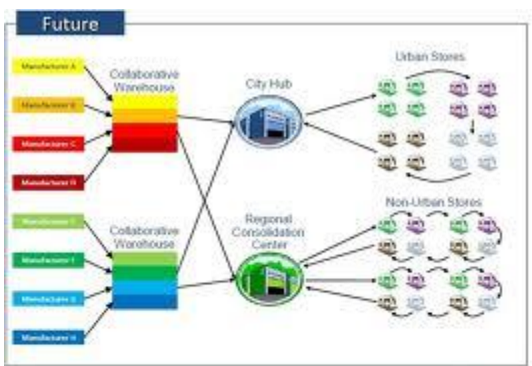
ISE/OR 766 NETWORK FLOWS

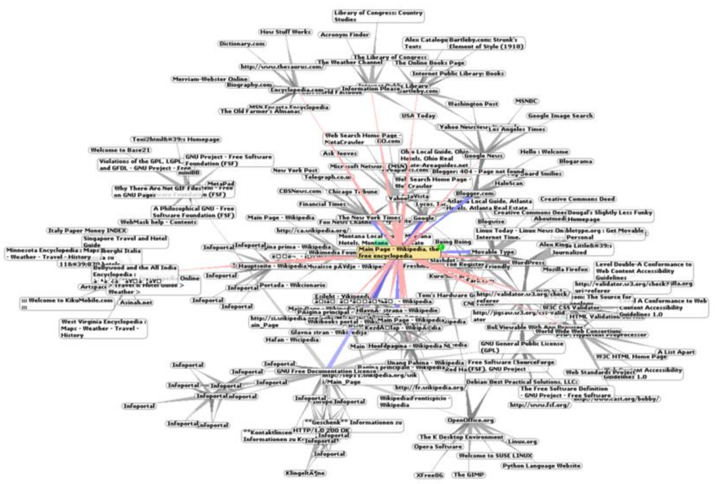
LECTURE 1: INTRODUCTION

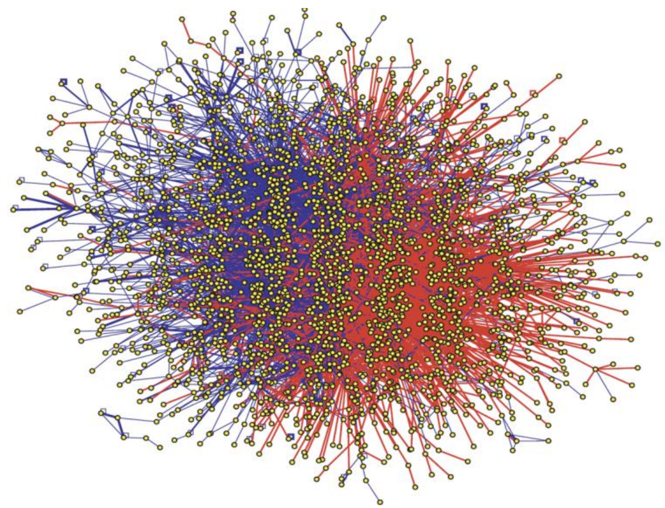
Shu-Cherng Fang
North Carolina State University – Fall 2026

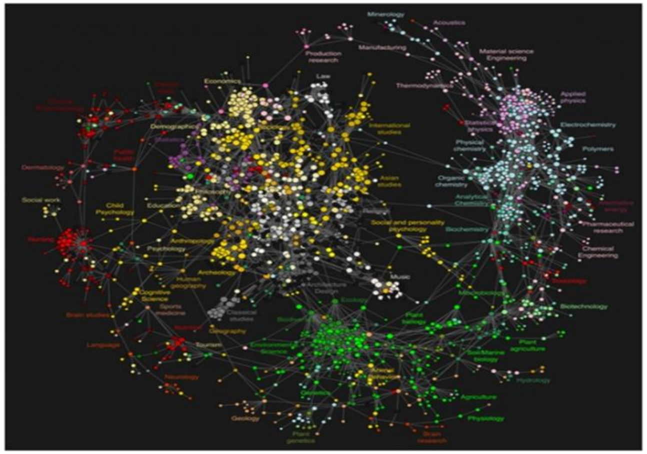
What is a Network?

