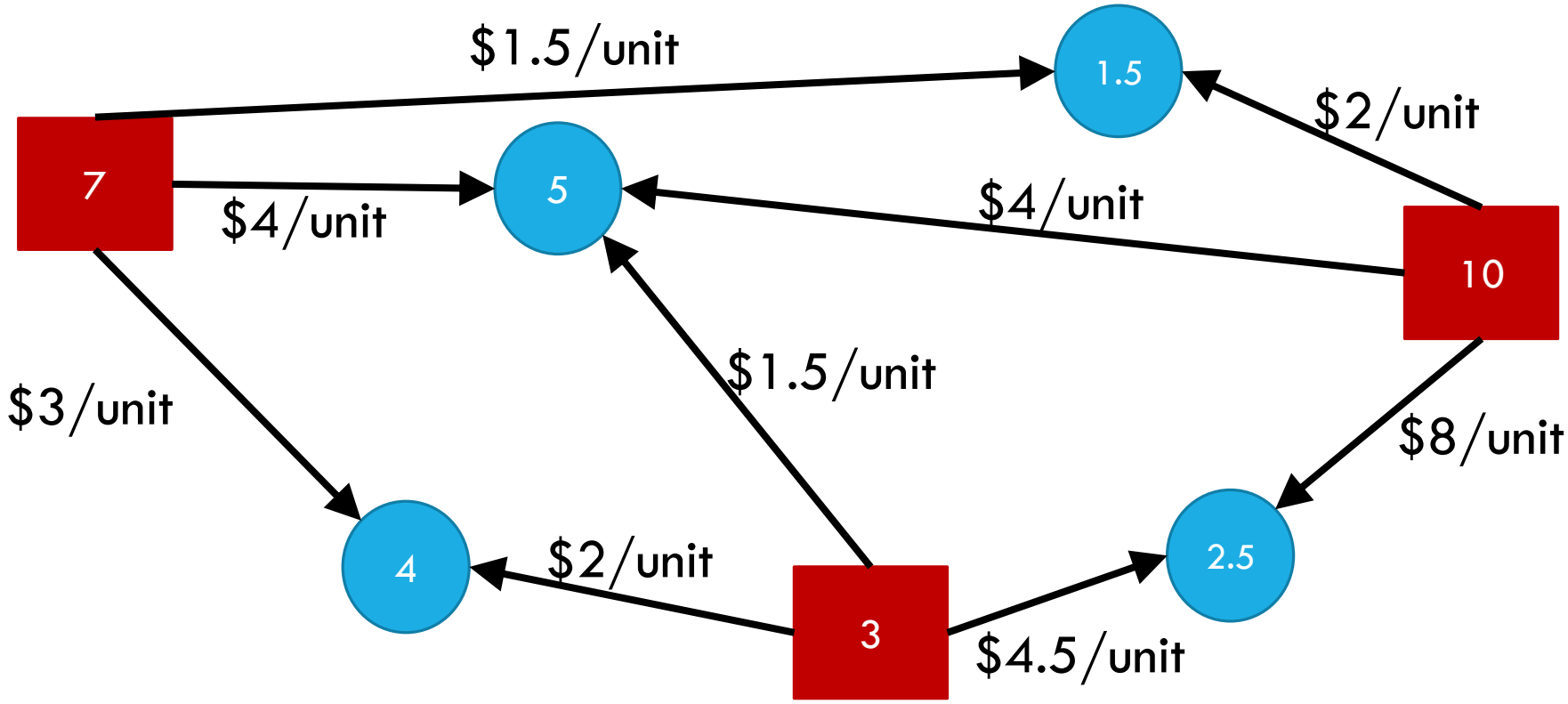
Example Problem

You're laying down soil for a bunch of new gardens. You got a few big piles of soil delivered (more than enough to cover the gardens)



Example Problem

What variables should we use?

