Parameterized Complexity

Every instance comes with a parameter k.

Often k is solution size, but could be many other things

The problem is fixed parameter tractable (FPT) if exists algorithm with running time $f(k)n^c$.

So Vertex Cover parameterized by solution size is fixed parameter tractable.

Alternative Parameters

So far we have only seen the solution size as the parameter.

Often other parameters also make sense, or even make more sense than solution size.

k-Coloring

A valid k-coloring is a funcion $f : V(G) \rightarrow \{1...k\}$ such that no edge has same colored endpoints.

Input: G, k

Question: Does G have a valid k-coloring?

Parameter: k

Cannot have FPT algorithm – NP-hard for k=3!

k-Coloring parameterized by VC

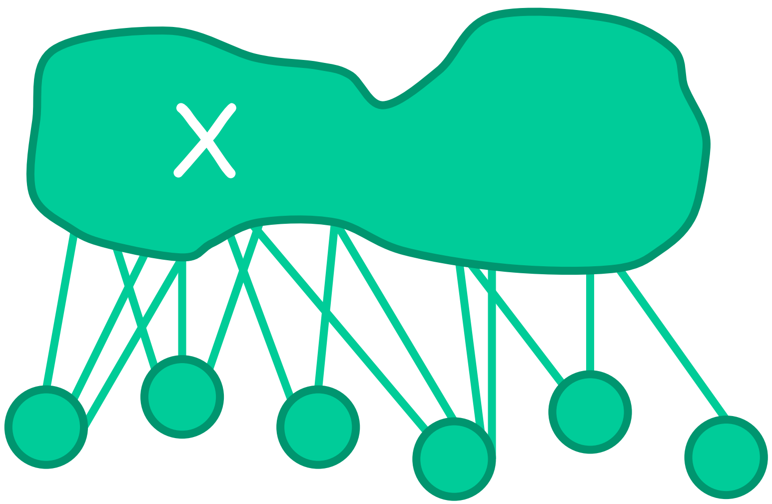
Input: G, integer k, set $X \subseteq V(G)$ such that X is a vertex cover of G, integer $x = |X|$.

Question: Does G have a proper k-coloring?

Parameter: x

FPT now means $f(x)n^{O(1)}$.

k-Coloring parameterized by VC



$I = V(G) \setminus X$

If $x+1 \leq k$ say YES

Thus, assume $k \leq x$.

Branch on k^x colorings of X.

For each guess, color I greedily.

Total running time: $O(k^x \cdot (n+m)) = O(x^x \cdot (n+m))$.

Dynamic Programming

Steiner Tree

Input: Graph G, vertex set Q, integer k.

Question: Is there a set S of size at most k such that $Q \subseteq S$ and $G[S]$ is connected?

Parameter: $|Q|$

Will see $4^{|Q|}n^{O(1)}$ time algorithm.

DP for Steiner Tree

vertex

$\leq k$, solution size

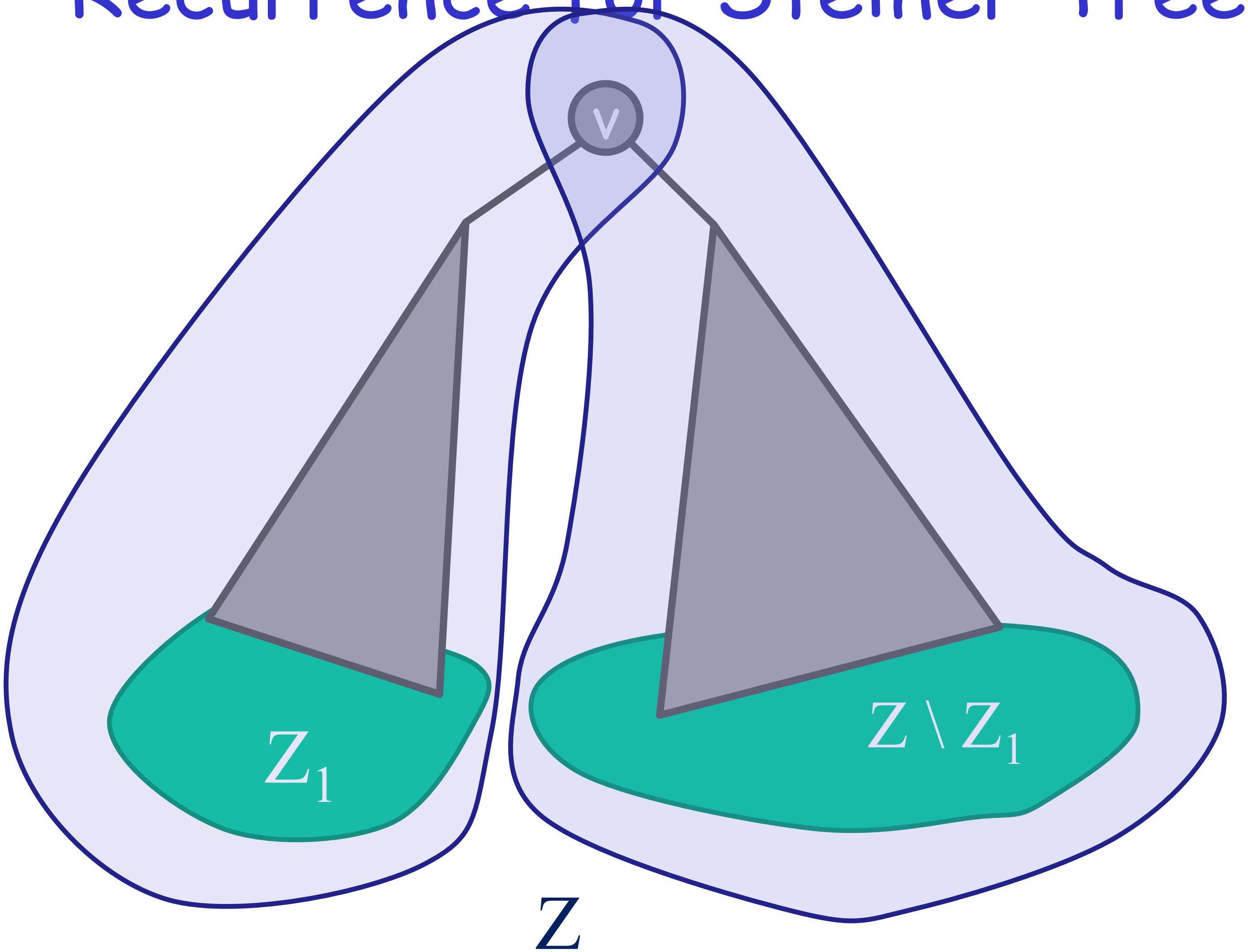
$\subseteq Q$

$T[v,p,Z]$ = **True** if there exists a set S of size at most p such that that $Z \cup \{v\} \subseteq S$ and $G[S]$ is connected.

