Network Flow Algorithms:
Applications to Metabolic Flux Analysis

Chem 274B Micro Presentation

Authors: Christian Fernandez, Kirk Ehmsen, David Houshangi, Yash Maheshwaran

12/5/25

Background: Digraphs with Edge Capacities

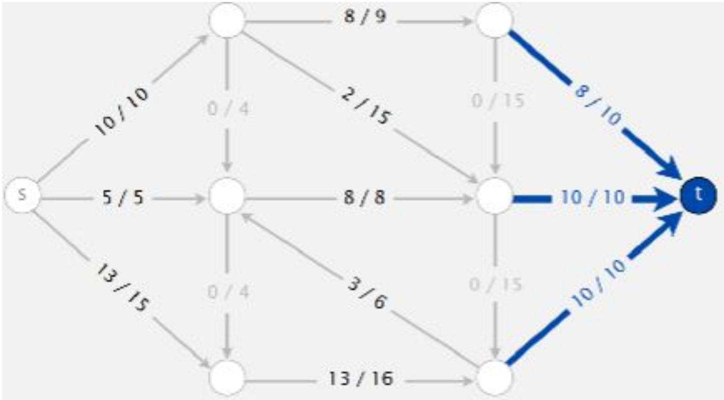
Features

- Source Vertex (s)
- Target vertex (t)
- Edges between nodes with capacity limits on flow

Importance to Biology?

Metabolic pathways are naturally from edge-weighted digraphs, allowing us to analyze:

- Bottlenecks
- Essential reactions (steps)
- Optimal flow in biological pathways.



Applicable Algorithms

Ford-Fulkerson Algorithm: Finds Max Flow

- Starts with 0 flow
- Identify Augmenting Paths:
  - Path from source to target where additional flow can be pushed
- While there exists an Augmenting Path:
  - Find an Augmenting Path: BFS
    - There are other options
  - Compute Bottleneck Capacity
  - Increase Flow on that path by Bottleneck Capacity
- Flow = Max Flow if no augmenting paths remain

Residual graph: Find Min-Cut Edges from Max Flow

- Set Flow = Max Flow
- Build residual graph where:
  - Residual Capacity = Capacity - Flow
- Traverse graph along edges where Residual Capacity > 0, marking all possible nodes reached
- In original graph:
  - Edges from reachable → unreachable nodes are min-cut edges
- Represent bottlenecks

Complexity

- Finding Augmenting Paths
  - Multiple ways to identify augmenting paths which lead to various runtime complexities
- For BFS implementation:
  - **Time:**
    - FF must find all augmented paths: **O(V * E)**
    - Each BFS takes **O(V + E)**
    - In total: O(V*E) * O(V + E) = **O(V^2E + VE^2)**
    - If V >> E: **O(V^2 * E)**
    - If V << E: **O(V * E^2)**
  - **Space:**
    - Implement BFS via a queue: **O(V)**
    - However, we also store edge capacities: **O(E)**
    - In total: **O(V + E)**

Augmenting Path	Number of Paths
Random Path	<= E*U
DFS Path	<= E * U
Shortest Path (BFS)	<= ½ E * V
Fattest Path	< E ln(E * U)

Digraph with V vertices, E edges, and integer capacities between 1 and U

Advantages / Limitations of FF + BFS

Use of BFS for finding Augmenting paths:

Advantages

- BFS guarantees polynomial runtime and convergence
  - As opposed to DFS

Limitations

- Slower, as compared to DFS
- Does not take into account costs associated with edges, if applicable

Ford-Fulkerson Algorithm in General:

Advantages

- Relatively simple logic
- Flexibility in choice of how to find augmenting paths
- Always computes a max flow if FF terminates