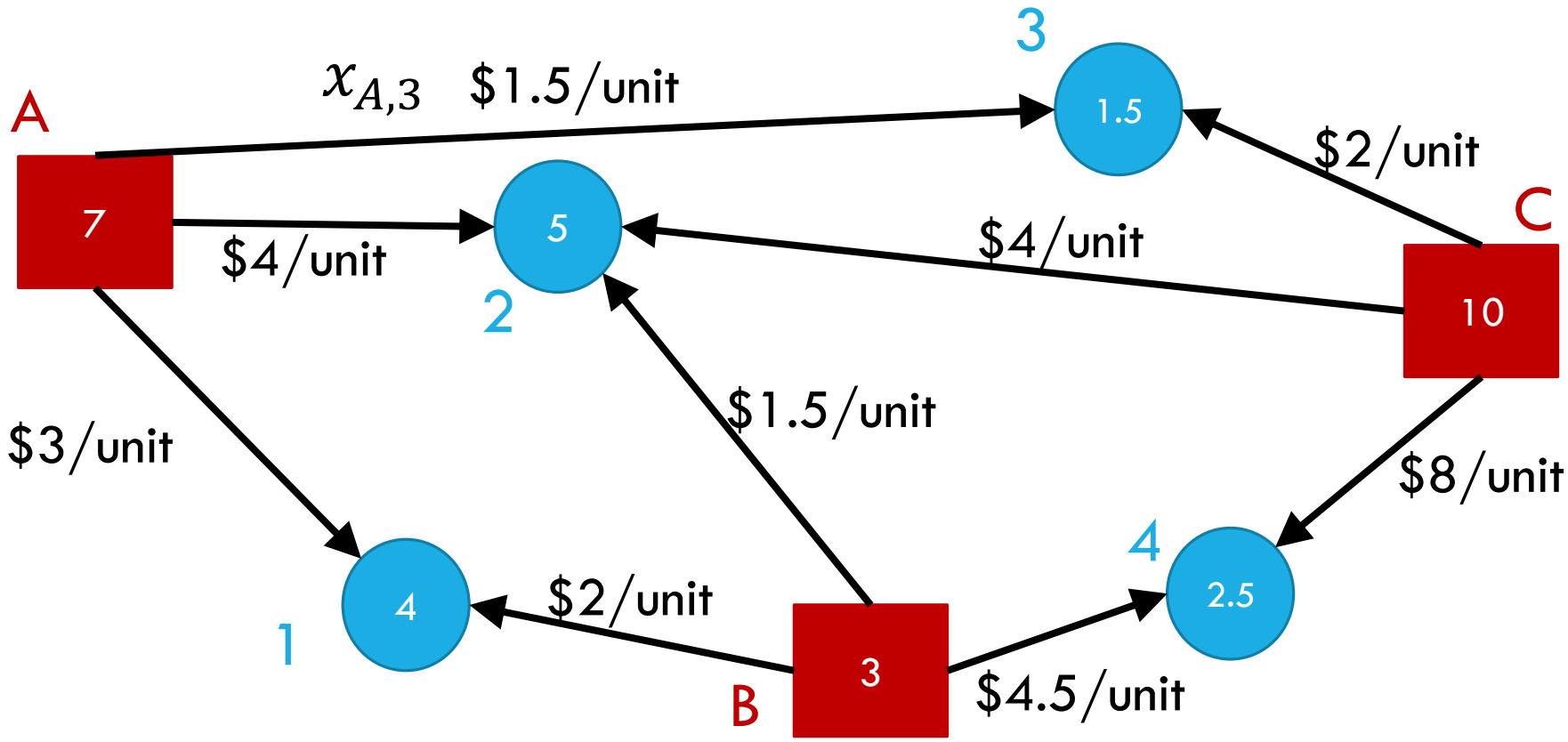


Example Problem

What variables should we use?

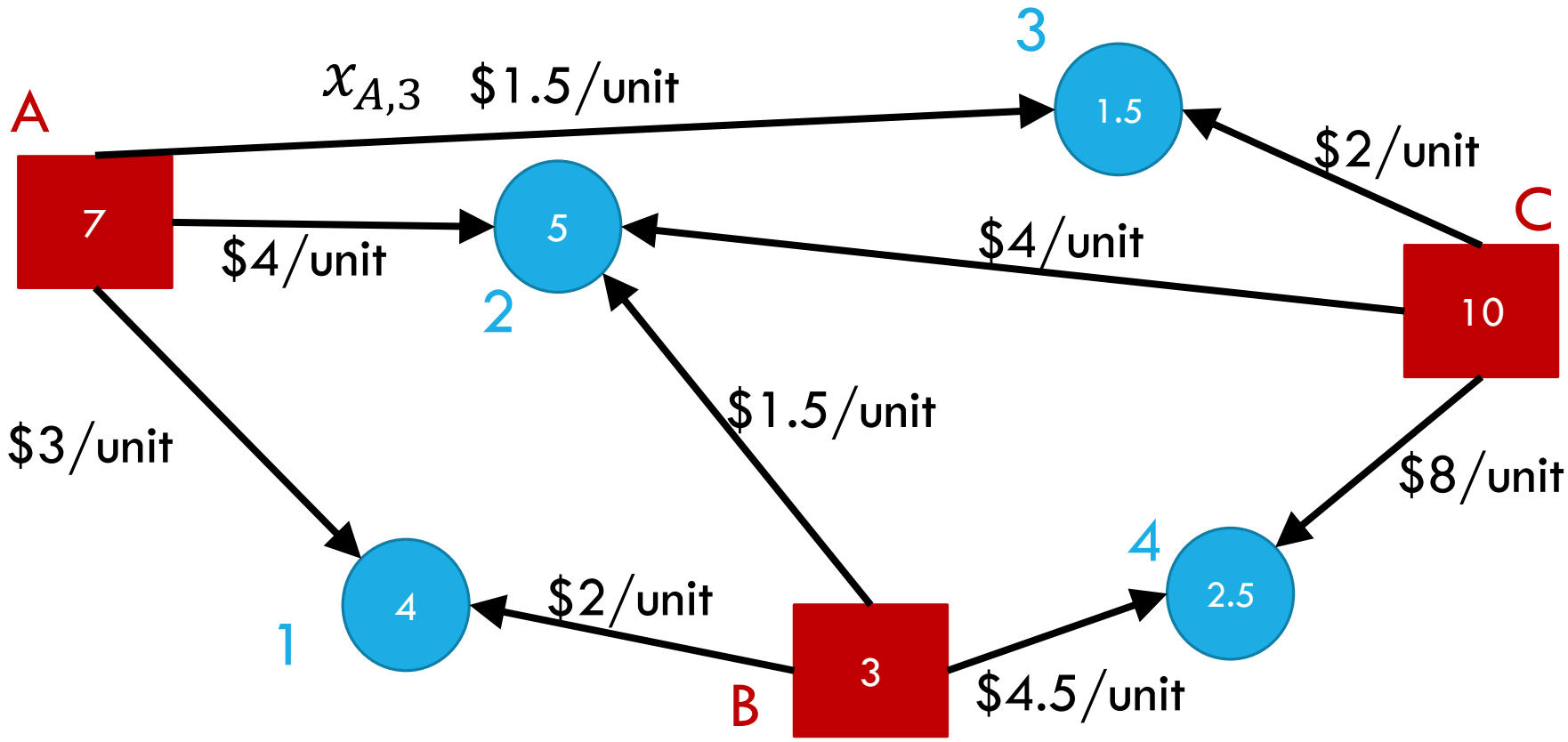
One for each edge (how much to move from a pile to a garden)

E.g. $x_{A,3}$ is how many units moved from A to 3.