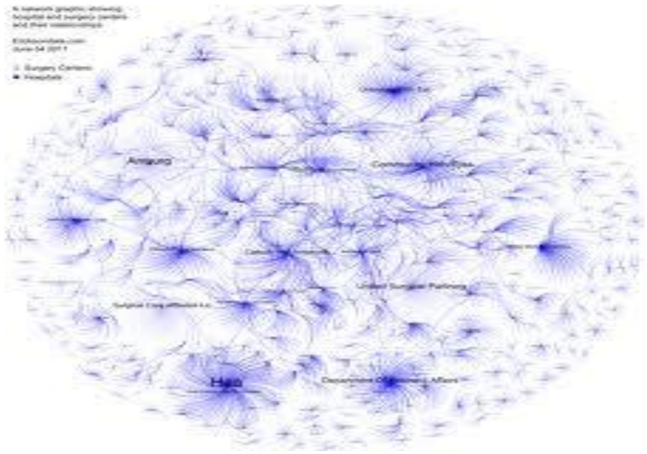


What is a Network?









What is the meaning of Network Flows?

In our course, a network G=(N,A) consists of a finite number of nodes (in N) which are connected by arcs (in A) and there may associate some capacity and cost with each arc and node.

Commodities, information, cash, manpower etc. are moving from one node to another along the arcs subject to capacity and other constraints.

Where Network Flows Arise?

- **Transportation**
  - **Transportation of goods over transportation networks**
  - **Scheduling of fleets of airplanes: time/space networks**
- **Manufacturing**
  - **Scheduling of goods for manufacturing**
  - **Flow of manufactured items within inventory systems**
- **Communications**
  - **Design and expansion of communication systems**
  - **Flow of information across networks**
- **Personnel Assignment**
  - **Assignment of crews to airline schedules**
  - **Assignment of drivers to vehicles**
- **Other Networks: Power, Hydraulic, Electric, …**

What is Network Optimization about?

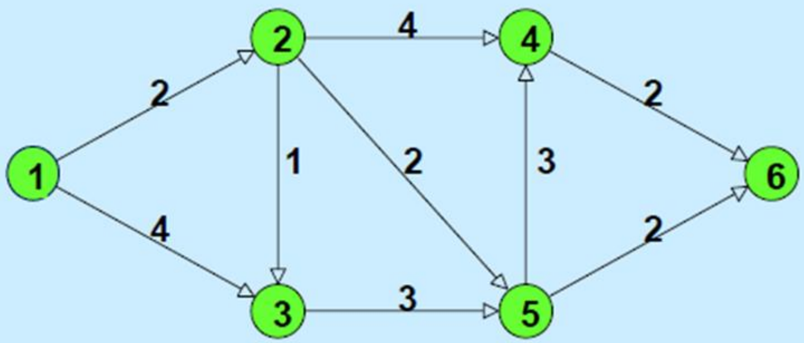
Network optimization deals with problems with at least one objective function to be optimized over a network structure.

What's Special about Network Optimization?

- Network optimization lies in the middle of the great divide that separates the continuous and discrete optimization problems.
- It strongly ties linear programming and combinatorial optimization – the extreme points of the underlying polyhedron are of integer values and represent solutions of combinatorial problems that are seemingly unrelated to linear programming.
- Network models provide ideal vehicles for explaining many of the fundamental ideas in both continuous and discrete optimization.

Example 1

The shortest path problem



Consider a network G = (N, A) in which there is an origin node s and a destination node t.

