CYCLE CANCELLING ALGORITHM: MATH 7825

Paul Guidas

Notation and Assumptions

- The Cycle Cancelling Algorithm is used for solving min-cost flow problems

Let:

$\mathbf{G} = (\mathbf{N}, \mathbf{A})$ be a directed network with N nodes and A arcs.

$\mathbf{c_{ij}} \geq \mathbf{0}$ represent the cost of arc $(i, j) \in A$

$\mathbf{u_{ij}}$ represent the capacity of arc $(i, j) \in A$

$\mathbf{s(i)}$ represent the supply or demand of node i depending on whether $s(i) > 0$ or $s(i) < 0$, $\forall i \in N$

$\mathbf{C}$ denote the largest magnitude of any arc cost.

$\mathbf{U}$ denote the largest magnitude of any supply/demand or finite arc capacity.

So, the minimum cost problem is:

$$\text{Minimize } z(x) = \sum_{(i,j)\in A} c_{ij} x_{ij}$$

subject to:

$$\sum_{\{f:(i,j)\in A\}} x_{ij} - \sum_{\{f:(j,i)\in A\}} x_{ji} = s(i) \quad \forall i \in N,$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A.$$

We will also assume:

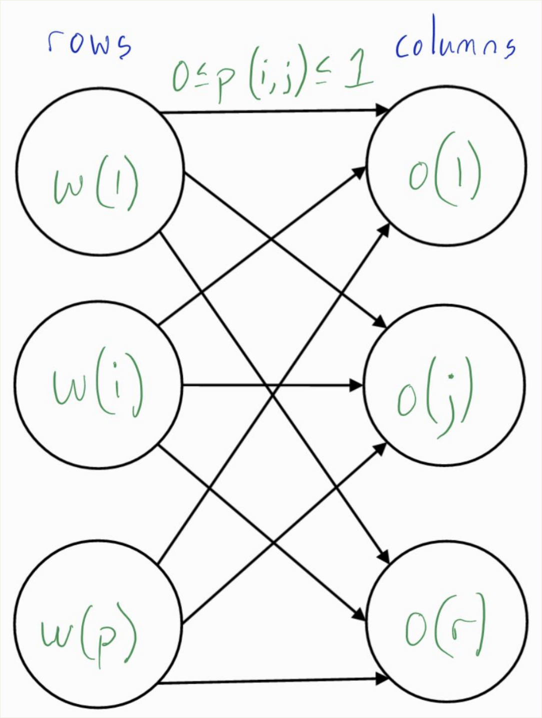
The network is directed. (This is necessary for us to have feasible flow.)

The data is all integral. (Data being defined as arc capacities/costs and node supplies/demands)

