CSE 521: Design & Analysis of Algorithms I

Linear Programming

Paul Beame

Linear Programming

- The process of minimizing a linear objective function subject to a finite number of linear equality and inequality 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 Programming

- Reading:
  - Chapter 7 of text by Dasgupta, Papadimitriou, Vazirani linked in on web page.
  - “Linear Programming”, by Howard Karloff
    - First 34 pages on Simplex Algorithm available through Google books preview
  - “Linear Programming”, by Vasek Chvatal
  - “Introduction to Linear Optimization”, by Dimitris Bertsimas and John Tsitsiklis

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/pound, **$3**/pound
  - peanut butter: **1** unit of protein/pound, **$2**/pound
- In proper diet, need **4** units protein/day.

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

Let **y** = # pounds 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:

$x + 2y \geq 4$

$x \geq 0, \quad y \geq 0$

- 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 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:

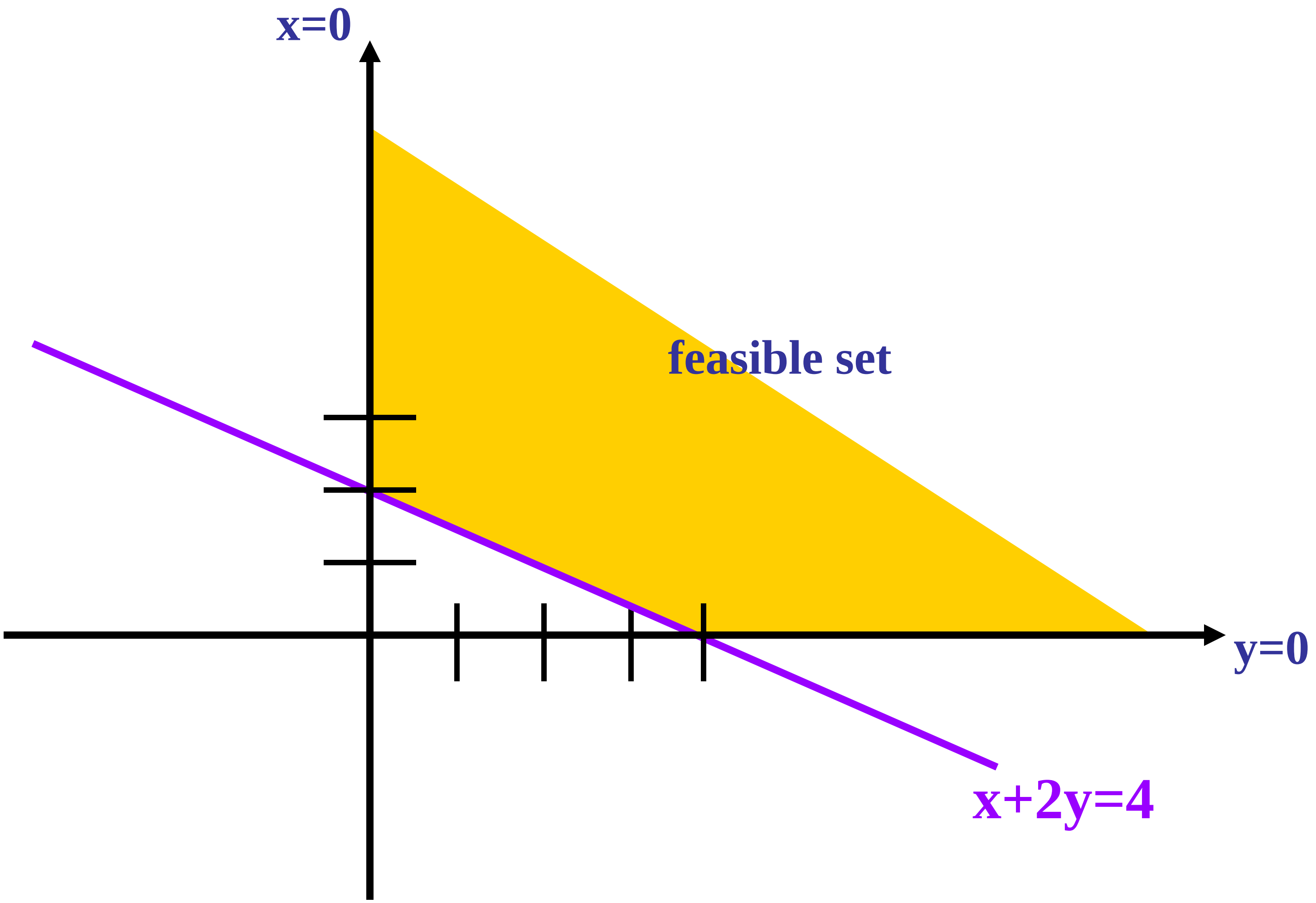
   $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \le b_i$ (or $\ge$ or **=**)

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

Feasible Set

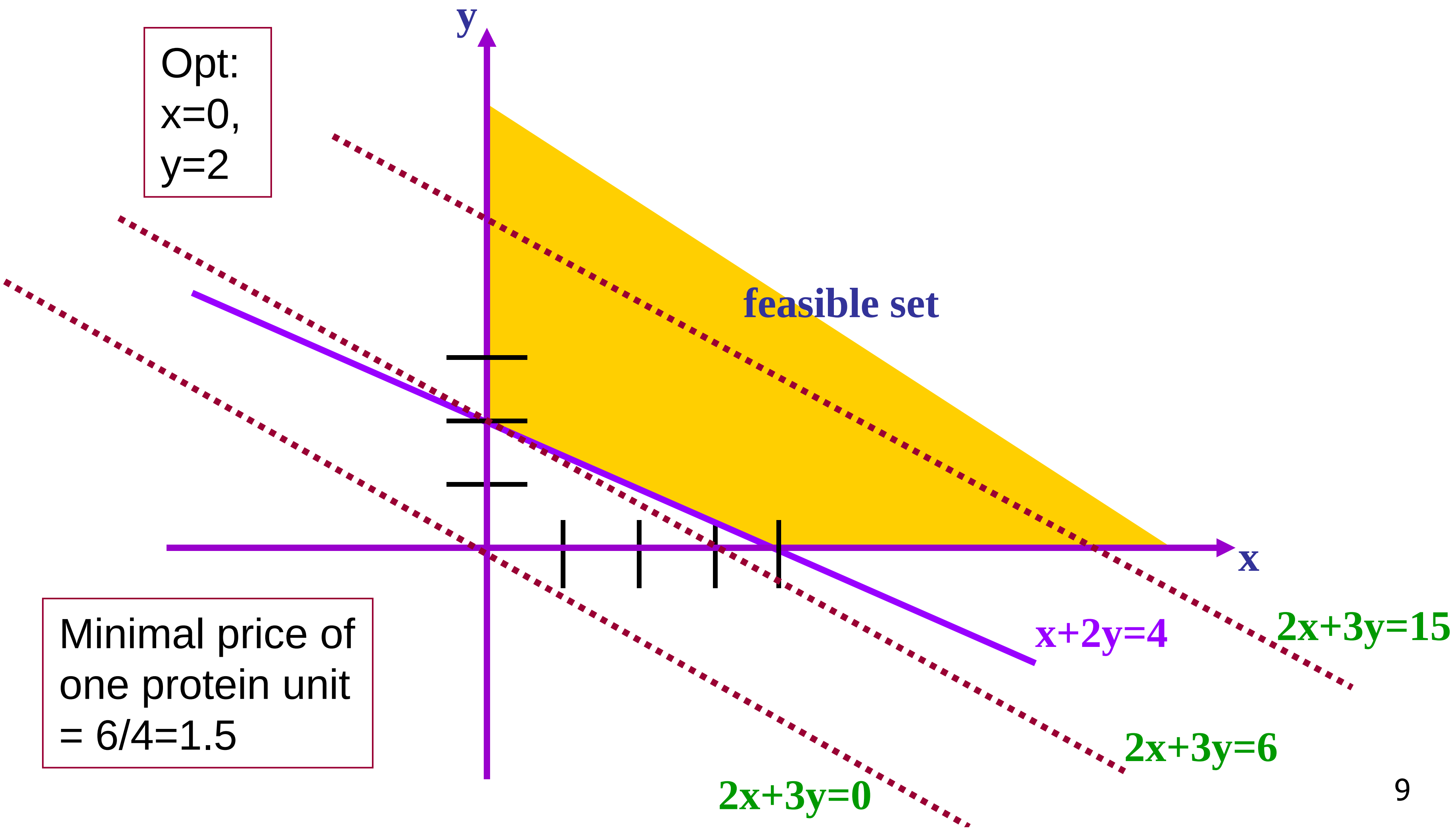
- Each linear inequality divides n-dimensional space into two halfspaces, one where the inequality is satisfied, and one where it's not.
- Feasible Set : solutions to a family of linear inequalities.
- The linear cost functions, defines a family of parallel hyperplanes (lines in 2D, planes in 3D, etc.). Want to find one of minimum cost → must occur at corner of feasible set.

