Data Structures and Algorithm

Xiaoqing Zheng

zhengxq@fudan.edu.cn

Polynomial time algorithms

Almost all the algorithms we have studied thus far have been ***polynomial-time*** algorithms.

- On input of size n, their worst-case running time is $O(n^k)$ for some constant k.

Undecidable

HALTS(P, X)

DIAGONAL(X)

1. a: **if** HALTS(X, X)
2. **then goto** a
3. **else** halt

How about

DIAGONAL(*DIAGONAL*)

Shortest vs. longest simple paths

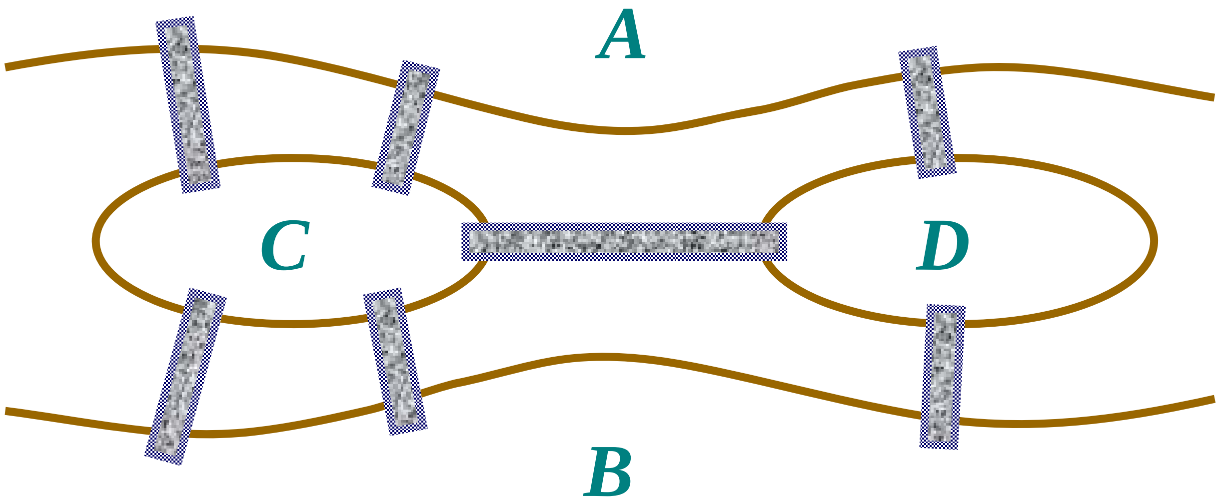
Shortest simple path

Given a directed graph $G = (V, E)$, find the ***shortest*** simple path between two vertices.

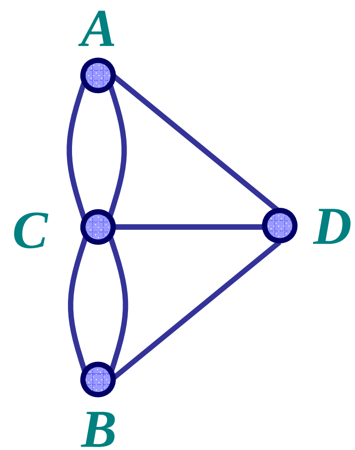
Longest simple path

Given a directed graph $G = (V, E)$, find the ***longest*** simple path between two vertices.

Konigsberg's Bridges Problem



The river Pregel divides the town of Konigsberg into four separate land masses, *A*, *B*, *C*, and *D*. Seven bridges connect the various parts of town, and some of the town's curious citizens wondered if it were possible to take a journey across all seven bridges without having to cross any bridge more than once.



