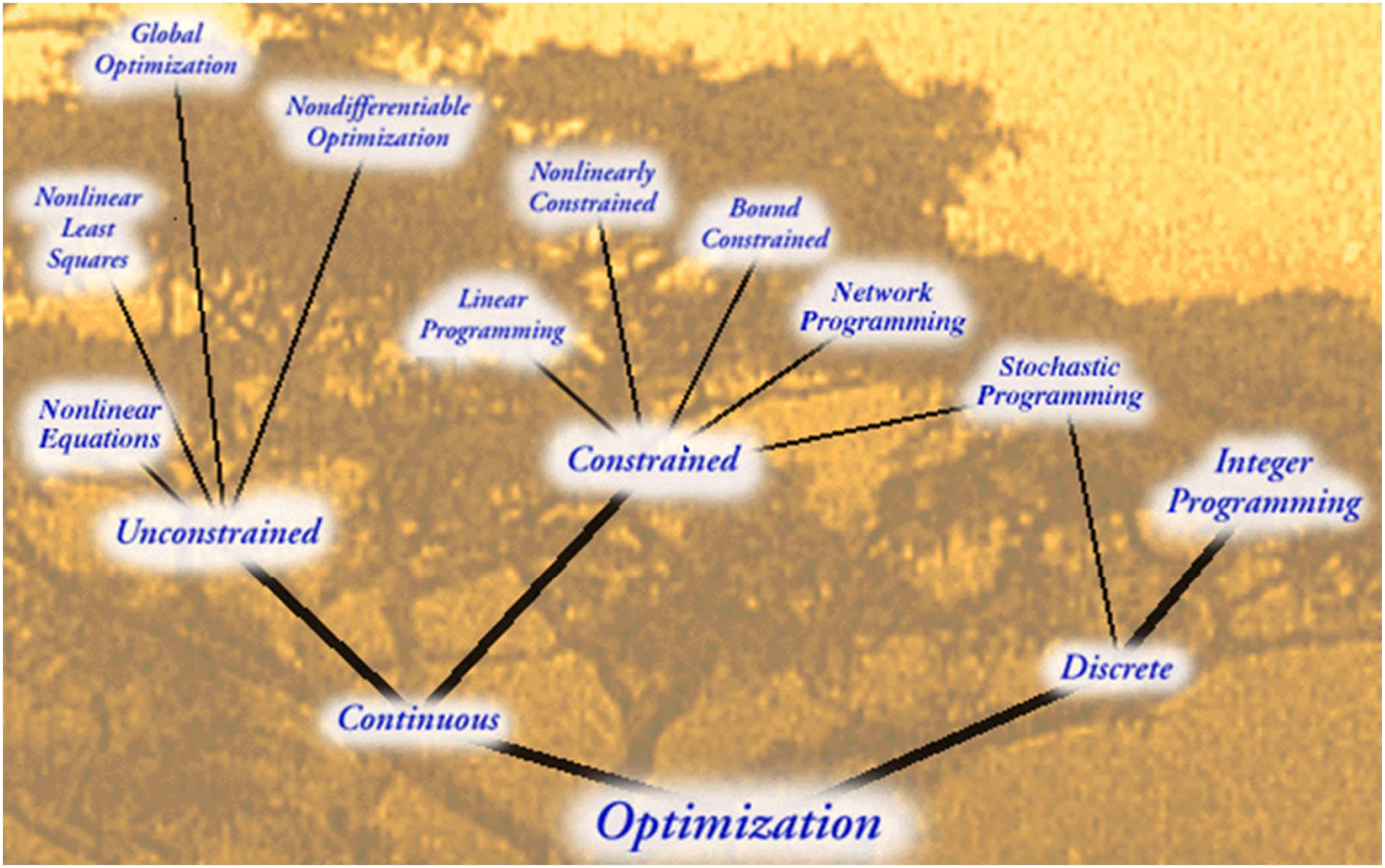
Lecture 3

3B1B Optimization Michaelmas 2016 A. Zisserman

Linear Programming

- Extreme solutions
- Simplex method
- Interior point method
- Integer programming and relaxation

The Optimization Tree



Linear Programming

The name is historical, it should really be called Linear Optimization.

The problem consists of three parts:

A linear function to be maximized

$$\text{maximize } f(\mathbf{x}) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

Problem constraints

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

Non-negative variables

$$\text{e.g. } x_1, x_2 \ge 0$$

Linear Programming

The problem is usually expressed in matrix form and then it becomes:

$$\begin{aligned} \text{Maximize} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0 \end{aligned}$$

where $\mathbf{A}$ is a $m \times n$ matrix.

The objective function, $\mathbf{c}^\top \mathbf{x}$, and constraints, $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, are all linear functions of $\mathbf{x}$.

(Linear programming problems are convex, so a local optimum is the global optimum.)

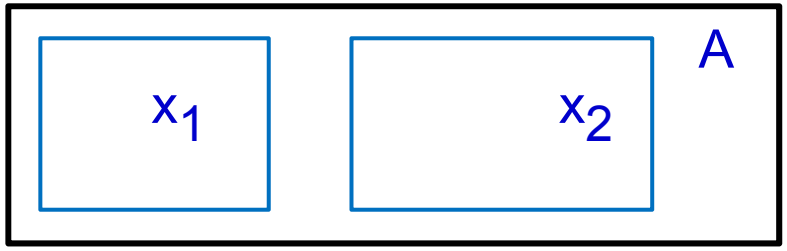
Example 1

A farmer has an area of A square kilometers to be planted with a combination of wheat and barley.

A limited amount F of fertilizer and P of insecticide can be used, each of which is required in different amounts per unit area for wheat (F_1, P_1) and barley (F_2, P_2).

Let S_1 be the selling price of wheat, and S_2 the price of barley, and denote the area planted with wheat and barley as x_1 and x_2 respectively.

The optimal number of square kilometers to plant with wheat vs. barley can be expressed as a linear programming problem.



Example 1 cont.

$$\text{Maximize} \quad S_1x_1 + S_2x_2$$ (the revenue – this is the "objective function")

subject to

$$x_1 + x_2 \leq A$$ (limit on total area)

$$F_1x_1 + F_2x_2 \leq F$$ (limit on fertilizer)

$$P_1x_2 + P_2x_2 \leq P$$ (limit on insecticide)

$$x_1 >= 0,\ x_2 \geq 0$$ (cannot plant a negative area)

which in matrix form becomes

maximize

$$\begin{bmatrix} S_1 & S_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

subject to

$$\begin{bmatrix} 1 & 1 \\ F_1 & F_2 \\ P_1 & P_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} A \\ F \\ P \end{bmatrix}, \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq 0$$

Example 2: Max flow

Given: a weighted directed graph, source s, destination t

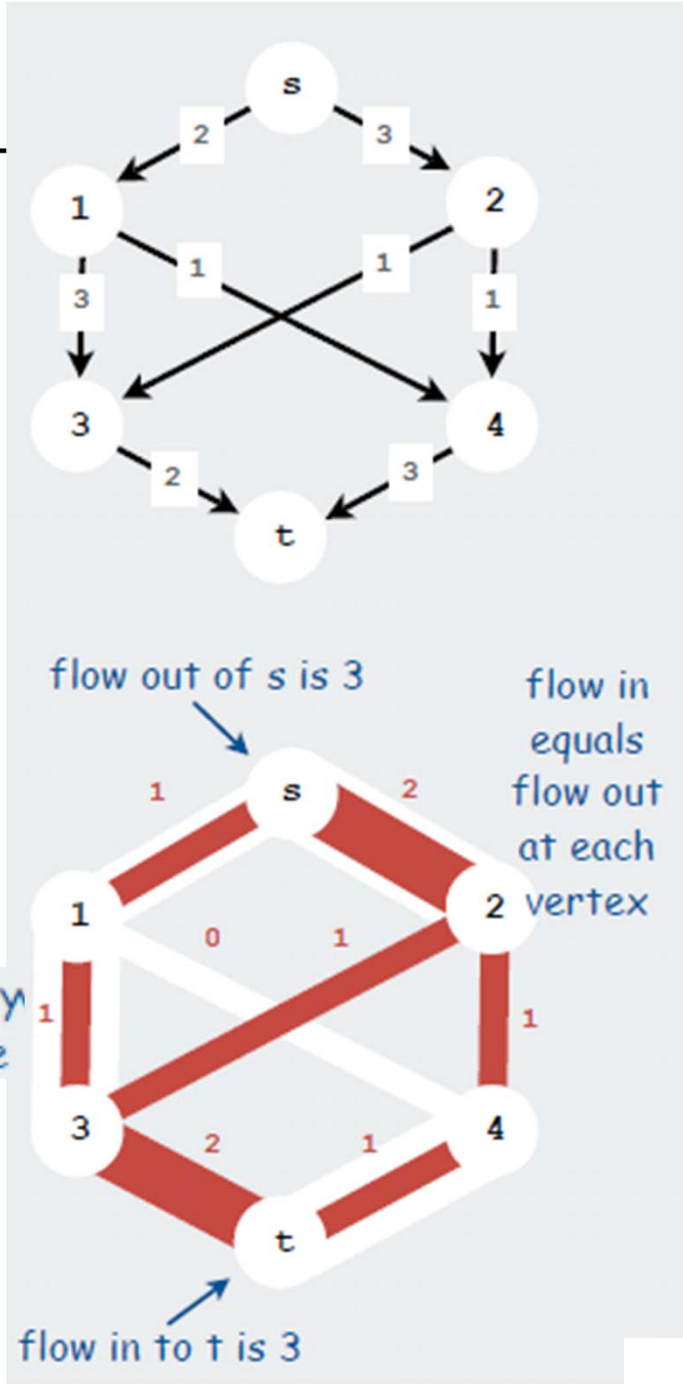
Interpret edge weights as capacities

Goal: Find maximum flow from s to t

- Flow does not exceed capacity in any edge
- Flow at every vertex satisfies equilibrium

[flow in equals flow out]

e.g. oil flowing through pipes, internet routing



LP formulation of maxflow problem

One variable per edge.