Visually…
x= peanut butter, y = steak



Optimal vector occurs at some corner of the feasible set!

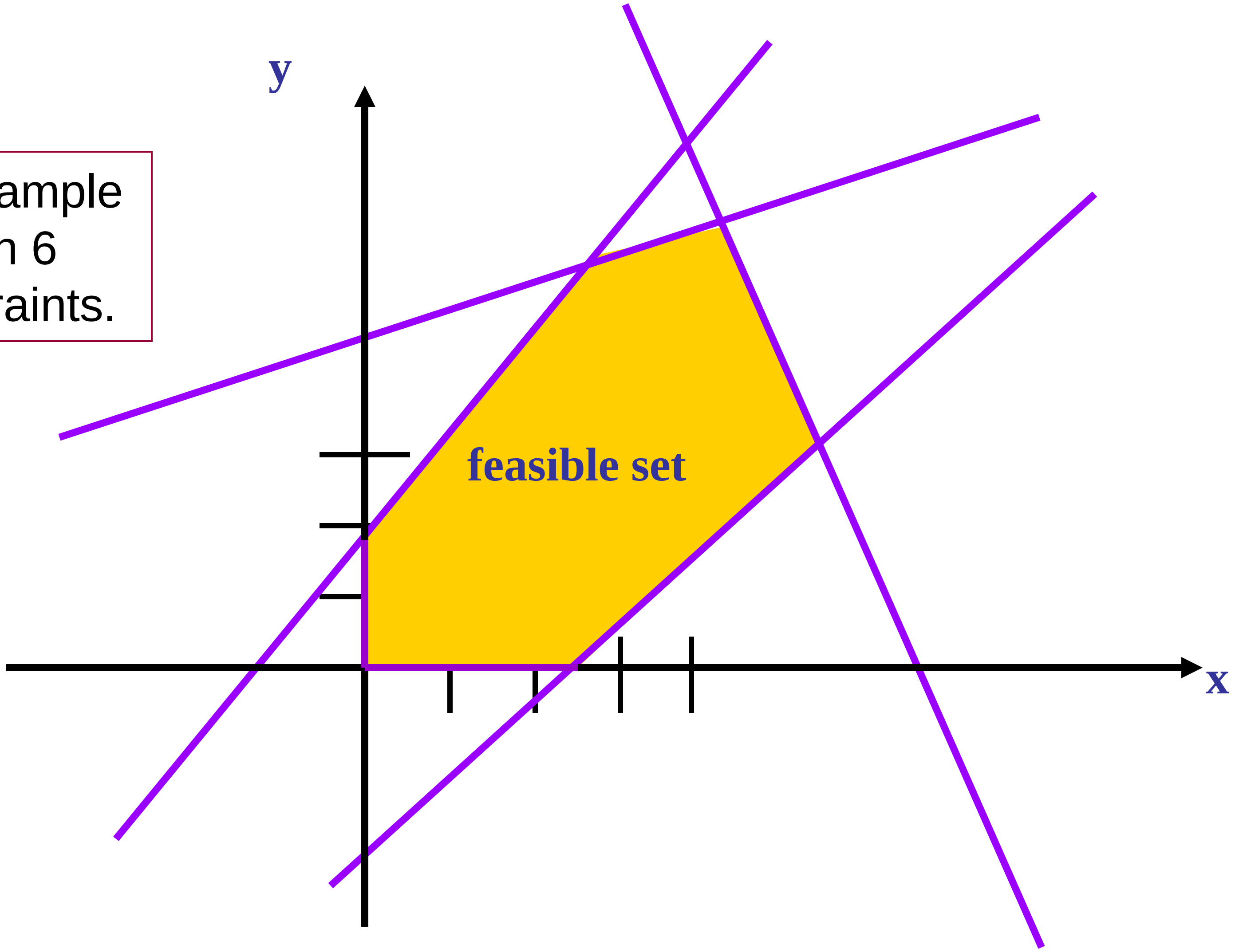
Opt:
x=0,
y=2



Minimal price of one protein unit = 6/4=1.5

Optimal vector occurs at some corner of the feasible set

An Example with 6 constraints.



Standard Form of a Linear Program.

Maximize $c_1x_1 + c_2x_2 + \ldots + c_nx_n$

subject to $\Sigma_{1\leq j \leq n} a_{ij}x_j \leq b_j$ $\quad i=1..m$

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

or

Minimize $b_1y_1 + b_2y_2 + \ldots + b_my_m$

subject to $\Sigma_{1\leq i\leq m} a_{ij}y_i \geq c_j$ $\quad j=1\ldots n$

$y_i \geq 0$ $\quad i=1..m$

The Feasible Set

- Intersection of a set of half-spaces, called a polyhedron.
- If it’s bounded and nonempty, it’s a polytope.

There are 3 cases:

- feasible set is empty.
- cost function is unbounded on feasible set.
- cost has a minimum (or maximum) on feasible set.

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

Solving LPs

- There are several algorithms that solve any linear program optimally.
  - The Simplex method (class of methods, usually very good but worst-case exponential for known methods)
  - The Ellipsoid method (polynomial-time)
  - More
- These algorithms can be implemented in various ways.
- There are many existing software packages for LP.
- It is convenient to use LP as a ``black box'' for solving various optimization problems.

LP formulation: another example

Bob’s bakery sells bagel and muffins.

To bake a dozen bagels Bob needs **5** cups of flour, **2** eggs, and **1** cup of sugar.

To bake a dozen muffins Bob needs **4** cups of flour, **4** eggs and **2** cups of sugar.

Bob can sell bagels at **$10**/dozen and muffins at **$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	

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

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

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 $\mathbf{c}\cdot\mathbf{x}$

s.t. $A\mathbf{x} \leq \mathbf{b}$

$\mathbf{x} \geq \mathbf{0}$.

Example: Max Flow

Variables: **f(e)** - the flow on edge e.

$$\text{Max} \;\; \Sigma_{e \in \mathrm{in}(t)} f(e)$$

s.t.

$$f(e) \le c(e), \;\; \forall e \in E$$

$$\Sigma_{e \,\in\, \mathrm{in}(v)} f(e) - \Sigma_{e \,\in\, \mathrm{out}(v)} f(e) = 0, \quad \forall v \in V\text{-}\{s,t\}$$

$$f(e) \ge 0, \;\; \forall e \in E$$

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 $\mathbf{x_1},\mathbf{x_2}$ denote the number of cars and dolls produced by Danny.

The Toy Factory Problem

Let $\mathbf{x_1,x_2}$ denote the number of cars and dolls produced by Danny.