Table size is $2^{|Q|}kn$

We want to know the minimum p such that $T[v,p,Q] = \text{true}$, for some $v \in V(G)$

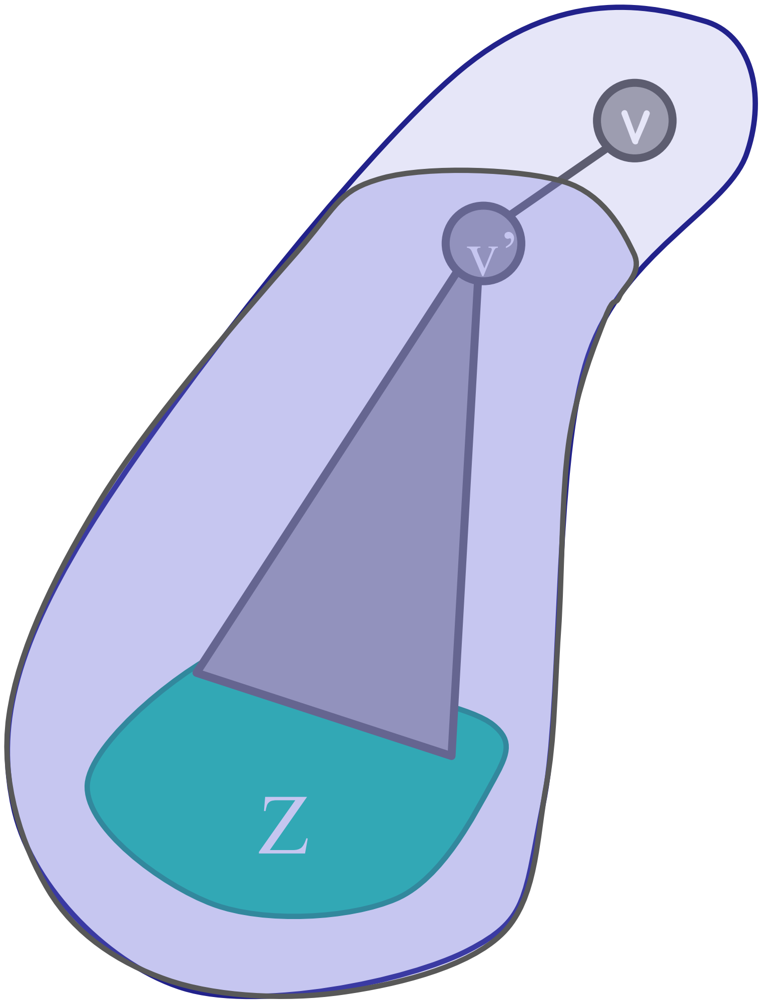
Recurrence for Steiner Tree



Recurrence for Steiner Tree

$$T[v,p,Z] = \bigvee_{1\leq p_1 \leq p} \; \bigvee_{\emptyset \subset Z_1 \subset Z} \begin{array}{c} T[v,p_1,Z_1] \\ + T[v,p-p_1+1,Z \setminus Z_1] \end{array}$$

Recurrence for Steiner Tree



Recurrence for Steiner Tree

$$T[v,p,Z] = \bigvee_{1 \le p_1 \le p} \bigvee_{\emptyset \subset Z_1 \subset Z} T[v,p_1,Z_1] + T[v, p - p_1 + 1, Z \setminus Z_1] \;\; \bigvee_{u \in N(v)} T[u, p-1, Z]$$

Steiner Tree, Analysis

Table size: $2^{|Q|}nk$

Time to fill one entry: $O(k2^{|Q|} + n)$

Total time: $O(4^{|Q|}nk^2 + 2^{|Q|}n^2k)$

Treewidth as parameter

We saw a $O(4^{k}\cdot n)$ time algorithm for max. independent set in treewidth k graphs

This is an example for a very broad situation:

Croucelle's theorem: any graph property expressible in monadic second order logic is FPT with treewidth as the parameter (by DP)

Monadic second order logic:

- Quantification over vertices, sets of vertices, edges, sets of edges.
- Adjacency and incidence checks
- Or, and, not

Example MSO2:

3 coloring:

$\exists X \subseteq V, Y \subseteq V$ s.t.

$[\ (\forall\, x \in X\ x \notin Y)$ X,Y are disjoint

$\quad \wedge\ \forall u,v:$

$\quad\quad \{\ (u,v) \in E \Rightarrow [\quad (u \in X \Rightarrow v \notin X)$ endpoints of an edge

$\quad\quad\quad \wedge\ (u \in Y \Rightarrow v \notin Y)$ do not have same color

$\quad\quad\quad \wedge\ (u \in V-(X \cup Y) \Rightarrow v \notin V-(X \cup Y))$

$\quad\quad\quad]$

$\quad \}$

$]$

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:

$$x + 2y \geq 4$$
$$x \geq 0, \quad y \geq 0$$

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:

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

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.

