Advanced Algorithms
Linear Programming

Reading:

CLRS, Chapter29 (2^{nd} ed. onward).

"Linear Algebra and Its Applications", by Gilbert Strang, chapter 8

"Linear Programming", by Vasek Chvatal

"Introduction to Linear Optimization", by Dimitris Bertsimas and John Tsitsiklis

- Lecture notes by John W. Chinneck: http://www.sce.carleton.ca/faculty/chinneck/po.html

An Example: The Diet Problem

- A student is trying to decide on lowest cost diet that provides sufficient amount of protein, with two choices:
  - steak: 2 units of protein/kg, \$3/kg
  - peanut butter: 1 unit of protein/kg, \$2/kg
- In proper diet, need 4 units protein/day.

Let x = # kgs peanut butter/day in the diet.

Let y = # kgs steak/day in the diet.

Goal: minimize $2x + 3y$ (total cost)

subject to constraints:

$x + 2y \geq 4$

$x \geq 0, \; y \geq 0$

This is an LP- formulation of our problem

An Example: The Diet Problem

Goal: minimize $2x + 3y$ (total cost)
subject to constraints:

$x + 2y \geq 4$

$x \geq 0, \; y \geq 0$

- This is an optimization problem.
- Any solution meeting the nutritional demands is called a *feasible solution*
- A feasible solution of minimum cost is called the *optimal solution.*

Linear Programming

- The process of optimizing a linear objective function subject to a finite number of linear constraints.
- The word "programming" is historical and predates computer programming.
- Example applications:
  - airline crew scheduling
  - manufacturing and production planning
  - telecommunications network design
- "Few problems studied in computer science have greater application in the real world."

Linear Program - Definition

A linear program is a problem with n variables $x_1,\dots,x_n$, that has:

1. A linear objective function, which must be minimized/maximized. Looks like:

   $$\min\ (\max)\ c_1x_1+c_2x_2+\dots+c_nx_n$$

2. A set of m linear constraints. A constraint looks like:

   $$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \text{ (or } \geq \text{ or } = \text{)}$$

Note: the values of the coefficients c_i, $a_{i,j}$ are given in the problem input.

LP – Matrix form

$\max c^T x$ s.t.

$Ax \leq b$

x – vector of n variables

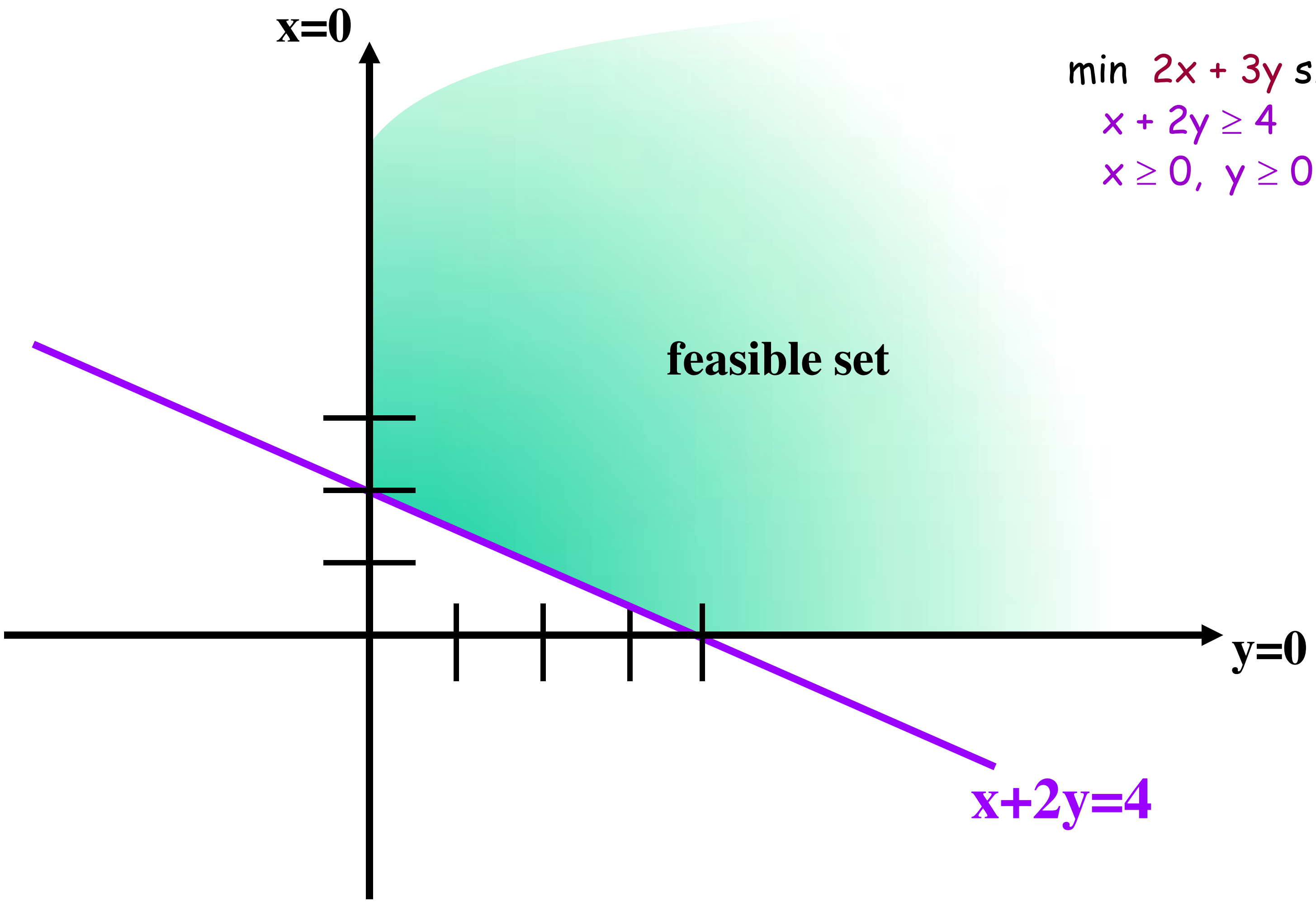
c – vector of n objective function coefficients

A – m-by-n matrix

b – vector of dimension m

Geometric intuition
x= peanut butter, y = steak

min $2x + 3y$ s.t.
$x + 2y \geq 4$
$x \geq 0,\ y \geq 0$



Feasible Set

- Each linear inequality divides n-dimensional space into two halfspaces, one where the inequality is satisfied, and one where it's not.
- Feasible Set : solutions to a family of linear inequalities.

Feasible Set

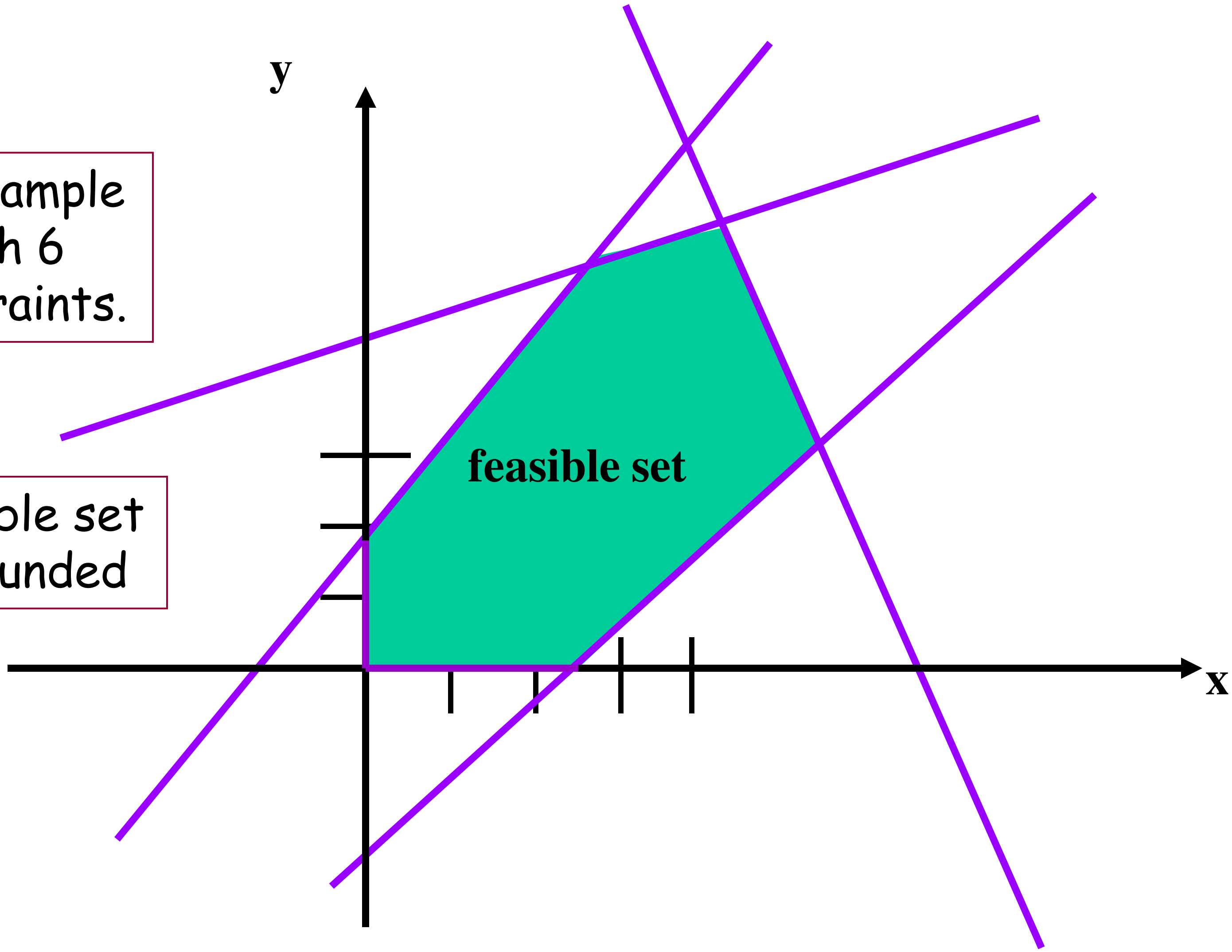
- Each linear inequality divides n-dimensional space into two halfspaces, one where the inequality is satisfied, and one where it's not.
- The feasible set is the intersection of the halfspaces where all inequalities are satisfied.
- An intersection of halfspaces is called a convex polyhedron. So the feasible set is a convex polyhedron.
- Fact: every point p in a convex polytope can be represented as a convex combination of the vertices v_i of the polytope.

$$p = \sum \lambda_i v_i \qquad \left(0 \le \lambda_i \le 1\,;\, \sum \lambda_i = 1\right)$$

Feasible set!

