

Graph Theory Applied in Ski Resort Geospatial

From Mathematical Concepts to Real-World Applications

Author: mircea_stefan.simoiu@upb.ro

Reviewer: stephane.ploix@grenoble-inp.fr

Table of Contents

1. Introduction
2. Undirected Graphs
3. Directed Graphs
4. Assigning Attributes to Edges and Nodes
5. Connectivity
6. Shortest Paths

Introduction

Context

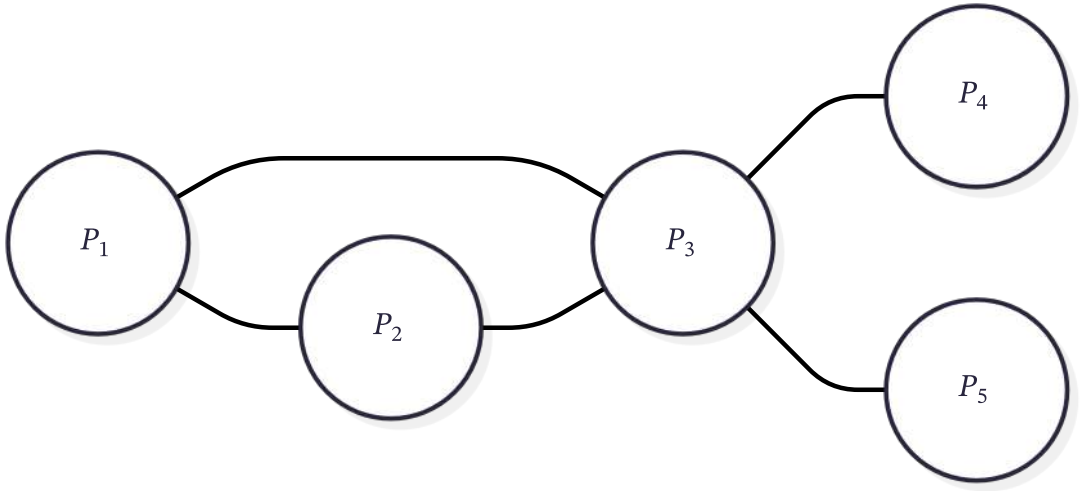
- Many complex systems involve networks of interconnected elements
    - Examples: traffic networks, social networks, etc.
- To design or analyze such systems, we need a proper tool for modeling
- Graph theory provides a powerful mathematical framework for modeling and analyzing networks
- It can also be used in ski-resort energy management

Undirected Graphs

Basic concepts

Definition: A graph $G = (V, E)$ consists of:

- Node (Vertex) Set V: Finite set of nodes
- Edge Set E: Set of unordered pairs $\{u, v\}$ where $u, v \in V$

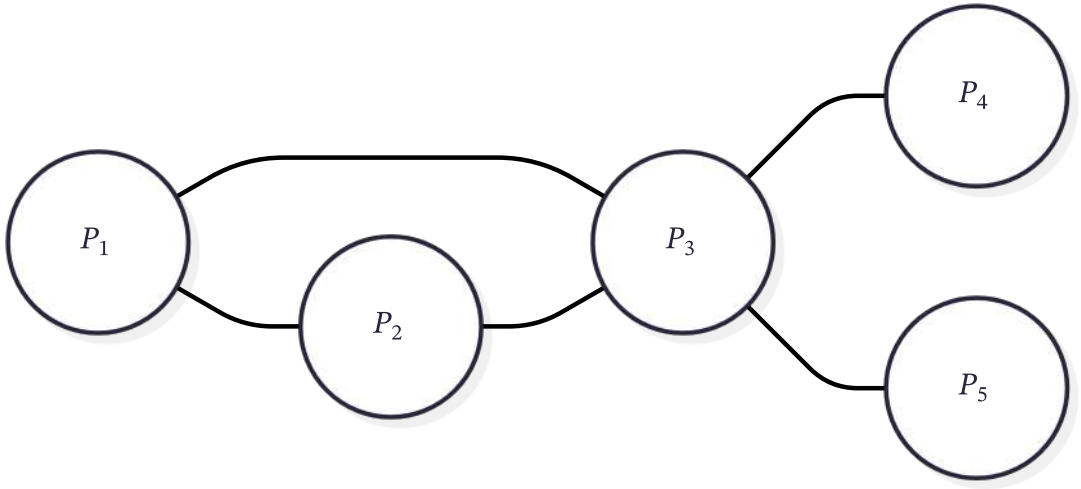


In this example, the graph G consits of:

- Node Set V: $\{P_1, P_2, P_3, P_4, P_5\}$
- Edge Set E: $\{\{P_1, P_2\}, \{P_1, P_3\}, \{P_2, P_3\}, \{P_3, P_4\}, \{P_3, P_5\}\}$

Properties of the graph

- Order of the graph: $|V| = n$ represents the number of nodes
- Size of the graph: $|E| = m$ represents the number of edges
- Degree of a node: $deg(v)$ represents the number of edges connected to a node v