Euler tour vs. Hamiltonian cycle

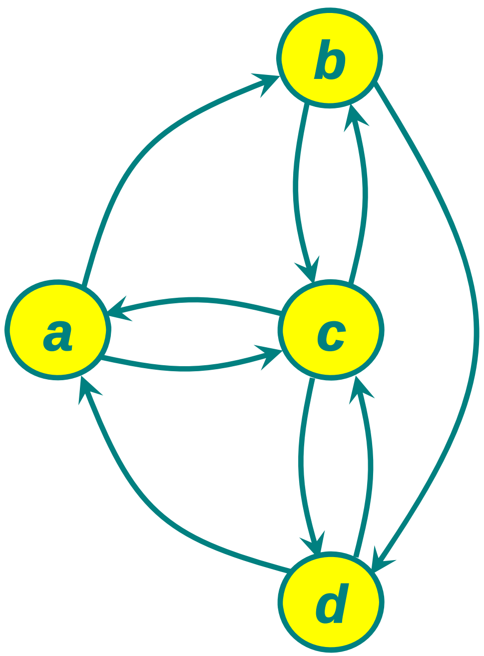
Euler tour

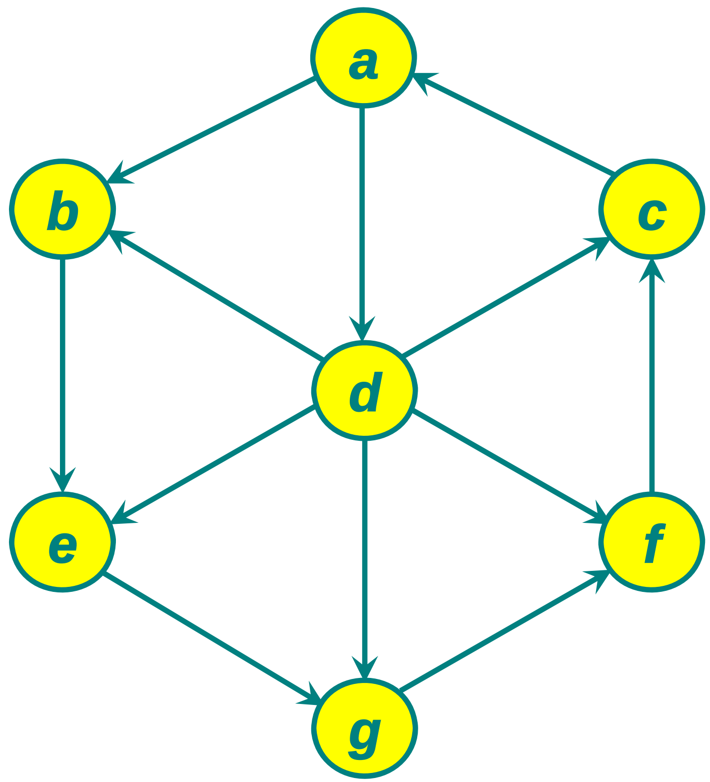
An Euler tour of a connected, directed graph $G = (V, E)$ is a cycle that traverses ***each edge*** of G exactly once.

Hamiltonian cycle

An Hamiltonian cycle of a directed graph $G = (V, E)$ is a simple cycle that contains ***each vertex*** in V.

Euler tour





Traveling salesman problem

Analysis of traveling salesman problem

Given a finite number of "cities" along with the cost of travel between each pair of them, find the ***cheapest way*** of visiting each city exactly once and finishing at the city he starts from.

- 19 ***Cities***
- ***Possible routes*** $= 18! = 6.40237 \times 10^{15}$
- ***life time*** $= 80 \times 365 \times 24 \times 60 \times 60$
  $= 3{,}1536 \times 10^{9}$
- ***Computer speed*** = 10000 ***routes/second***

253.77 ***Generation!***

Certificate

- The class **NP** consists of those problems that are ***"verifiable"*** in ***polynomial time***.
- If we were given a ***certificate*** of a solution, then we could verify that the ***"certificate"*** is correct in time polynomial in the size of the input to the problem.

$$\mathbf{P} \subseteq \mathbf{NP}$$

$$\mathbf{P} \subset \mathbf{NP} \text{ or } \mathbf{P} = \mathbf{NP}$$

The Clay Mathematics Institute is offering a US$1 million reward to anyone who has a formal proof that problem

NP-complete

- a problem is in the class **NPC**—and we refer to it as being ***NP-complete***—if it is in **NP** and is as ***"hard"*** as any problem in **NP**.
- Most theoretical computer scientists believe that the NP-complete problem are ***intractable***.
- Given the wide range of **NP-complete** problems that have been studied to date without anyone having discovered a ***polynomial time solution*** to any of them.
- It is important to become familiar with this remarkable class of problems.