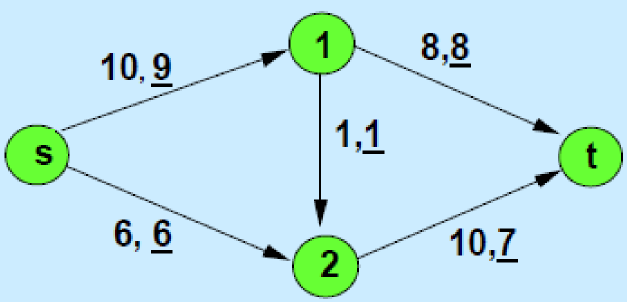
standard notation: n = |N|, m = |A|

What is the shortest path from s to t?

Example 2

The Maximum Flow Problem

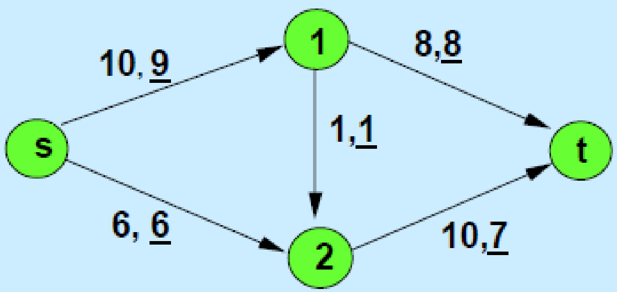
- Directed Graph G = (N, A).
  - Source s
  - Sink t
  - Capacities u_{ij} on arc (i,j)
  - Maximize the flow from s to t, subject to
- Flow out of i = Flow into i, for i ≠ s or t.



A Network with Arc Capacities and Flows

Example 2 (cont.)

Representing the Max Flow as an LP



Flow out of i - Flow into i = 0 for $i \neq s$ or t.

$$
\begin{aligned}
\max \quad & v \\
\text{s.t} \quad & x_{s1} + x_{s2} = v \\
& -x_{s1} + x_{12} + x_{1t} = 0 \\
& -x_{s2} - x_{12} + x_{2t} = 0 \\
& -x_{1t} - x_{2t} = -v
\end{aligned}
$$

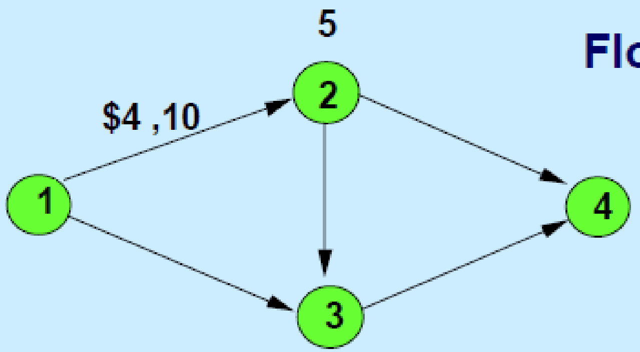
$0 \leq x_{ij} \leq u_{ij}$ for all (i,j)

$$
\begin{aligned}
\max \quad & v \\
\text{s.t.} \quad & \textstyle\sum_j x_{sj} = v \\
& \textstyle\sum_j x_{ij} - \sum_j x_{ji} = 0 \quad \text{for each } i \neq s \text{ or } t \\
& -\textstyle\sum_i x_{it} = -v
\end{aligned}
$$

$0 \leq x_{ij} \leq u_{ij}$ for all (i,j)

Example 3

Min Cost Flows



Flow out of i - Flow into i = b(i)

Each arc has a linear cost and a capacity

min $\sum_{i,j} c_{ij}x_{ij}$

s.t $\sum_j x_{ij} - \sum_j x_{ji} = b(i)$ **for each i**

$0 \le x_{ij} \le u_{ij}$ **for all (i,j)**

Covered in detail in Chapter 1 of AMO

What is Combinatorial Optimization?

Combinatorial analysis is the mathematical study of the arrangement, grouping, ordering, or selection of discrete objects, usually finite in number. Traditionally, combinatorialists have been concerned with questions of existence or of enumeration.

Recently, a new line of combinatorial investigation has gained increasing importance. The question asked is "What is a *best* arrangement?"

Some Representative Optimization Problems

- Arc-Covering Problem

  An arc (i, j) is said to "cover" nodes i and j. What is the smallest possible subset of arcs that can be chosen, such that each node of G is covered by at least one arc of the subset?