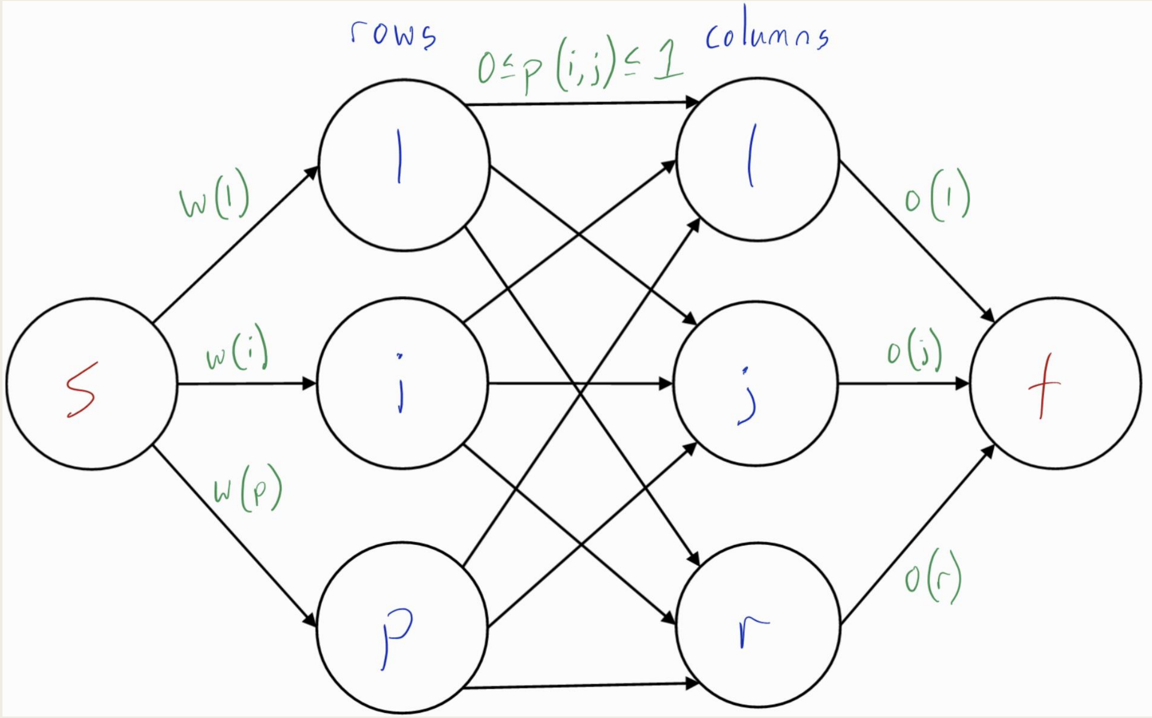
Feasibility and Algorithm setup

- Verify that the sum of the supply/demand values = 0
- Then, convert the problem to a max flow problem

$$\sum_{\{i \in N\}} s(i) = 0$$

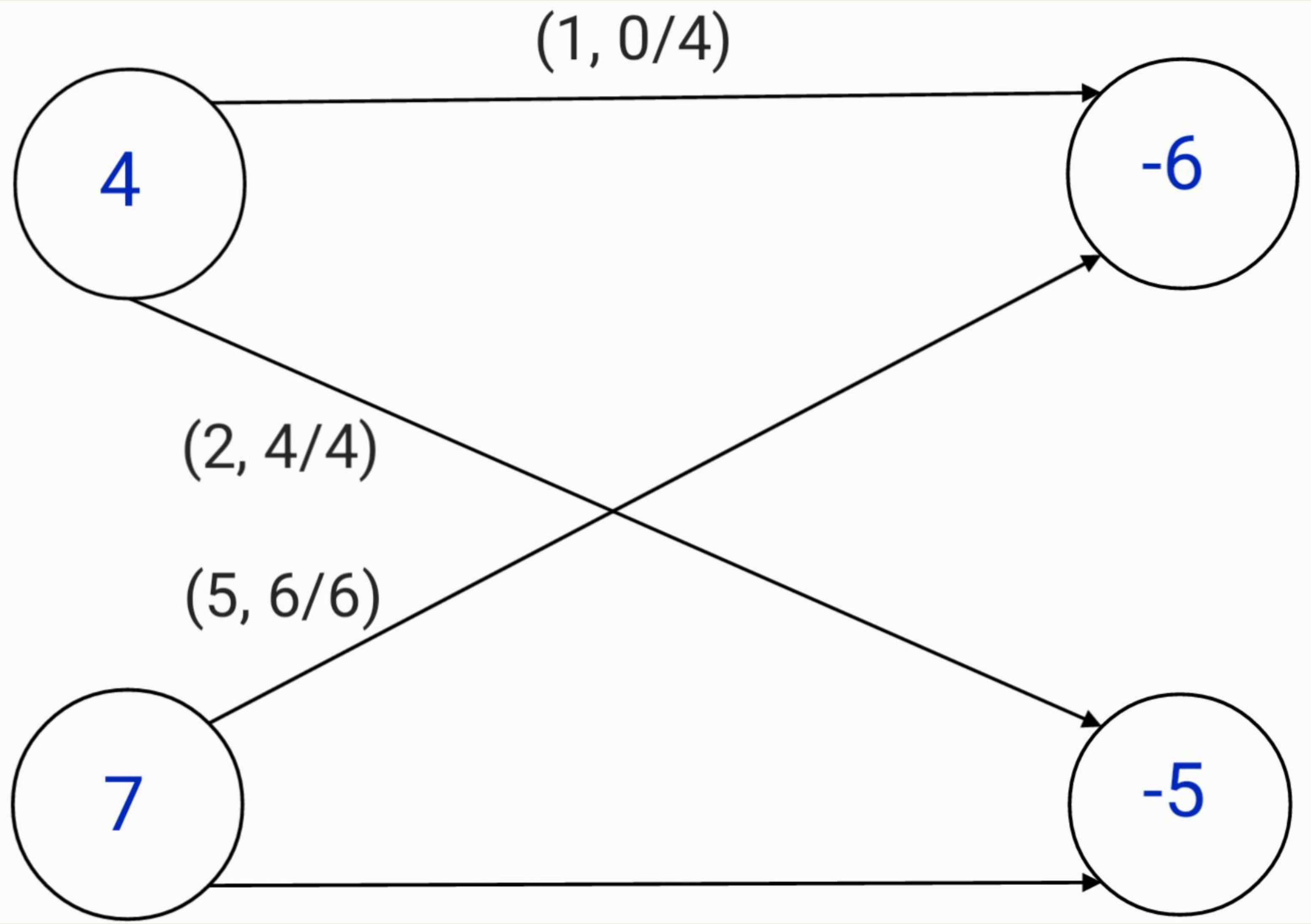


→



Initial Feasible Solution

- Verify a feasible flow exists using a max flow algorithm and use as a starting point for the main algorithm
- This example shows a **feasible** but not **optimal** solution