One inequality per edge, one equality per vertex.

maximize x_{ts}

subject to the constraints

capacity constraints:

$x_{s1} \le 2$

$x_{s2} \le 3$

$x_{13} \le 3$

$x_{14} \le 1$

$x_{23} \le 1$

$x_{24} \le 1$

$x_{3t} \le 2$

$x_{4t} \le 3$

interpretation:
x_{ij} = flow in edge i-j

equilibrium constraints:

$x_{ts} = x_{s1} + x_{s2}$

$x_{s1} = x_{13} + x_{14}$

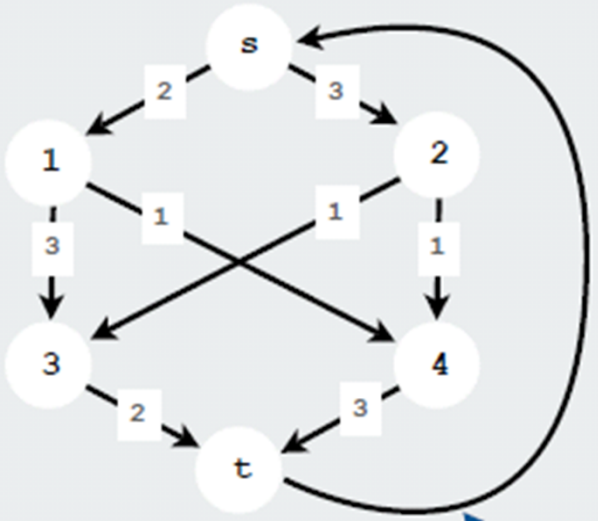
$x_{s2} = x_{23} + x_{24}$

$x_{13} + x_{23} = x_{3t}$

$x_{14} + x_{24} = x_{4t}$

$x_{3t} + x_{4t} = x_{ts}$

all $x_{ij} \ge 0$



solution

$x_{s1} = 2$

$x_{s2} = 2$

$x_{13} = 1$

$x_{14} = 1$

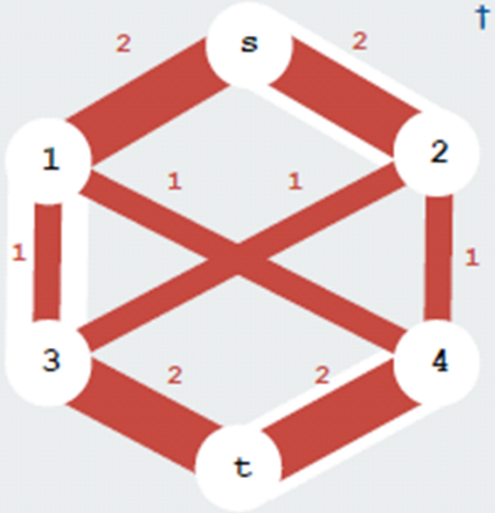
$x_{23} = 1$

$x_{24} = 1$

$x_{3t} = 2$

$x_{4t} = 2$

$x_{ts} = 4$ ← maxflow value

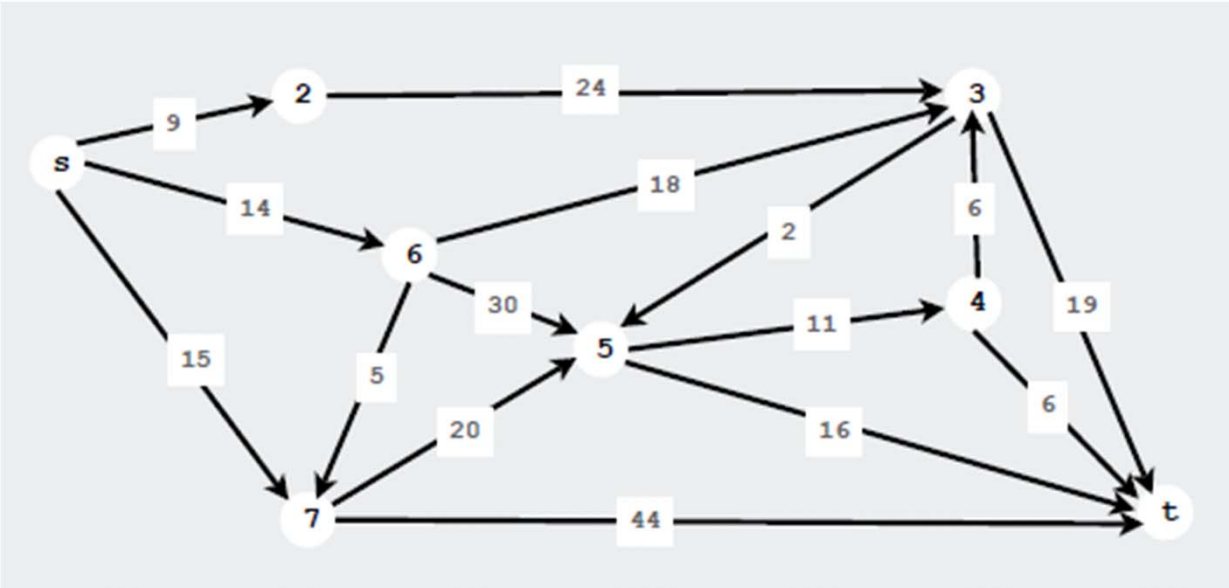


Example 3: shortest path

Given: a weighted directed graph, with a single source s

Distance from s to v: length of the shortest part from s to v

Goal: Find distance (and shortest path) to every vertex



e.g. plotting routes on Google maps

Application

Minimize number of stops (lengths = 1)

Minimize amount of time (positive lengths)

LP – why is it important?

We have seen examples of:

- allocating limited resources
- network flow
- shortest path

Others include:

- matching
- assignment …

It is a widely applicable problem-solving model because:

- non-negativity is the usual constraint on any variable that represents an amount of something
- one is often interested in bounds imposed by limited resources.

Dominates world of industry

Linear Programming – 2D example

$$\max_{x_1,x_2} f(x_1, x_2)$$

Cost function:

$$f(x_1, x_2) = 0x_1 + 0.5x_2$$

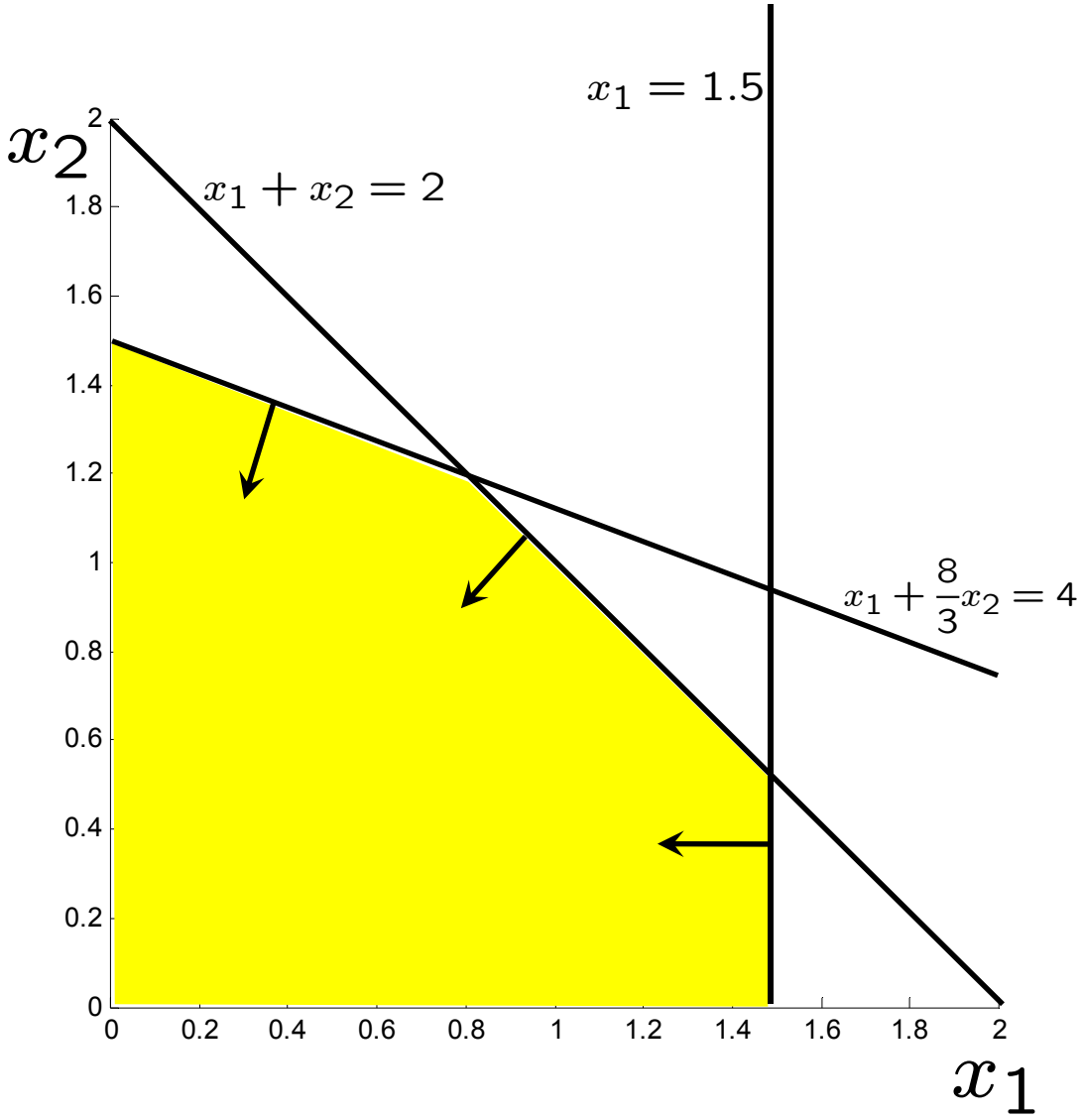
Inequality constraints:

$$\begin{aligned} x_1 + x_2 &\leq 2 \\ x_1 &\leq 1.5 \\ x_1 + \frac{8}{3}x_2 &\leq 4 \\ \\ x_1, x_2 &\geq 0 \end{aligned}$$

Feasible Region

Inequality constraints:

$$
\begin{aligned}
x_1 + x_2 &\leq 2 \\
x_1 &\leq 1.5 \\
x_1 + \frac{8}{3}x_2 &\leq 4 \\
x_1, x_2 &\geq 0
\end{aligned}
$$



LP example

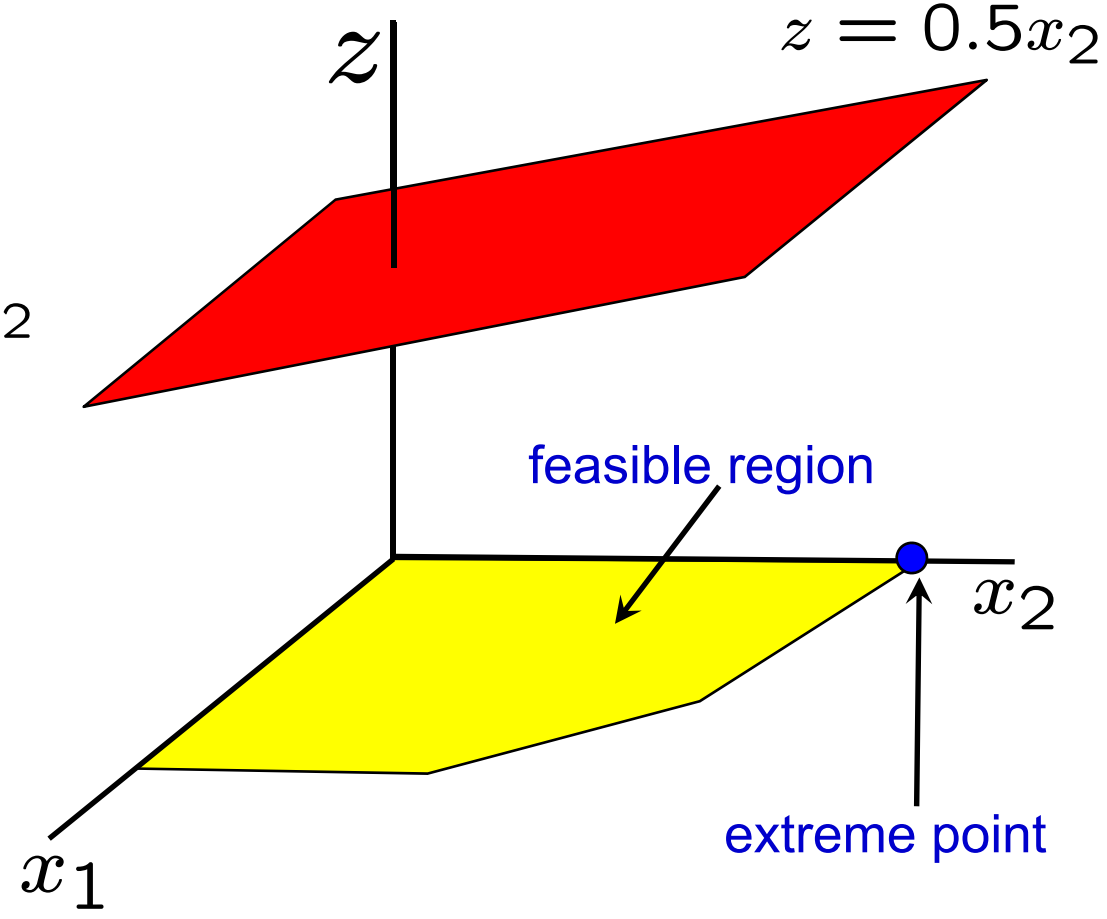
$$\max_{x_1,x_2} f(x_1, x_2)$$

Cost function:

$$f(x_1, x_2) = 0x_1 + 0.5x_2$$

Inequality constraints:

$$\begin{aligned} x_1 + x_2 &\leq 2 \\ x_1 &\leq 1.5 \\ x_1 + \frac{8}{3}x_2 &\leq 4 \\ x_1, x_2 &\geq 0 \end{aligned}$$