Objective:

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

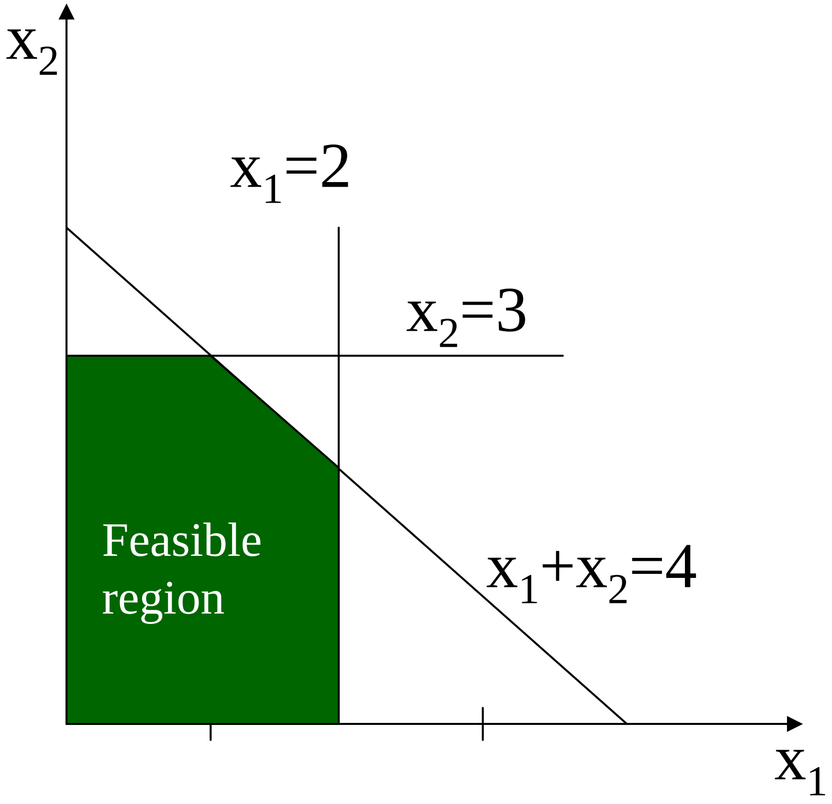
s.t $\mathbf{x_1 \leq 2}$

$\mathbf{x_2 \leq 3}$

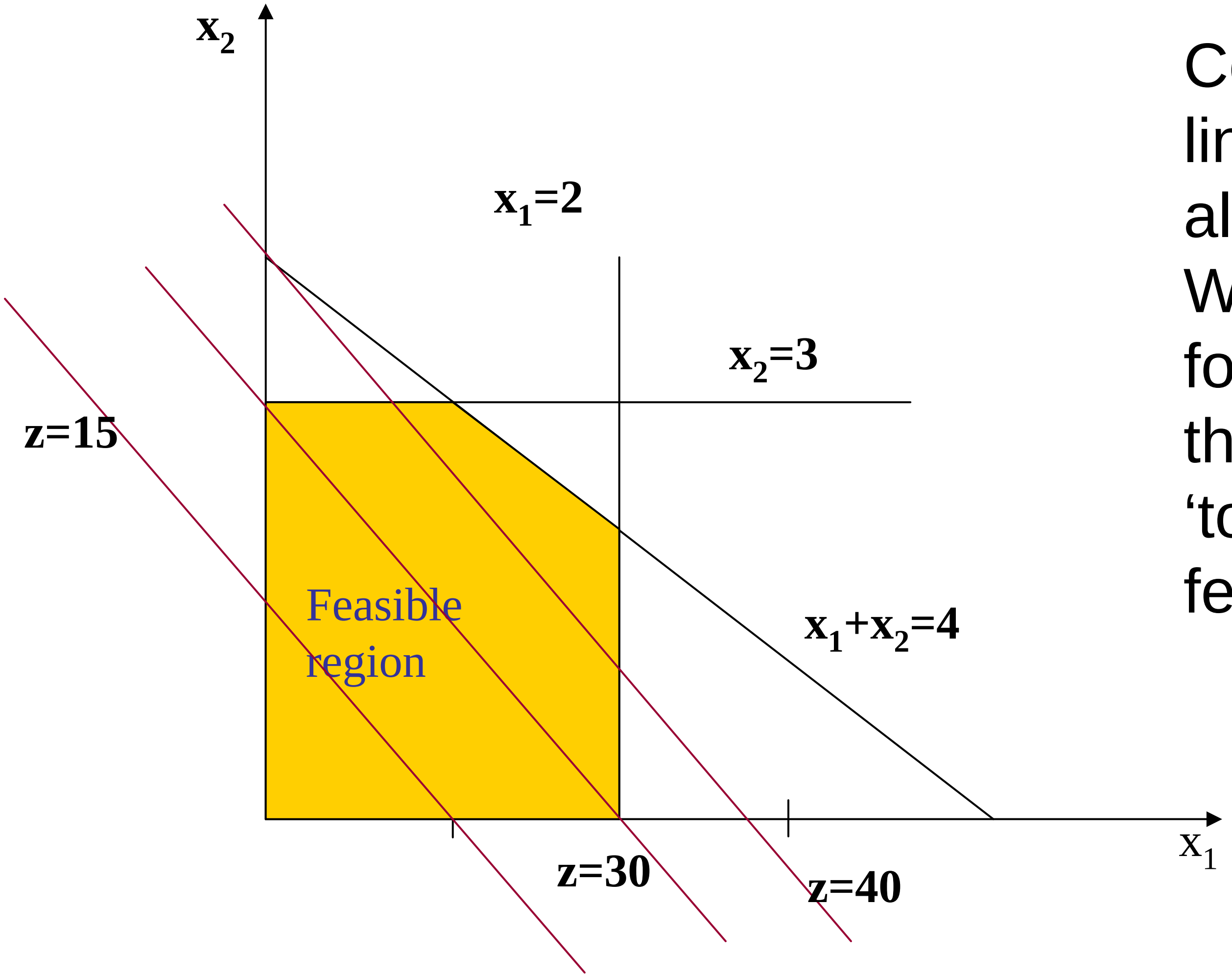
$\mathbf{x_1+x_2 \leq 4}$

$\mathbf{x_1 \geq 0}$

$\mathbf{x_2 \geq 0}$



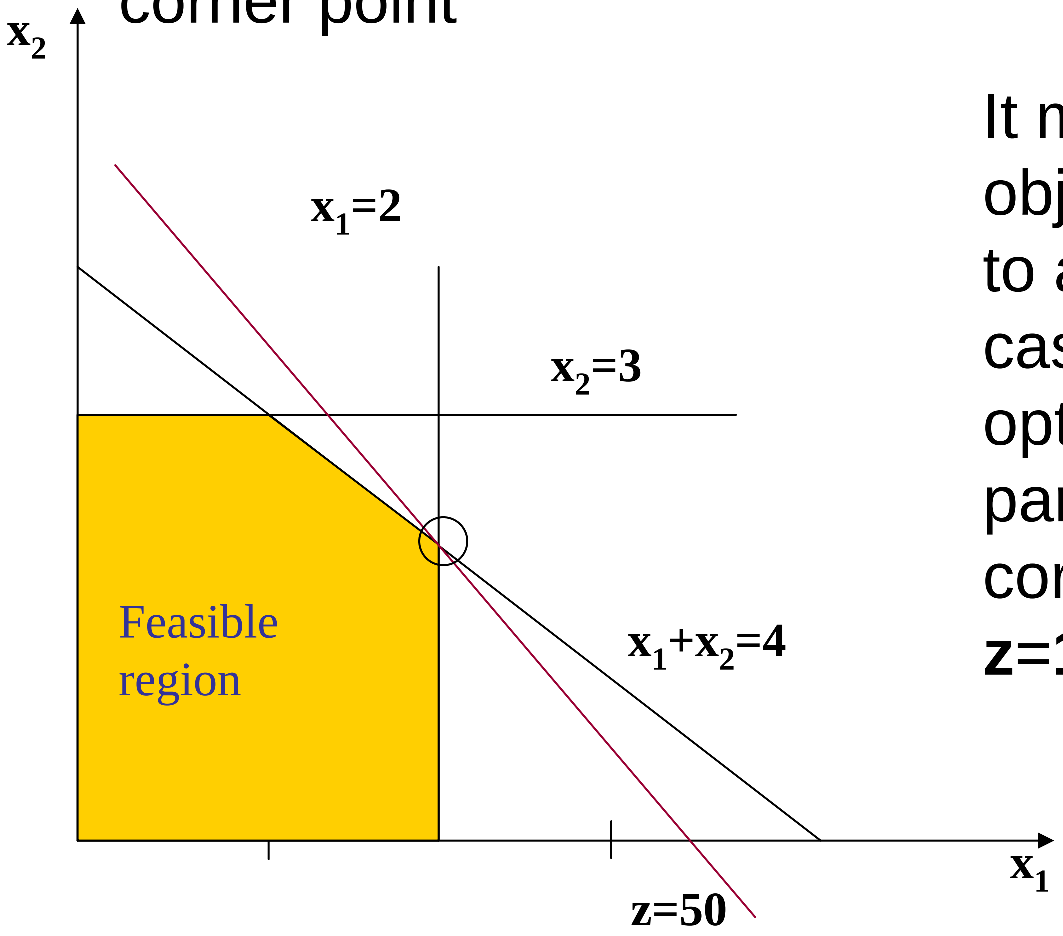
The Toy Factory Problem



Constant profit lines – They are always parallel. We are looking for the best one that still ‘touches’ the feasible region.