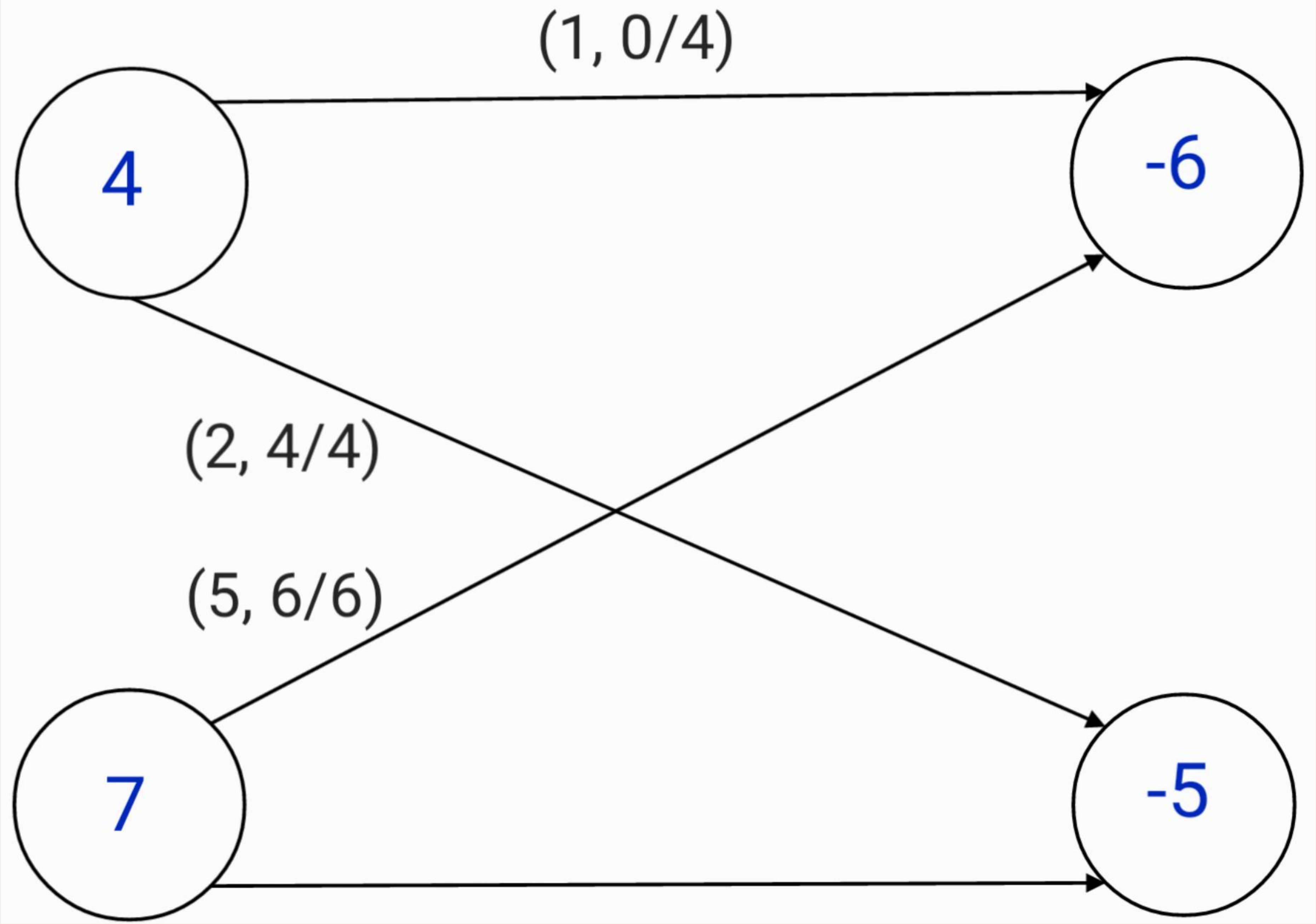
- Arcs: (Cost, Flow/Capacity)
- Nodes: Supply/Demand