The clique problem

A ***clique*** in an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge in E.

- The ***size*** of a clique is the number of vertices it contains.
- The clique problem is the optimization problem of finding a ***clique of maximum size*** in a graph.

CLIQUE = {<G, k>: G is a graph with a clique of size k}.

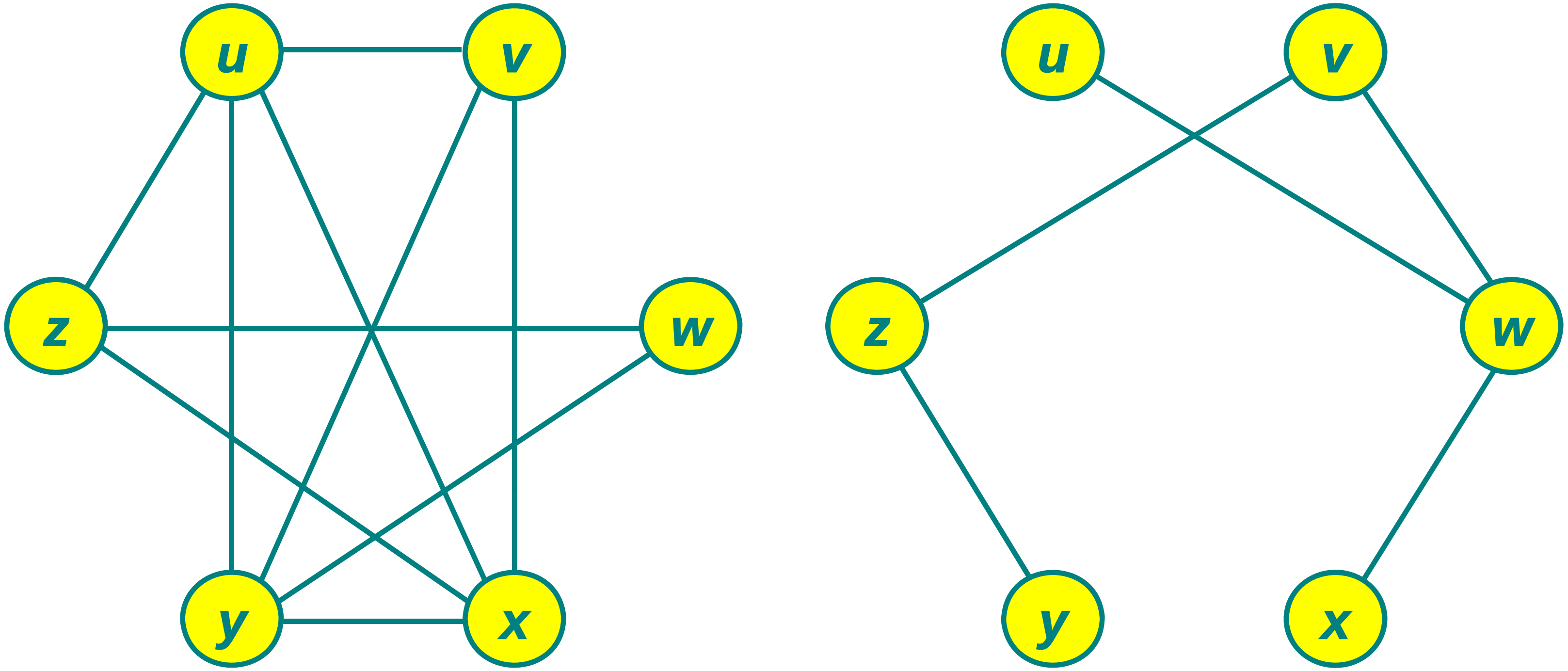
The vertex-cover problem

A ***vertex-cover*** of an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ such that if $(u, v) \in E$, then $u \in V'$ or $v \in V'$ (or both).

- The ***size*** of a vertex cover is the number of vertices in it.
- The vertex cover problem is to find a ***vertex cover of minimum size*** in a given graph.