Important Observations:

1. An optimum solution to the LP is always at a corner point

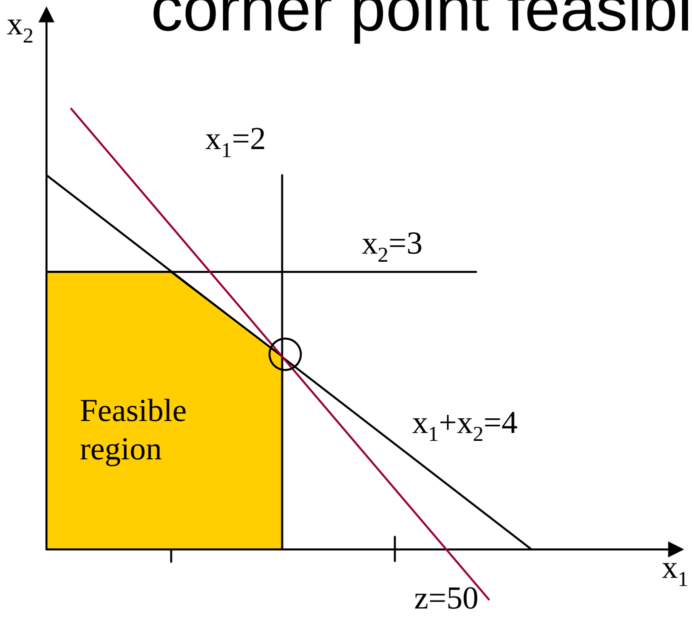


It might be that the objective line is parallel to a constraint. In this case there are many optimum points, in particular at the relevant corner points (consider $\mathbf{z=15x_1+15x_2}$).

Important Observations:

2. If a corner point feasible solution has an objective function value that is better than or equal to all its adjacent corner point feasible solutions then it is optimal.

3. There is a finite number of corner point feasible solutions.



The Simplex method: Travel along the corner points till a local maximum.

The Simplex Method

Phase 1 (start-up): Find any corner point feasible solution. In many standard LPs the origin can serve as the start-up corner point.

Phase 2 (iterate): Repeatedly move to a better adjacent corner point feasible solution until no further better adjacent corner point feasible solution can be found. The final corner point defines the optimum point.

Example: The Toy Factory Problem

Phase 1: start at (0,0)

Objective value = Z(0,0)=0

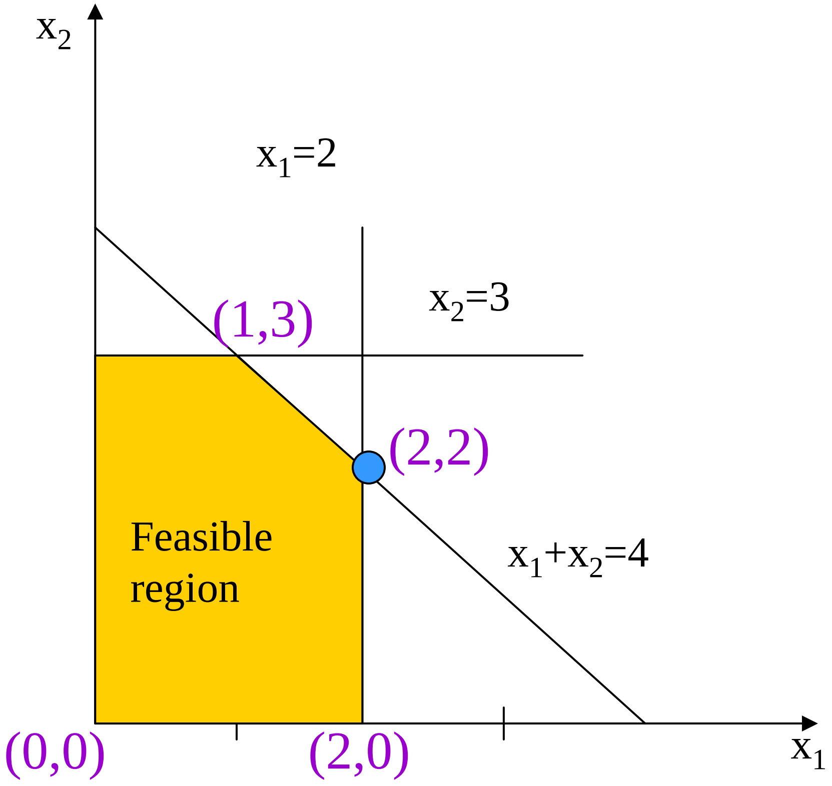
Iteration 1: Move to (2,0).

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 Corner Points Algebraically

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

Notes:

- In a corner point a subset of the inequalities are equations.
- It is possible to find the intersection of linear equations.
- We will add slack variables – to determine which inequality is active and which is not active

Adding Slack Variables

Let $\mathbf{s_1,s_2,s_3}$ be the slack variables

Objective:

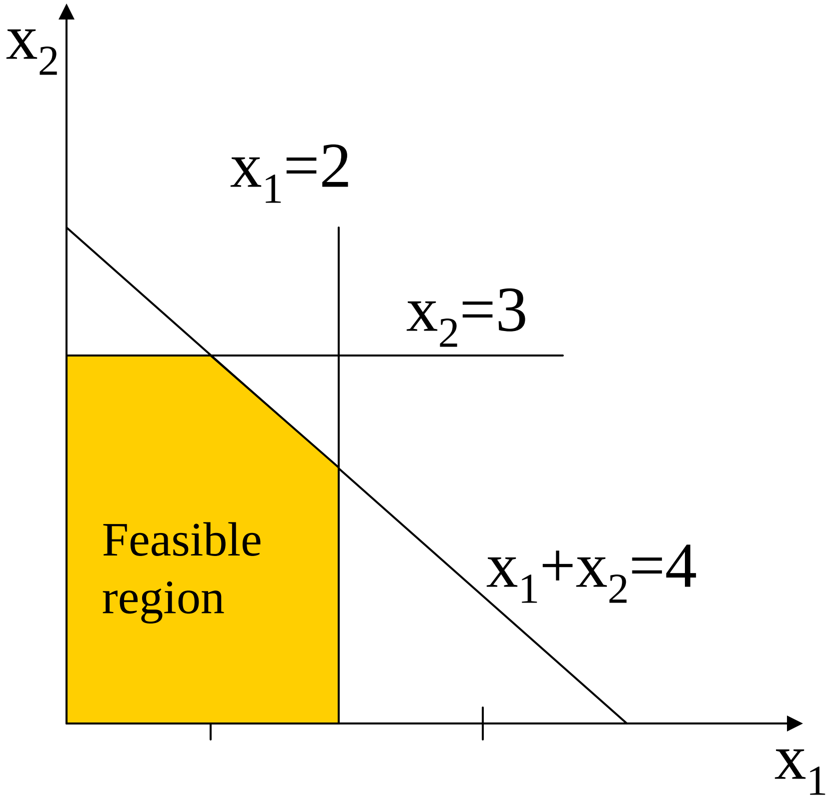
$\mathbf{Max\ z=15x_1+10x_2}$

s.t $\mathbf{x_1+s_1 = 2}$

$\mathbf{x_2 +s_2 = 3}$

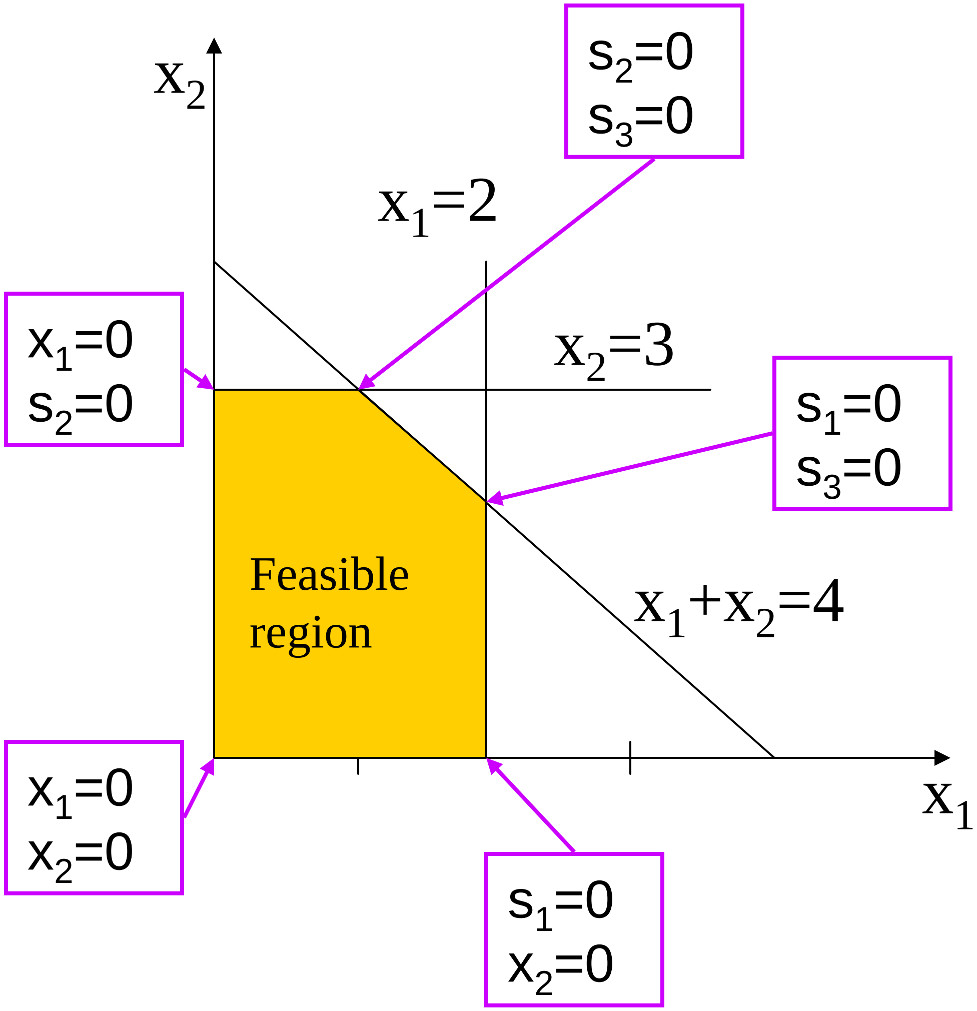
$\mathbf{x_1+x_2 +s_3 = 4}$

$\mathbf{x_1, x_2, s_1, s_2, s_3 \geq 0}$



A corner point: Some slack variables or original variables 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 corner points: 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.

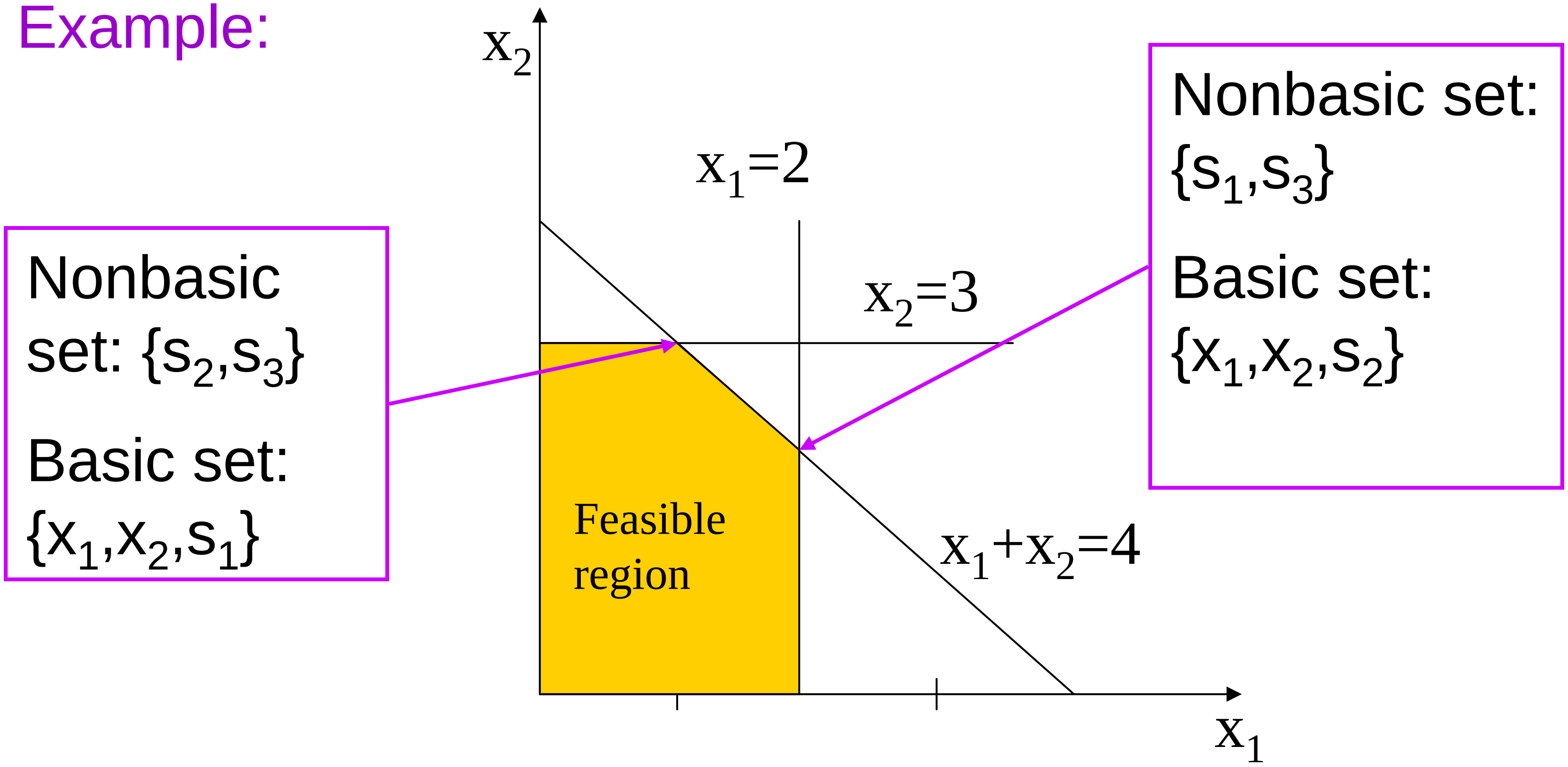
A basis: As simplex proceeds, the variables are always assigned to the basic set or the nonbasic set. The current assignment of the variables is called the basis.

Nonbasic, variables set to zero, corresponding constraint is active.

The Simplex Method

In two adjacent cornerpoints, the nonbasic sets and the basic sets are identical except for one member.