In this example, the graph has 5 nodes and 5 edges

Types of graphs

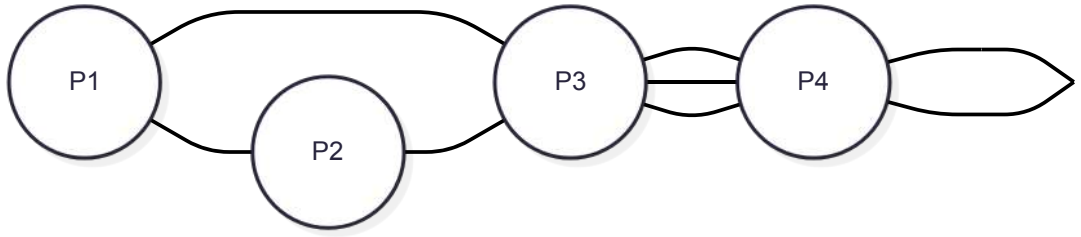
Depending on the connection types

Simple Graph: A graph $G = (V, E)$ where:

- No Self-loops: $\{v, v\} \notin E$ for any $v \in V$
- No multiple edges: $\{u, v\}$ appears at most once in E

Multigraph: A graph $G = (V, E)$ where:

- Multiple edges: $\{u, v\}$ can appear multiple times in E



In this example, we have:

- *3 edges between P_3 and P_4: $\{P_3, P_4\}$ appears 3 times in E*
- *a self-loop: $\{P_4, P_4\}$ appears once in E*

Types of graphs

Depending on the number of edges

Complete Graph: A graph $G = (V, E)$ where:

- Every pair of nodes is connected: $\{u, v\} \in E$ for any $u, v \in V$

Bipartite Graph: A graph $G = (V, E)$ where:

- Nodes can be partitioned into two independent sets: $V = V_1 \cup V_2$ and $E \subseteq V_1 \times V_2$

Types of graphs

Depending on the types of edges

Undirected Graph: A graph $G = (V, E)$ where:

- Edges are undirected: $E \subseteq \{u, v\}$ for any $u, v \in V$

Directed Graph: A graph $G = (V, E)$ where:

- Edges are directed: $E \subseteq V \times V$

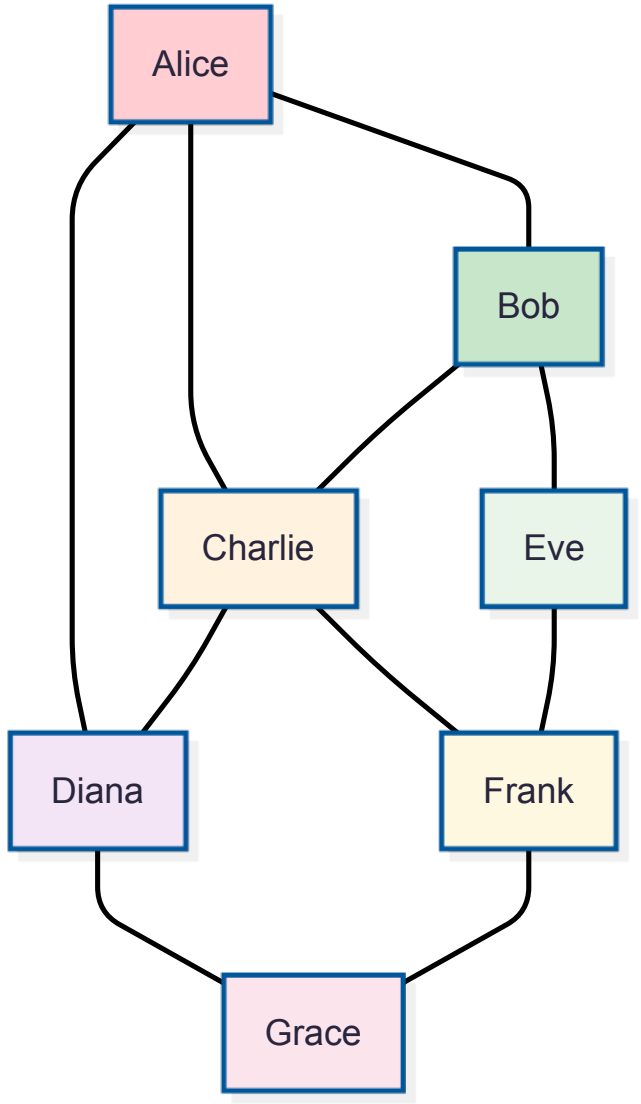
Applications

- Social Networks: Facebook, Twitter, LinkedIn
- Power Grid: Electricity distribution networks
- Biology: Protein-protein interactions, gene regulatory networks

Exercise:

Determine the type of the following graph, depending on:

- *the connection types*
- *the number of edges*
- *the types of edges*

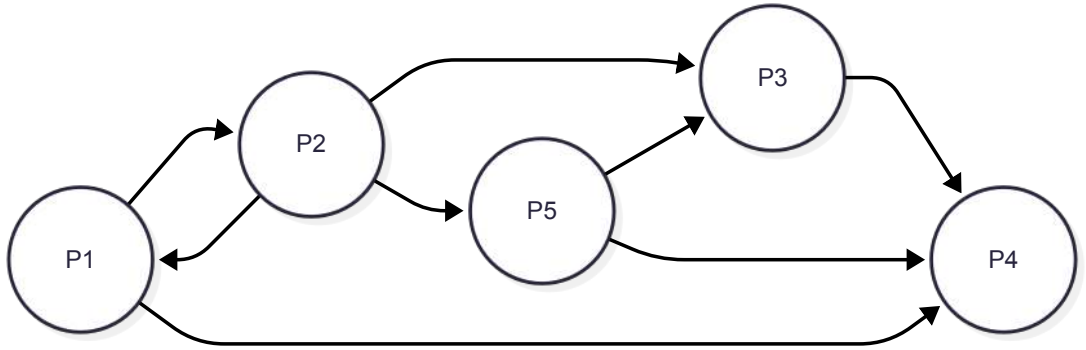


Directed graphs

Basic concepts

Directed Graph (Digraph): A graph $G = (V, E)$ where:

- Ordered Pairs: $E \subseteq V \times V$ (ordered edges)
- Directed edges: (u, v) means u points to v

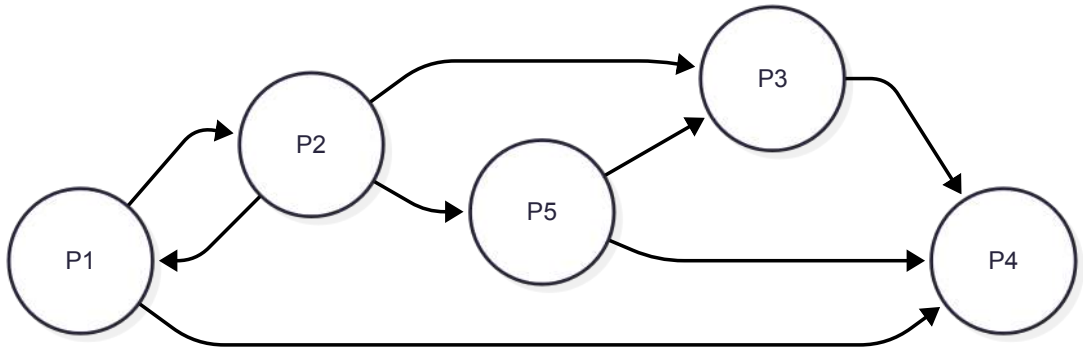


In this example, the graph G consits of:

- Vertex Set V: $\{P_1, P_2, P_3, P_4, P_5\}$
- Edge Set E: $\{(P_1, P_2), (P_2, P_3), \dots\}$

Properties

- Asymmetry: $(u, v) \in E$ does not imply $(v, u) \in E$
- In-degree: $deg^-(v) = |\{u : (u, v) \in E\}|$
- Out-degree: $deg^+(v) = |\{u : (v, u) \in E\}|$
- Degrees represent measures for the influence of a node on the graph



$$deg^-(P_1) = 1$$

$$deg^+(P_1) = 2$$

Properties

Degree centrality refers to the number of links held by a node.

$$C_d(v) = \frac{deg(v)}{n-1}$$

where n is the number of nodes in the graph.

Betweenness centrality measures how often a node appears on shortest paths between other nodes.

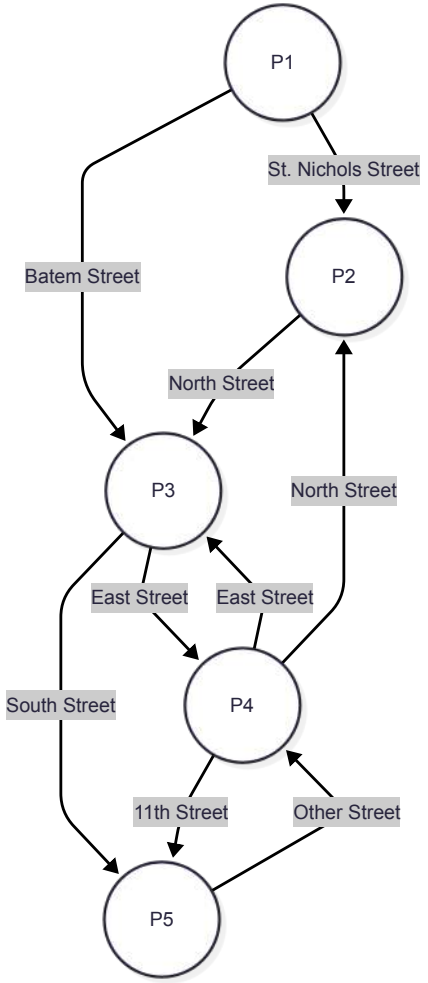
$$C_b(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Where:

- σ_{st} is the number of shortest paths from s to t
- $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through v

Applications

- Ski Lifts: One-way transportation (up only)
- Ski Slopes: One-way descent paths
- Traffic Flow: One-way roads and paths
- Water Flow: Natural drainage patterns



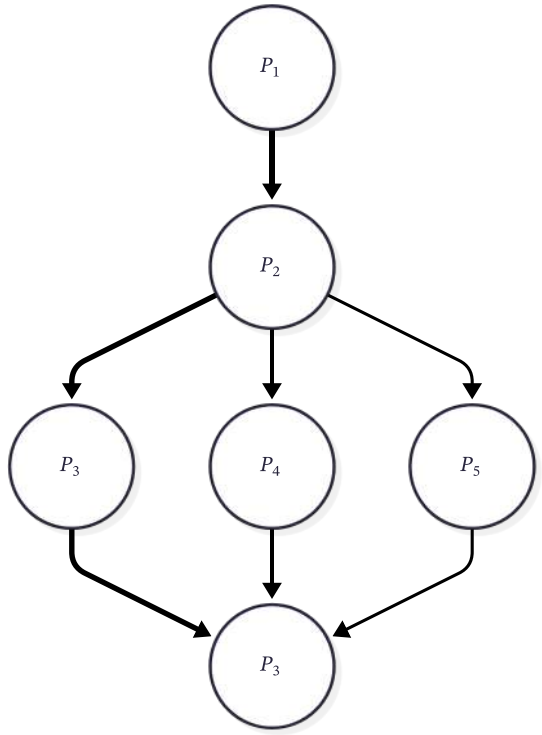
Paths & Cycles

Path: Sequence $(v_0, v_1, \ldots, v_k)$ where $(v_i, v_{i+1}) \in E$ where the nodes are distinct

Walk: a sequence of nodes who do not have to be distinct

Cycle: Path where $v_0 = v_k$ and $k > 0$

Reachability: Vertex v is reachable from u if there exists a directed path from u to v