LP example

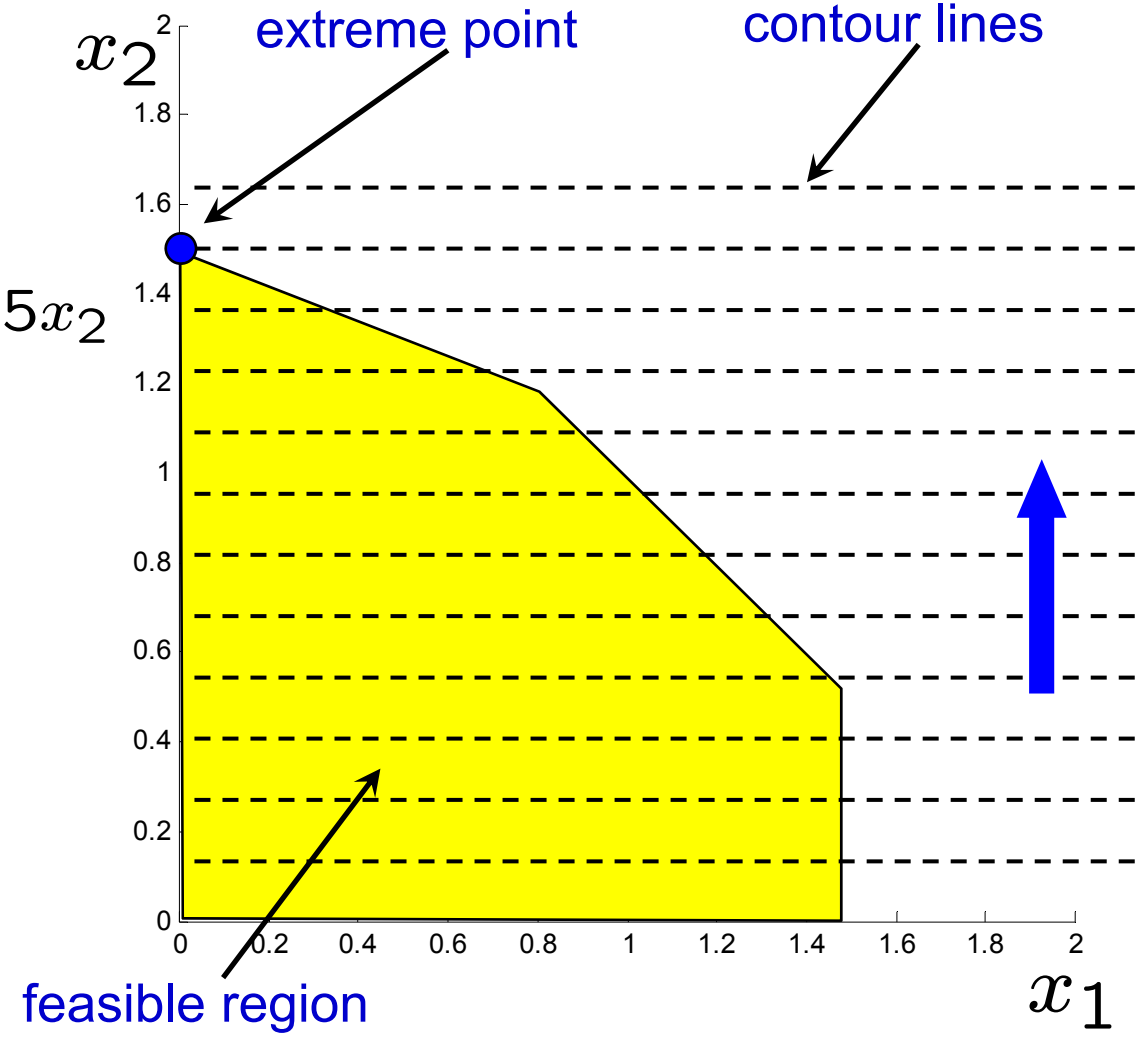
$$\max_{x_1,x_2} f(x_1, x_2)$$

Cost function:

$$f(x_1, x_2) = 0x_1 + 0.5x_2$$

Inequality constraints:

$$\begin{aligned} x_1 + x_2 &\leq 2 \\ x_1 &\leq 1.5 \\ x_1 + \frac{8}{3}x_2 &\leq 4 \\ x_1, x_2 &\geq 0 \end{aligned}$$



LP example – change of cost function

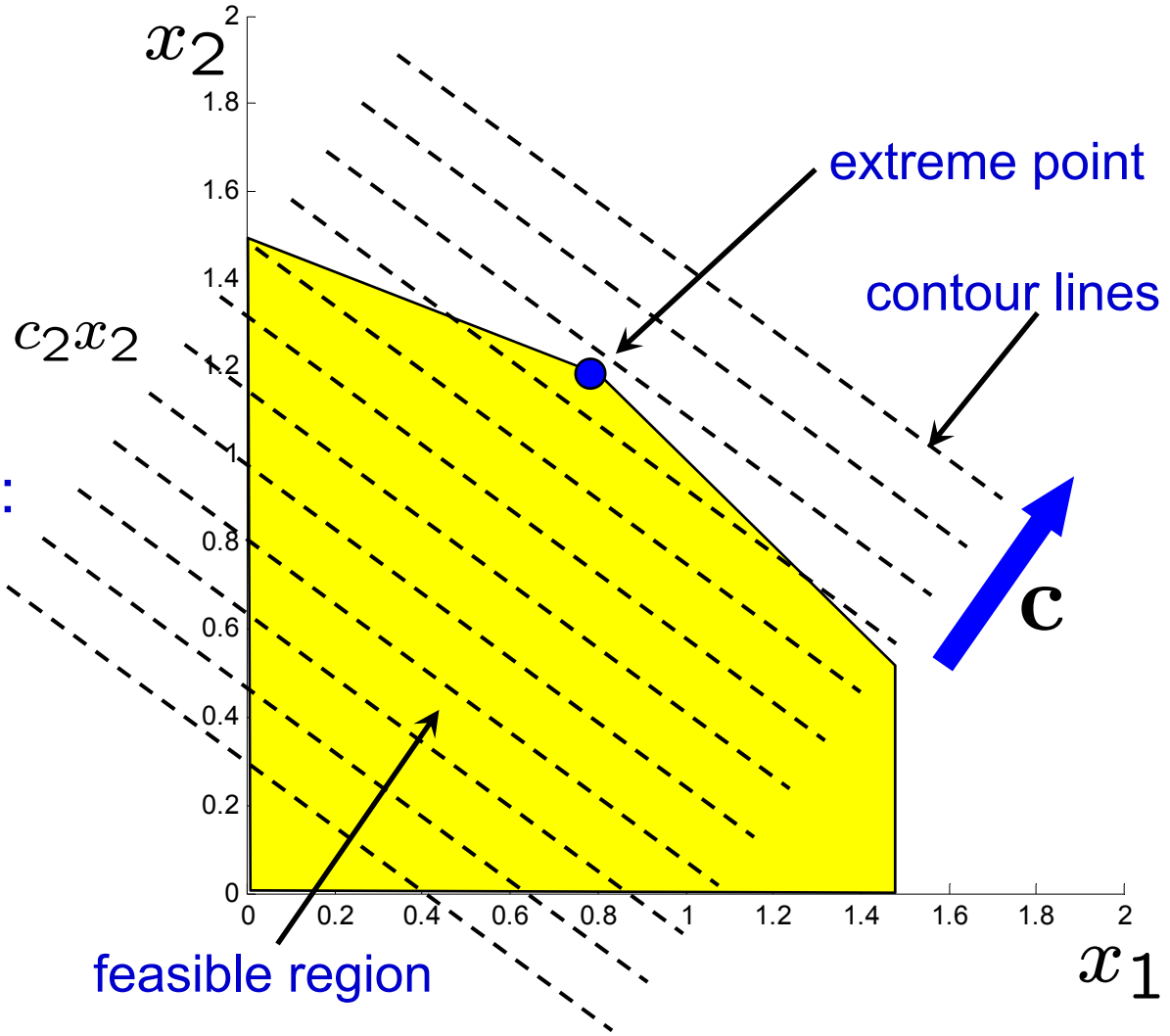
$$\max_{x_1,x_2} f(x_1, x_2)$$

Cost function:

$$\begin{aligned} f(x_1, x_2) &= \mathbf{c}^\top \mathbf{x} \\ &= c_1 x_1 + c_2 x_2 \end{aligned}$$

Inequality constraints:

$$\begin{aligned} x_1 + x_2 &\leq 2 \\ x_1 &\leq 1.5 \\ x_1 + \frac{8}{3} x_2 &\leq 4 \\ \\ x_1, x_2 &\geq 0 \end{aligned}$$

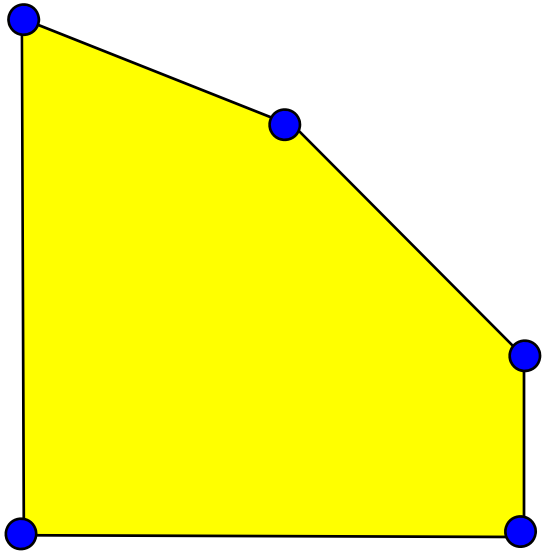


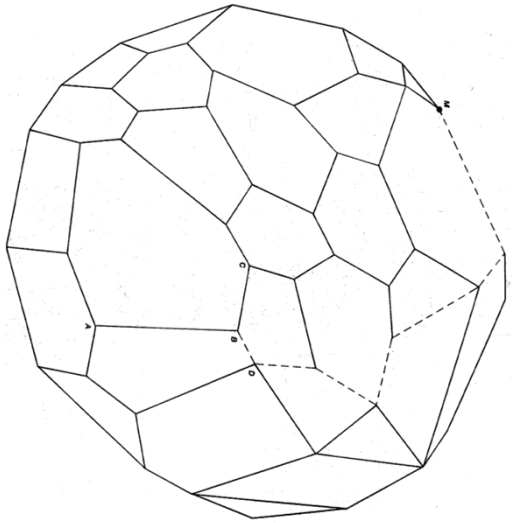
Linear Programming – optima at vertices

- The key point is that for any (linear) objective function the optima only occur at the corners (vertices) of the feasible polygonal region (never on the interior region).