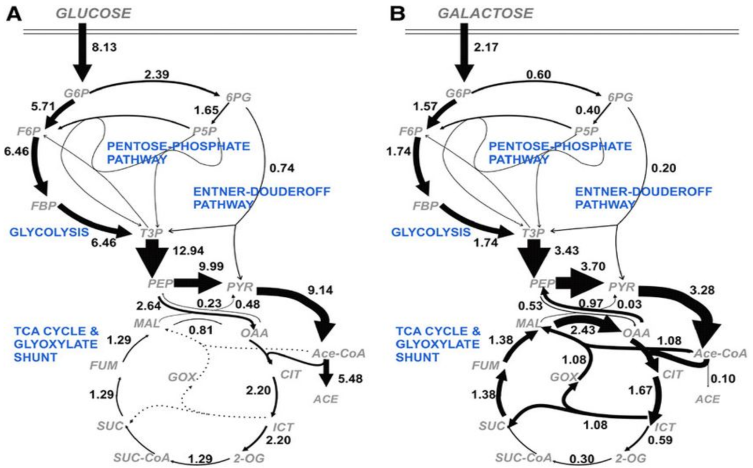
Limitations

- Inefficient runtime for large graphs
- May not terminate if edge capacities are not integers

Biological Applications

- **Molecular docking**
  a. Identifying what the most ideal paths (best paths) are for ligands to interact with active sites in proteins
- **Chemical Reaction Networks**
  a. Model reaction as nodes and flows as rate of reactions of molecules between reactions
     i. Find bottlenecks or critical reactions that limit pathway efficiency
- **Network Pharmacology**
  a. Be able to better evaluate disease pathways by looking at where cuts are present that disrupt disease pathways
  b. Be able to identify drug combinations and evaluate their effects

- **Specific Example in Literature: E-coli**
  a. **Goal ⇒** Identifying which transcription factor controls the central metabolic pathway flux in E. Coli on glucose vs galactose and how this controls changes with conditions.
  b. **Algorithm ⇒** Used 13C-based metabolic flux analysis (13C - MFA) with the Fiat Flux software (plus whole isotopologue modeling) to estimate intracellular fluxes from ^13C labeling data.



Reference:
https://www.researchgate.net/publication/50936992_Large-scale_13C-flux_analysis_reveals_distinct_transcriptional_control_of_respiratory_and_fermentative_metabolism_in_Escherichia_coli

Example: Glycolysis (1/2)

What is Glycolysis?

Glycolysis is a pathway in which glucose is broken down into pyruvate through a sequence of enzyme-catalyzed steps.

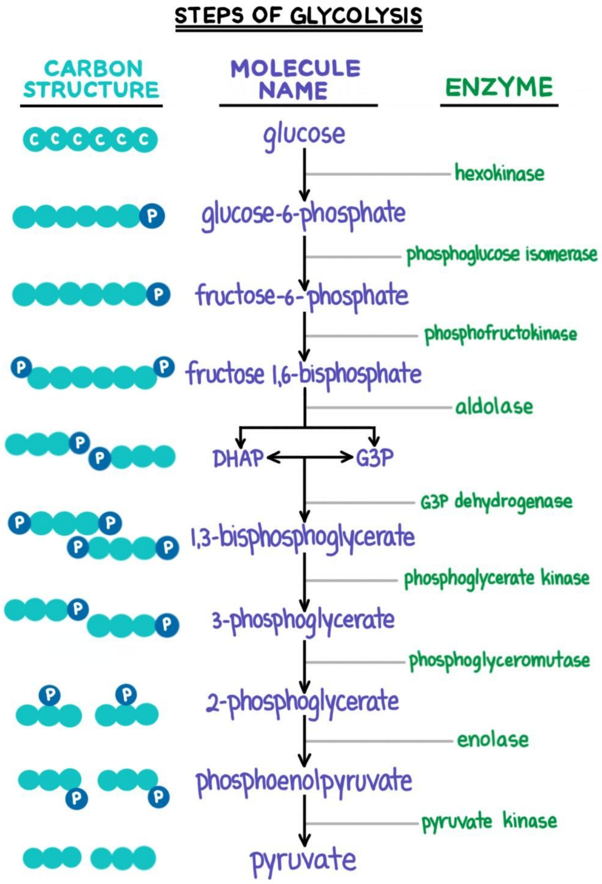
Goals:

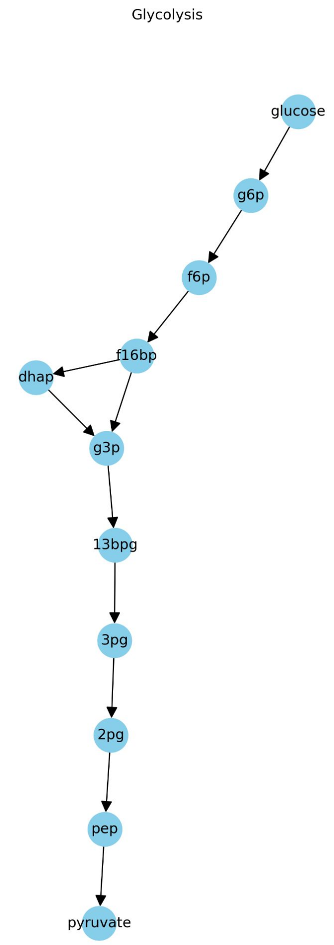
Identify max flux of our metabolite:

- **0.68 → Highest possible throughput from glucose to pyruvate given the reaction capacities**

Identify rate limiting enzyme in metabolic pathway:

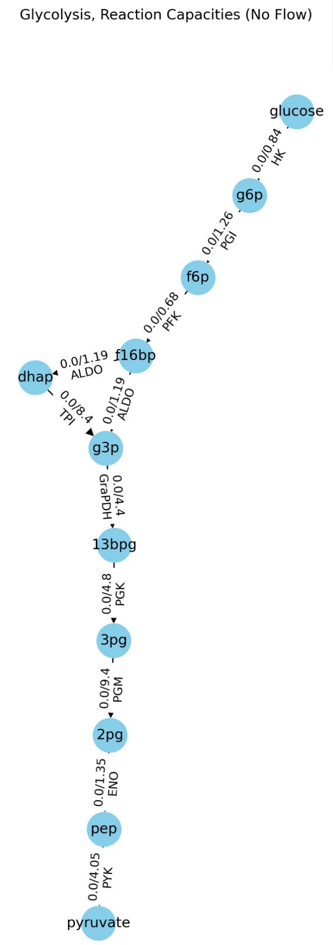
- **Phosphofructokinase (PFK)**





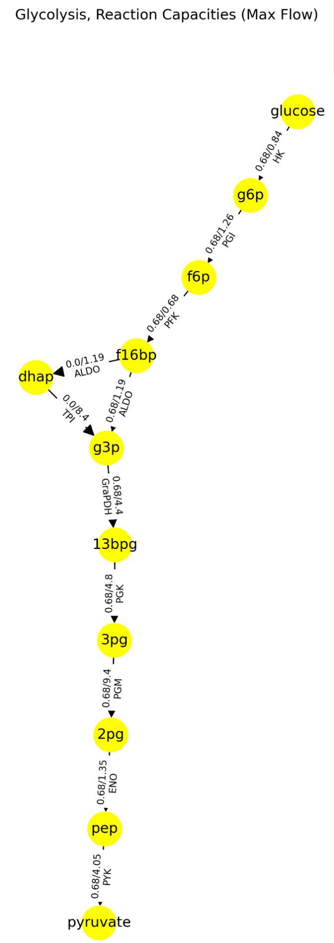
Glycolysis

Glycolysis Graph



Glycolysis, Reaction Capacities (No Flow)