Example:



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

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

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

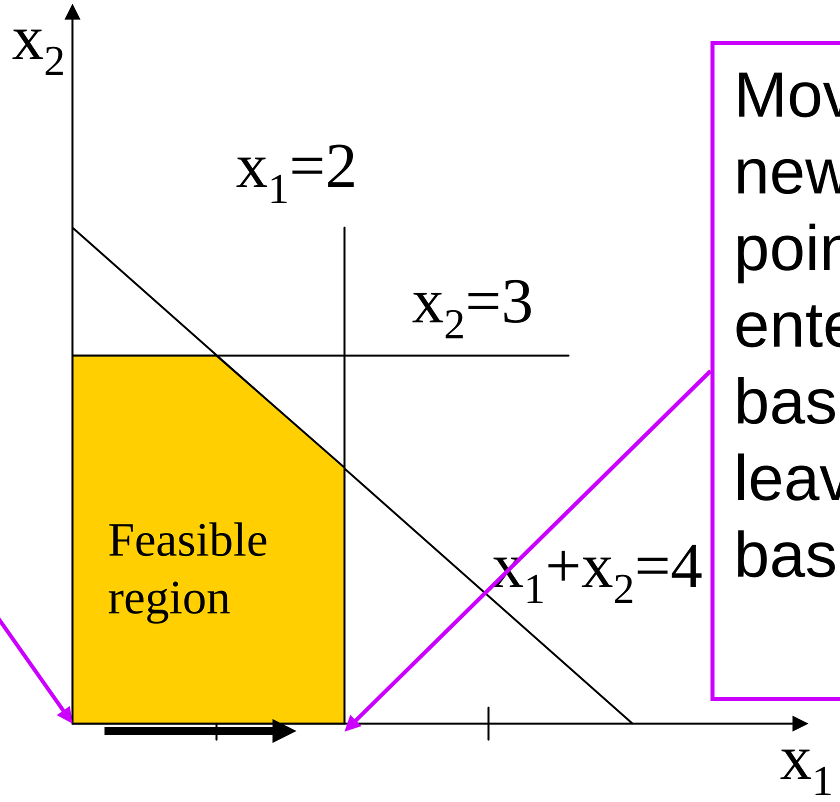
Basic set: $\{x_1,x_2,s_2\}$

The Simplex Method

The process is therefore based on swapping a pair of variables between the basic and the nonbasic sets.

It is done in a way that best improves the objective function.

Example:



Current cornerpoint

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

The Simplex Method – more details

Phase 1 (start-up): Initial corner point feasible solution.

Phase 2 (iterate):

1. Can the current objective value be improved by swapping a basic variable? If not - stop.
2. Select entering basic variable: choose the nonbasic variable that gives the fastest rate of increase in the objective function value.
3. Select the leaving basic variable by applying the minimum ratio test.
4. Update the equations to reflect the new basic feasible solution.

The Simplex Method – example (1)

Objective:

Max $z=15x_1+10x_2$

s.t **$x_1+s_1 = 2$**

$x_2 + s_2 = 3$

$x_1+x_2 + s_3 = 4$

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

Phase 1 (start-up): Initial cornerpoint feasible solution:

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

Nonbasic set = {$\mathbf{x_1}$, $\mathbf{x_2}$}
Basic set = {$\mathbf{s_1}$, $\mathbf{s_2}$, $\mathbf{s_3}$}

Phase 2 (iterate):

1. Are we optimal? NO, **z**'s value can increase by increasing both $\mathbf{x_1}$ and $\mathbf{x_2}$.
2. Select entering basic variable: $\mathbf{x_1}$ has a better ratio of improving the objective value (**15** > **10**).

The Simplex Method – example (2)

3. Select the leaving basic variables: 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 $\mathbf{x_1}$ increases)?

Answer: For $\mathbf{s_1}$ the ratio is 2/1=**2**, for $\mathbf{s_2}$ the ratio is infinite, for $\mathbf{s_3}$ the ratio is 4/1=**4**. s_1 has the smallest ratio.

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

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 values of $\mathbf{x_2}$.

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

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

End of iteration 2.

The Simplex Method – example (4)

Phase 2 (iteration 3):

1. Are we optimal? YES, **z**’s value cannot increase by increasing the value of either $\mathbf{s_1}$ or $\mathbf{s_3}$.

End of example.

Remarks:

There is a quick way (The Simplex tableau) to find out the variable to enter and the variable to leave the basis.

In case of a tie, both directions are ok, there is no heuristic to determine which will end up first.

Simplex Algorithm: An Example in 3D

Maximize $5x + 4y + 3z$

subject to $2x + 3y + z \leq 5$

$4x + y + 2z \leq 11$

$3x + 4y + 2z \leq 8$

$x, y, z \geq 0$.

Simplex Algorithm: An Example in 3D

Maximize $5x + 4y + 3z$

subject to $2x + 3y + z \le 5$

$4x + y + 2z \le 11$

$3x + 4y + 2z \le 8$

$x, y, z \ge 0$.

Step 0: convert inequalities into equalities by introducing slack variables **a,b,c**.

Define: $a = 5-2x-3y-z \quad \Rightarrow \quad a \ge 0$

$b = 11-4x-y-2z \quad \Rightarrow \quad b \ge 0$

$c = 8-3x-4y-2z \quad \Rightarrow \quad c \ge 0$

$F = 5x+4y + 3z$, **objective function**

Example of Simplex Method, continued.

Step 1: Find initial feasible solution:

x=0,y=0,z=0 ⇒ **a=5**, **b=11**, **c=8** ⇒ **F=0**.

Step 2: Find feasible solution with higher value of **F**

For example, can increase **x** to get **F=5x**.

How much can we increase **x**?

$a = 5-2x-3y-z \geq 0 \Rightarrow x \leq 5/2$ **most stringent**

$b = 11-4x-y-2z \geq 0 \Rightarrow x \leq 11/4$

$c = 8-3x-4y-2z \geq 0 \Rightarrow x \leq 8/3$

⇒ increase **x** to **5/2** ⇒ **F= 25/2**, **a=0**, **b=1**, **c=1/2**

Example of Simplex Method, continued.

Want to keep doing this, need to get back into state where **x**,**b**,**c** on l.h.s. of equations.

$\mathbf{a = 5\text{-}2x\text{-}3y\text{-}z \Rightarrow x = 5/2 - 3/2\ y - 1/2\ z - 1/2\ \ a}$ (*)

Substituting (*) into other equations:

$\mathbf{b = 11\text{-}4x\text{-}y\text{-}2z \geq 0 \Rightarrow b = 1 + 5y + 2a}$

$\mathbf{c = 8\text{-}3x\text{-}4y\text{-}2z \geq 0 \Rightarrow c = 1/2 + 1/2\ y - 1/2\ z + 3/2\ a}$

$\mathbf{F = 5x + 4y + 3z \Rightarrow F = 25/2 - 7/2\ y + 1/2\ z - 5}/2\ a$

In order to increase **F** again, should increase **z**

Example of Simplex Method, continued.

How much can we increase **z**?

$x = 5/2 - 3/2\, y - 1/2\, z - 1/2\, a \quad \Rightarrow z \leq 5$

$b = 1 + 5y + 2a \quad \Rightarrow$ no restriction

$c = 1/2 + 1/2\, y - 1/2\, z + 3/2\, a \quad \Rightarrow z \leq 1$ **most stringent (^)**

Setting **z** = **1** yields

$x=2$, $y=0$, $z=1$, $a=0$, $b = 1$, $c = 0$.

$F = 25/2 - 7/2\, y + 1/2\, z - 5/2\, a \quad \Rightarrow F = 13$.

Again, construct system of equations.

From (^) $\quad z = 1 + y + 3a - 2c$.

Example of Simplex Method, continued.

Substituting back into other equations:

$z = 1 + y + 3a - 2c.$

$x = 5/2 - 3/2\ y - 1/2\ z - 1/2\ a \quad \Rightarrow x = 2 - 2y - 2a + c$

$b = 1 + 5y + 2a \quad \Rightarrow b = 1 + 5y + 2a$

$F = 25/2 - 7/2\ y + 1/2\ z - 5/2\ a \Rightarrow F = 13 - 3y - a - c$

And we’re done.