$Ax \le b$

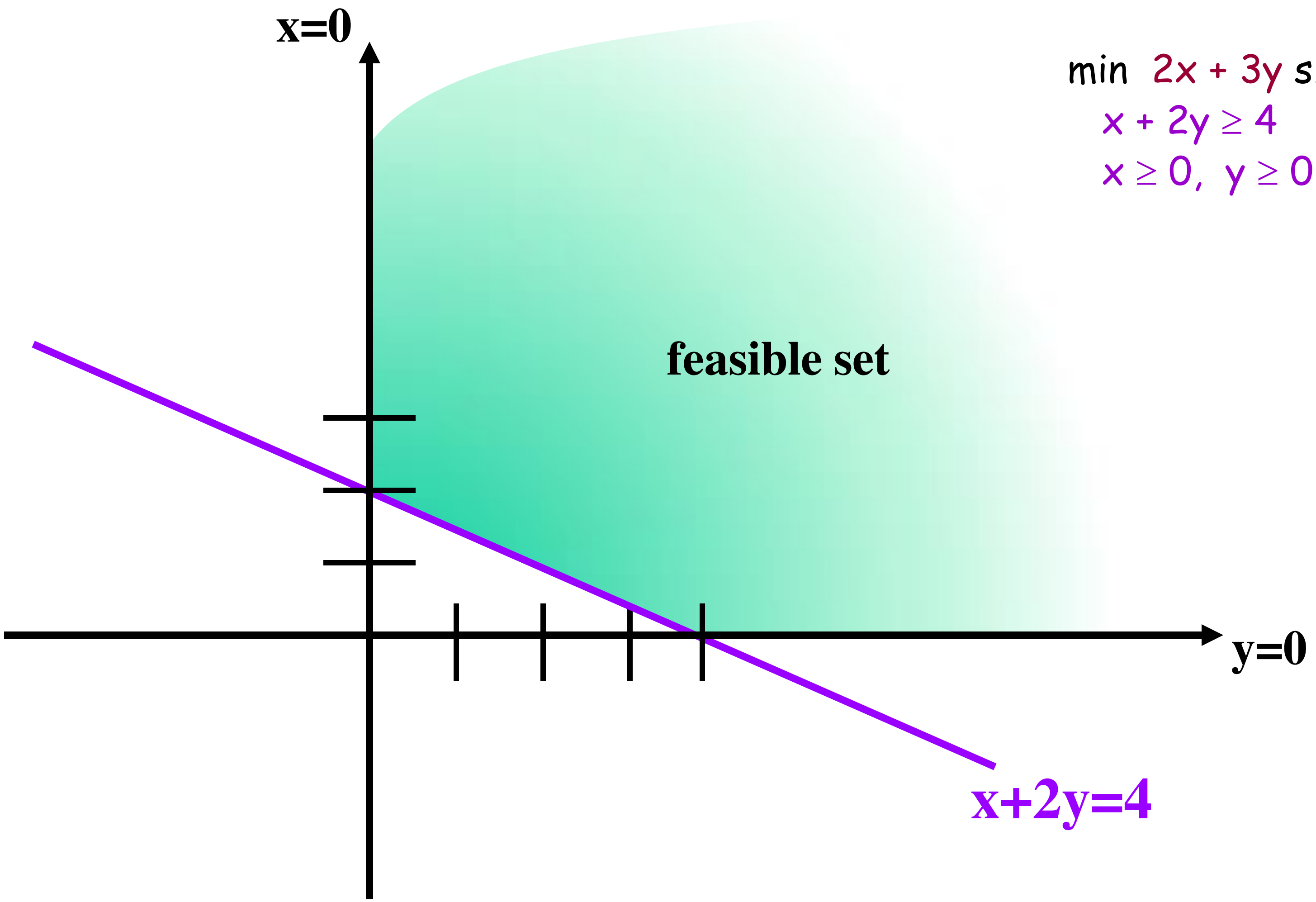
x – vector of n variables

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.

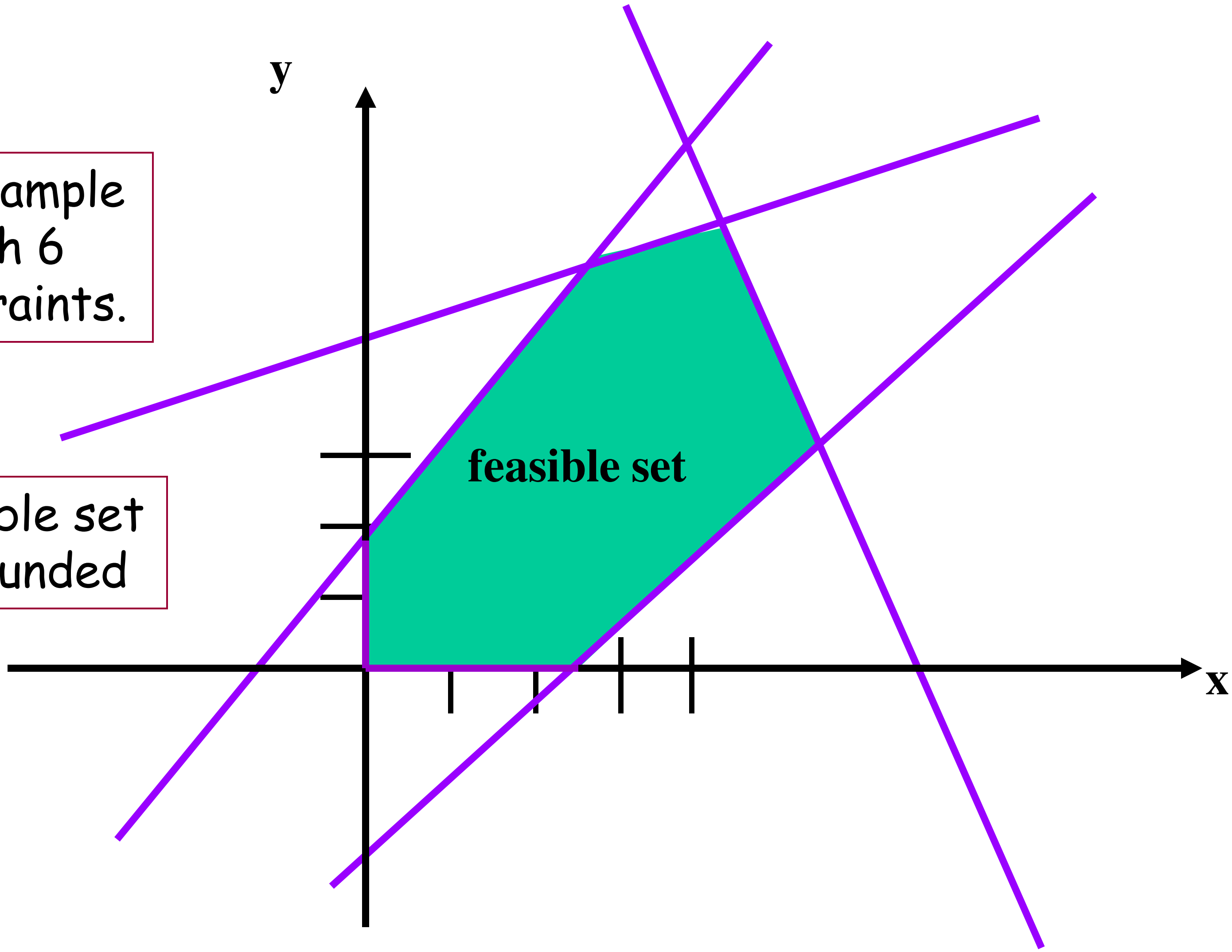
- Fact: every point p in a convex polytope can be represented as a convex combination of the vertices v_i of the polyhedron.

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

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.