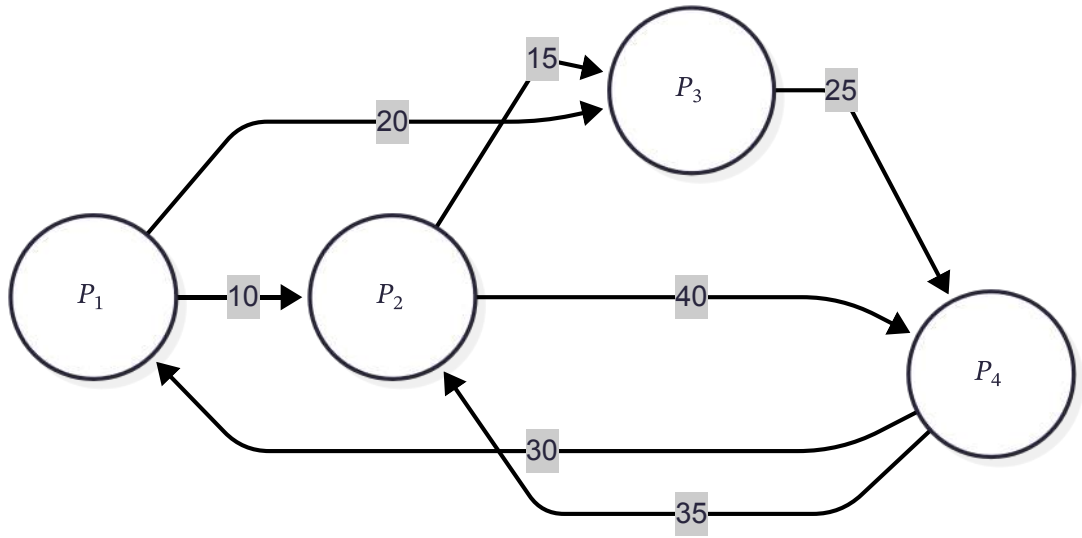


Assigning Attributes to Edges and Nodes

Weighted Graph Theory

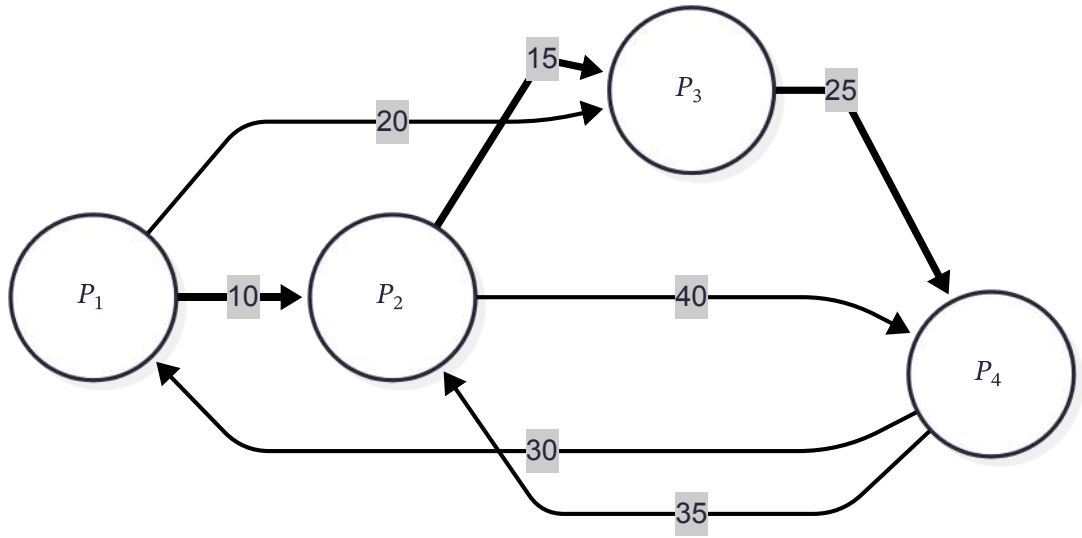
Weighted Graph: A graph $G = (V, E, w)$ where:

- Weight Function: $w : E \to \mathbb{R}$ (or $\mathbb{R}^+$)
- Node Attributes: Function $f : V \to \mathbb{R}^d$ (d-dimensional attributes)
- Edge Attributes: Function $g : E \to \mathbb{R}^k$ (k-dimensional attributes)



Paths in Directed Graphs

- In directed graphs, paths are defined similarly to undirected graphs, but the edges must be traversed in the correct direction.
- The length of a path is the sum of the weights of the edges in the path