- Arc-Coloring Problem

  It is desired to paint the arcs of G various colors, subject to the constraint that not all the arcs in any cycle are painted the same color. What is the smallest number of colors that will suffice?

Some Representative Optimization Problems

- Spanning-Tree Problem

  It is desired to find a subset of arcs that when these arcs are removed from G, the graph remains connected. For what subset of arcs is the sum of the arc lengths maximized? (The complementary set of arcs is a "minimal spanning tree".)

- Shortest Path Problem

  What is the shortest path between two specified nodes of G?

- Chinese Postman's Problem

  It is desired to find a tour (a closed path) that passes through each arc in G at least once. What is the shortest such tour?

Some Representative Optimization Problems

- Assignment Problem

  An $n \times n$ matrix $W = (w_{ij})$ is given. It is desired to find a subset of the elements in W, with exactly one element in each row and in each column. For what subset is the sum of the elements minimized?

- Machine Sequencing Problem

  A number of jobs are to be processed by a machine. For each job a processing time and a deadline are specified. How should the jobs be sequenced, so that the number of late jobs is minimized?

When is a problem solved?

- A combinatorial problem is NOT solved if we cannot live long enough to see the answer!

- The challenge is to develop algorithms for which the required number of elementary operations is acceptably small – polynomial time algorithms!

The Criterion of Polynomial Boundedness

- One generally accepted standard in the realm of combinatorial optimization is that of "polynomial boundedness".

- An algorithm is considered *"good"* if the required number of elementary computational steps is bounded by a polynomial in the size of the problem.

- The *size of a problem instance* is the number of *bits* required to encode it.

Size of Problem - Example

•

- The size of an *m by n* matrix *A* is not *m x n*.

* If each element takes *K* bits, the size is *mnK*

* e.g., if $2^{17} < Max\{|a_{ij}|\} < 2^{18}$,

then $K = 18 = O(\log(|a_{max}|))$.

Polynomial Time Algorithm

An algorithm runs in polynomial time if the number of steps taken by an algorithm on any instance *I* is bounded by a polynomial in the size of *I*.

An algorithm runs in exponential time if it does not run in polynomial time.

Solution Methods

- Linear Programming
- Recursion and Enumeration
- Heuristics/Soft Computing
- Statistical Sampling
- Special and Ad Hoc Techniques