An Example with 6 constraints.

Feasible set is bounded



The Feasible Set

- Feasible set is a convex polyhedron.
- A bounded and nonempty polyhedron is called a convex polytope.

There are 3 cases:

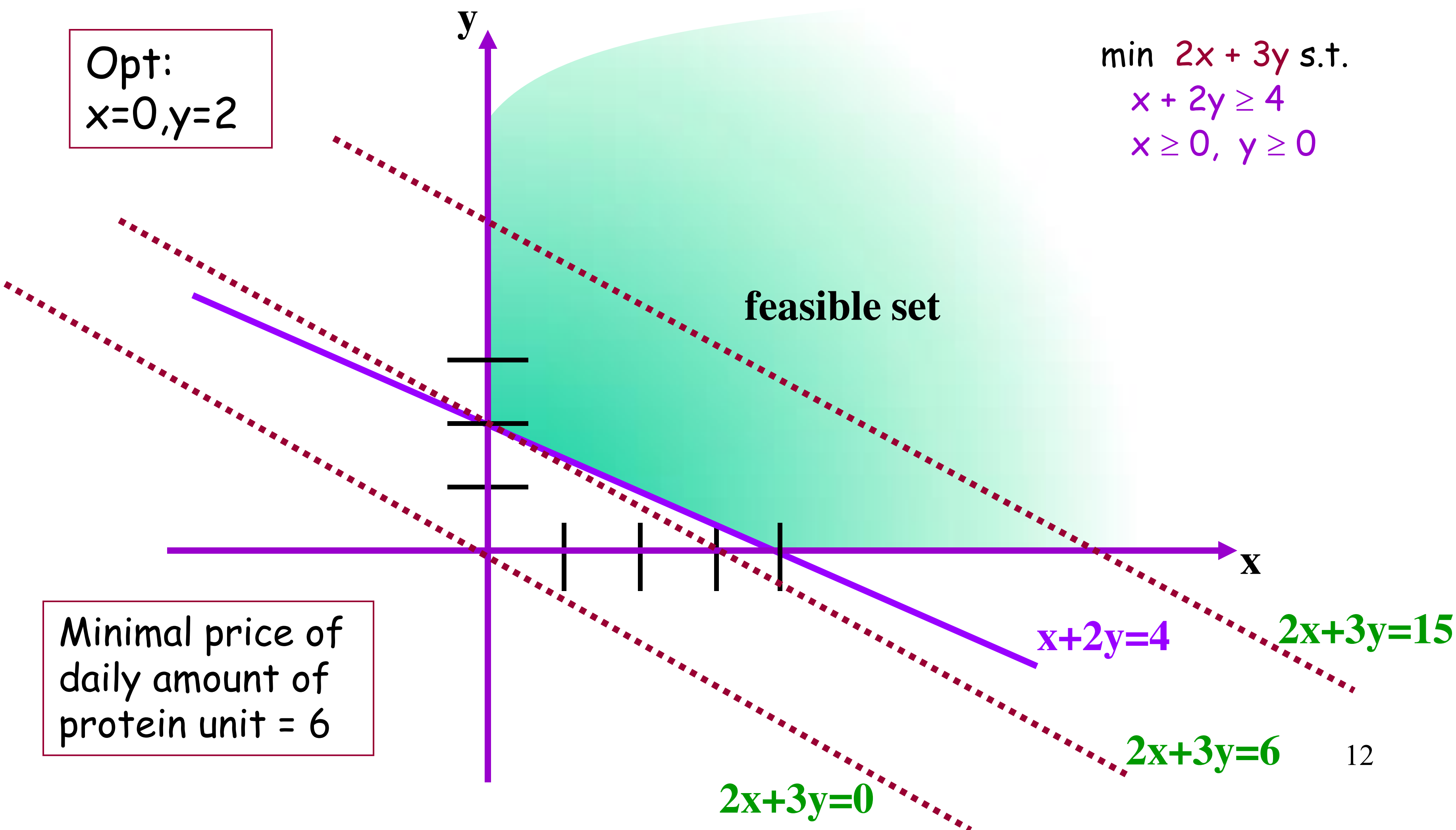
- feasible set is empty (problem is not feasible)
- Feasible set is unbounded
- Feasible set is bounded and nonepmty (a polytope)

- First two cases very uncommon for real problems in economics and engineering.

Lines of constant objective function

Opt:
x=0,y=2

min $2x + 3y$ s.t.
$x + 2y \geq 4$
$x \geq 0, \ y \geq 0$



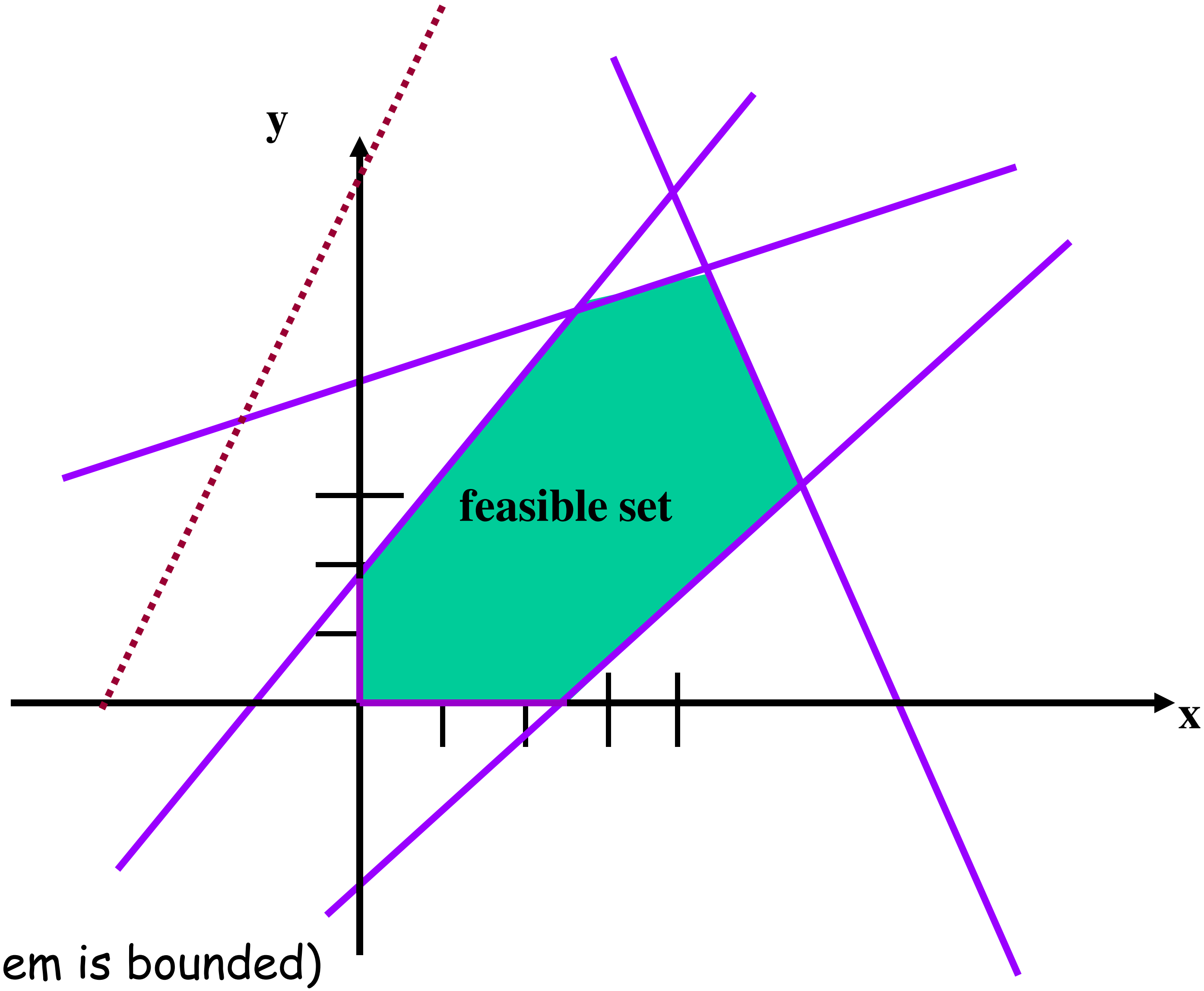
Minimal price of daily amount of protein unit = 6

The optimal objective value

There are 3 cases:

- feasible set is empty (problem is not feasible)
- cost function is unbounded on feasible set.
- cost has a minimum (or maximum) on feasible set.

Optimal value occurs at some vertex of the feasible set!