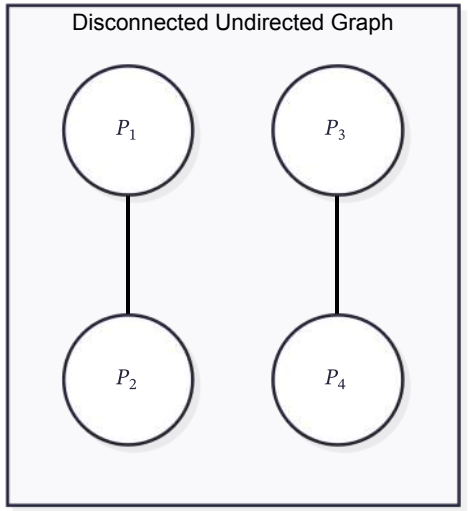


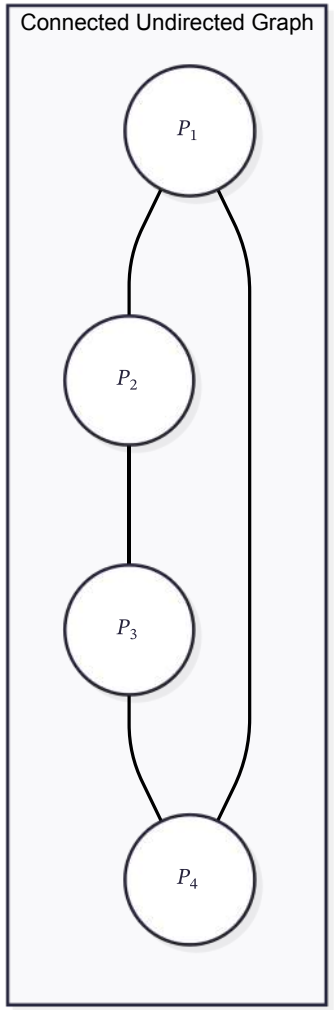
Connectivity

Reachability

Reachability: Node v is reachable from u if there exists a path from u to v

Connected Graph: A graph where every node is reachable from every other node

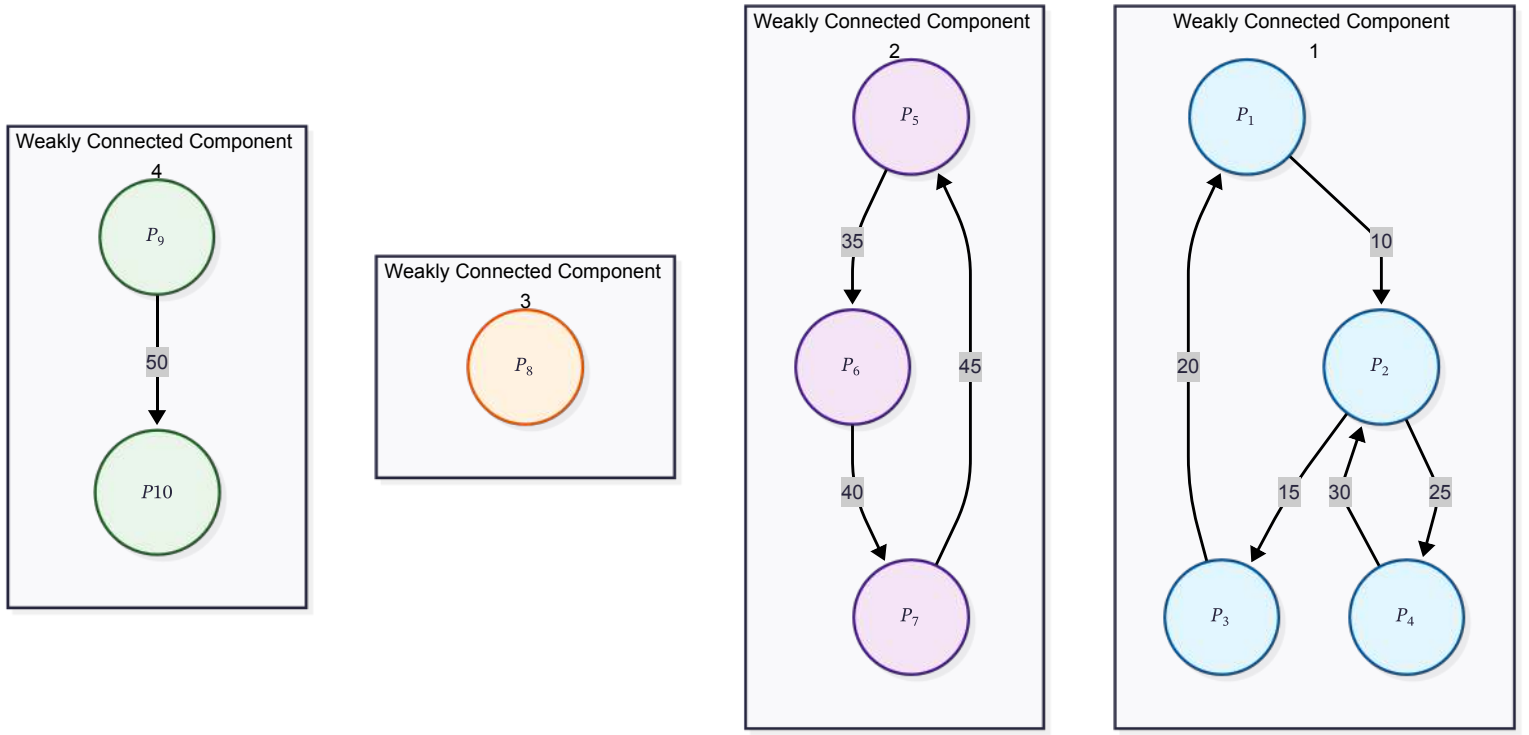




Weakly Connected Components

Weakly Connected: A directed graph where the underlying undirected graph is connected

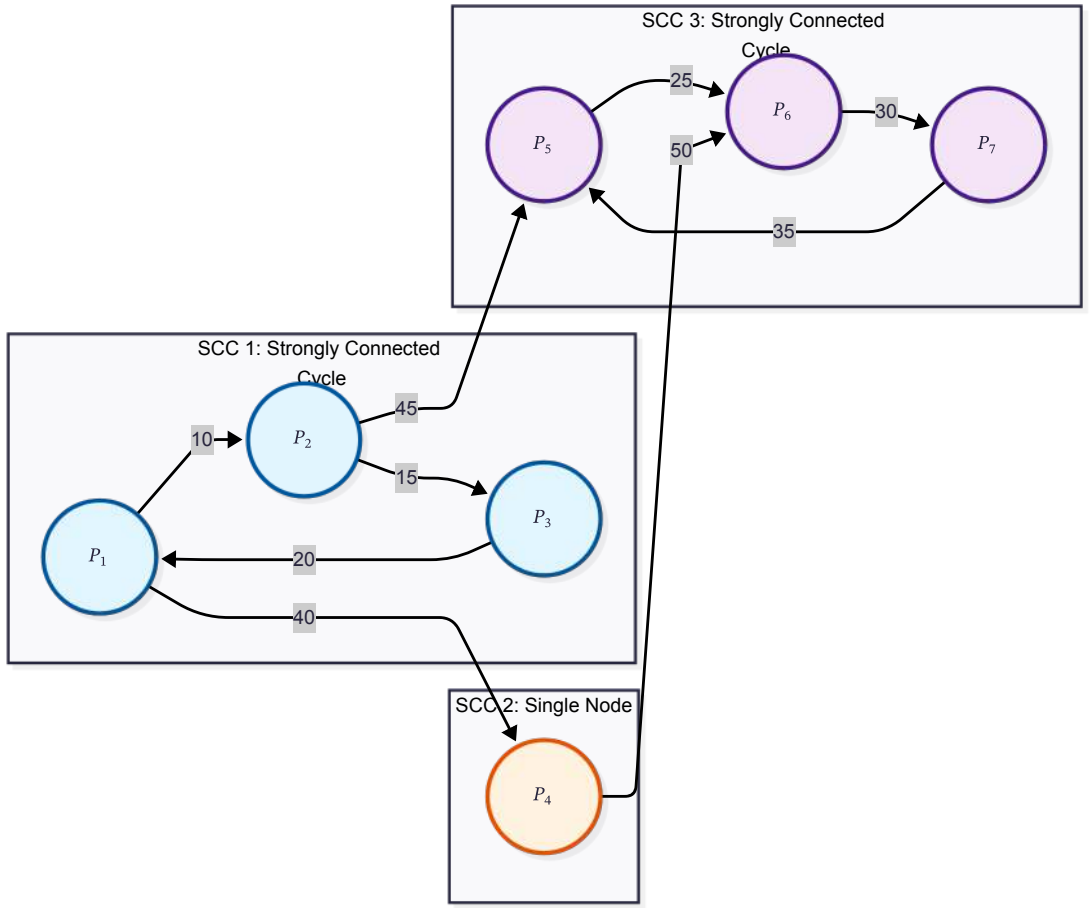
Weakly Connected Component: Maximal weakly connected subgraph (adding a node will make it disconnected)