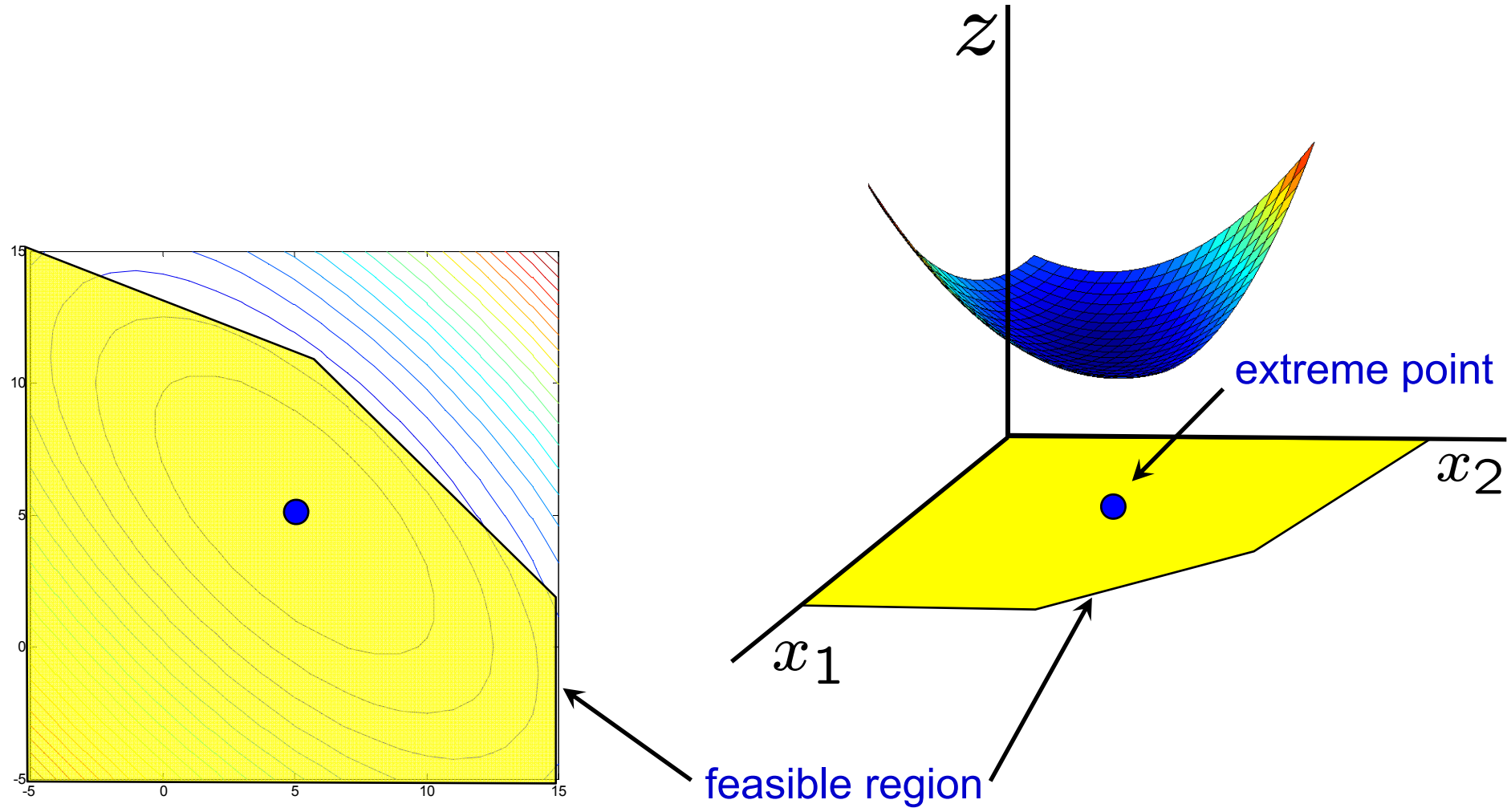
- Similarly, in 3D the optima only occur at the vertices of a polyhedron (and in nD at the vertices of a polytope).

- However, the optimum is not necessarily unique: it is possible to have a set of optimal solutions covering an edge or face of a polyhedron.





Cf Constrained Quadratic Optimization

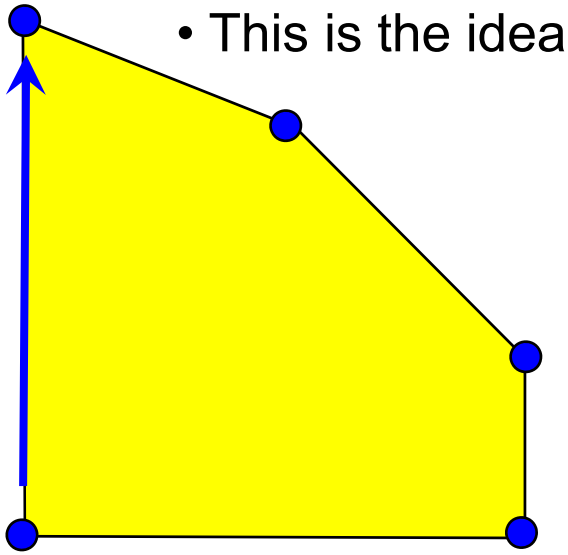


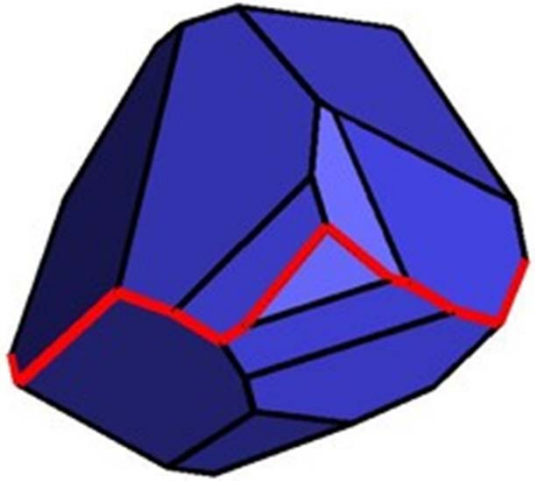
Linear Programming – solution idea

So, even though LP is formulated in terms of continuous variables, x, there are only a finite number of discrete possible solutions that need be explored.

How to find the maximum??

- Try every vertex? But there are too many in large problems.
- Instead, simply go from one vertex to the next increasing the cost function each time, and in an intelligent manner to avoid having to visit (and test) every vertex.
- This is the idea of the simplex algorithm





Sketch solutions for optimization methods

We will look at 2 methods of solution:

1. Simplex method
   - Tableau exploration of vertices based on linear algebra

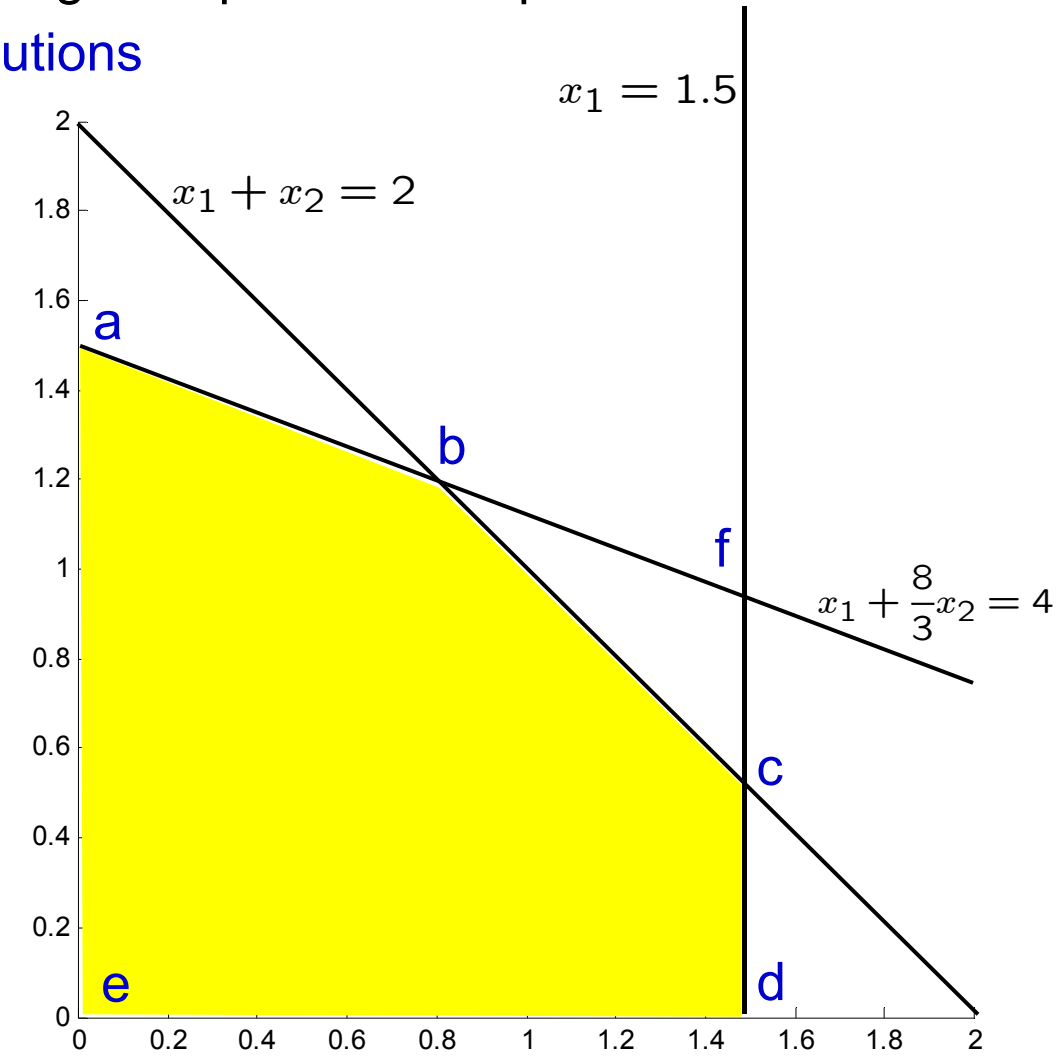
2. Interior point method
   - Continuous optimization with constraints cast as barriers