Example Problem

What constraints are there on the variables?



Example Problem

What constraints are there on the variables?

Gardens each get enough soil:

$x_{A,1} + x_{B,1} \geq 4$

$x_{A,2} + x_{B,2} + x_{C,2} \geq 5$

$x_{A,3} + x_{C,3} \geq 1.5$

$x_{B,4} + x_{C,4} \geq 2.5$

Can't overuse a pile:

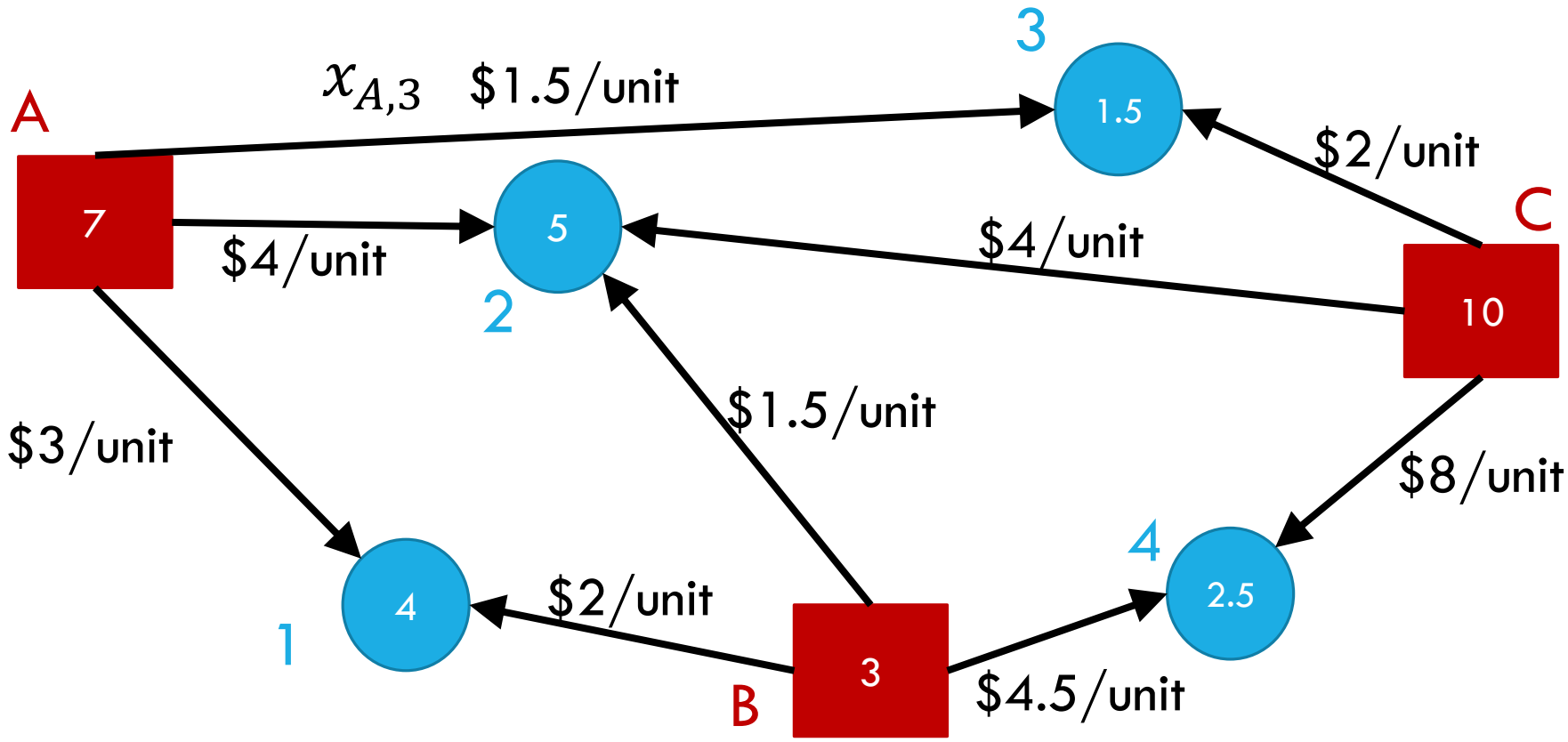
$x_{A,1} + x_{A,2} + x_{A,3} \leq 7$

$x_{B,1} + x_{B,2} + x_{B,4} \leq 3$

$x_{C,2} + x_{C,3} + x_{C,4} \leq 10$

No anti-soil:

$x_{i,j} \geq 0$ for all i, j

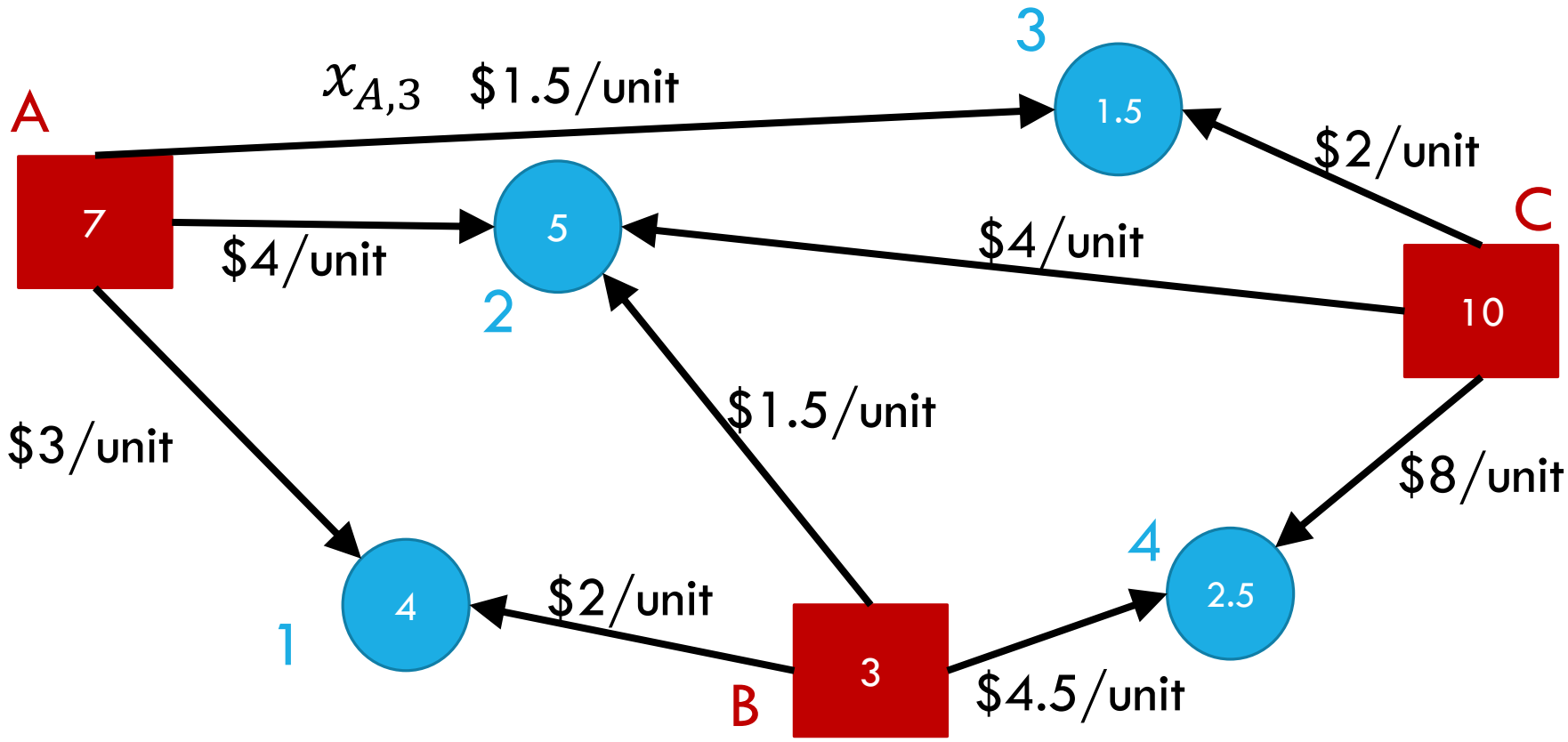


Example Problem

What's the cost (in terms of the variables)?

$$(x_{A,1} \cdot 3 + x_{A,2} \cdot 4 + x_{A,3} \cdot 1.5) +$$
$$(x_{B,1} \cdot 2 + x_{B,2} \cdot 1.5 + x_{B,4} \cdot 4.5) +$$
$$(x_{C,2} \cdot 4 + x_{C,3} \cdot 2 + x_{C,4} \cdot 8)$$

Sum cost*var for all the variables



Full Definition

Minimize: $(x_{A,1} \cdot 3 + x_{A,2} \cdot 4 + x_{A,3} \cdot 1.5) + (x_{B,1} \cdot 2 + x_{B,2} \cdot 1.5 + x_{B,4} \cdot 4.5) + (x_{C,2} \cdot 4 + x_{C,3} \cdot 2 + x_{C,4} \cdot 8)$

Subject To:

$x_{A,1} + x_{B,1} \geq 4$

$x_{A,2} + x_{B,2} + x_{C,2} \geq 5$

$x_{A,3} + x_{C,3} \geq 1.5$

$x_{B,4} + x_{C,4} \geq 2.5$

$x_{A,1} + x_{A,2} + x_{A,3} \leq 7$

$x_{B,1} + x_{B,2} + x_{B,4} \leq 3$

$x_{C,2} + x_{C,3} + x_{C,4} \leq 10$

$x_{i,j} \geq 0$ for all i, j

A Linear Program

A linear program is defined by:

Real-valued **variables**

Subject to satisfying **everything** in a list of **linear constraints**

A linear constraint is a statement of the form: $\sum a_i x_i \leq c_i$
where a_i are constants, the x_i are variables and c_i is a constant.

Maximizing or minimizing a linear objective function

A linear objective function is a function of the form: $\sum b_i x_i$
where b_i are constants and the x_i are variables.

Linear constraints

Pollev.com/robbie

Can you write each of these requirements as linear constraint(s)?

Some of these are tricks…

x_i times x_j is at least 5

$5x_i$ is equal to 1

$x_i \leq 5$ OR $x_i \geq 7$

x_i is non-negative.

x_i is an integer.

Linear constraints

Can you write each of these requirements as linear constraint(s)?

Some of these are tricks…

x_i times x_j is at least 5	No way to represent ☹
$5x_i$ is equal to 1	$5x_i \leq 1$ and $-5x_i \leq -1$
$x_i \leq 5$ OR $x_i \geq 7$	No way to represent ☹
x_i is non-negative.	$x_i \geq 0$
x_i is an integer.	No way to represent ☹

What are we looking for?

A solution (or point) is a setting of all the variables

A **feasible point** is a point that satisfies all the constraints.

An **optimal point** is a point that is feasible and has at least as good of an objective value as every other feasible point.