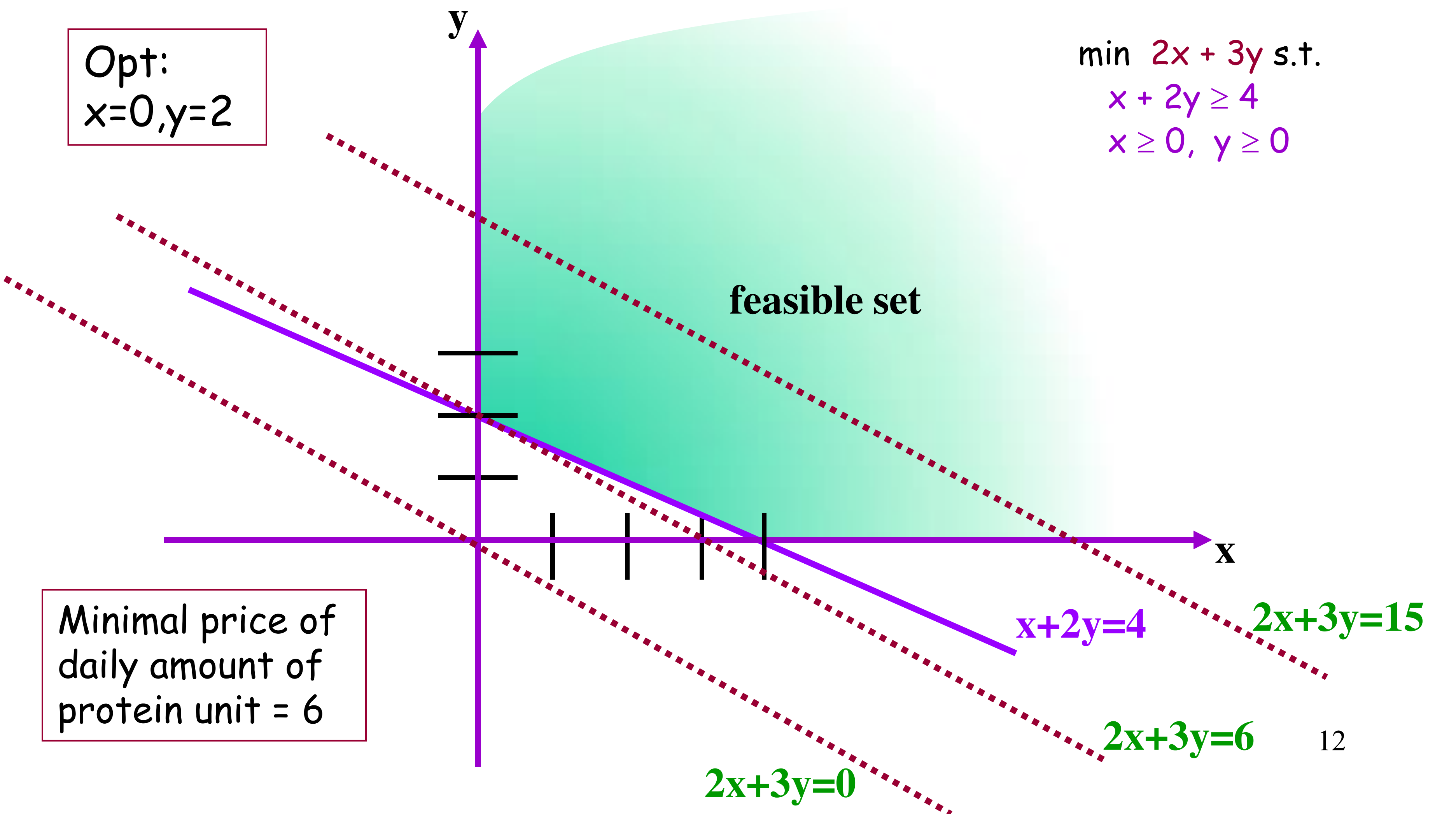
There are 3 cases:

- 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.
$x + 2y \geq 4$
$x \geq 0,\ y \geq 0$

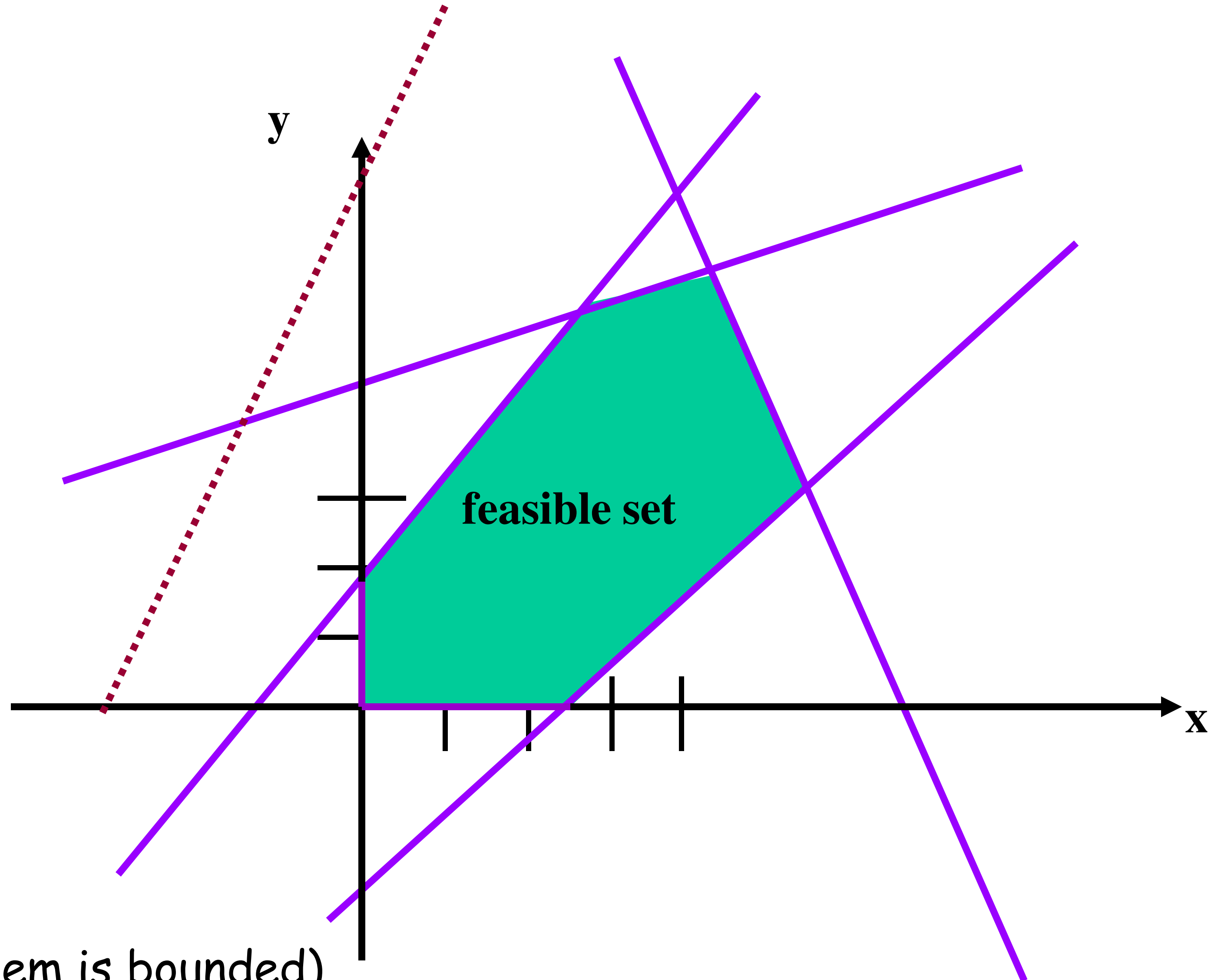
Minimal price of daily amount of protein unit = 6