A Central Result of LP Theory: Duality Theorem

- Every linear program has a dual
- If the original is a minimization, the dual is a maximization and vice versa
- Solution of one leads to solution of other

Primal: Maximize $\mathbf{c}^T\mathbf{x}$ subject to $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{x} \geq 0$

Dual: Minimize $\mathbf{b}^T\mathbf{y}$ subject to $\mathbf{A}^T\mathbf{y} \geq \mathbf{c}$, $\mathbf{y} \geq 0$

If one has optimal solution so does the other, and their values are the same.

Proof of Weak Duality

- Suppose that
  - $\mathbf{x}$ satisfies $A\mathbf{x} \le \mathbf{b}$, $\mathbf{x} \ge 0$
  - $\mathbf{y}$ satisfies $A^T\mathbf{y} \ge \mathbf{c}$, $\mathbf{y} \ge 0$
- Then
  - $\mathbf{c}^T\mathbf{x} \le (A^T\mathbf{y})^T \mathbf{x}$ since $\mathbf{x} \ge 0$ and $A^T\mathbf{y} \ge \mathbf{c}$

    $= \mathbf{y}^T A \mathbf{x}$ by definition

    $\le \mathbf{y}^T\mathbf{b}$ since $\mathbf{y} \ge 0$ and $A\mathbf{x} \le \mathbf{b}$

    $= \mathbf{b}^T\mathbf{y}$ by definition
- This says that any feasible solution to the primal (maximization problem) has an objective function value at most that of any feasible solution of the dual (minimization) problem.
- Strong duality says that the optima of the two are equal

Primal: **Maximize** c^Tx **subject to** $Ax \leq b$**,** $x \geq 0$

Dual: **Minimize** b^Ty **subject to** $A^Ty \geq c$**,** $y \geq 0$

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

Primal:

maximize $2x + 3y$

s.t $x+2y \leq 4,$

$2x + 5y \leq 1,$

$x - 3y \leq 2,$

$x \geq 0, y \geq 0$

Dual:

minimize $4p + q + 2r$ s.t

$p+2q + r \geq 2,$

$2p+5q -3r \geq 3,$

$p,q,r \geq 0$

Simple Example

- Diet problem: minimize $2x + 3y$

  subject to $x+2y \geq 4$,

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

- Dual problem: maximize $4p$

  subject to $p \leq 2$,

  $2p \leq 3$,

  $p \geq 0$

- Dual: the problem faced by a druggist who sells synthetic protein, trying to compete with peanut butter and steak

Simple Example

- The druggist 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 or he won't sell any
  - revenue to druggist will be 4p
- Solution: $p \le 3/2$ → 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 grocer and the druggist the result is always a tie.
- Optimal solution to primal tells purchaser what to do.
- Optimal solution to dual fixes the natural prices at which economy should run.
- The diet x and vitamin prices y are optimal when
  - grocer sells zero of any food that is priced above its vitamin equivalent.
  - druggist charges 0 for any vitamin that is oversupplied in the diet.

Duality Theorem

Druggist’s max revenue = Purchasers 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.
- Important implications for economists.

Example: Max Flow

Variables: f_{uv} - the flow on edge $e=(u,v)$.

Max $\Sigma_u f_{su}$ — Dual variables

s.t.

$f_{uv} \le c_{uv}, \ \forall (u,v) \in E$ — h_{uv}

$\Sigma_u f_{uv} - \Sigma_w f_{vw} = 0, \ \forall v \in V-\{s,t\}$ — g_v

$f_{uv} \ge 0, \ \forall (u,v) \in E$

Dual to: Max Flow

Variables: g_v, h_{uv} - the flow on edge $e=(u,v)$

$$\text{Min} \quad \Sigma_{uv} c_{uv} h_{uv}$$

s.t.

$$h_{sv} + g_v \geq 1, \quad \forall v \in V-s$$

$$h_{uv} + g_v - g_u \geq 0, \quad \forall u,v \in V-\{s,t\}$$

$$h_{ut} - g_u \geq 0, \quad \forall u \in V-t$$

$$h_{uv} \geq 0, \quad \forall (u,v) \in E$$

WLOG at minmum

$h_{uv}=\max(g_u-g_{v,}0)$

for $u,v \neq s,t$

$h_{ut}=\max(g_u,0)$

$h_{sv}=\max(1-g_{v,}0)$

Dual Solution: Given st-cut (S,T) with $S=s \cup A$

Set $g_v=1$ for $v \in A$ and $g_v=0$ otherwise

Set $h_{uv}=1$ for $u \in A$ and v not in A

Set $h_{uv}=0$ otherwise

Value is exactly the value of the cut

Properties of LP optima

Maximize $c_1x_1 + c_2x_2 + \ldots + c_nx_n$

subject to

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\le b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\le b_2 \\
&\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\le b_m \\
-x_1 \qquad\qquad\qquad\qquad &\le 0 \\
-x_2 \qquad\qquad &\le 0 \\
\ldots \qquad &\\
-x_n &\le 0
\end{aligned}
$$

$$\begin{bmatrix} A \\ -I \end{bmatrix} x \le \begin{bmatrix} b \\ 0 \end{bmatrix}$$

Properties of LP optima

- Every corner point **x*** of the feasible region **$Ax \le b$, $x \ge 0$** satisfies:
  - **$\bar{A}x^* = b'$** where **$\bar{A}$** and **b'** are some subset of **n** rows of [**A**|-**I**] and [**b**|**0**] that yield an invertible square submatrix**.**
- Now **$x^* = \bar{A}^{-1}b'$** and **$\bar{A}^{-1} = \text{adj}(\bar{A})/\det(\bar{A})$** where $\text{adj}(\bar{A})_{ij}$ is the determinant of a submatrix of **$\bar{A}$**
  - Implies that if **A**,**b** are integers then every corner point has rational coordinates
- Def: **A** is totally unimodular iff every square submatrix of **A** has determinant **+1,0**, or **-1**
  - $\det(\bar{A}) = \pm\det(A')$ where **A'** is a submatrix of **A**
  - So **if** **A** is totally unimodular then all corner points have integer coordinates

Minimum Cost Perfect Matching

- Given two sets of vertices U,V with $|U|=|V|$ and a cost $c_{uv} \geq 0$ for each pair/edge, find a perfect matching between U and V of minimum total cost
- LP: variables x_{uv}

  Minimize $\Sigma_{uv}\, c_{uv}\, x_{uv}$ s.t.

  $$\Sigma_v\, x_{uv} \geq 1$$

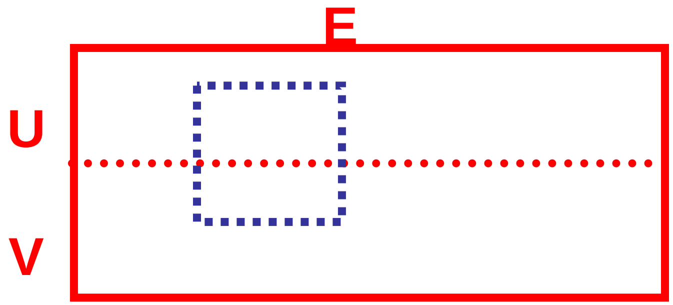
  $$\Sigma_u\, x_{uv} \geq 1$$

  $$x_{uv} \geq 0$$

- Note: No need to say that $x_{uv} \leq 1$ since any solution with an $x_{uv} > 1$ can be reduced to $x_{uv}=1$ and get a smaller objective function value since $c_{uv} \geq 0$

Incidence Matrix of a Bipartite Graph is Totally Unimodular

Matrix is



Induction on size of submatrix: m=1 True

- Square submatrix has a 0 column
  - Determinant=0 True
- Square submatrix of size m has one 1 in some column
  - Expanding along the column gives +1 or -1 times Determinant of size m-1 square submatrix.