(if problem is bounded)

Optimal solution always at a vertex

The linear cost function defines a family of parallel hyperplanes (lines in 2D, planes in 3D, etc.).

Want to find one of minimum cost.

If exists, must occur at a vertex of the feasible set.

Proof: Let p be any point in the feasible set.
Write $p = \sum \lambda_i v_i \qquad (0 \leq \lambda_i \leq 1 \, ; \sum \lambda_i = 1)$
By linearity of the objective function z,
$\mathrm{z}(p) = \sum \lambda_i z(v_i) \leq z(v_{max})$, where v_{max} is the vertex that maximizes z.

Standard Form of a Linear Program.

maximize $\sum_{j=1}^{n} c_j x_j$

subject to:

$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1 \ldots m$

$x_j \geq 0 \quad j = 1 \ldots n$

$\max c^T x$ s.t.

$Ax \leq b$

$x \geq 0$

Converting to Standard Form

minimize $\displaystyle\sum_{j=1}^{n} c_j x_j$

subject to:

$\sum_{j=1}^{n} a_{1j} x_j \geq b_1$

$\sum_{j=1}^{n} a_{2j} x_j = b_2$

maximize $\displaystyle\sum_{j=1}^{n} -c_j x_j$

subject to:

$\sum_{j=1}^{n} -a_{1j} x_j \leq -b_1$

$\sum_{j=1}^{n} a_{2j} x_j \leq b_2$

$\sum_{j=1}^{n} -a_{2j} x_j \leq -b_2$

Solving LP

- There are several algorithms that solve any linear program optimally.
  - The Simplex method (to be discussed)
  - The Ellipsoid method
  - The interior point method
- These algorithms can be implemented in various ways.
- There are many existing software packages for LP.
- LP can be used as a "black box" for solving various optimization problems.

LP formulation: another example

Bob's bakery sells bagels and muffins.

To bake a dozen bagels Bob needs 5 cups of flour, 2 eggs, and one cup of sugar.

To bake a dozen muffins Bob needs 4 cups of flour, 4 eggs and two cups of sugar.

Bob can sell bagels for 10$/dozen and muffins for 12$/dozen.

Bob has 50 cups of flour, 30 eggs and 20 cups of sugar.

How many bagels and muffins should Bob bake in order to maximize his revenue?

LP formulation: Bob's bakery

	Bagels	Muffins	Avail.
Flour	5	4	50
Eggs	2	4	30
Sugar	1	2	20
Revenue	10	12	

$$A = \begin{pmatrix} 5 & 4 \\ 2 & 4 \\ 1 & 2 \end{pmatrix}$$

$$c = \begin{pmatrix} 10 \\ 12 \end{pmatrix} \qquad b = \begin{pmatrix} 50 \\ 30 \\ 20 \end{pmatrix}$$

Maximize $10x_1+12x_2$

s.t. $5x_1+4x_2 \leq 50$

$2x_1+4x_2 \leq 30$

$x_1+2x_2 \leq 20$

$x_1 \geq 0, x_2 \geq 0$

Maximize $c^T \cdot x$

s.t. $Ax \leq b$

$x \geq 0.$

In class exercise:

Write the maximum flow problem an LP

Input: directed graph $G=(V,E)$ with non-negative arc capacities $c(e)$,

source and sink vertices s,t

Output: maximum flow from s to t in G.

Towards the Simplex Method

The Toy Factory Problem (TFP):

A toy factory produces dolls and cars.

Danny, a new employee, is hired. He can produce 2 cars and 3 dolls a day. However, the packaging machine can only pack 4 items a day. The company's profit from each doll is 10$ and from each car is 15$. What should Danny be asked to do?

Step 1: Describe the problem as an LP problem.

Let x_1,x_2 denote the number of cars and dolls produced by Danny.

The Toy Factory Problem

Let x_1,x_2 denote the number of cars and dolls produced by Danny.

Objective:

Max $z=15x_1+10x_2$

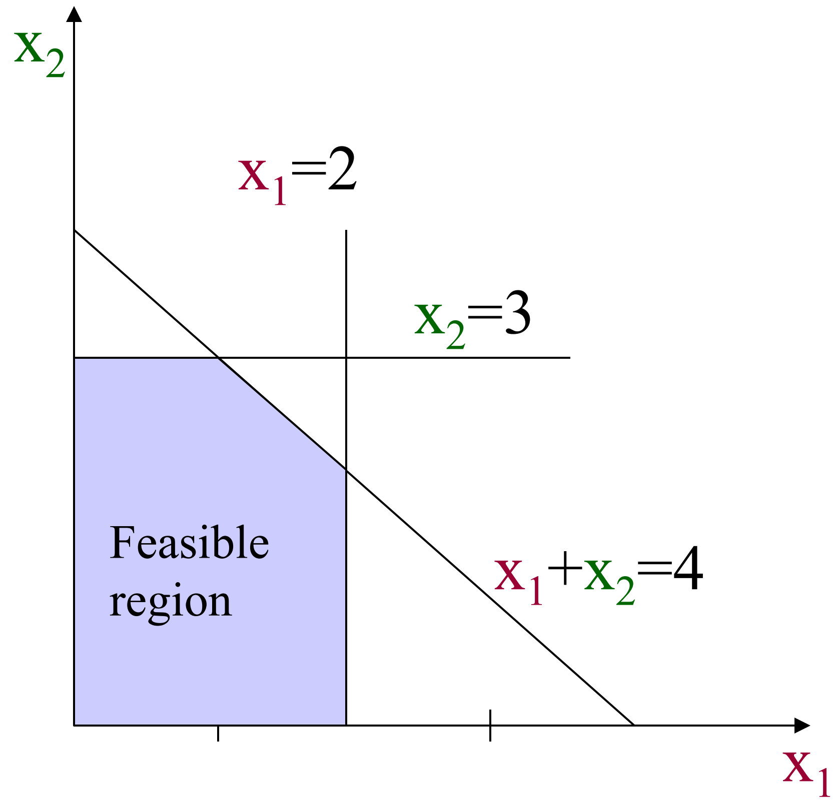
s.t $x_1 \leq 2$

$x_2 \leq 3$

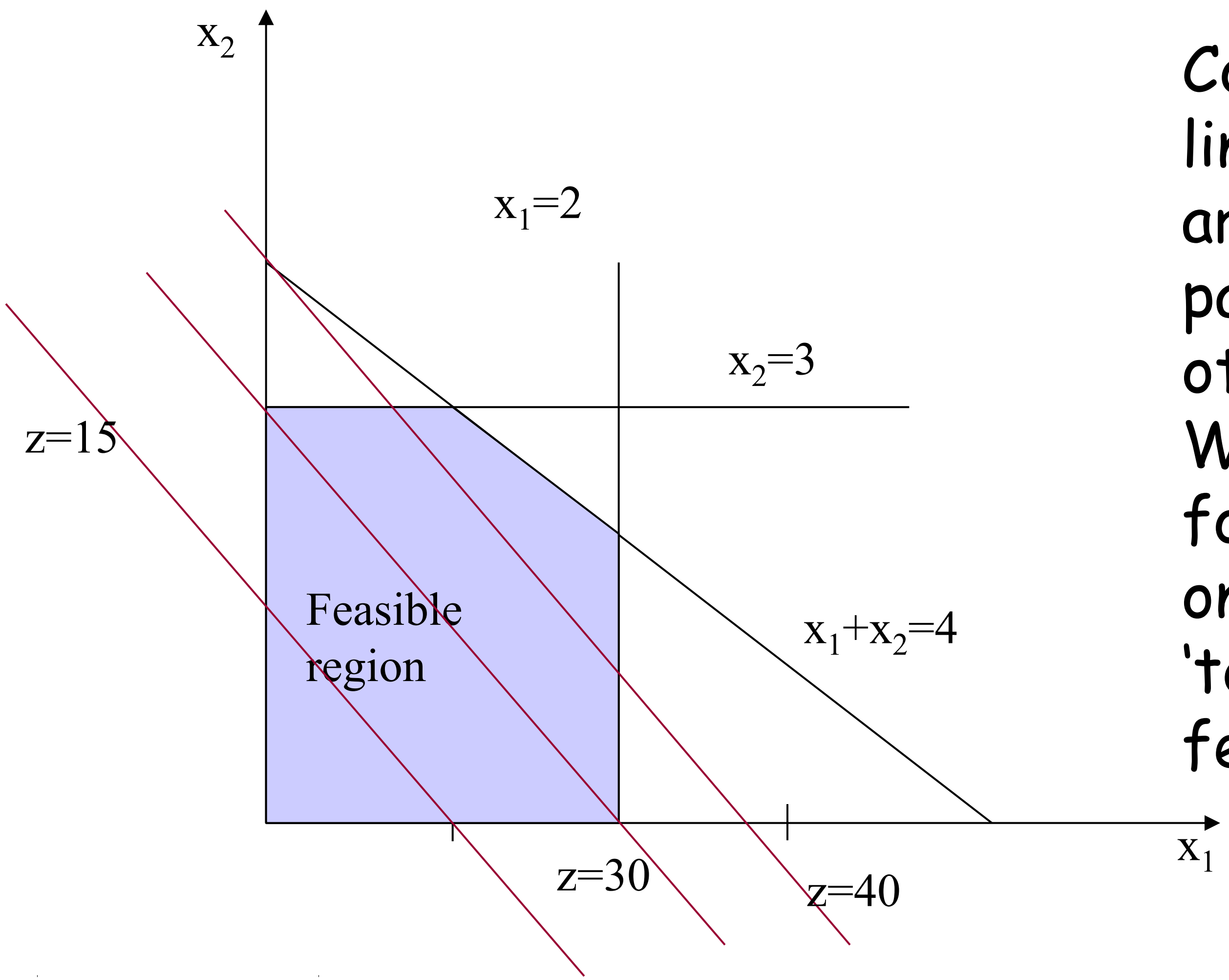
$x_1+x_2 \leq 4$

$x_1 \geq 0$

$x_2 \geq 0$



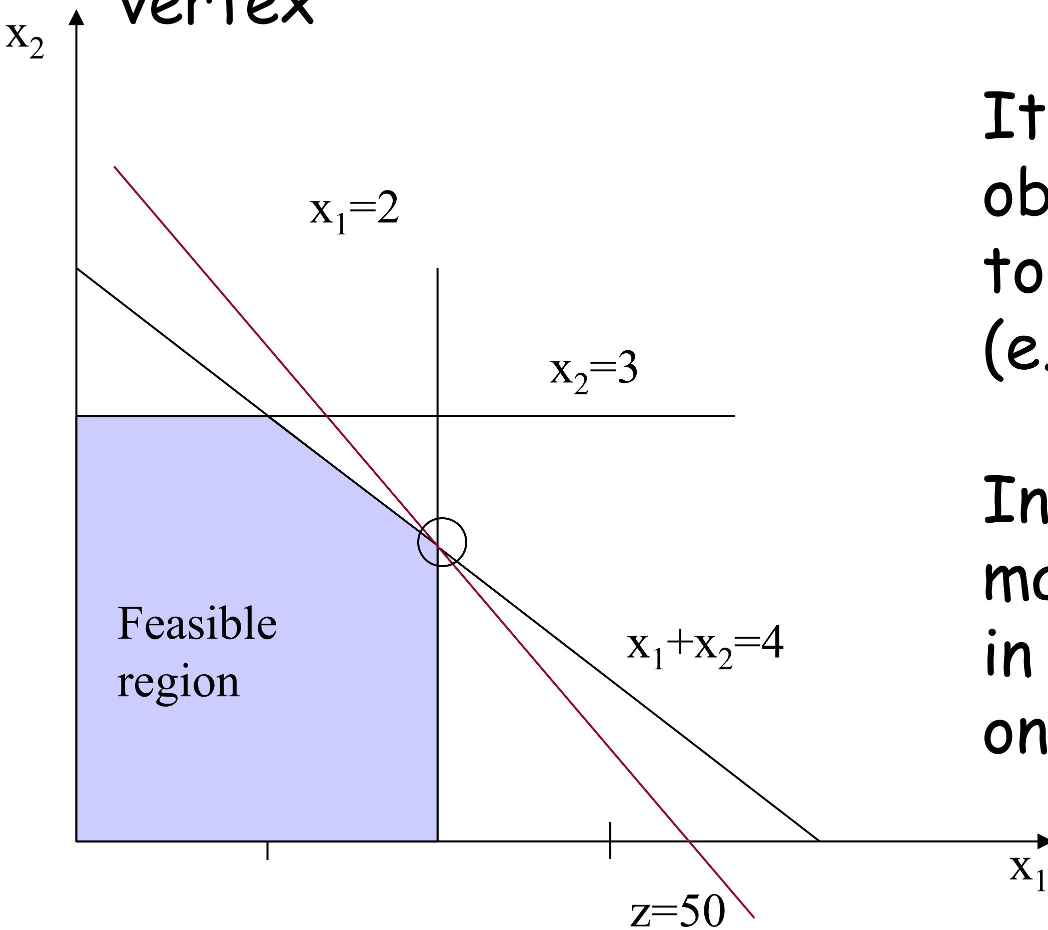
The Toy Factory Problem



Constant profit lines – They are always parallel to each other.
We are looking for the best one that still 'touches' the feasible region.

Important Observations:

1. We already know that the optimum occurs at a vertex



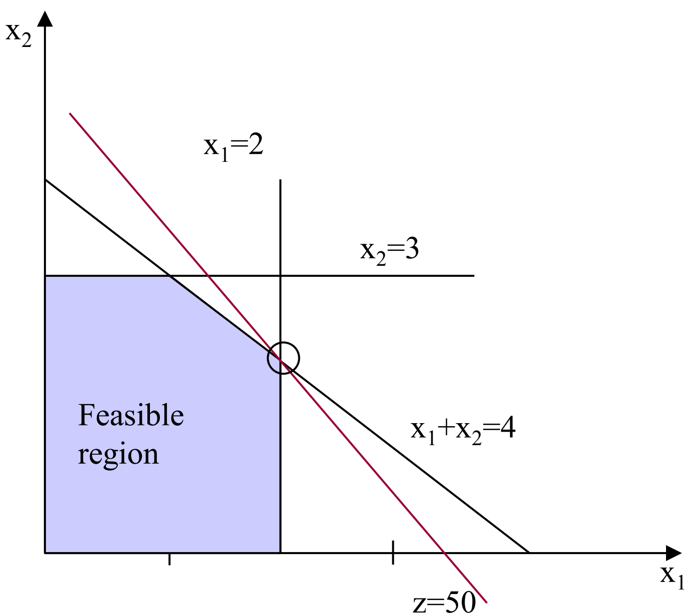
It might be that the objective line is parallel to a constraint.
(e.g. $z=15x_1+15x_2$).

In this case there are many optimal solutions, in particular there is one at a vertex.

Important Observations:

2. If the objective function at a vertex is not smaller than that of any of its adjacent vertices, then it is optimal. (i.e., local optimum is also global)

3. There is a finite number of vertices.



The Simplex method: Travel along the vertices till a local maximum!!!

The Simplex Method

Phase 1 (start-up): Find Any vertex. In standard LPs the origin can serve as the start-up vertex. (why?)

Phase 2 (iterate): Repeatedly move to a better adjacent vertex until no further better adjacent vertex can be found. The optimum is at the final vertex.

Example: The Toy Factory Problem

Objective: $z=15x_1+10x_2$

Phase 1: start at (0,0)

Objective value = Z(0,0)=0

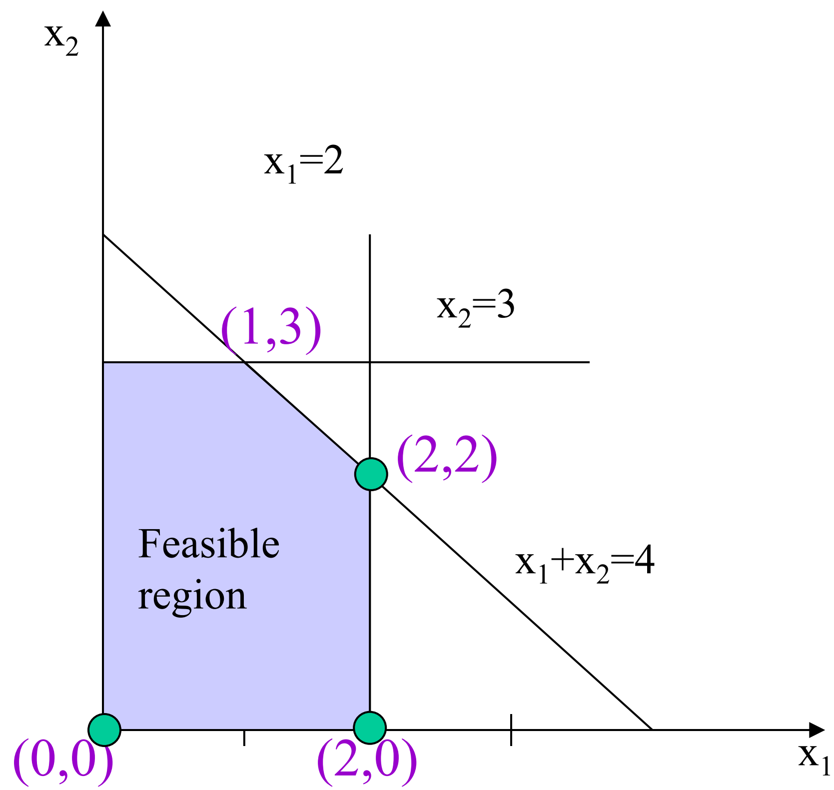
Iteration 1: Move to (2,0).

Z(2,0)=30. An Improvement

Iteration 2: Move to (2,2)

Z(2,2)=50. An Improvement

Iteration 3: Consider moving to (1,3), Z(1,3)=45 < 50. Conclude that (2,2) is optimum!



Finding CornerPoints Algebraically

The simplex method is easy to follow graphically. But how is it implemented in practice?