Example Problem

Gardens each get enough soil:

$x_{A,1} + x_{B,1} \geq 4$

$x_{A,2} + x_{B,2} + x_{C,2} \geq 5$

$x_{A,3} + x_{C,3} \geq 1.5$

$x_{B,4} + x_{C,4} \geq 2.5$

Can't overuse a pile:

$x_{A,1} + x_{A,2} + x_{A,3} \leq 7$

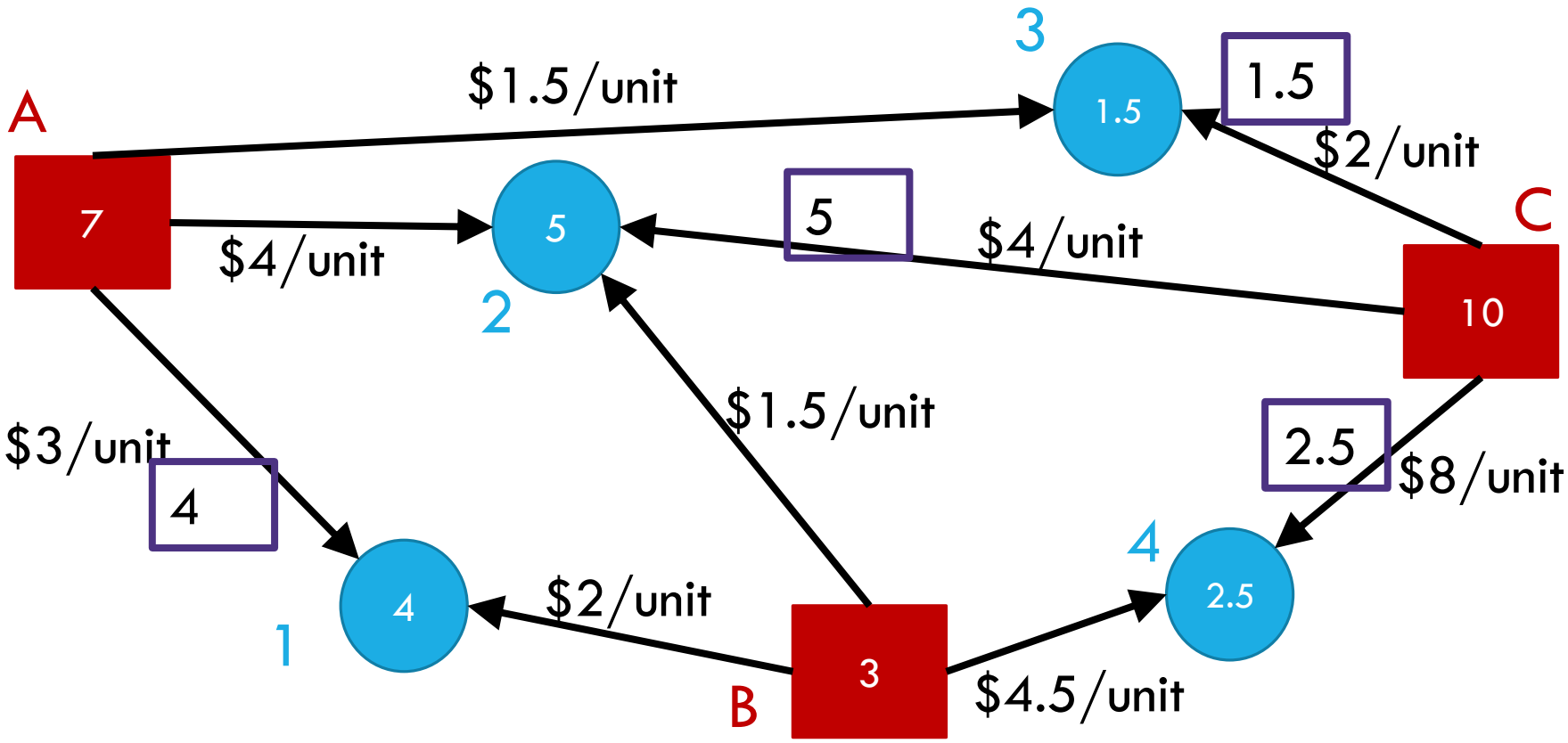
$x_{B,1} + x_{B,2} + x_{B,4} \leq 3$

$x_{C,2} + x_{C,3} + x_{C,4} \leq 10$

No anti-soil:

$x_{i,j} \geq 0$ for all i, j

A feasible point.
Objective: 55



Example Problem

Gardens each get enough soil:

$x_{A,1} + x_{B,1} \geq 4$

$x_{A,2} + x_{B,2} + x_{C,2} \geq 5$

$x_{A,3} + x_{C,3} \geq 1.5$

$x_{B,4} + x_{C,4} \geq 2.5$

Can't overuse a pile:

$x_{A,1} + x_{A,2} + x_{A,3} \leq 7$

$x_{B,1} + x_{B,2} + x_{B,4} \leq 3$

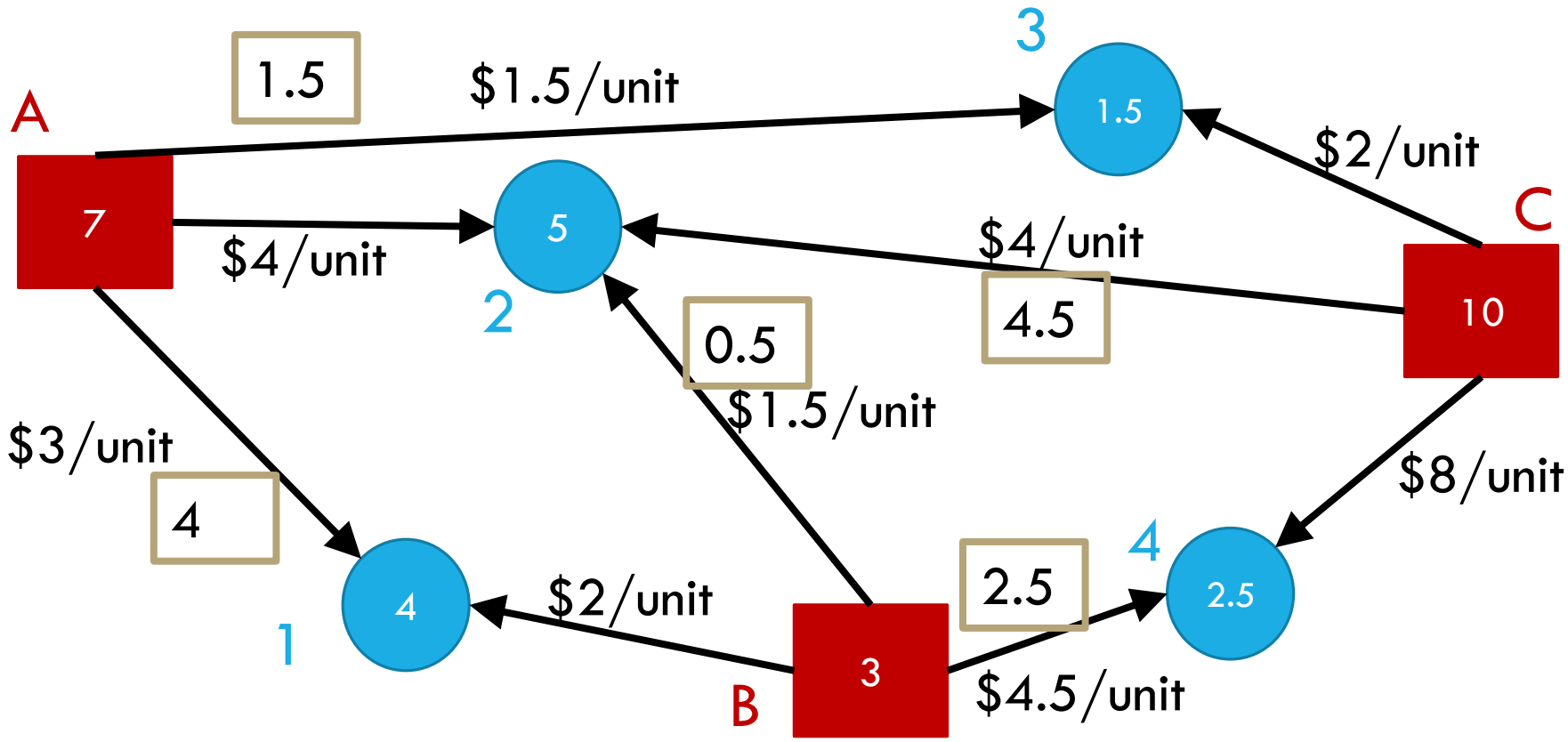
$x_{C,2} + x_{C,3} + x_{C,4} \leq 10$

No anti-soil:

$x_{i,j} \geq 0$ for all i, j

A feasible point.
Objective: 44.25

This is an optimal point. There are others!



Example Problem

Gardens each get enough soil:

$x_{A,1} + x_{B,1} \geq 4$

$x_{A,2} + x_{B,2} + x_{C,2} \geq 5$

$x_{A,3} + x_{C,3} \geq 1.5$

$x_{B,4} + x_{C,4} \geq 2.5$

Can't overuse a pile:

$x_{A,1} + x_{A,2} + x_{A,3} \leq 7$

$x_{B,1} + x_{B,2} + x_{B,4} \leq 3$

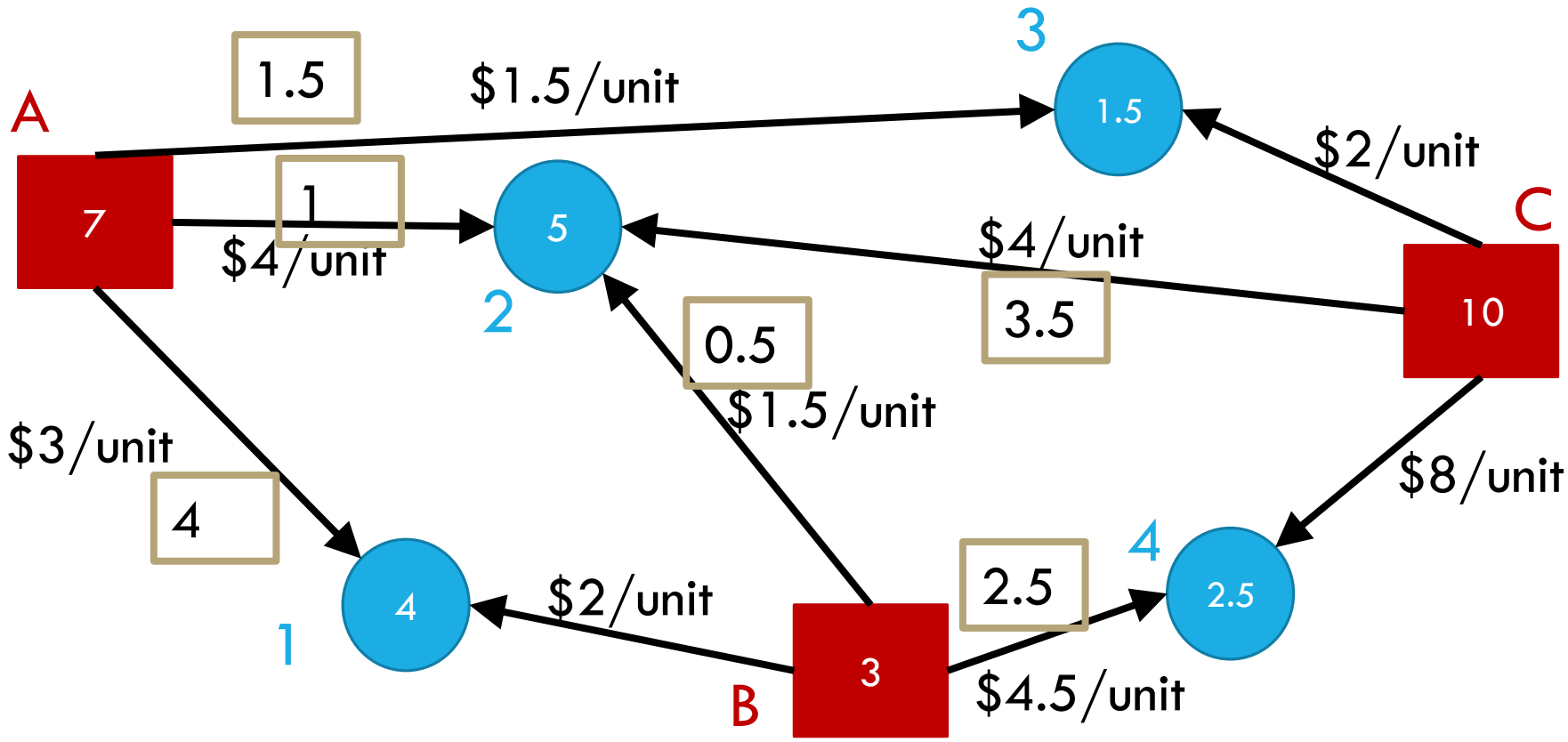
$x_{C,2} + x_{C,3} + x_{C,4} \leq 10$

No anti-soil:

$x_{i,j} \geq 0$ for all i, j

A feasible point.
Objective: 44.25

Here's another optimal point!!



Solving LPs

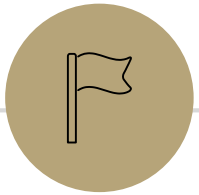
For this class, we're only going to think about library functions to solve linear programs (i.e. we won't teach you how any of the algorithms work)

The most famous is the **Simplex Method** – can be quite slow (exponential time) in the worst case. But rarely hits worst-case behavior.

Very fast in practice. Idea: jump from extreme point to extreme point.

The **Ellipsoid Method** was the first theoretically polynomial time algorithm $O(n^6)$ where n is the number of bit needed to describe the LP (usually $\approx$ the number of constraints)

Interior Point Methods are faster theoretically, and starting to catch up practically. $O(n^{2.373})$ theoretically



Some Practice

Write an LP for finding the biggest flow

Let $c(e)$ be the capacity on edge e

What are your variables? (How do you represent a flow)

What are your constraints? (When is a flow valid)

What is your objective? (what do you want to maximize or minimize)

Write an LP for finding the biggest flow

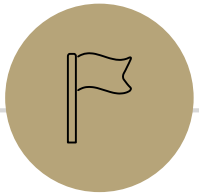
Let $c(e)$ be the capacity on edge e

Max $\sum_{e:e \text{ enters } t} x_e$

Subject to:

$\sum_{e:e \text{ enters } u} x_e = \sum_{e:e \text{ leaves } u} x_e$ for every u other than s and t

$0 \leq x_e \leq c(e)$ for every edge e



Integrality

Another Question

Change the problem

Instead of infinitely divisible dirt…

What if instead we're moving whole unit things (the dirt is in bags we can't open or we're moving bikes or plants or anything else that can't be split)

Or if we're assigning people to shifts (can have 1/3 of a person on a shift)

Well, the constraints will change (your "demand" and "supplies" will be integers)

Non-Integrality

Some linear programs only have optimal solutions that have some (or all) variables as non-integers (even with only integers in the objective function and constraints).

For dirt or water or anything arbitrarily divisible, no big deal!

For cell phones or bicycles…only possibly a big deal!

In practice: if the optimal thing to manufacture 999,999.8 widgets per day, rounding up or down probably isn't going to make a huge difference in your profits.

But sometimes rounding isn't ok…

What do you do if you need integers?

Integer Programs are linear programs where you can mark some variables as needing to be integers.

In practice – often still solvable (Excel also has a solver for these problems). But no longer guaranteed to be efficient.

Indeed, solving an integer program is NP-hard.

Lots of theory has been done for when the optimum will be all integers. (MATH 407 or MATH 514)

But sometimes you get a fractional solution…what can you do?

A nicer example

Sometimes we can round fractional solutions into integral ones.

Minimum Weight Vertex Cover

We've seen how to solve the problem with DP on trees.

Let's try it now with linear programming.

A set S of vertices is a vertex cover if for every edge (u, v), u is in S, v is in S or both are in S.

Vertex Cover LP

Pollev.com/robbie

Write an LP for finding the minimum weight vertex cover

A set S of vertices is a vertex cover if for every edge (u, v), u is in S, v is in S or both are in S.

What are your variables, then how do you constrain them?

Let $w(u)$ be the weight for a vertex u. You can treat $w(u)$ as a constant.