Strongly Connected Components

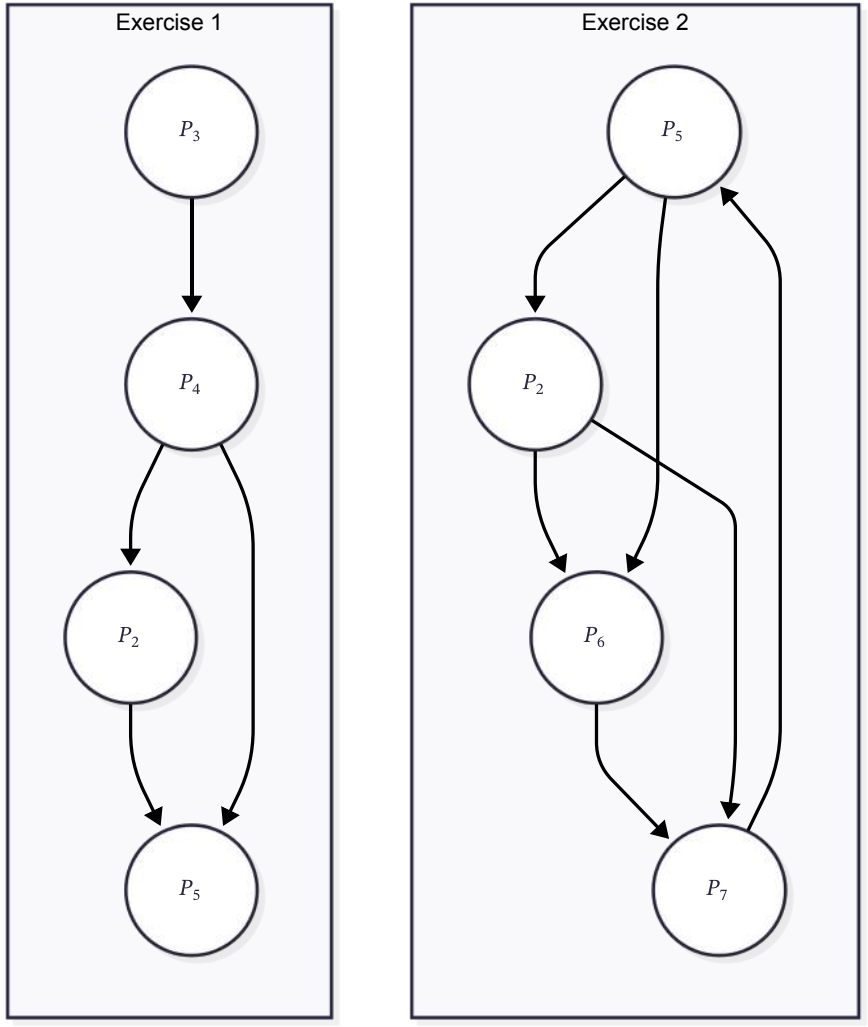
Strongly Connected: A directed graph where for each pair of nodes u and v, there exists a path from u to v and a path from v to u

Strongly Connected Component (SCC): Maximal strongly connected subgraph



Exercise

Determine if the following graphs are strongly connected or weakly connected.