Notes:

- At a vertex a subset of the inequalities are equalities.
- It is easy to find the intersection of linear equalities (solutio to a system of equations).
- We will add slack variables – to determine which inequality is active and which is not active

Adding Slack Variables

Let s_1, s_2, s_3 be the slack variables

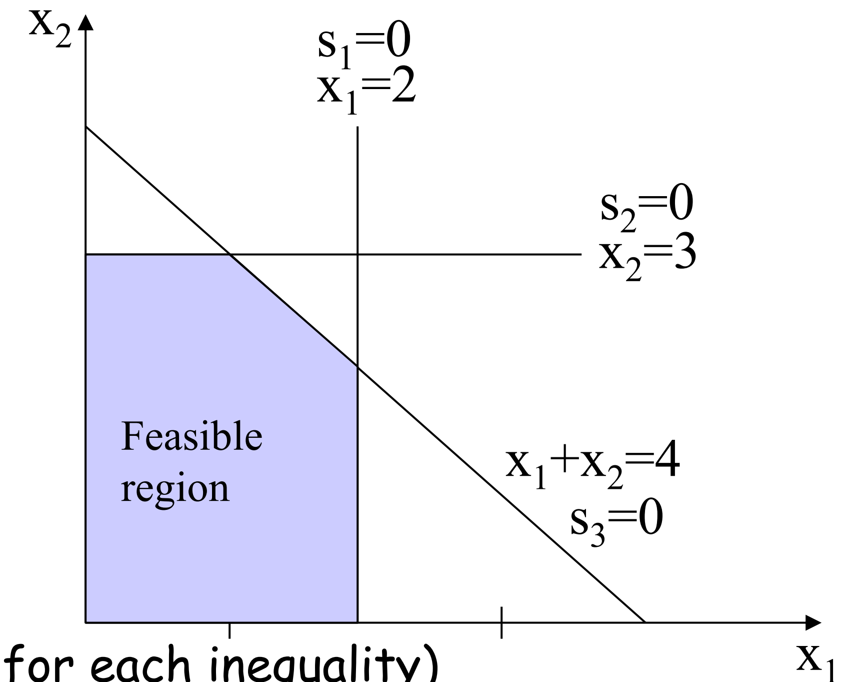
Objective: Max $z = 15x_1 + 10x_2$

s.t $x_1 + s_1 = 2$

$x_2 + s_2 = 3$

$x_1 + x_2 + s_3 = 4$

$x_1, x_2, s_1, s_2, s_3 \geq 0$



n – number of (original) variables

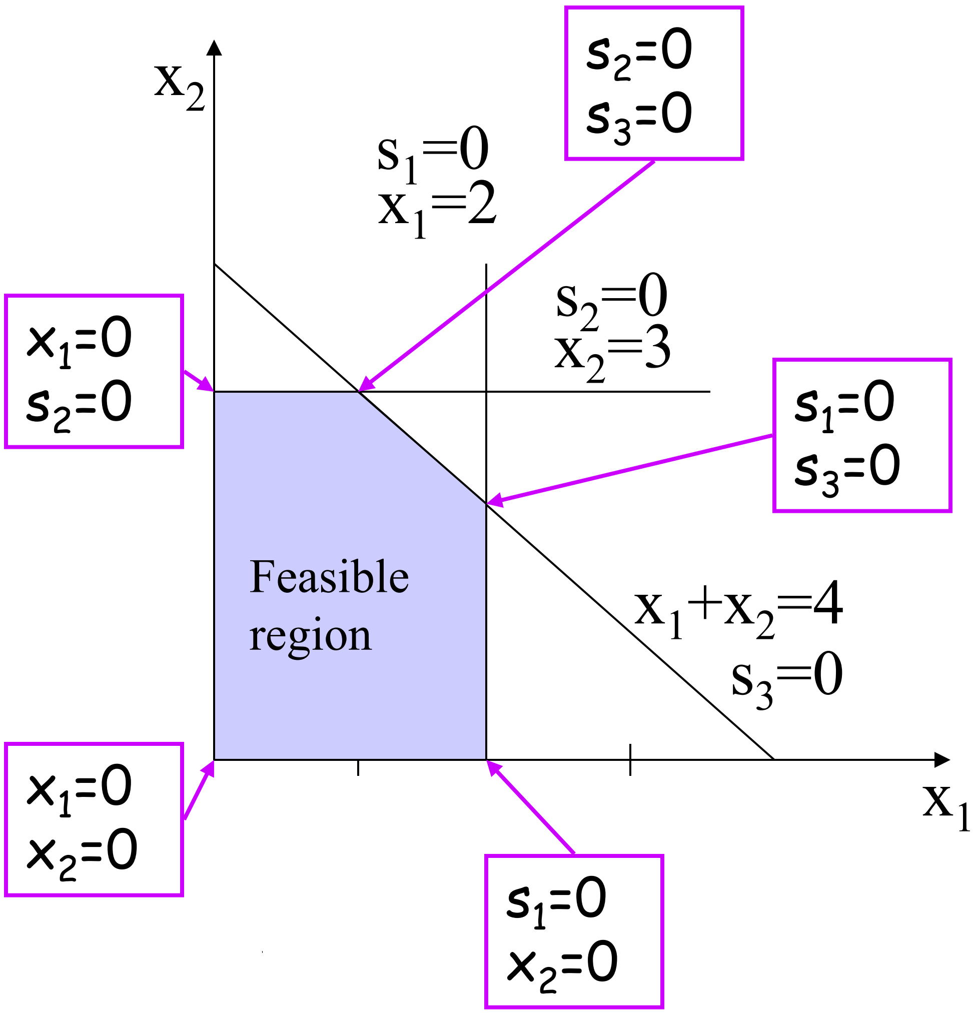
m – number of inequalities

Number of slack variables is m (one for each inequality)

m equations, n+m variables. Setting n vars uniquely determines the values of the other variables.

A vertex: n variables (slack or original) are zero.

Adding Slack Variables



$x_1 + s_1 = 2$

$x_2 + s_2 = 3$

$x_1+x_2 +s_3 = 4$

$x_1, x_2, s_1, s_2, s_3 \geq 0$

Moving between vertices: Decide which two variables are set to zero.

The Simplex Method - Definitions

Nonbasic variable: a variable currently set to zero by the simplex method.

Basic variable: a variable that is not currently set to zero by the simplex method.

The values of basic variable is determined by the nonbasic variables

A basis: The current set of basic variables.

If a slack variable is nonbasic (i.e., is set to zero), the corresponding constraint is active.

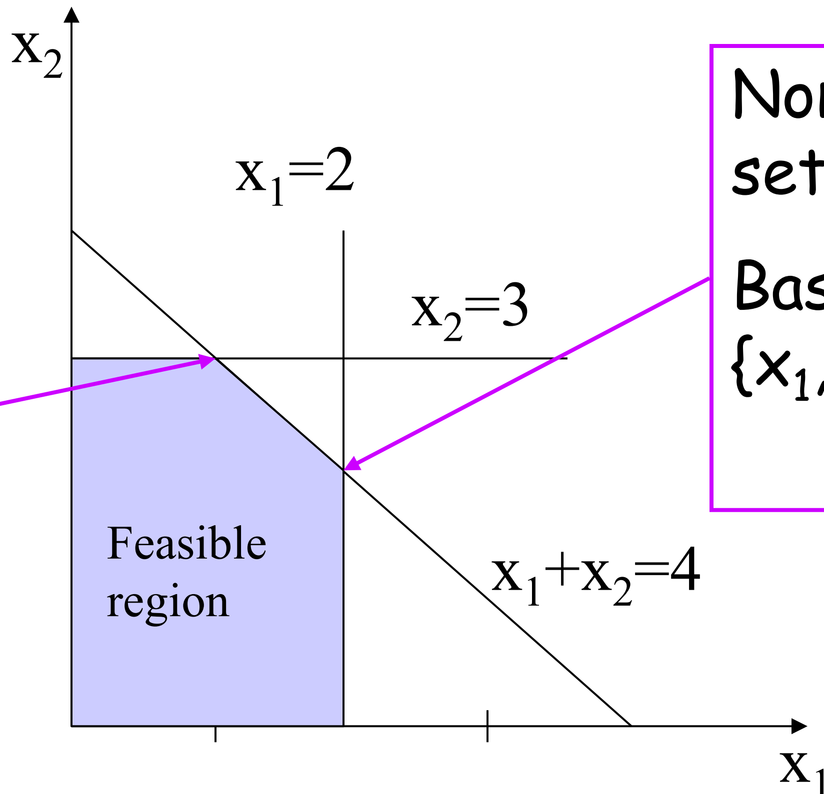
The Simplex Method

In two adjacent vertices, the basis is identical except for one member.

Example:

Nonbasic set: $\{s_2, s_3\}$

Basic set: $\{x_1, x_2, s_1\}$



Nonbasic set: $\{s_1, s_3\}$

Basic set: $\{x_1, x_2, s_2\}$

The Simplex Method

At each step - swap a pair of basic and nonbasic variables

The variable that enters the basic set is the one that yields the greatest improvement to the objective function.

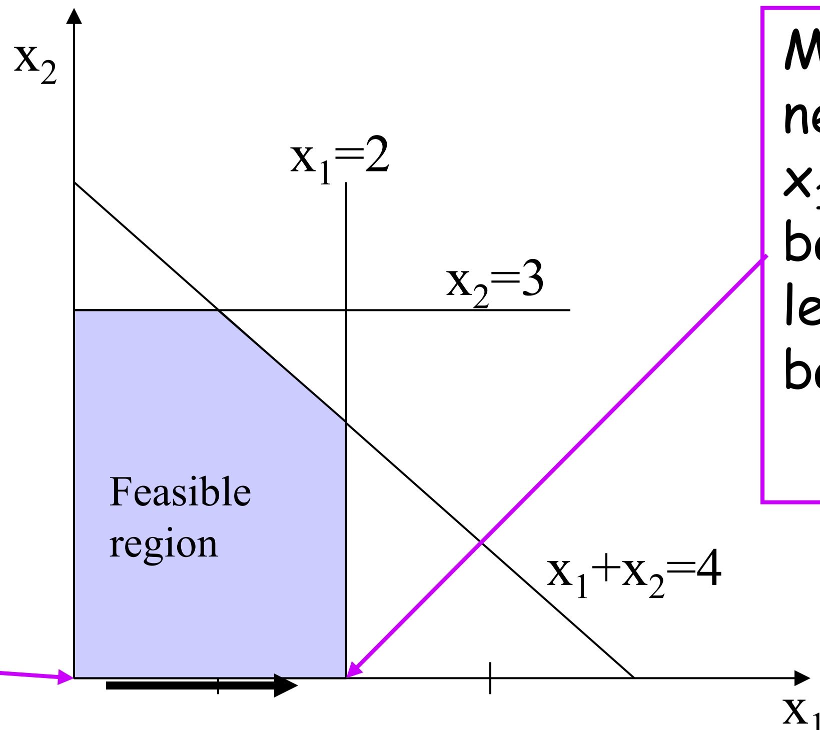
Objective: $z=15x_1+10x_2$

$s_1 = 2 - x_1$

$s_2 = 3 - x_2$

$s_3 = 4 - x_1 - x_2$

$x_1, x_2, s_1, s_2, s_3 \geq 0$



Moving to a new vertex: x_1 enters the basic set, s_1 leaves the basic set

Current vertex

$x_1=x_2=0$

The Simplex Method – more details

Phase 1 (start-up): Initial vertex.