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 $p = \sum \lambda_i v_i$ $\quad (0 \le \lambda_i \le 1\,;\, \sum \lambda_i = 1)$
By linearity of the objective function z,
$z(p) = \sum \lambda_i z(v_i) \le z(v_{max})$, 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 $\sum_{j=1}^{n} c_j x_j$

subject to:

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

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

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

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 $z = 15x_1 + 10x_2$

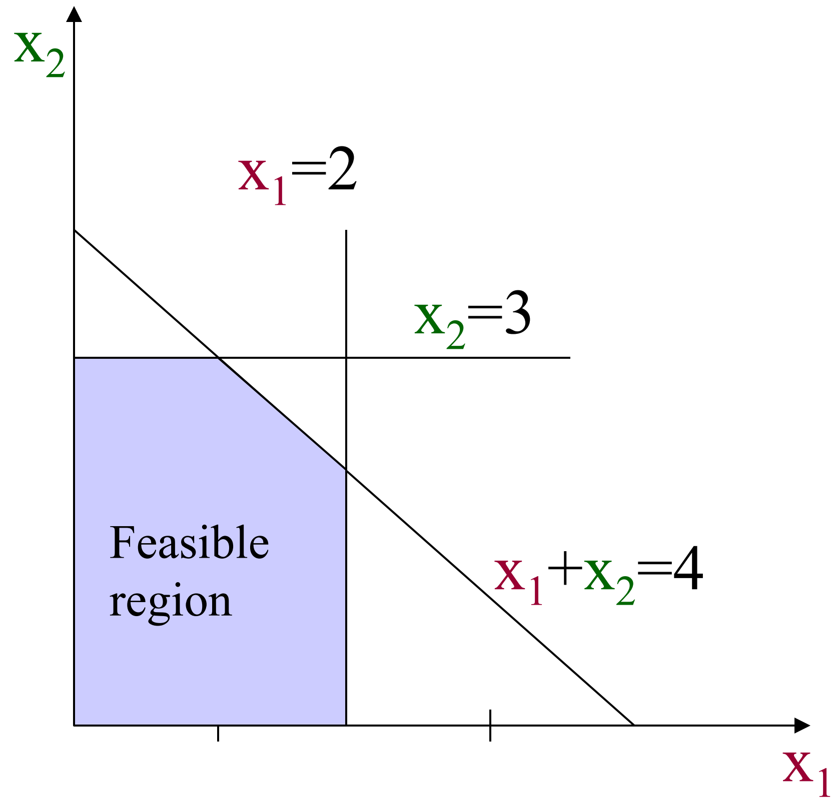
s.t $x_1 \leq 2$

$x_2 \leq 3$

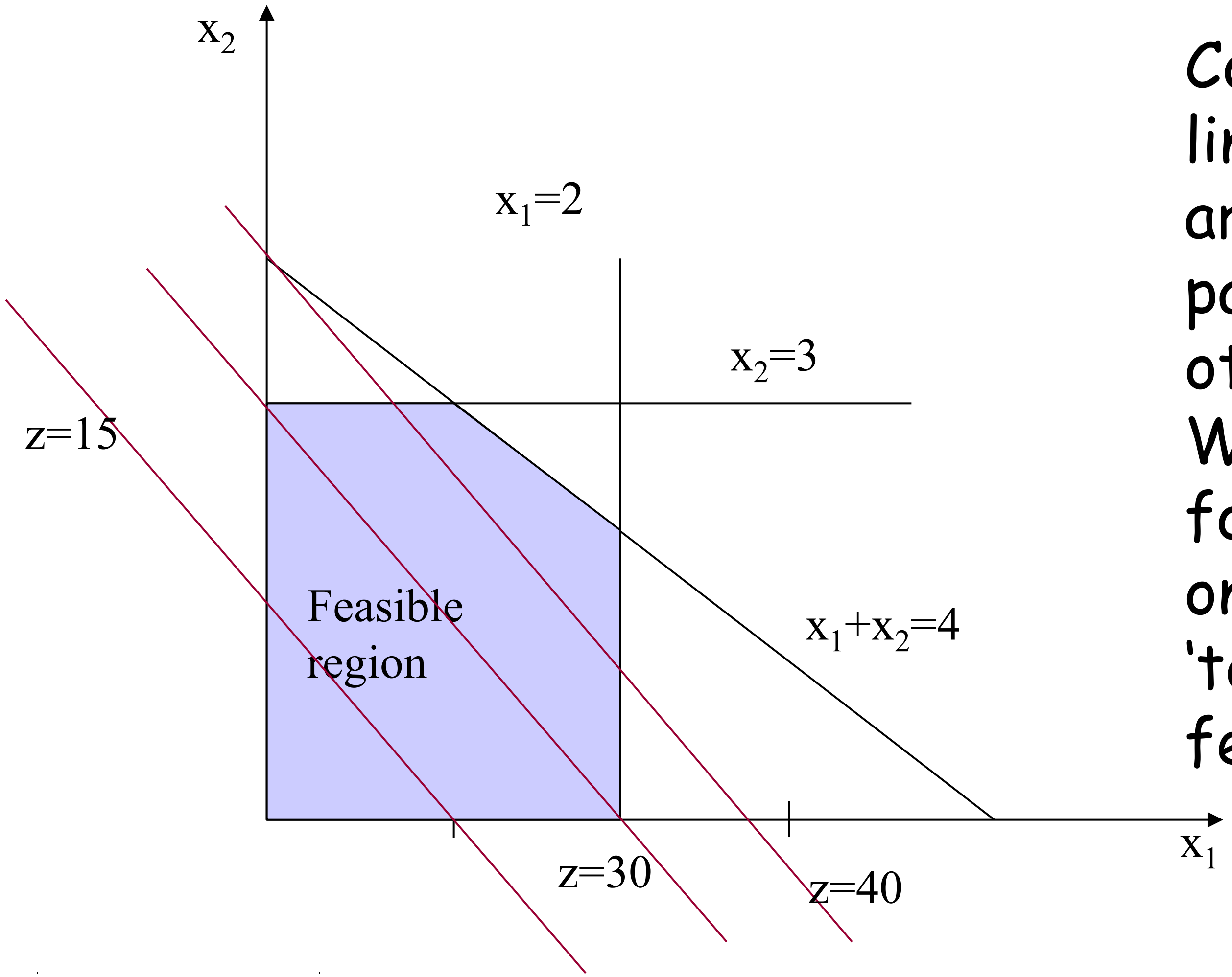
$x_1 + x_2 \leq 4$

$x_1 \geq 0$

$x_2 \geq 0$



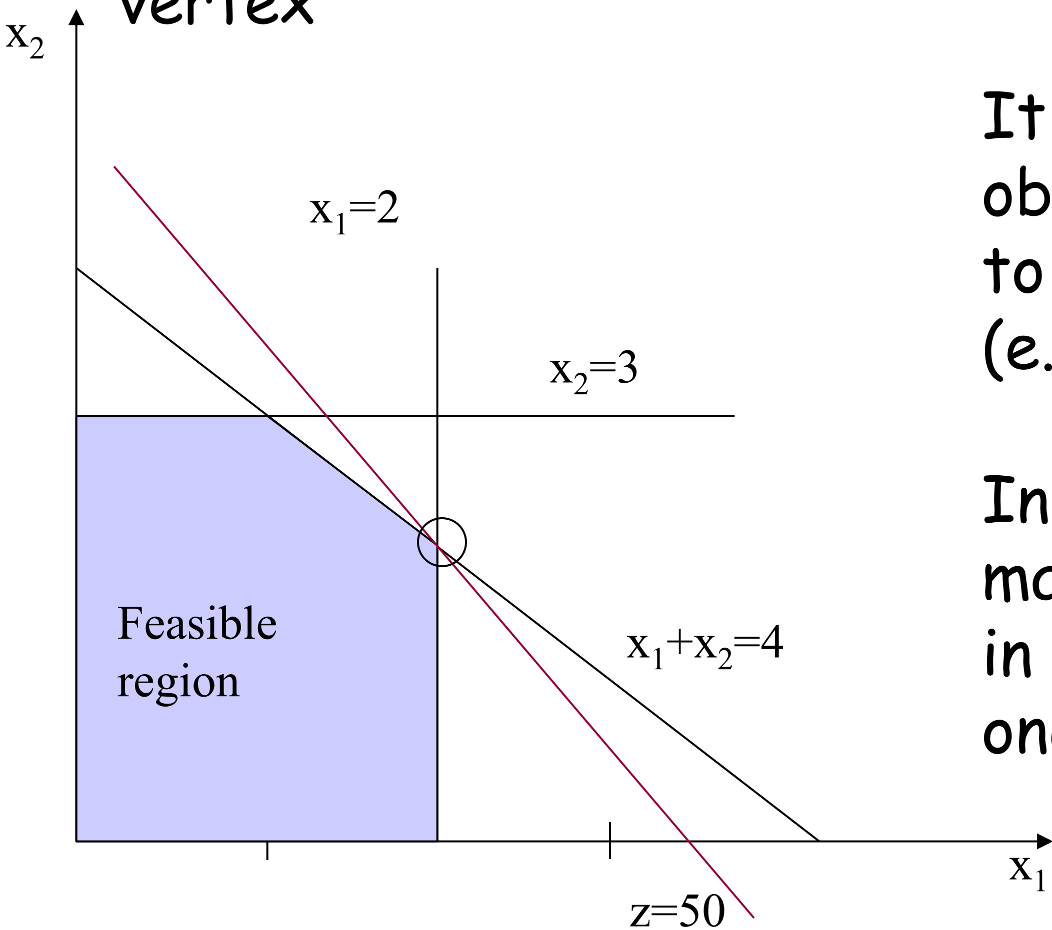
The Toy Factory Problem



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



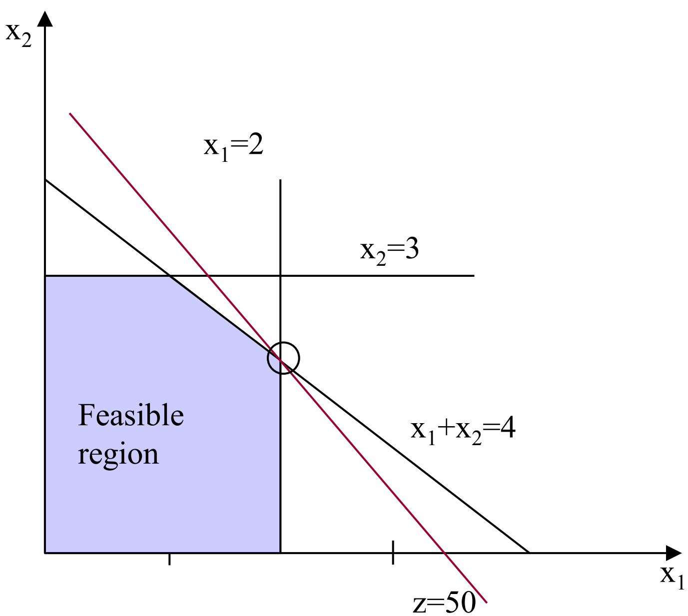
It might be that the objective line is parallel to a constraint.
(e.g. $z=15x_1+15x_2$).

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

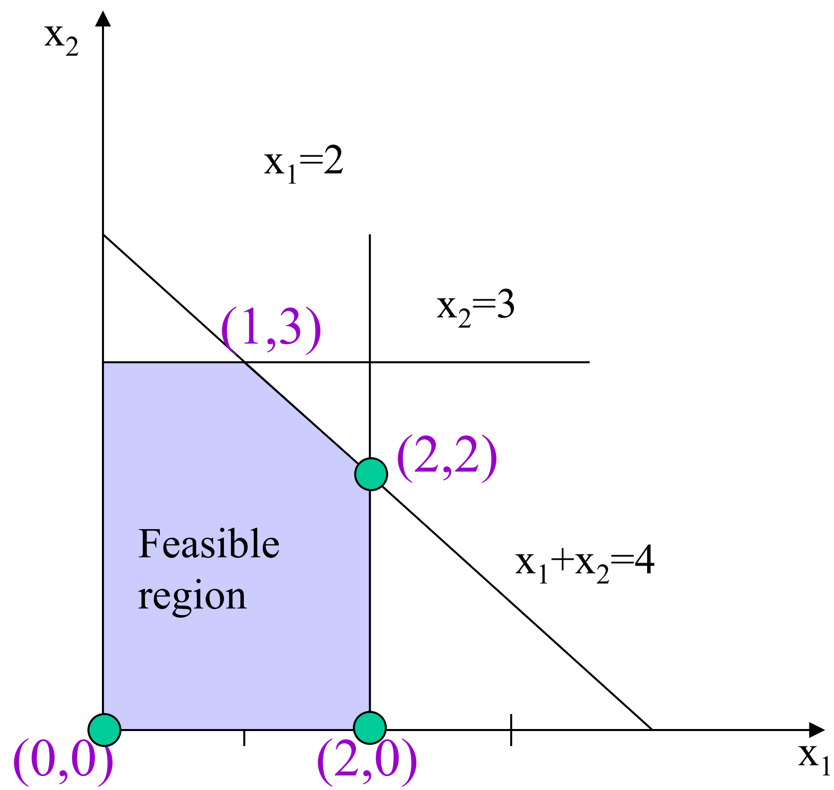
Iteration 1: Move to (2,0).