Simplex method

- Optimum must be at the intersection of constraints
- Intersections are easy to find, change inequalities to equalities
- Intersections are called basic solutions

- some intersections are outside the feasible region (e.g. f) and so need not be considered

- the others (which are vertices of the feasible region) are called basic feasible solutions



Worst complexity …

There are

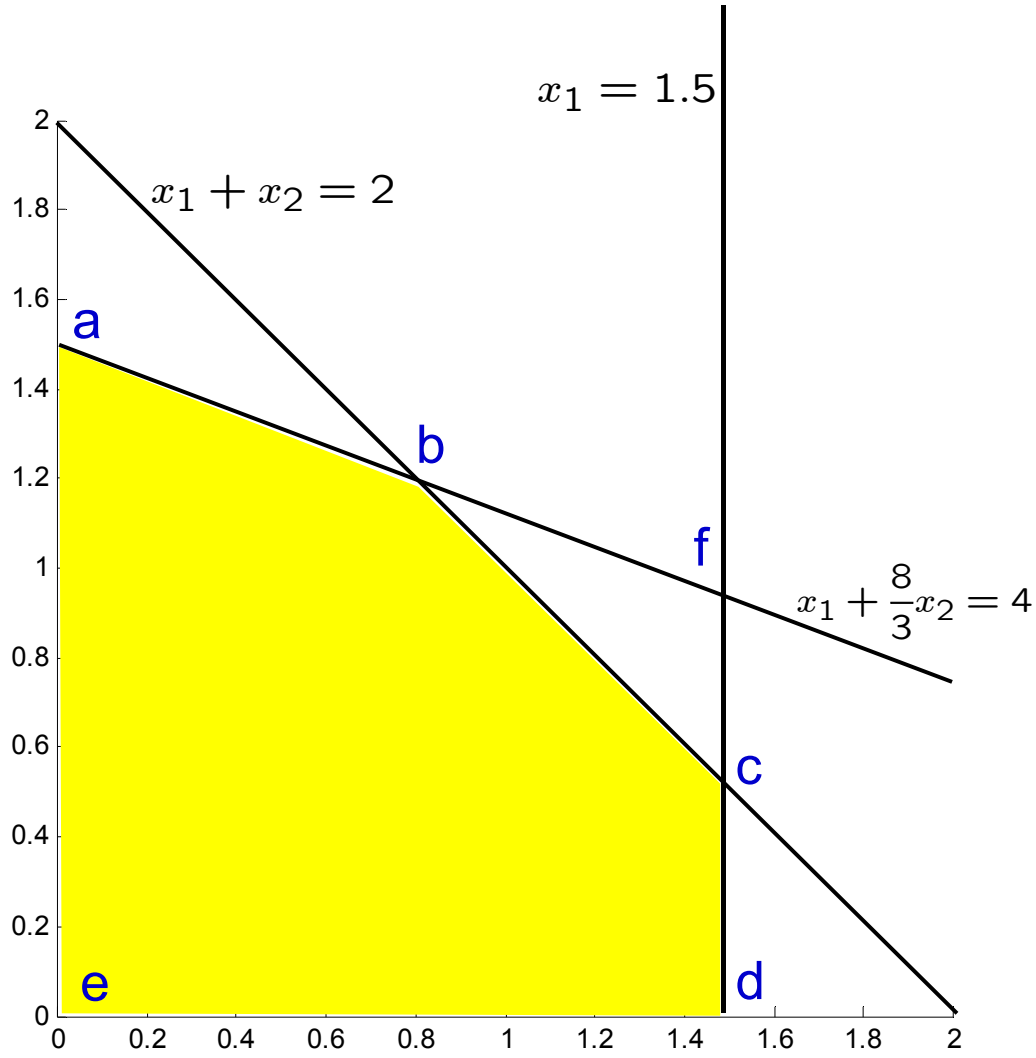
$$C_n^m = \frac{m!}{(m-n)!\,n!}$$

possible solutions, where m is the total number of constraints and n the dimension of the space.

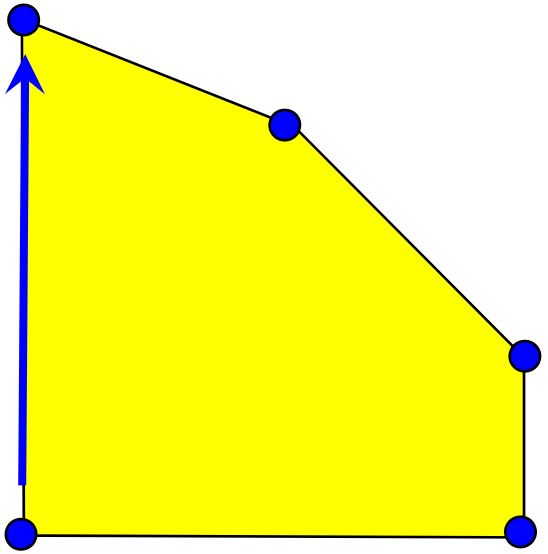
In this case

$$C_2^5 = \frac{5!}{(3)!\,2!} = 10$$

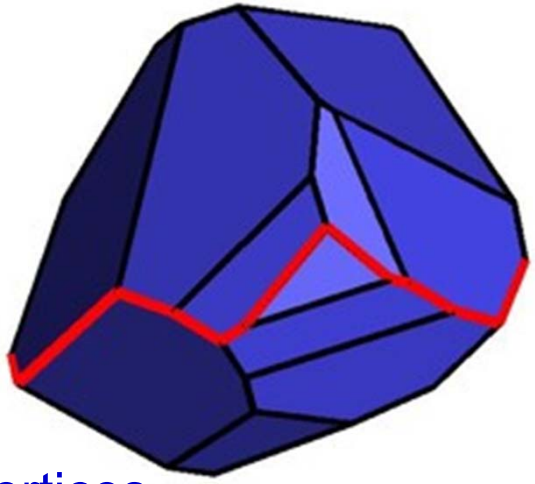
However, for large problems the number of solutions can be huge, and it is not realistic to explore them all.



The Simplex Algorithm



- Start from a basic feasible solution (i.e. a vertex of feasible region)
- Consider all the vertices connected to the current one by an edge
- Choose the vertex which increases the cost function the most (and is still feasible of course)
- Repeat until no further increases are possible



In practice this is very efficient and avoids visiting all vertices

Matlab LP Function linprog

Linprog() for medium scale problems uses the Simplex algorithm

Example

Find x that minimizes

$f(x) = -5x_1 - 4x_2 - 6x_3,$

subject to

$x_1 - x_2 + x_3 \le 20$

$3x_1 + 2x_2 + 4x_3 \le 42$

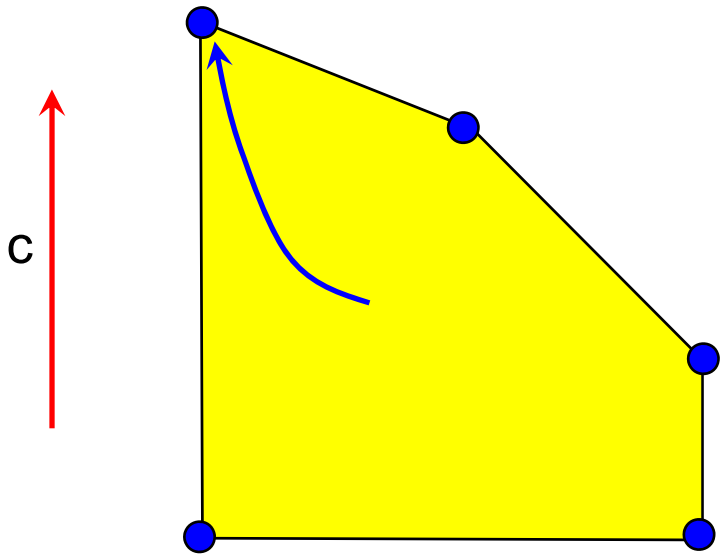
$3x_1 + 2x_2 \le 30$

$0 \le x_1,\ 0 \le x_2,\ 0 \le x_3.$

```
>> f = [-5; -4; -6];
>>  A = [1 -1 1
         3 2 4
         3 2 0];
>> b = [20; 42; 30];
>> lb = zeros(3,1);
>> x = linprog(f,A,b,[],[],lb);
>> Optimization terminated.
>> x
x =
    0.0000
   15.0000
    3.0000
```