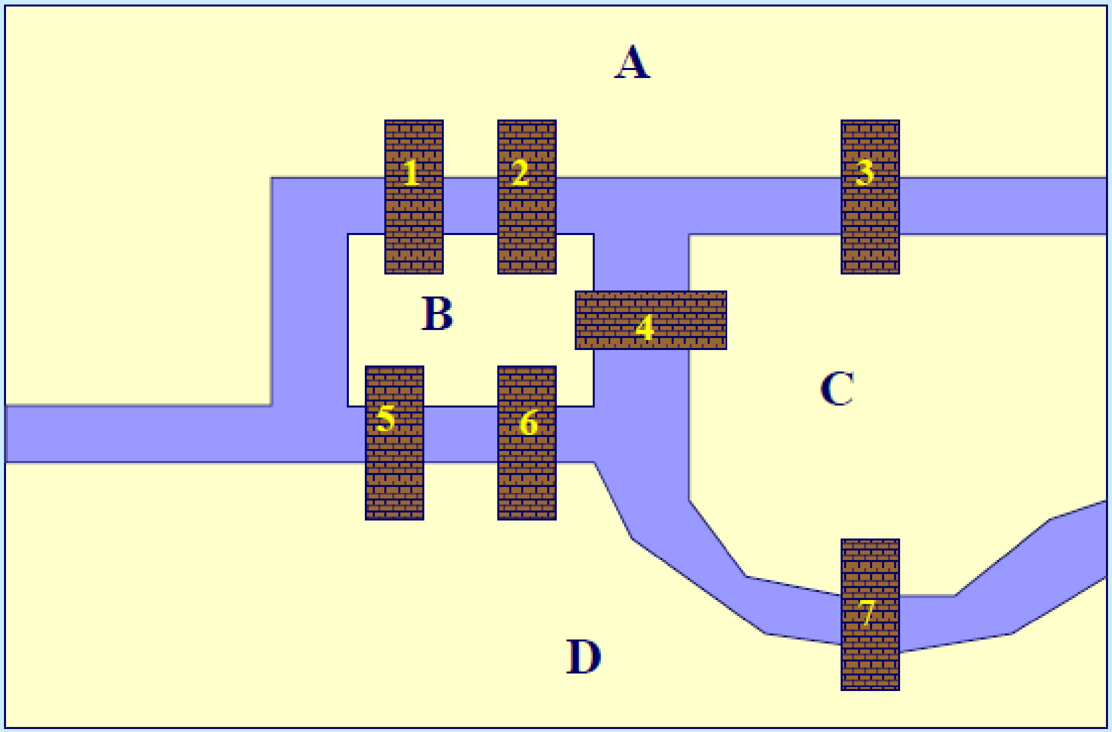
An old problem in history

The Bridges of Koenigsberg: Euler 1736

- "Graph Theory" began in 1736
- Leonard Eüler
  - Visited Koenigsberg
  - People wondered whether it is possible to take a walk, end up where you started from, and cross each bridge in Koenigsberg exactly once
  - Generally it was believed to be impossible