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

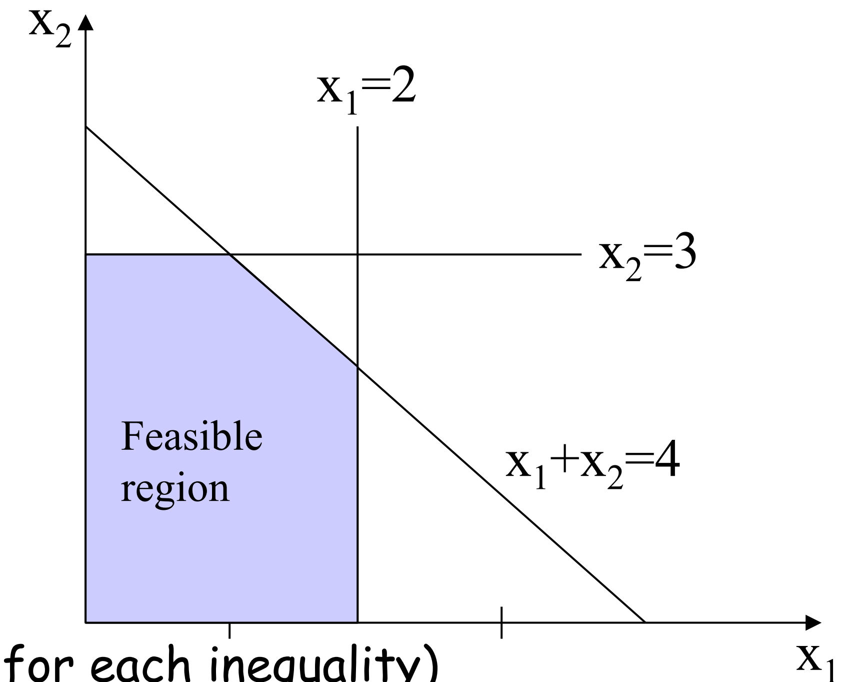
Adding Slack Variables

Let s_1, s_2, s_3 be the slack variables

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

s.t

$$x_1 + s_1 = 2$$
$$x_2 + s_2 = 3$$
$$x_1 + x_2 + s_3 = 4$$
$$x_1, x_2, s_1, s_2, s_3 \geq 0$$

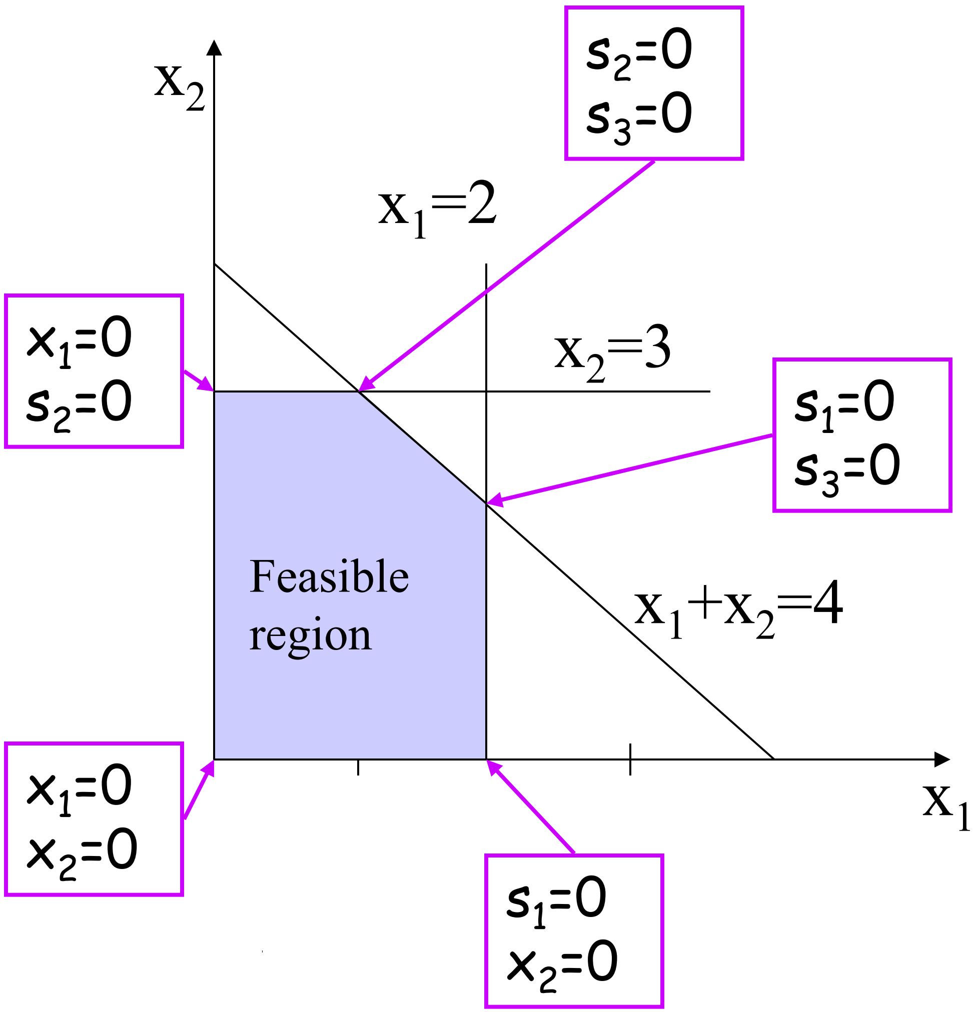


n – number of (original) variables
m – number of inequalities
Number of slack variables is m (one for each inequality)

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 along 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