Initialization of FF

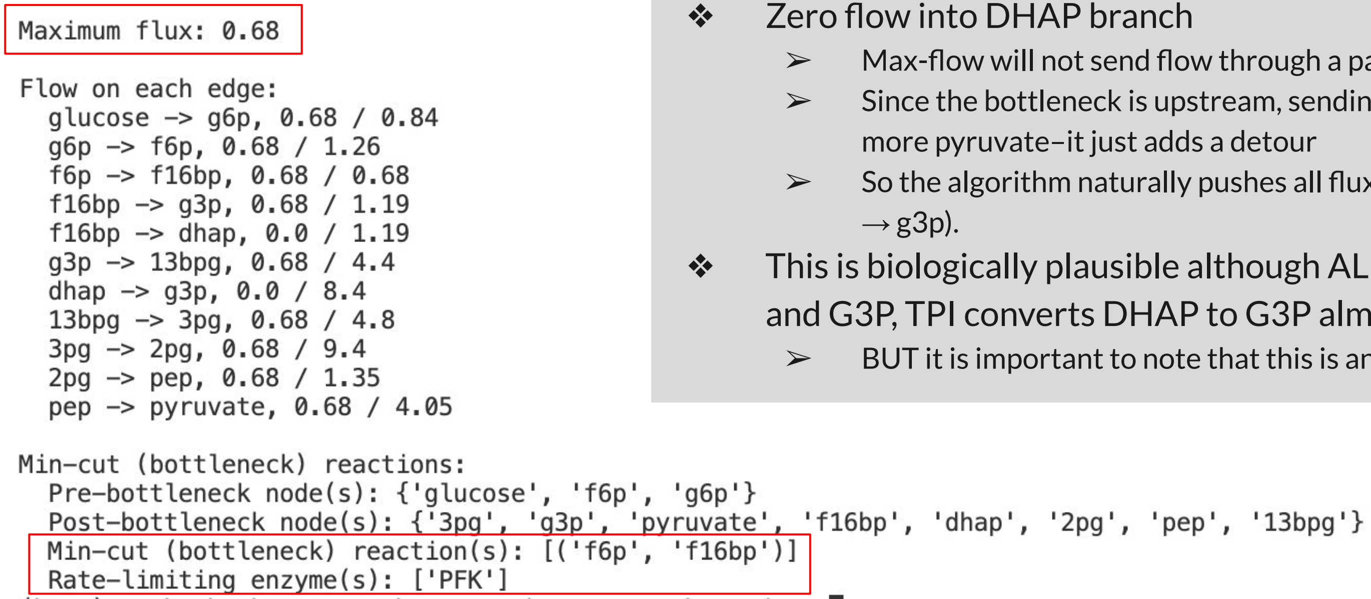


Glycolysis, Reaction Capacities (Max Flow)

Termination of FF

Example: Glycolysis (2/2)

Code Output:



```
Maximum flux: 0.68

Flow on each edge:
  glucose -> g6p, 0.68 / 0.84
  g6p -> f6p, 0.68 / 1.26
  f6p -> f16bp, 0.68 / 0.68
  f16bp -> g3p, 0.68 / 1.19
  f16bp -> dhap, 0.0 / 1.19
  g3p -> 13bpg, 0.68 / 4.4
  dhap -> g3p, 0.0 / 8.4
  13bpg -> 3pg, 0.68 / 4.8
  3pg -> 2pg, 0.68 / 9.4
  2pg -> pep, 0.68 / 1.35
  pep -> pyruvate, 0.68 / 4.05

Min-cut (bottleneck) reactions:
  Pre-bottleneck node(s): {'glucose', 'f6p', 'g6p'}
  Post-bottleneck node(s): {'3pg', 'g3p', 'pyruvate', 'f16bp', 'dhap', '2pg', 'pep', '13bpg'}
  Min-cut (bottleneck) reaction(s): [('f6p', 'f16bp')]
  Rate-limiting enzyme(s): ['PFK']
```

- ❖ PFK is the universally recognized rate-limiting enzyme in glycolysis and our algorithm properly shows this.
- ❖ The bottleneck reaction from f6p → f16bp is catalyzed by this rate limiting enzyme and our algorithm properly accounts for this as well.
- ❖ Zero flow into DHAP branch
  - ➢ Max-flow will not send flow through a pathway unless it increases total output
  - ➢ Since the bottleneck is upstream, sending any flow into dhap doesn't create more pyruvate–it just adds a detour
  - ➢ So the algorithm naturally pushes all flux through the most direct path (f16bp → g3p).
- ❖ This is biologically plausible although ALDO does produce both DHAP and G3P, TPI converts DHAP to G3P almost instantaneously
  - ➢ BUT it is important to note that this is an idealized version of real biology

Thank You!