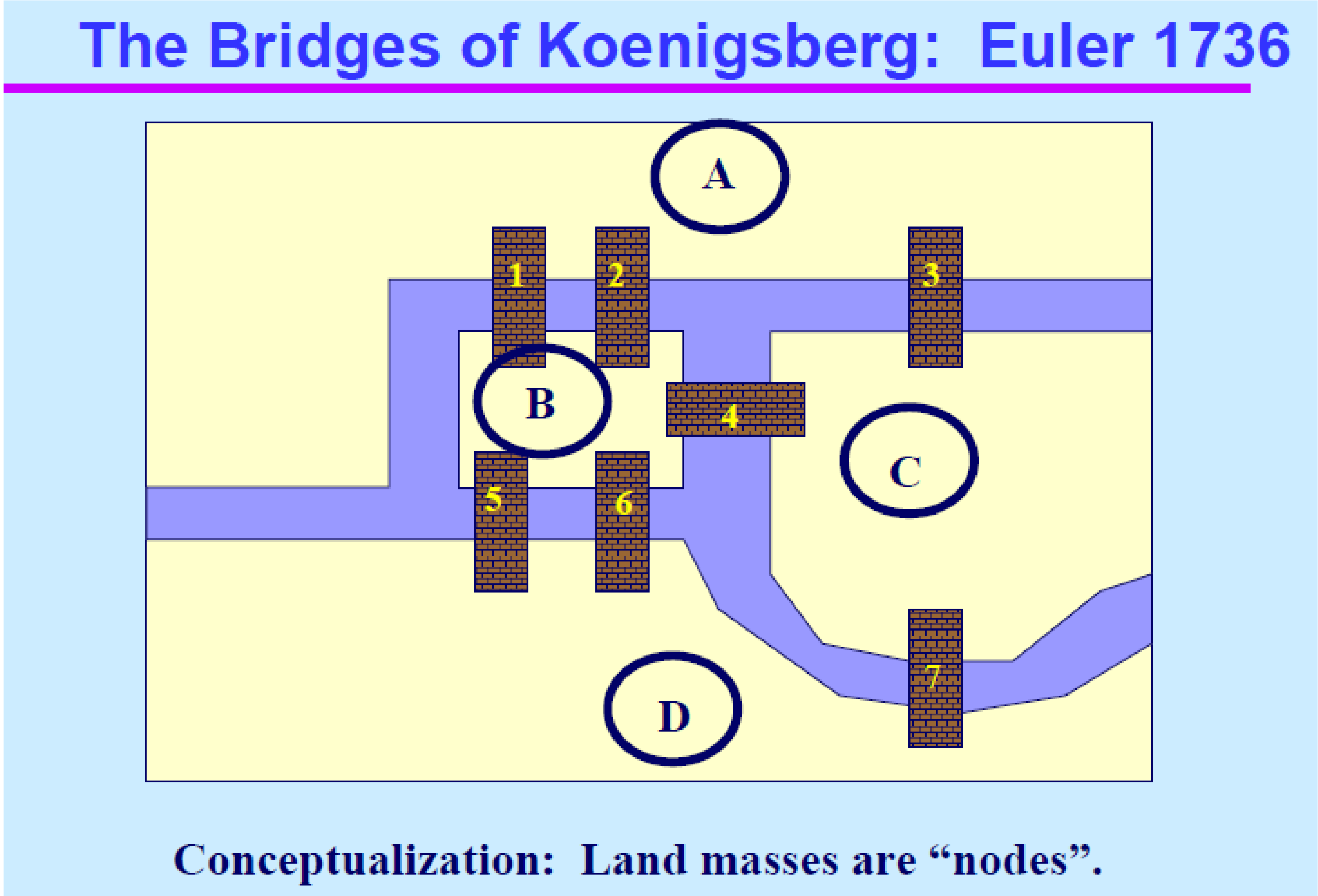
Step 1

The Bridges of Koenigsberg: Euler 1736

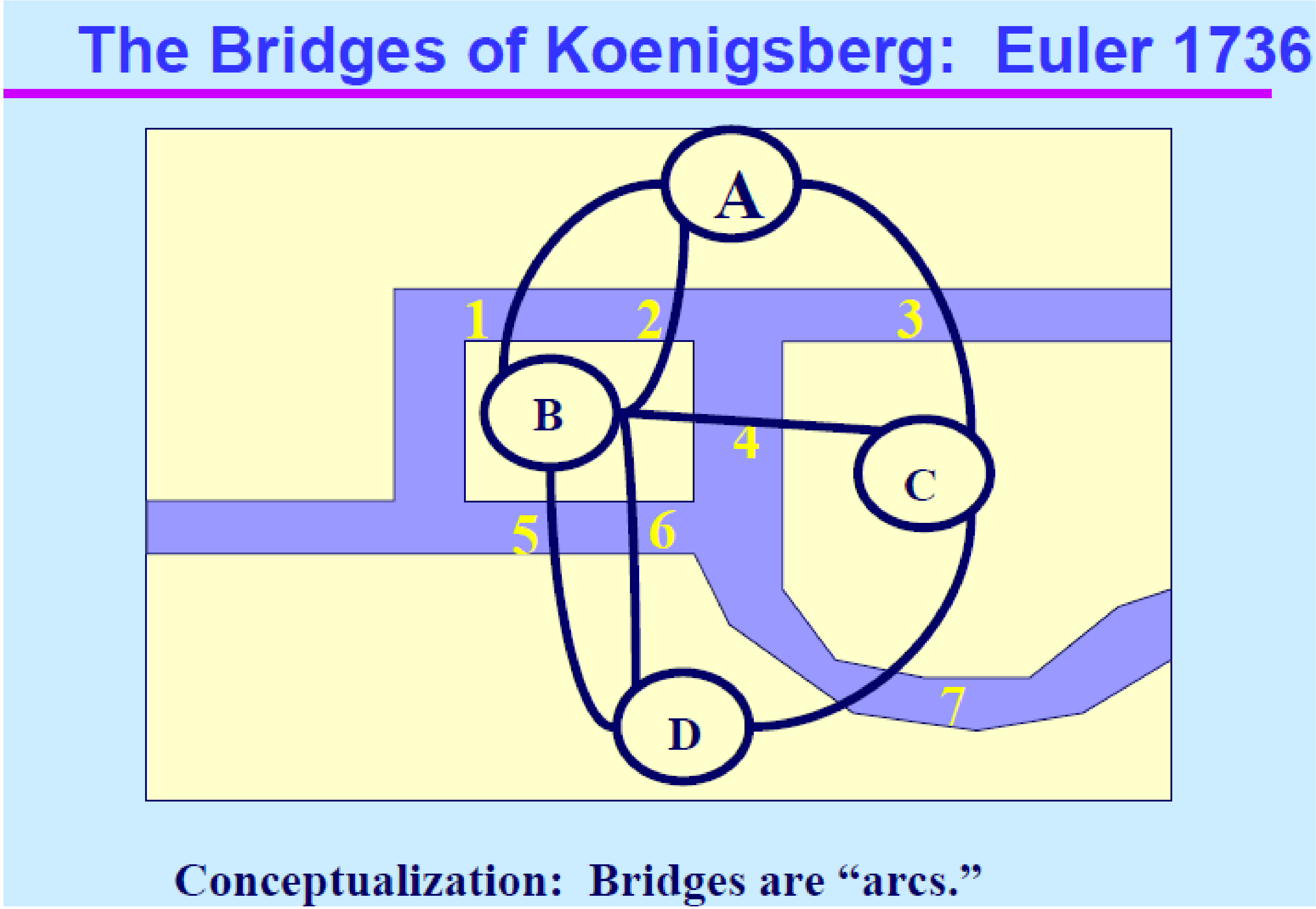


Is it possible to start in A, cross over each bridge exactly once, and end up back in A?

Step 2

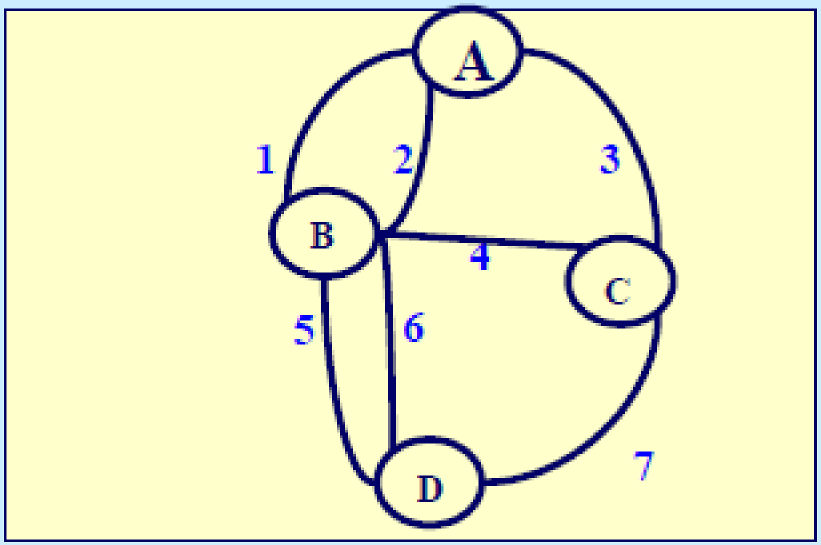

The Bridges of Koenigsberg: Euler 1736
A
B
C
D
1
2
3
4
5
6
7
Conceptualization: Land masses are "nodes".

Step 3


The Bridges of Koenigsberg: Euler 1736
A
B
C
D
1
2
3
4
5
6
7
Conceptualization: Bridges are "arcs."

Step 4

The Bridges of Koenigsberg: Euler 1736



Is there a "walk" starting at A and ending at A and passing through each arc exactly once?
Such a walk is called an *eulerian cycle*.

Step 5

- What's your answer? Why?
- Can you make it work? How?
- What's the insight?
- Can you generalize it?