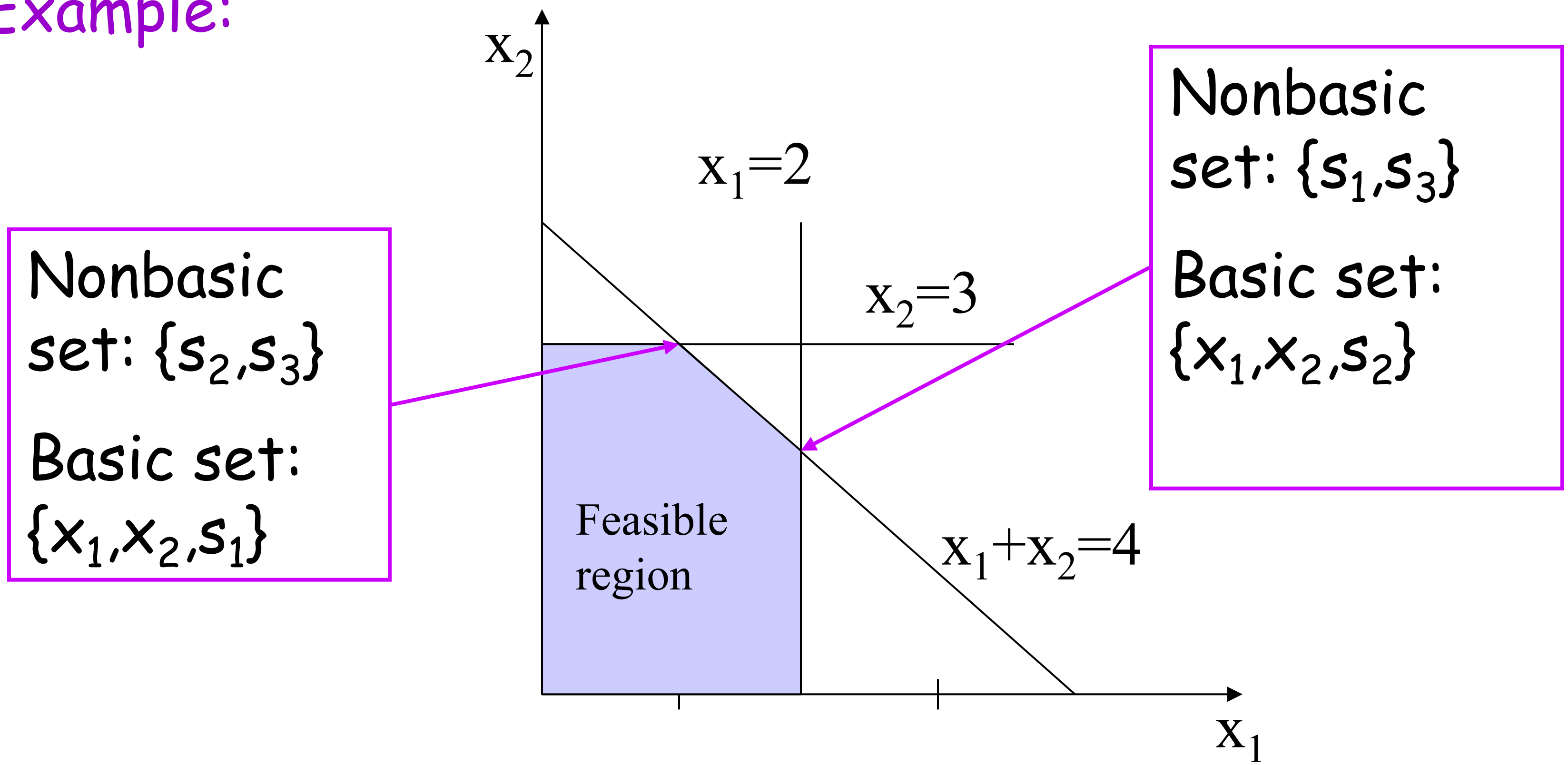
A basis: The current set of basic variables.

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:

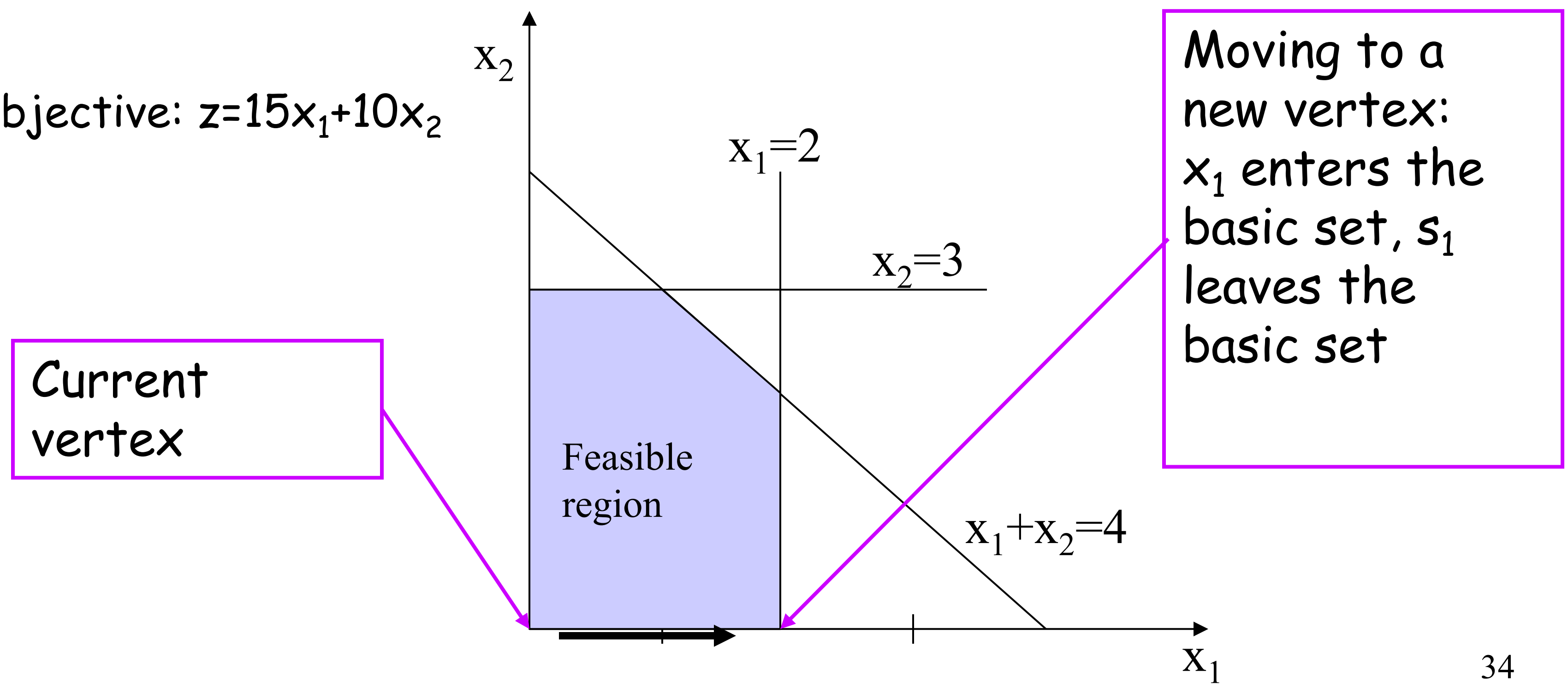


The Simplex Method

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: $z=15x_1+10x_2$



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. Select 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.