    True by I.H.
- Square submatrix has two 1's in every column
  - Sum rows in U = sum of rows in V
  - Rows are linearly dependent so Determinant=0

    True

Minimum Cost Perfect Matching

- Given two sets of vertices **U,V** with **|U|=|V|** and a cost $c_{uv} \geq 0$ for each pair/edge, find a perfect matching between **U** and **V** of minimum total cost
- LP: variables x_{uv}

Minimize $\Sigma_{uv}\, c_{uv}\, x_{uv}$ s.t. Dual variables

$\Sigma_{v}\, x_{uv} \geq 1$ p_u

$\Sigma_{u}\, x_{uv} \geq 1$ p_v

$x_{uv} \geq 0$

- Note: No need to say that $x_{uv} \leq 1$ since any solution with an $x_{uv} > 1$ can be reduced to $x_{uv} = 1$ and get a smaller objective function value since $c_{uv} \geq 0$

Minimum Cost Perfect Matching

- Dual LP: variables p_u, p_v

  Maximize $\Sigma_u \, p_u + \Sigma_v \, p_v$ s.t.

  $$p_u + p_v \leq c_{uv}$$

  $$p_u, p_v \geq 0$$

- Think of p_u and p_v as prices that the endpoints can charge for participating in the matching
- In optimal solution for both Primal and Dual the budget exactly balances.

Primal-Dual Algorithms

- Sometimes one can cleverly work with both primal and dual to get an efficient algorithm that is purely combinatorial
  - e.g. Min-cost perfect matching from Section 7.13 of the K&T text
    - Repeatedly select cheapest augmenting path where notion of 'cheapest' depends on current values of prices from the dual for the maxflow version of the problem, i.e. the $\mathbf{g_u}$,$\mathbf{g_v}$ values, update prices

Primal-Dual Algorithm for Min-cost Perfect Matching

- Maintain residual graph as in usual reduction to maxflow
- Start with all $f_{uv}=0$, $p_u=0$ and $p_v= \min_u c_{uv}$
- Maintain modified costs $\check{c}_{uv} = c_{uv} – p_u – p_v \geq 0$
  - Matched edges **uv** will have $\check{c}_{uv} = 0$
- Find shortest **st**-path in residual using modified cost as measure of distance (other edges have length 0)
  - If shortest path has length **0** then augment flow along path to increase size of matching
  - If shortest path has length >**0** then only its last edge **ab in** bipartite graph will have $\check{c}_{ab} > 0$
    - Add $\check{c}_{ab}$ to p_u on all $u \in U$ reachable by paths of length **0**
    - Add $-\check{c}_{ab}$ to p_v on all $v \in V$ reachable by paths of length **0**
      - Need to argue that this keeps prices ≥ **0**
    - Path will now have length **0** and set of edges **uv** with $\check{c}_{uv} = 0$ will strictly increase to include **ab**

Ellipsoid Algorithm

- Running time is polynomial but depends on the # of bits **L** needed to represent numbers in **A, b,** and **c**
  - Like capacity-scaling for network flow but a much bigger polynomial
  - Interior point methods running times also depend on **L**

- Method applies to large class of convex programs
  - Can be efficient for LPs with exponentially many constraints

- Open whether a strongly polynomial-time algorithm exists for LP
  - One where running time has # of operations polynomial in just **m** and **n**

Integer Programming (IP)

- An LP problem with an additional requirement that variables will only get an integral value, maybe from some range.
- 01P – binary integer programming: variables should be assigned only 0 or 1.
- Can model many problems.
- NP-hard to solve!

01P Example: Vertex Cover

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

Minimize $\quad \boldsymbol{\Sigma_v x_v}$

Subject to: $\mathbf{x_v} \in \{0,1\}$

$\mathbf{x_u + x_v} \geq \mathbf{1} \quad \forall (\mathbf{u}, v) \in E$

01P Example: Weighted 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: $x_i \in \{0,1\}^n$

$\sum_{j \in Si} x_i \geq 1 \quad \forall j=1..m$

01P Example: Shortest Path

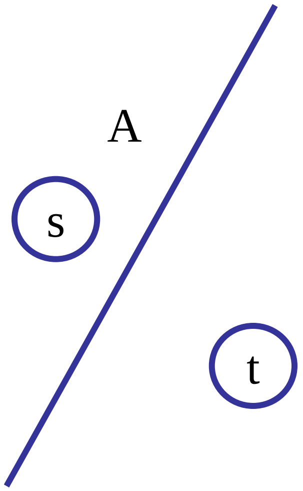
Given a directed graph **G**(**V**,**E**), **s**,**t** $\in$ **V** and length $\mathbf{p_e}$ for edge **e**.

Variables: For each edge, $\mathbf{x_e}$ – is **e** in the path?

Minimize $\quad \sum_{\mathbf{e}} \mathbf{p_e} \mathbf{x_e}$

Subject to: $\quad \mathbf{x_e} \in \{\mathbf{0},\mathbf{1}\} \quad \forall \mathbf{e} \in \mathbf{E}$

$\sum_{\mathbf{e} \in \mathbf{A}} \mathbf{x_e} \geq \mathbf{1} \quad \forall \mathbf{s} \rightarrow \mathbf{t}$ cut **A**



LP-based approximations

- We don’t know any polynomial-time algorithm for any NP-complete problem
- We know how to solve LP in polynomial time
- We will see that LP can be used to get approximate solutions to some NP-complete problems.

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: Weighted Vertex Cover is NP-complete.

The best known approximation factor is

2- 1/sqrt(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.

$x_v + x_u \geq 1, \quad \forall (u,v) \in E$

$x_v \in \{0,1\} \quad \forall v \in V$

The LP Relaxation

This is **not** a linear program: the constraints of type

$\mathbf{x_v} \in \{\mathbf{0},\mathbf{1}\}$ are not linear. We got an LP with integrality constraints on variables – an **integer linear program (IP)** that is NP-hard to solve.

However, if we replace the constraints $\mathbf{x_v} \in \{\mathbf{0},\mathbf{1}\}$

by $\mathbf{x_v} \geq \mathbf{0}$ and $\mathbf{x_v} \leq \mathbf{1}$, we will get a linear program.

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

LP Relaxation of Weighted Vertex Cover

Min $\Sigma_{v \in V} w_v x_v$

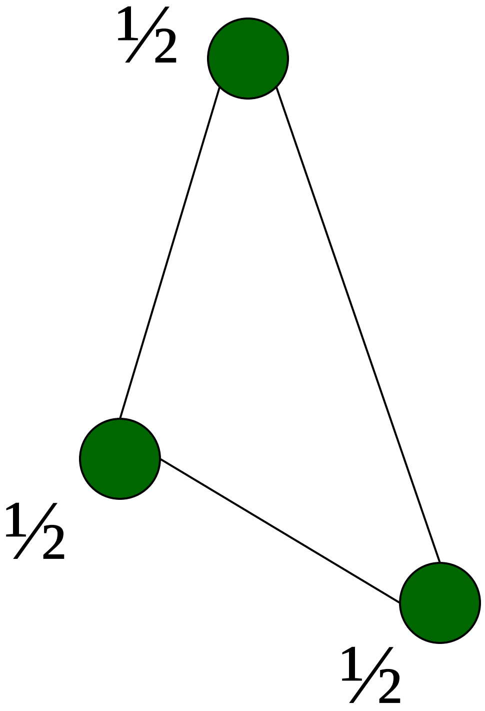
s.t.

$x_v + x_u \geq 1, \quad \forall (u,v) \in E$

$x_v \geq 0, \quad \forall v \in V$

$x_v \leq 1, \quad \forall v \in V$

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 $\mathbf{x}_v$=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.

Approximation of Weighted Vertex Cover Using LP-Rounding

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: $\Sigma_{v \in s} w_v = \Sigma_{\{v|\, x_V \geq 1/2\}} w_v$

$\leq 2\, \Sigma_{v \in V} w_v x_v = 2\, OPT_{LP}$

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

$\leq 2\, OPT_{VC}$.

Linear Programming -Summary

- Of great practical importance to solve linear programs:
  - they model important practical problems
    - production, approximating the solution of inconsistent equations, 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, but the simplex algorithm works very well in practice.
- One problem where you really do not want to roll your own code.