Phase 2 (iterate):

1. Can the current objective value be improved by swapping a basic variable? If not - stop.
2. Select **nonbasic variable to enter basic set**: choose the nonbasic variable that gives the fastest rate of increase in the objective function value.
3. Find the **leaving basic variable** – as we increase the chosen nonbasic variable, the value of the basic variables changes. Move the first one to become zero to the nonbasic set. (aka minimum ratio test).
4. Update the equations to reflect the new basic feasible solution.

The Simplex Method - example (1)

Objective:

$$\text{Max } z=15x_1+10x_2$$

s.t

$$x_1+s_1 = 2$$

$$x_2+s_2 = 3$$

$$x_1+x_2+s_3 = 4$$

$$x_1, x_2, s_1, s_2, s_3 \geq 0$$

Phase 1 (start-up):

Initial vertex: $x_1=0$, $x_2=0$

Basic Variable	Constraint
$s_1=$	$2-x_1$
$s_2=$	$3-x_2$
$s_3=$	$4-x_1-x_2$
$z=$	$15x_1+10x_2$

Phase 2 (iterate):

1. Are we optimal? NO, z's value can increase by increasing both x_1 and x_2.
2. Select entering nonbasic variable: x_1 has a better rate of improving the objective value (15 > 10).

The Simplex Method – example (2)

3. Select the leaving basic variable: The minimum ratio test. We ask: which constraint most limits the increase in the value of the entering basic variable (will first reduce to zero as the value of x_1 increases)?

Answer: For s_1 the ratio is 2/1=2, for s_2 the ratio is infinite, for s_3 the ratio is 4/1=4. s_1 has the smallest ratio.

Basic Variable	Constraint	Bound on Increase
$s_1=$	$2-x_1$	$x_1 \le 2$
$s_2=$	$3-x_2$	No limit
$s_3=$	$4-x_1-x_2$	$x_1 \le 4$
$z=$	$15x_1+10x_2$	

$x_1=2-s_1$

Basic Variable	Constraints
$x_1=$	$2-s_1$
$s_2=$	$3-x_2$
$s_3=$	$2-x_2+s_1$
$z=$	$30-15s_1+10x_2$

4. Update the equations to reflect the new basic feasible solution: $x_1=2$, $x_2=0$, $s_1=0$, $s_2=3$, $s_3=2$. $z=30$. Nonbasic set = $\{s_1, x_2\}$, Basic set = $\{x_1, s_2, s_3\}$,

End of iteration 1.

The Simplex Method - example (3)

Phase 2 (iteration 2):

1. Are we optimal? NO, z's value can increase by increasing the value of x_2. ($z = 30-15s_1+10x_2$)

2. Select entering nonbasic variable: the only candidate is x_2.
3. Select the leaving basic variables: The minimum ratio test. For x_1 the ratio is infinite, for s_2 the ratio is 3/1=3, for s_3 the ratio is 2/1=2. s_3 has the smallest ratio.

Basic Variable	Constraints	Bound on Increase
x_1=	$2-s_1$	No limit
s_2=	$3-x_2$	$x_2 \leq 3$
s_3=	$2-x_2+s_1$	$x_2 \leq 2$
z=	$30-15s_1+10x_2$	

$x_2=2-s_3+s_1$

Basic Variable	Constraints
x_1=	$2-s_1$
s_2=	$1+s_3-s_1$
x_2=	$2-s_3+s_1$
z=	$50-5s_1-10s_3$

4. Update the equations to reflect the new basic feasible solution: $x_1=2$, $x_2=2$, $s_1=0$, $s_2=1$, $s_3=0$. $z=50$. Nonbasic set = $\{s_1, s_3\}$, Basic set = $\{x_1 , s_2, x_3\}$,

End of iteration 2.

The Simplex Method - example (4)

Phase 2 (iteration 3):

1. Are we optimal? YES, z's value cannot increase. ($z=50-5s_1-10s_3$)

End of example.

Simplex Algorithm: Another Example

Maximize $2x_1 + 8x_2$

subject to $10x_1 + 4x_2 \leq 77$

$x_1 + 8x_2 \leq 40$

$x_1, x_2 \geq 0.$

Solution: In Class

The Simplex Algorithm

- Does the simplex algorithm always terminate?
- We improve the objective function at every step, so we don't visit the same vertex twice.
- How many vertices are there?
- How many different basis sets are there?
- $\binom{n+m}{n}$ i.e., an exponential number…

- Indeed, the simplex algorithm is not polynomial.
- However, it is polynomial on most inputs and is fast in practice.
- The ellipsoid and interior point methods are polynomial.

Remarks

- For simplicity, we assumed that the matrix A is non-singular (otherwise omit linearly dependent constraints)
- If the value of some basic variable happens to already be zero, a simplex step may not increase the objective function. This is called a degenerate step. Need to handle these cases properly to make sure the algorithm doesn't cycle forever.
  We do not worry about degenerate steps in this class.

Example: Vertex Cover

Write the minimum vertex cover problem as a linear program

Example: Vertex Cover

Variables: for each $v \in V$, x_v - is v in the cover?

Minimize $\Sigma_v x_v$

Subject to: $x_i + x_j \geq 1 \quad \forall_{\{i,j\} \in E}$

$x_v \in \{0,1\}$

Integer Programming (IP)

- An LP problem with an additional constraint that variables will only get an integral value, maybe from some range.
- BIP – binary integer programming: variables should be assigned only 0 or 1.
- Can model many problems.
- NP-hard to solve!

BIP Example: Set Cover

Input: a Collection $S_1, S_2, \ldots, S_n$ of subsets of $\{1,2,3,\ldots,m\}$ a cost p_i for set S_i.

Output: A collection of subsets whose union is $\{1,2,\ldots,m\}$.

Objective: Minimum total cost of selected subsets.

Variables: For each subset, x_i – is subset S_i selected for the cover?

Minimize $\sum_i p_i \cdot x_i$

Subject to: $x_i \in \{0,1\}$

$$\forall j = 1 \ldots m: \sum_{i: j \in S_i} x_i \geq 1$$

BIP Example: Shortest Path

Given a directed graph $G(V,E)$, $s,t \in V$ and nonnegative length p_e for each edge e.

Variables: For each edge, x_e – is e in the path?

Minimize $\quad \sum_e p_e \cdot x_e$

Subject to: $\quad x_e \in \{0,1\}$

$$\sum_v x_{(s,v)} - \sum_u x_{(u,s)} \geq 1$$

$$\sum_u x_{(u,t)} - \sum_v x_{(t,v)} \geq 1$$

$$\forall u \in V-\{s,t\}: \quad \sum_p x_{(u,p)} - \sum_q x_{(q,u)} = 0$$

At least 1 more edge leaving s than entering

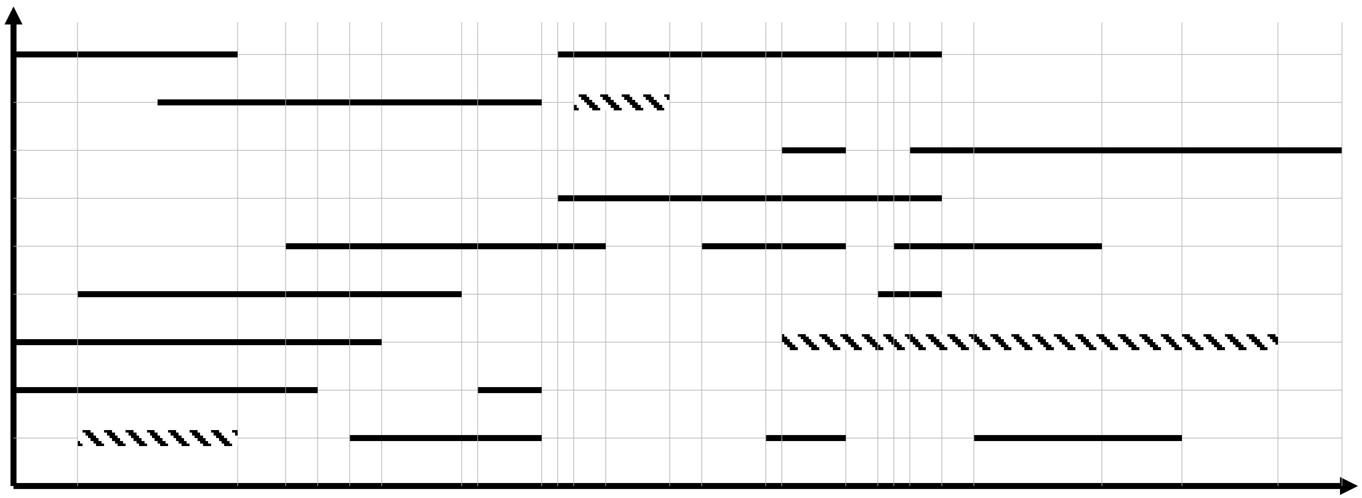
At least 1 more edge entering t than leaving

All other nodes have same number of edges entering and leaving

BIP example: Single machine scheduling of interval jobs.