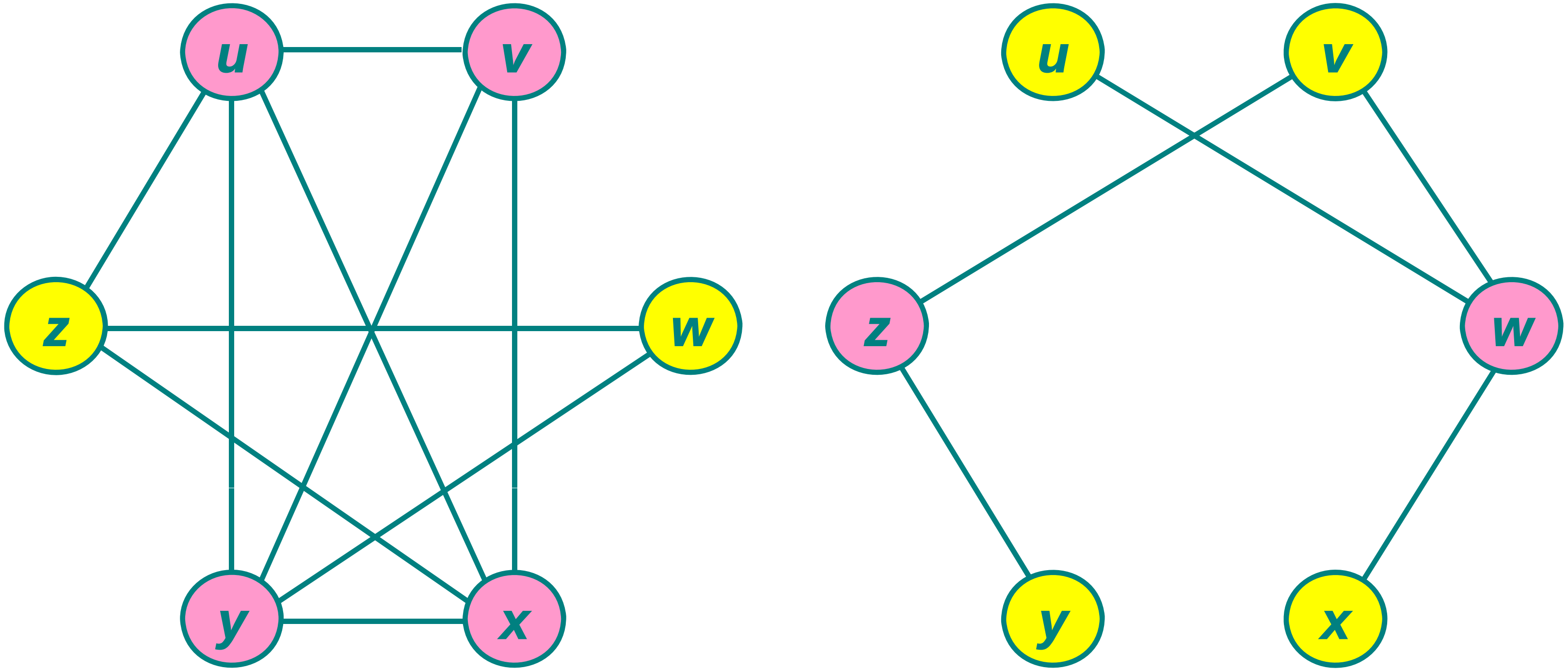
VERTEX-COVER = $\{\langle G, k\rangle$: G is a graph has a vertex cover of size $k\}$.