Theorem

An undirected graph has an Eulerian cycle *if and only if*

(i) every node degree is even,

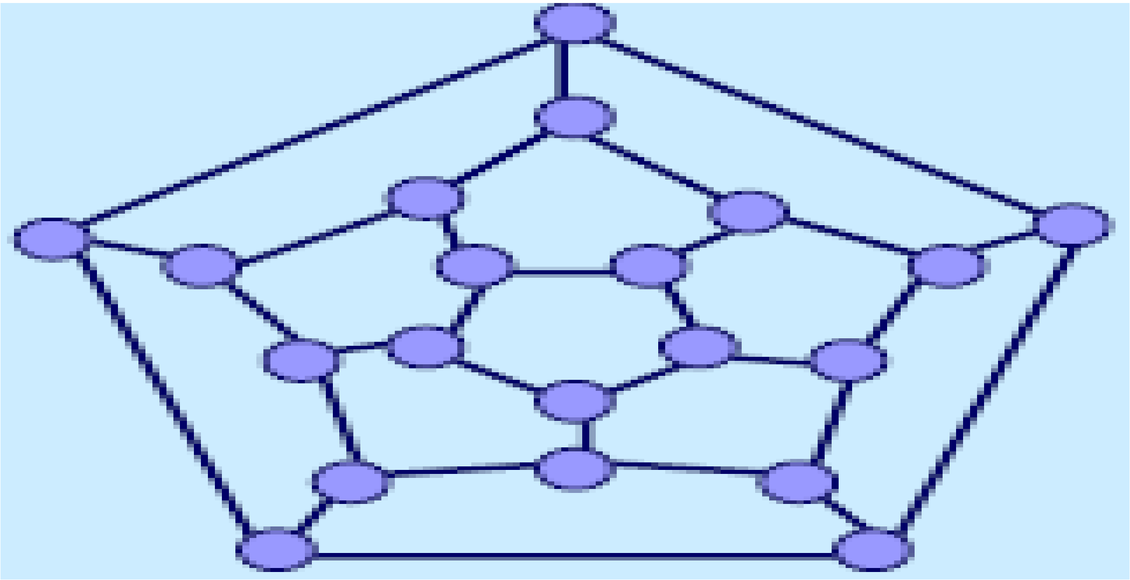
(ii) the graph is connected (i.e., there is a path from each node to each other node).

Another old problem in history

- **In 1857, the Irish mathematician, Sir William Rowan Hamilton invented a puzzle that he hoped would be very popular.**
- **A *Hamiltonian cycle*** is a cycle that passes through each node of the graph exactly once.
- **This is often called a *traveling salesman tour*.**

Hamilton’s Around the World Game

The objective was to construct a Hamiltonian cycle.
The game was not a commercial success, but the mathematics of traveling salesman is very popular today



Polynomial Time Algorithm

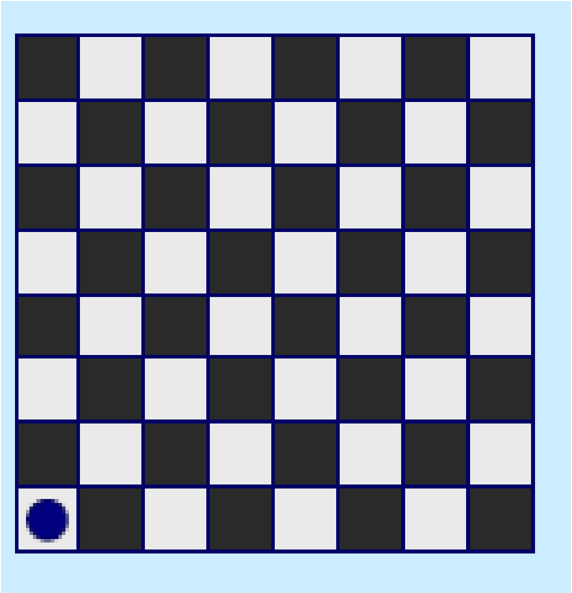
An algorithm runs in polynomial time if the number of steps taken by an algorithm on any instance *I* is bounded by a polynomial in the size of *I*.

One can find an Eulerian cycle in O(m) steps.

An algorithm runs in exponential time if it does not run in polynomial time.

There is no polynomial time algorithm known for finding a Hamiltonian cycle.

Exercise – Knight's Tour Problem



- Can a knight visit all squares of a chessboard exactly once, starting at some square, and by making 63 legitimate moves?
- Hint: The knight's tour problem is a special case of the Hamiltonian tour problem.