Enhancing connectivity

- An important problem in network design is to enhance the connectivity of a graph
- We can do this by adding edges to the graph, i.e. $G' = (V, E \cup \{(u, v)\})$
- We can also do this by removing edges from the graph, i.e. $G' = (V, E \setminus \{(u, v)\})$
- Connectivity has an important impact in finding the shortest path between two nodes

Connectivity relevance in real-world applications

Network Design Problems:

- Facility Location: Optimal placement of new facilities
- Network Flow: Maximum flow between source and sink
- Capacitated Network: Edges have capacity constraints
- Traffic Engineering: Routing and load balancing

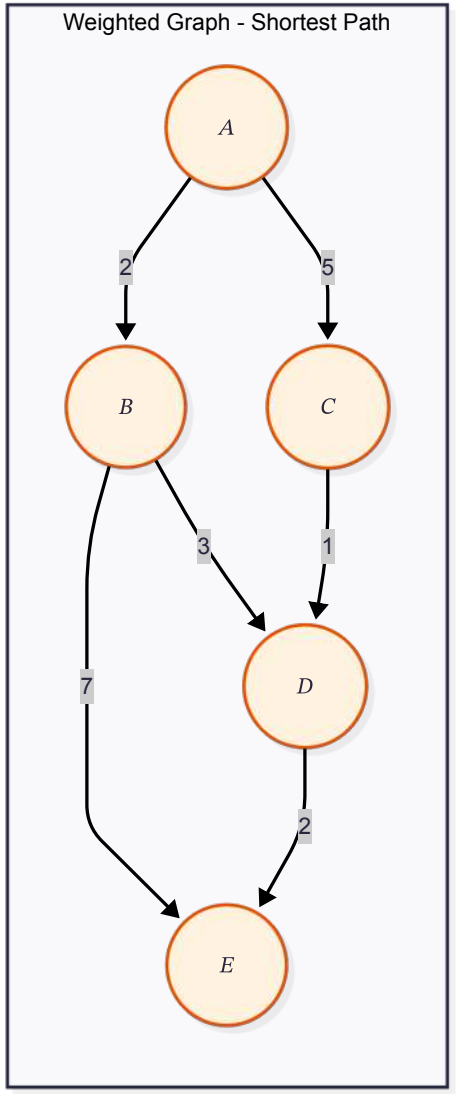
Shortest Paths

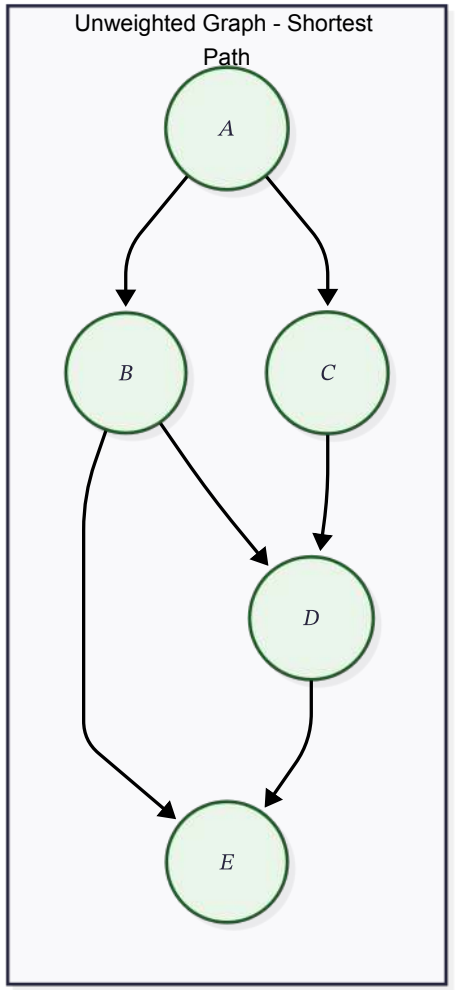
Shortest Path Problem

Shortest Path Problem: Find path with minimum number of edges between two nodes

Mathematical Formulation:

- Unweighted Case:
  $d(u, v) = \min\{k : \exists \text{ path of length } k \text{ from } u \text{ to } v\}$
- Weighted Case:
  $d(u, v) = \min\{\sum_{e \in P} w(e) : P \text{ is path from } u \text{ to } v\}$





Algorithmic Solutions:

- Breadth-First Search (BFS): Explores nodes in layers, finding shortest path by number of edges in unweighted graphs
- Dijkstra's Algorithm: Greedy approach that always selects the closest unvisited node, finding shortest path by total weight in graphs with non-negative weights
- Bellman-Ford: Relaxes all edges repeatedly, capable of handling negative weights and detecting negative cycles
- Floyd-Warshall: Computes shortest paths between all pairs of nodes using dynamic programming

Applications

- Routing: Finding the shortest path between two nodes
- Decision Making: Finding the shortest path in a graph with multiple paths, where the edges have different costs

Key Theoretical Takeaways

1. Graph Structure: $G = (V, E)$ with optional weight functions
2. Properties: Degree centrality, betweenness centrality
3. Connectivity: Weak vs. strong connectivity in directed graphs
4. Path Problems: Shortest path definition and importance in real-world applications

Application in ski resort management

We can model the infrastructure of a ski resort with directed graphs.

- Each edge represents a skilift/skiing path
- Each node represents a permanent geographical position
- The ski resort should be strongly connected.
- Each edge has attributes (i.e. power consumption, slope)

Geospatial modeling of the ski resort helps in solving the energy management problem.

Thank You!

Graph Theory: From Mathematical Concepts to Real-World Applications

Contact: mircea_stefan.simoiu@upb.ro

Repository: BATEM Project