Complement of graph



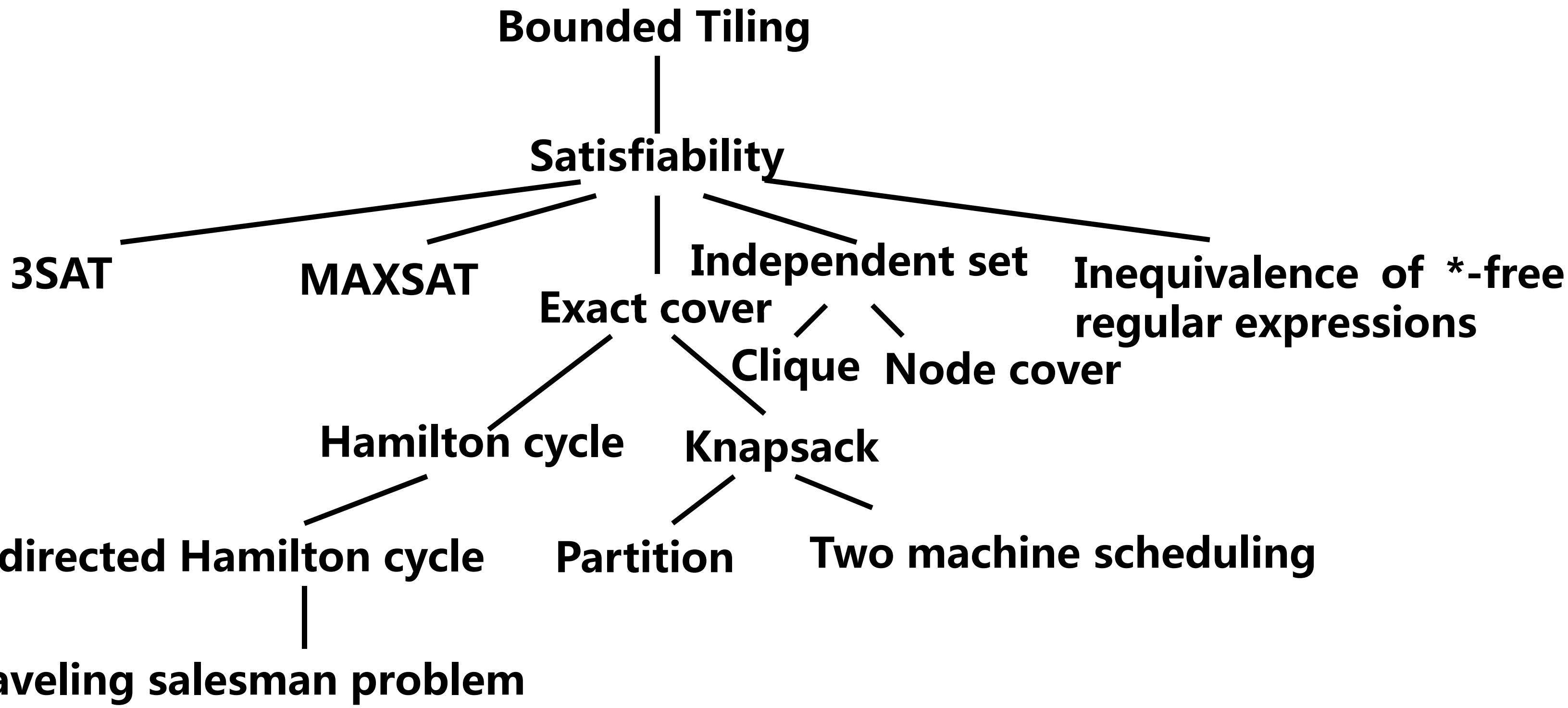
Given an undirected graph $G = (V, E)$, define the ***complement*** of G as $\overline{G} = (V, \overline{E})$, where $\overline{E} = \{(u, v): u, v \in V, u \neq v,$ and $(u, v) \notin E\}$.

Reduction



Graph G has a ***clique*** of size k if and only if the graph $\overline{G}$ has a ***vertex cover*** of size $|V| - k$.

NP complete


Bounded Tiling
Satisfiability
3SAT
MAXSAT
Exact cover
Independent set
Inequivalence of *-free regular expressions
Clique
Node cover
Hamilton cycle
Knapsack
Undirected Hamilton cycle
Partition
Two machine scheduling
Traveling salesman problem