Interior Point Method

- Solve LP using continuous optimization methods
- Represent inequalities by barrier functions
- Follow path through interior of feasible region to vertex



Barrier function method

We wish to solve the following problem

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & {\mathbf{a}_i}^\top \mathbf{x} \leq b_i,\ i = 1 \ldots m \end{aligned}$$

Problem could be rewritten as

$$\text{Minimize} \quad f(\mathbf{x}) + \sum_{i=1}^{m} I({\mathbf{a}_i}^\top \mathbf{x} - b_i)$$

where I is the indicator function

$$\begin{aligned} I(u) &= 0 \text{ for } u \leq 0 \\ &= \infty \text{ for } u > 0 \end{aligned}$$

The difficulty here is that $I(u)$ is not differentiable.

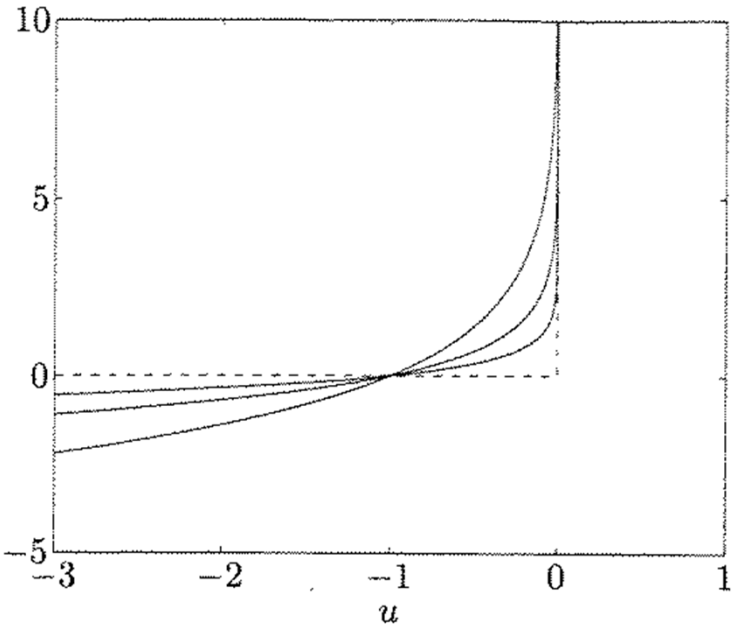
Approximation via logarithmic barrier functon

Approximate indicator function by logarithmic barrier

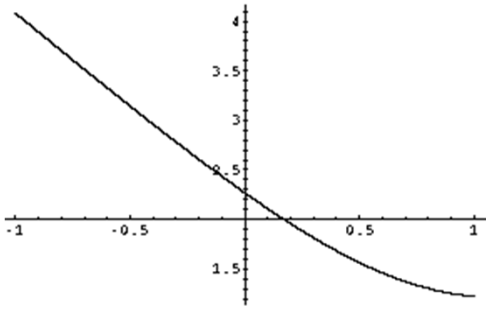
$$\text{Minimize} \quad f(\mathbf{x}) - \frac{1}{t}\sum_{i=1}^{m} \log(-\mathbf{a}_i{}^{\top}\mathbf{x} + b_i)$$

- for $t > 0$, $-(1/t)\log(-u)$ is a smooth approximation of $I(u)$
- approximation improves as $t \to \infty$

$$\frac{-\log(-u)}{t}$$



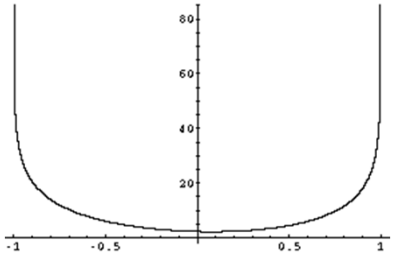
Barrier method – example

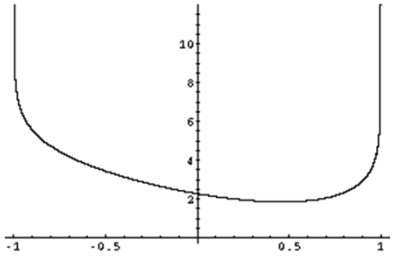


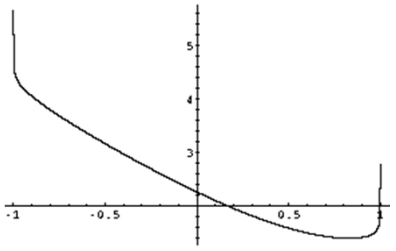
Function $f(x)$ to be minimized subject to $x \geq -1$, $x \leq 1$

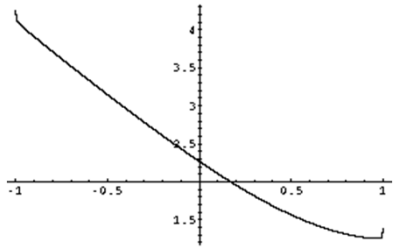
Minimizing function for different Values of t.

Minimas converge to the constrained minimum.









Function $tf(x) - \log(1 - x^2)$ for increasing values of t.

Algorithm for Interior Point Method

Problem becomes

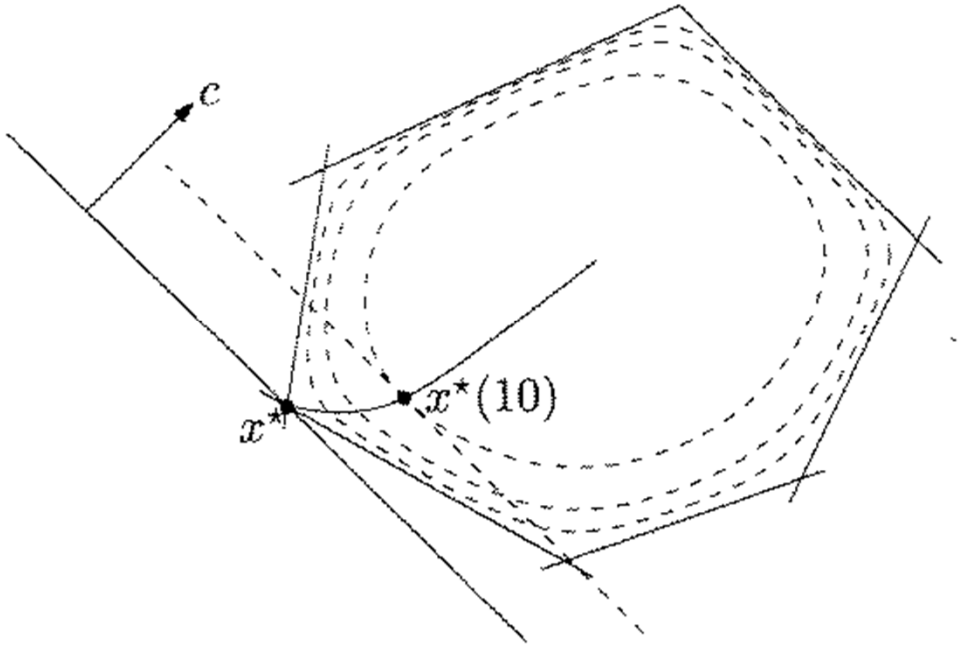
$$\text{Minimize} \quad tf(\mathbf{x}) - \sum_{i=1}^{m} \log(-\mathbf{a}_i{}^{\top}\mathbf{x} + b_i)$$

Algorithm

- Solve using Newton's method
- $t \rightarrow \mu t$
- repeat until convergence

As t increases this converges to the solution of the original problem

Example: interior point algorithm



Trace of the central path
– optimum for varying values of t.