- Schedule jobs (activities) on a single processor
- Each job can be scheduled in one of a finite collection of allowed time intervals
- Scheduling job j at interval I imposes $w(I)$ load, and yields a profit $p(I)$
- Find a maximum profit subset of intervals, at most one interval per job, such that the total load at each time is at most 1.
- Variables: x_I – for each possible interval I.

Single Machine Scheduling :



<u>Maximize</u> $\sum_I p(I) \cdot x_I$

s.t For each interval *I*: $x_I \in \{0,1\}$

For each time *t*: $\sum_{I:s(I)\leq t<e(I)} w(I) \cdot x_I \leq 1$

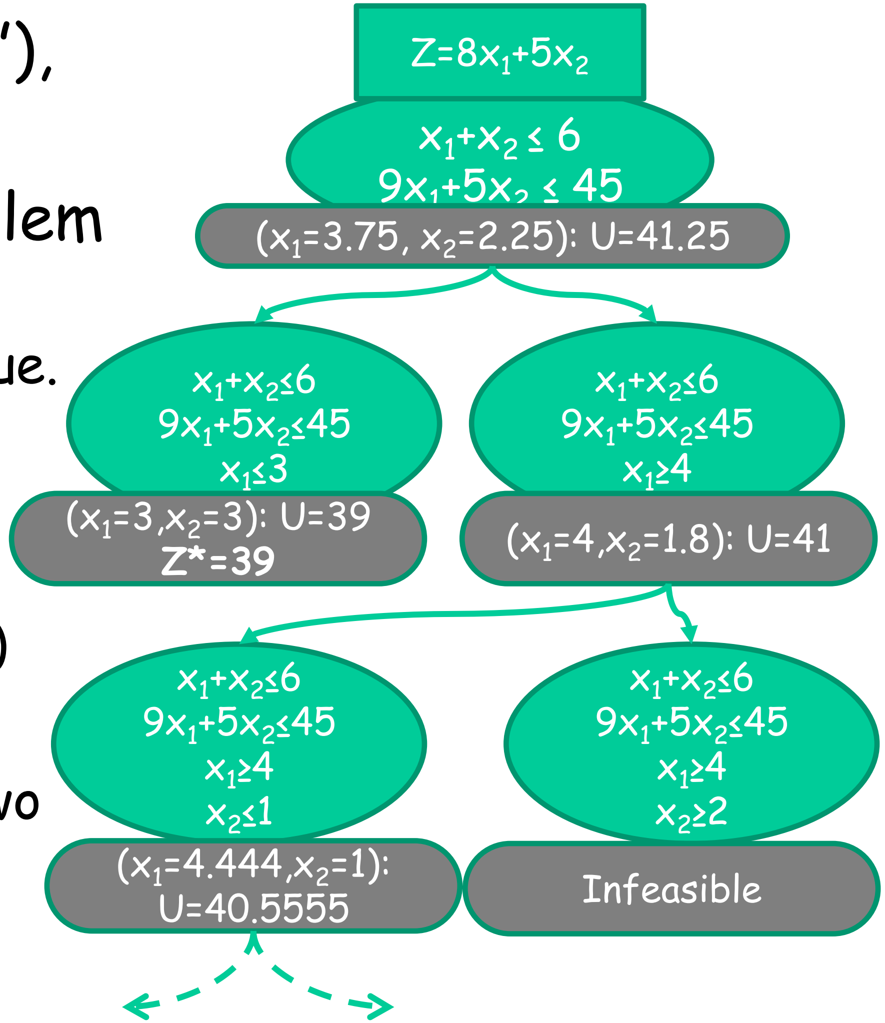
For each activity *A*: $\sum_{I\in A} x_I \leq 1$

Solving IPs is NP-Hard
What can we do?

- Heuristics
- Approximation algorithms
- Exploit special structure

Solving IP using Branch and Bound (described for maximization problems)

1. Set $Z^*=-\infty$ ("incumbent value"), Current node=root
2. Bound: Solve relaxed LP problem
   1. If infeasible, prune. Else, let U be the objective value.
      - U is an upper bound on OPT.
   2. If $U < Z^*$, prune. Else,
   3. If all variables are integral: Update Z^* to new value. (prune)
3. Branch: Select a leaf with a non-integral variable, x^*, branch into two sub-LPs: $x \le \lfloor x^* \rfloor$ and $x \ge \lceil x^* \rceil$.



Dakin's Algorithm

Animated Solution

More Branch and Bound examples

Maximize $8x_1 + 5x_2$

subject to $x_1 + x_2 \leq 6$

$9x_1 + 5x_2 \leq 45$

$x_1, x_2 \geq 0$ and integers.

Solution

Branch according to the binary value of a variable example.

Weighted Vertex Cover

Input: Graph $G=(V,E)$ with non-negative weights $w(v)$ on the vertices.

Goal: Find a minimum-cost set of vertices S, such that all the edges are covered. An edge is covered iff at least one of its endpoints is in S.

Recall: Vertex Cover is NP-complete.

The best known approximation factor is $2- (\log \log |V|/2 \log|V|)$.

Weighted Vertex Cover

Variables: for each $v \in V$, $x(v)$ – is v in the cover?

Min $\Sigma_{v \in V} w(v)x(v)$

s.t.

$x(v) + x(u) \geq 1, \ \forall (u,v) \in E$

$x(v) \in \{0,1\} \quad \forall v \in V$

The LP Relaxation

This is **not** a linear program: the constraints of type $x(v) \in \{0,1\}$ are not linear. We got an LP with integrality constraints on variables – an **integer linear programs (IP)** that is NP-hard to solve.

However, if we replace the constraints $x(v) \in \{0,1\}$ by $x(v) \geq 0$ and $x(v) \leq 1$, we will get a linear program.

The resulting LP is called a **Linear Relaxation** of IP, since we relax the integrality constraints.

LP Relaxation of Weighted Vertex Cover

Min $\Sigma_{v \in V} w(v)x(v)$

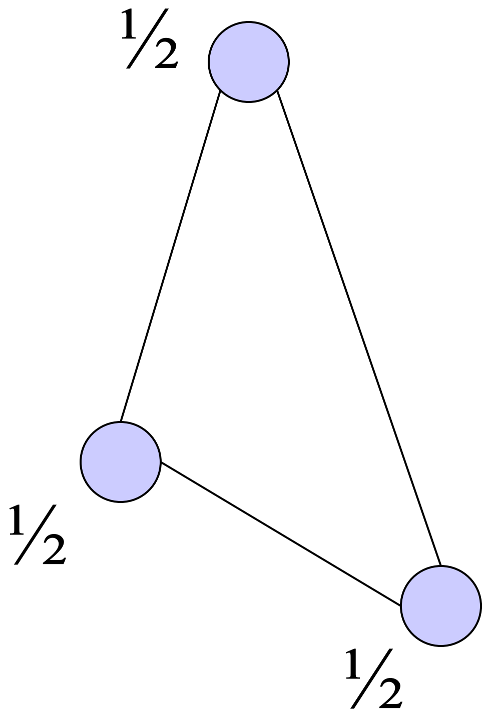
s.t.

$x(v) + x(u) \geq 1, \ \forall (u,v) \in E$

$x(v) \geq 0, \ \forall v \in V$

$x(v) \leq 1, \ \forall v \in V$

LP Relaxation of Weighted Vertex Cover - example



Consider the case of a 3-cycle in which all weights are 1.

An optimal VC has cost 2 (any two vertices)

An optimal relaxation has cost 3/2 (for all three vertices x(v)=1/2)

The LP and the IP are different problems. Can we still learn something about Integral VC?

Why LP Relaxation Is Useful ?

The optimal value of LP-solution provides a bound on the optimal value of the original optimization problem. OPT(LP) is always better than OPT(IP) (why?)

Therefore, if we find an integral solution within a factor r of OPT_{LP}, it is also an r-approximation of the original problem.