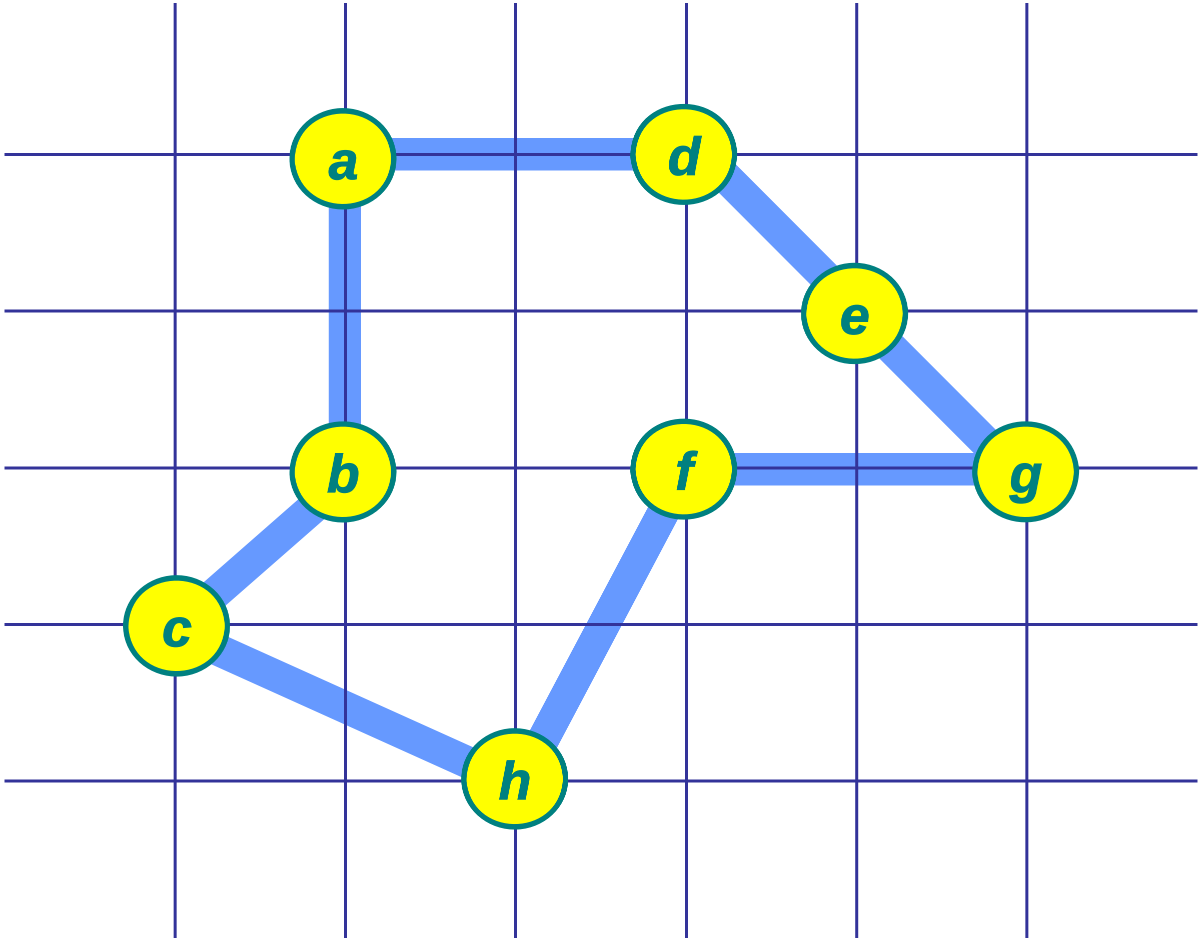
NP-complete and P

NP-Complete	Polynomial-Time
3SAT	2SAT
Traveling Salesman Problem	Minimum Spanning tree
Longest Path	Shortest Path
3D Matching	Bipartite Matching
Knapsack	Unary Knapsack
Independent Set	Independent Set on Trees
Integer Linear Programming	Linear Programming
Rudrata Path	Euler Path
Balanced Cut	Minimum Cut

Problem classification

- **P** (*polynomial*)
- **NPC** (*NP-Complete*)
- **EXP** (*exponential*)
- **Undecidables**

Approximation algorithms



Triangle inequality

Traveling salesman problem

Dealing with hard problems

What to do if:

- *Divide and conquer*
- *Dynamic programming*
- *Greedy algorithm*
- *Network Flows*
- *Linear Programming*
- ...

Linear programming

Products	**Machine time**	**Raw materials**	**Profits**
A	3	3	120
B	5	2	100
Total	600	500	

Maximize $120x_A + 100x_B$

subject to

$$3x_A + 5x_B \leq 600$$
$$3x_A + 2x_B \leq 500$$
$$x_A, x_B \geq 0$$

Topics

- *Number-theoretic algorithms*
- *Computational geometry*
- *Linear Programming*

Any question?

Xiaoqing Zheng
Fundan University