Vertex Cover LP

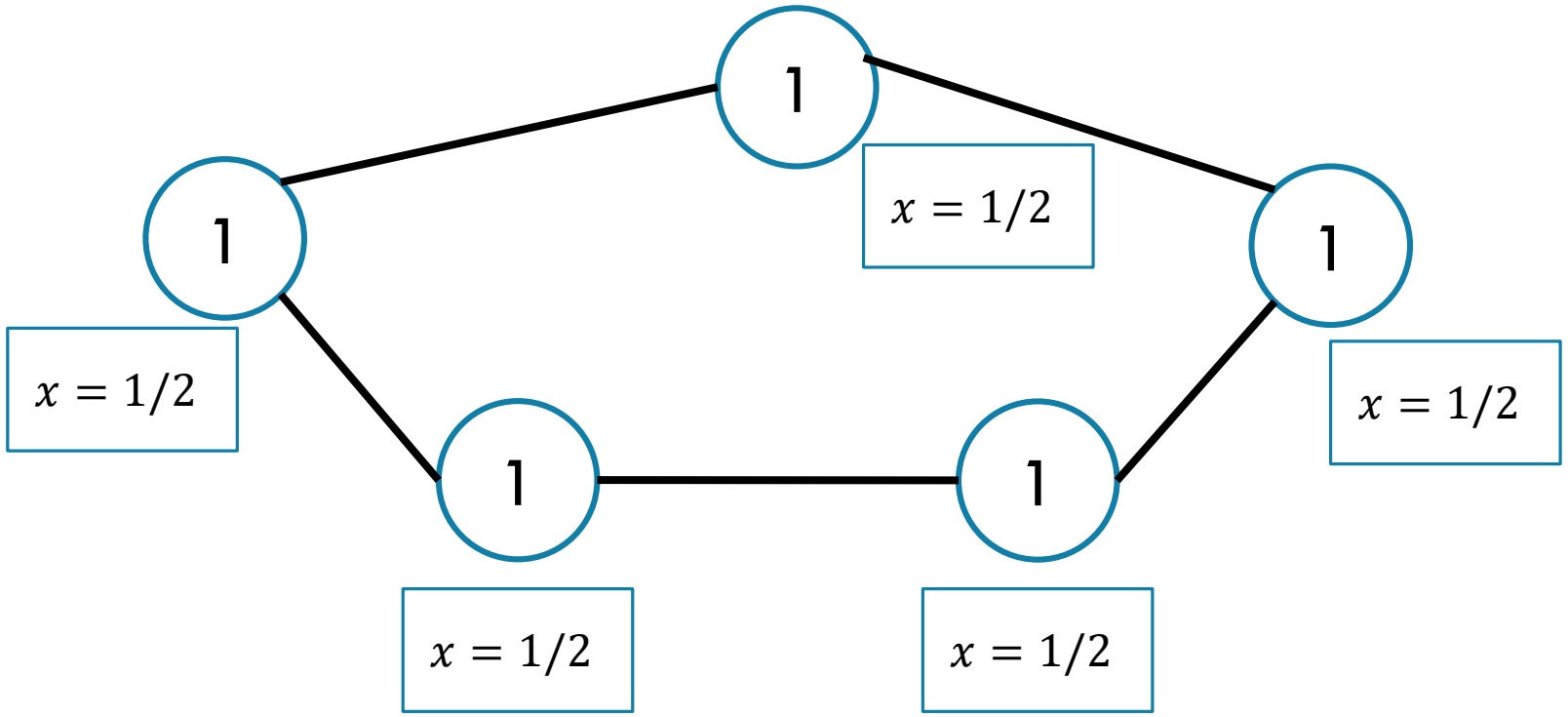
Minimize $\sum w(u) \cdot x_u$

Subject to:

$x_u + x_v \geq 1$ for all $(u, v) \in E$

$0 \leq x_u \leq 1$ for all u.

Non-Bipartite



The LP finds a fractional vertex cover of weight 2.5

There's no "real"/integral VC of weight 2.5. – lightest is weight 3.

There's a "gap" between integral and fractional solutions.

So, what if the graph isn't bipartite?

Big idea:

Just round!

Pollev.com/robbie

If $x_u \geq \frac{1}{2'}$ round up to 1.

If $x_u < \frac{1}{2}$, round down to 0

Minimize $\sum w(u) \cdot x_u$

Subject to:

$x_u + x_v \geq 1$ for all $(u, v) \in E$

$0 \leq x_u \leq 1$ for all u.

Two questions – is it a vertex cover? How far are we from the true minimum?

Is it a vertex cover?

Every edge was covered in the fractional matching

i.e. for every edge (u, v)

$x_u + x_v \geq 1$.

At least one of those is getting rounded up!

So every edge is covered.

And we've rounded to integers, so we have a "real" vertex cover.

How good of an approximation is it?

Well, we might have doubled the value of the LP when we rounded. But we definitely didn't do any more than that.

$2 \cdot LP \geq ALG$

And the value of the LP is definitely not bigger than the true size of the vertex cover (because otherwise the LP would have found that).

$OPT \geq LP$

Combining:

$2 \cdot OPT \geq ALG$

So we're safe in calling this a 2-approximation.

Comparing to the LP value

We did a weird thing on that last slide.

We were supposed to compare the value of our vertex cover to the best vertex cover.

But instead we compared it to the value of the LP…which we know isn't always the value of the vertex cover!

That wasn't laziness, it's a very common technique. We know very little about the true value of the vertex cover (if we knew what it looked like VERY VERY precisely, why couldn't we just write an algorithm to find it? We actually won't know much). So we start with what the algorithm gave us (that we do understand).

Side Note

Could we do better?

Not just with the LP.

If you take a graph with n vertices and every possible edge, the LP's minimum is $n/2$, the true minimum vertex cover is size $n - 1$.

The ratio is $2 - 1/n$. So if we don't at least double the value **sometimes** we won't get a vertex cover at all.

Getting a 1.9999999 approximation is an open problem!