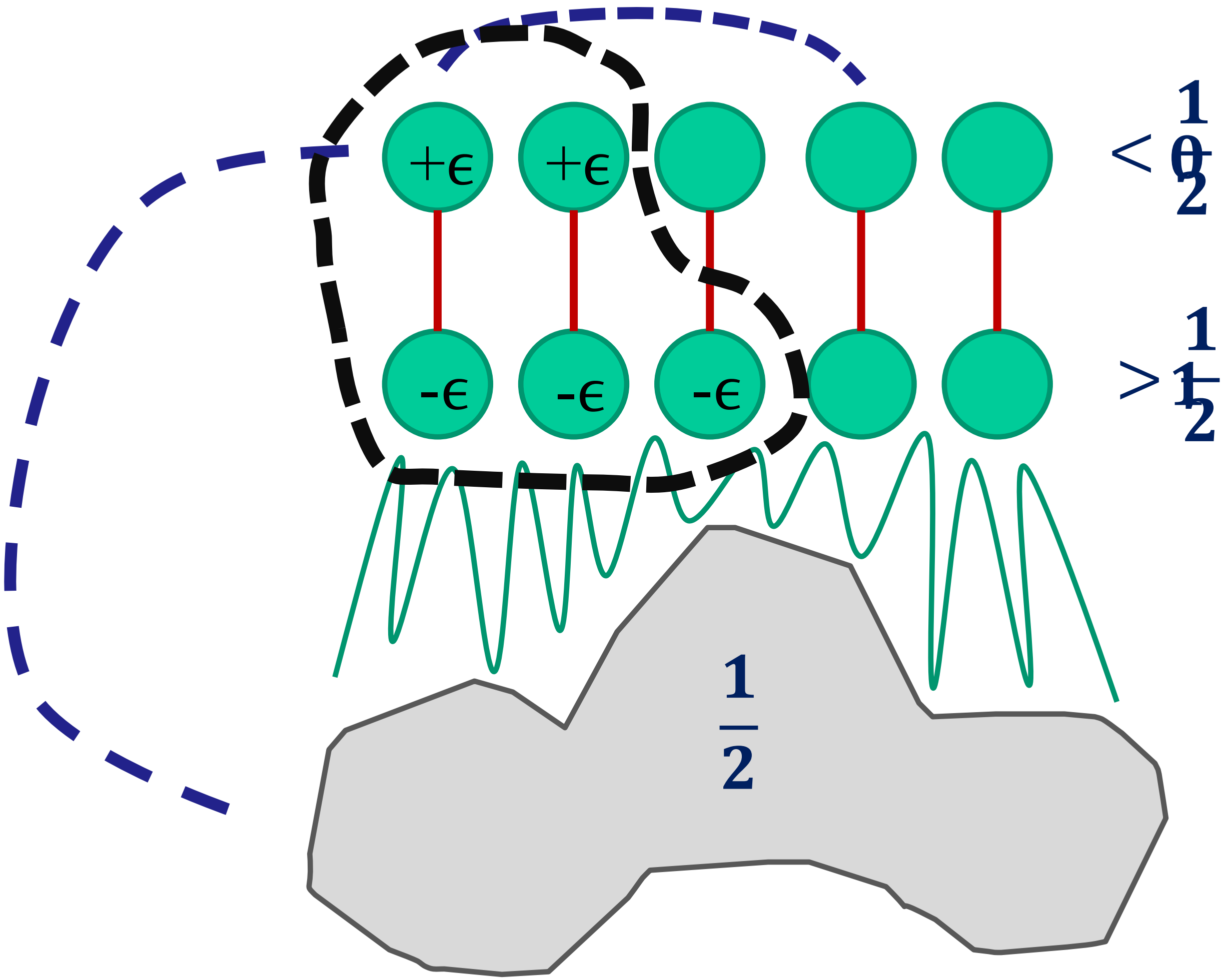
It can be done by 'wise' rounding.

Nemhauser Trotter Theorem

(a) There is always an optimal solution to Vertex Cover LP that sets variables to $\{0, \frac{1}{2}, 1\}$.

(b) For any $\{0, \frac{1}{2}, 1\}$-solution there is a matching from the 1-vertices to the 0-vertices, saturating the 1-vertices (i.e., every 1-vertex is matched).

Nemhauser Trotter Proof



2-approx. For Vertex Cover

Nemhauser-Trotter:

There is an optimal solution to Vertex Cover LP that sets variables to $\{0, \frac{1}{2}, 1\}$.

$\Rightarrow$ 2-approx algorithm:

Find optimal solution x* to LP relaxatoin.
Let y(v)=1 if $x^*(v) \neq 0$. y(v) = 0 otherwise.

y is a solution for VC IP (why?)

$\Sigma\, w(v)y(v) \leq \Sigma\, w(v)2x(v) = 2OPT_{LP} \leq 2OPT$

Even if we do not know the Nemhauser-Trotter thm!

1. Solve the LP-Relaxation.

2. Let S be the set of all the vertices v with $x(v) \geq 1/2$. Output S as the solution.

Analysis: The solution is feasible: for each edge $e=(u,v)$, either $x(v) \geq 1/2$ or $x(u) \geq 1/2$

The value of the solution is: $\Sigma_{v \in S} w(v) = \Sigma_{\{v | x(v) \geq 1/2\}} w(v) \leq \Sigma_{v \in V} w(v) 2x(v) = 2OPT_{LP}$

Since $OPT_{LP} \leq OPT_{VC}$, the cost of the solution is $\leq 2OPT_{VC}$.

LP Duality

Consider LP: max $\mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq 0$
n variables, m constraints

How large can the optimum be?

Take a vector $\mathbf{y}$ of m variables.
If $\mathbf{y} \geq 0$ then $\mathbf{y}^T A\mathbf{x} \leq \mathbf{y}^T\mathbf{b}$
If $\mathbf{c}^T \leq \mathbf{y}^T A$ then $\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T A\mathbf{x}$

So $\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T A\mathbf{x} \leq \mathbf{y}^T\mathbf{b}$
How small can $\mathbf{y}^T\mathbf{b}$ be?

minimize $\mathbf{b}^T\mathbf{y}$ s.t. $A^T\mathbf{y} \geq \mathbf{c}$, $\mathbf{y} \geq 0$ (called the dual LP)

Duality

Primal: maximize $\mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0$
Dual: minimize $\mathbf{b}^T\mathbf{y}$ s.t. $A^T\mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq 0$

- In the primal, **c** is cost function and **b** was in the constraint. In the dual, their roles are swaped.
- Inequality sign is changed and minimization turns to maximization.

Dual:

minimize $2x + 3y$

s.t
$$\begin{aligned} x + 2y &\geq 4, \\ 2x + 5y &\geq 1, \\ x - 3y &\geq 2, \\ x, y &\geq 0 \end{aligned}$$

Primal:

maximize $4p + q + 2r$

s.t
$$\begin{aligned} p + 2q + r &\leq 2, \\ 2p + 5q - 3r &\leq 3, \\ p, q, r &\geq 0 \end{aligned}$$

Duality - general form

Primal	Max $c^T x$	Min $b^T y$	Dual
	$\leq b_i$	≥ 0	
Constraints	$\geq b_i$	≤ 0	Variables
	$= b_i$	unconstrained	
	≤ 0	$\leq c_i$	
Variables	≥ 0	$\geq c_i$	Constraints
	unconstrained	$= c_i$	

max $\mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq 0$

If $\mathbf{y} \geq 0$ then $\mathbf{y}^T A\mathbf{x} \leq \mathbf{y}^T\mathbf{b}$

The Duality Theorem

Let P,D be an LP and its dual.
If one has optimal solution so does the other, and their values are the same.

We only saw $\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{b}$ (weak duality)
The duality thm: $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$ (proof not here)

Simple Example

- Diet problem: minimize $2x + 3y$
  subject to $x+2y \geq 4$,
  $x \geq 0, y \geq 0$

Peanut Butter

Steak

- Dual problem: maximize $4p$
  subject to $p \leq 2$,
  $2p \leq 3$,
  $p \geq 0$
- Dual: the problem faced by a pharmacist who sells synthetic protein, trying to compete with peanut butter and steak

Simple Example

- The pharmacist wants to maximize the price p, subject to constraints:
  - synthetic protein must not cost more than protein available in foods.
  - price must be non-negative
  - revenue to druggist will be 4p
- Solution: $p = 3/2$ → objective value = $4p = 6$
- Not coincidence that it's equal the minimal cost in original problem.

What's going on?

- Notice: feasible sets completely different for primal and dual, but nonetheless an important relation between them.
- Duality theorem says that in the competition between the grocery and the pharmacy the result is always a tie.
- Optimal solution to primal tells consumer what to do.
- Optimal solution to dual fixes the natural prices at which economy should run.

Duality Theorem

Druggist's max revenue = Consumers min cost

Practical Use of Duality:

- Sometimes simplex algorithm (or other algorithms) will run faster on the dual than on the primal.
- Can be used to bound how far you are from optimal solution.
- Interplay between primal and dual can be used in designing algorithms
- Important implications for economists.

Max Flow LP and its dual

Consider the max st-flow LP (add an arc from t to s):

$$\max\ f_{ts}\quad s.t.$$

$$f_{uv} \le c_{uv} \quad \forall uv \in E$$

$$\sum_{uv\in E} f_{uv} - \sum_{vu\in E} f_{vu} \le 0 \quad \forall v \in V$$

$$f_{uv} \ge 0$$

$$\min \sum_{uv\in E} c_{uv} d_{uv} \quad s.t.$$

$$d_{uv} - p_u + p_v \ge 0 \quad \forall uv \in E$$

$$p_s - p_t \ge 1$$

$$d_{uv} \ge 0\,, p_u \ge 0$$

IP version of dual = min st-cut

$$\min \sum_{uv \in E} c_{uv} d_{uv} \quad s.t.$$

$$d_{uv} - p_u + p_v \geq 0 \quad \forall uv \in E$$

$$p_s - p_t \geq 1$$

$$d_{uv} \in \{0,1\}, p_u \in \{0,1\}$$

Consider optimal solution (d^*, p^*): $p_s = 1, p_t = 0$
p^* naturally defines a cut: $S = \{v\colon p_v^* = 1\}, T = \{v\colon p_v^* = 0\}$
For $u \in S, v \in T$: $d_{uv}^* = 1$ for other $\mathrm{u}v$ can have $d_{uv}^* = 0$
So objective function is capacity of the minimum st-cut!

Back to LP Dual – still min-cut?

$$\min \sum_{uv \in E} c_{uv} d_{uv} \quad s.t.$$

$$d_{uv} - p_u + p_v \geq 0 \quad \forall uv \in E$$

$$p_s - p_t \geq 1$$

$$0 \leq d_{uv} \leq 1, 0 \leq p_u \leq 1$$

Dropping the upper bounds $d_{uv} \leq 1, p_u \leq 1$ cannot increase the objective value.

Can the objective function be improved when dropping the integrality constraints? In general – yes.
This specific matrix has a special property called total unimodularity
Such LPs have integral optimal solutions.
So optimum of dual LP remains value of min st-cut
By duality theorem: max-flow = mim-cut

Linear Programming -Summary

- Of great practical importance to solve linear programs:
  - they model important practical problems
    - production, manufacturing, network design, flow control, resource allocation.
  - solving an LP is often an important component of solving or approximating the solution to an **integer linear programming problem.**
- Can be solved in poly-time, the simplex algorithm works very well in practice.
- Use packages, you really do not want to roll your own code here.