Diagram copied from Boyd and Vandenberghe

Integer programming

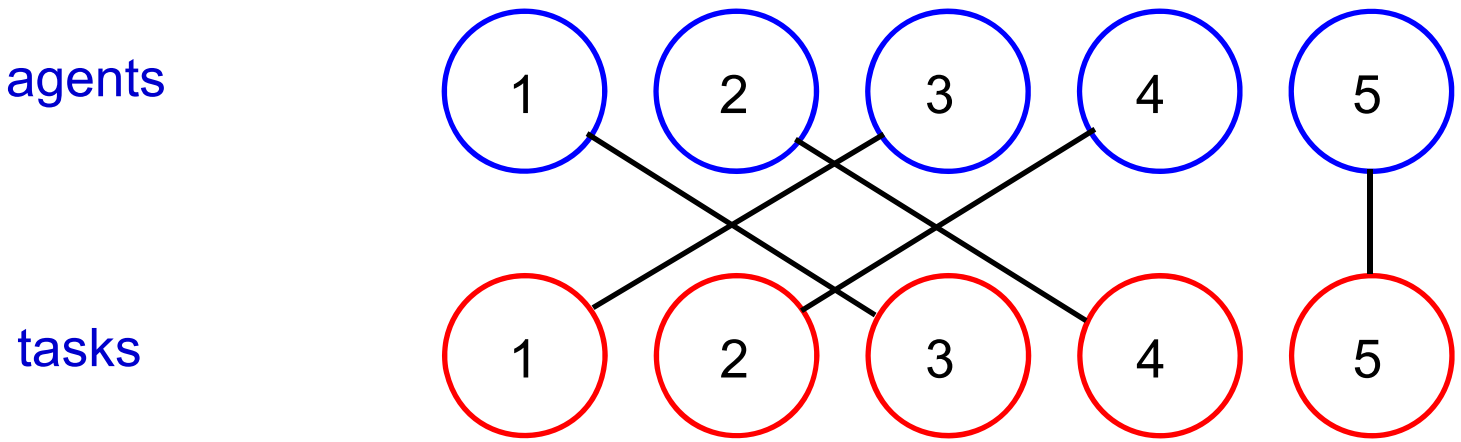
There are often situations where the solution is required to be an integer or have boolean values (0 or 1 only).

For example:

- assignment problems
- scheduling problems
- matching problems

Linear programming can also be used for these cases

Example: assignment problem



Objective:

- assign n agents to n tasks to minimize total cost

Assignment constraints:

- each agent assigned to one task only
- each task assigned to one agent only

cost matrix

tasks

agents

	1'	2'	3'	4'	5'
1	3	8	9	15	10
2	4	10	7	16	14
3	9	13	11	19	10
4	8	13	12	20	13
5	1	7	5	11	9

cost = 3 + 10 + 11 + 20 + 9 =53

tasks

agents

	1'	2'	3'	4'	5'
1	3	8	9	15	10
2	4	10	7	16	14
3	9	13	11	19	10
4	8	13	12	20	13
5	1	7	5	11	9

cost = 8 + 7 + 20 + 8 + 11 = 44

Example: assignment problem

Problem specification

x_{ij} is the assignment of agent i to task j (can take values 0 or 1)

c_{ij} is the (non-negative) cost of assigning agent i to task j

Minimize total cost $f(\mathbf{x}) = \sum_{ij} x_{ij} c_{ij}$ over the n^2 variables x_{ij}

Example solution for x_{ij} →

tasks j

agents i

0	1	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	0	0	1
1	0	0	0	0

each agent i assigned to one task only

- only one entry in each row

each task j assigned to one agent only

- only one entry in each column

Example: assignment problem

Linear Programme formulation

agents i \ tasks j					
	0	1	0	0	0
	0	0	0	1	0
	0	0	1	0	0
	0	0	0	0	1
	1	0	0	0	0

$$\min_{x_{ij}} \quad f(\mathbf{x}) = \sum_{ij} x_{ij} c_{ij}$$

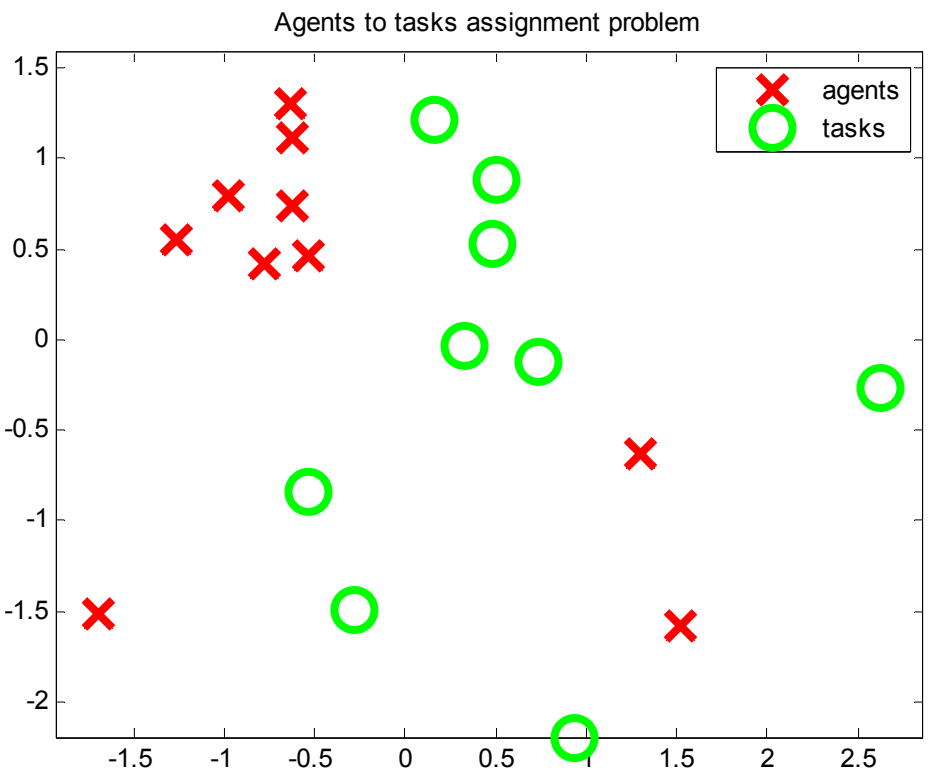
subject to

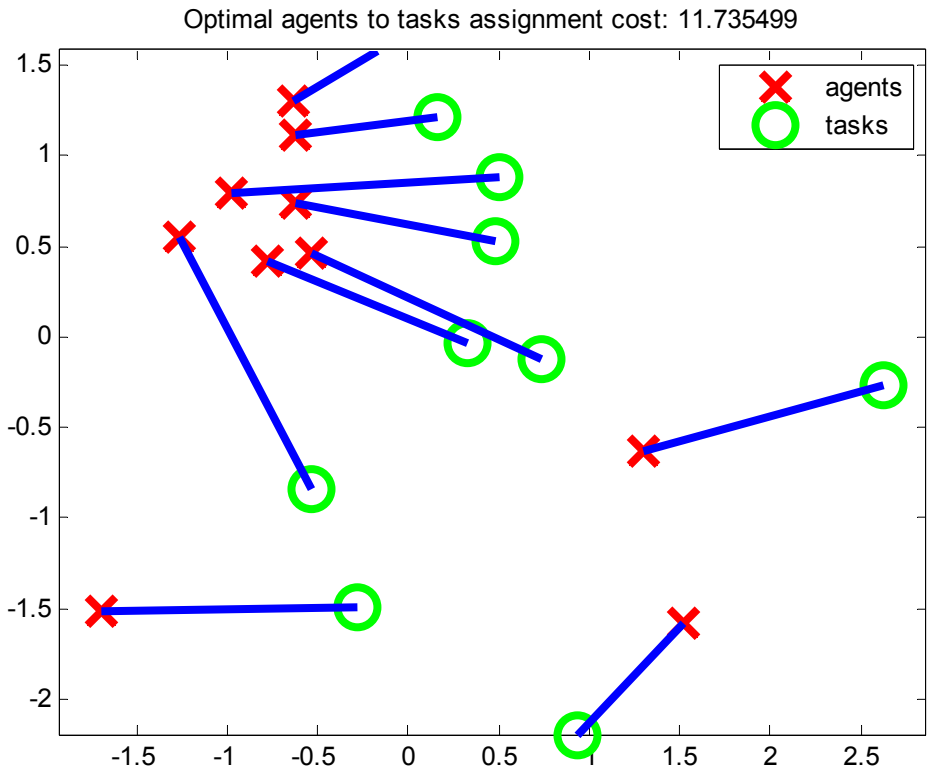
(inequalities) $\sum_j x_{ij} \leq 1, \forall i$ each agent assigned to at most one task

(equalities) $\sum_i x_{ij} = 1, \forall j$ each task assigned to exactly one agent

This is a relaxation of the problem because the variables x_{ij} are not forced to take boolean values.

However, it can be shown that the solution x_{ij} only takes the values 0 or 1





Cost of assignment = distance between agent and task

e.g. application: tracking, correspondence

Example application: tracking pedestrians

Multiple Object Tracking using Flow Linear Programming, Berclaz, Fleuret & Fua, 2009

What is next?

- Convexity (when does a function have only a global minimum?)
- Robust cost functions
- Stochastic algorithms