Psuedocode

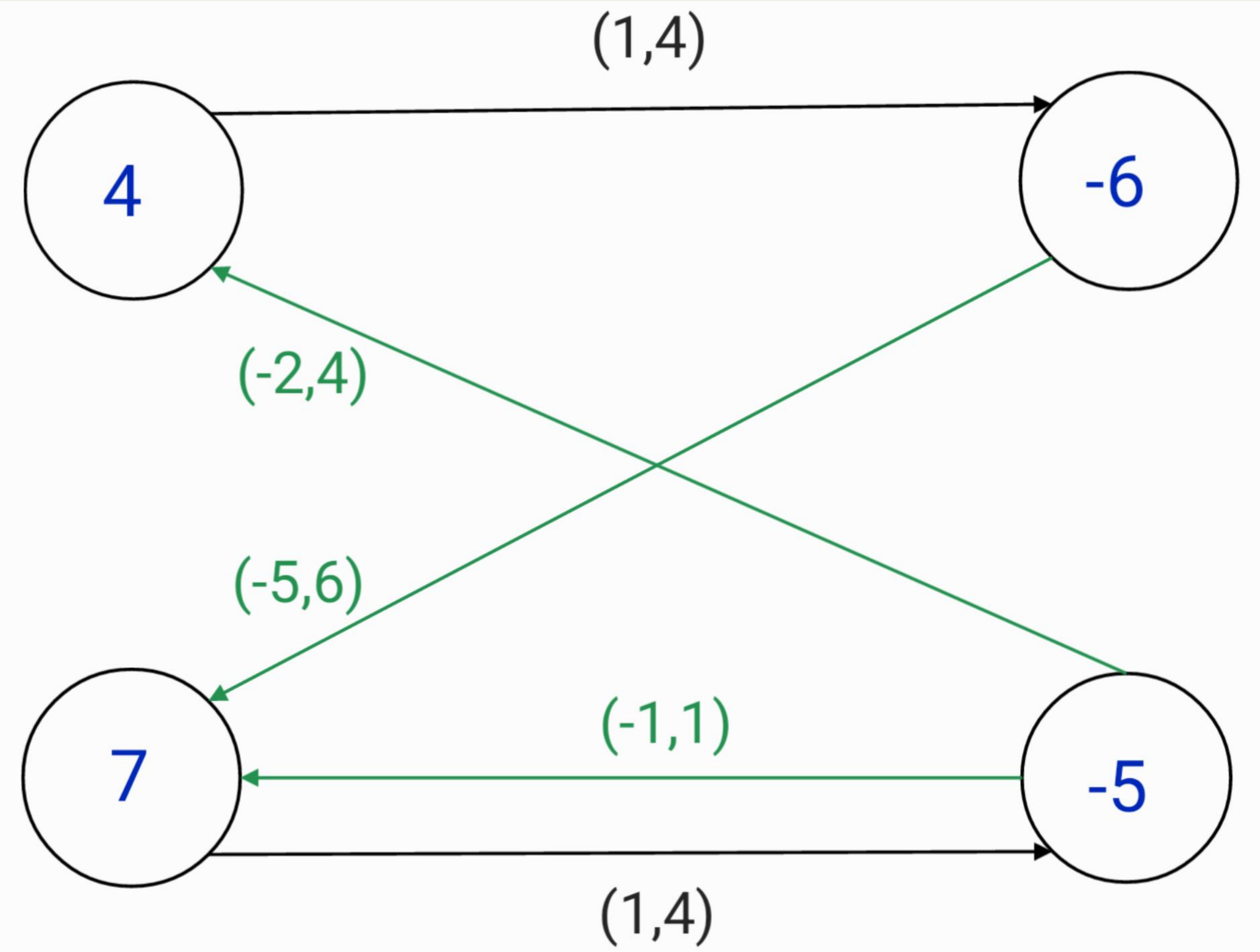
while the residual network G(x) contains a negative cost cycle:

1. Build a residual graph based on the initial feasible solution.
2. Use an algorithm (options discussed in complexity section) to detect a negative cost cycle W in the residual graph of the initial feasible solution. If no negative cost cycle is found **end algorithm.** (The current solution is the optimal solution.)
3. If a negative cost cycle W is found, set $\delta = min\{r_{ij} : (i, j) \in W\}$; (sets the amount of units to augment the path to the lowest remaining capacity along that path)
4. Augment δ units of flow along all arcs in W to "cancel" the negative cycle and update residual graph $G(x)$; (The updated $G(x)$ becomes the residual graph used for step 1 of the next iteration.)

Iteration 1: Graph

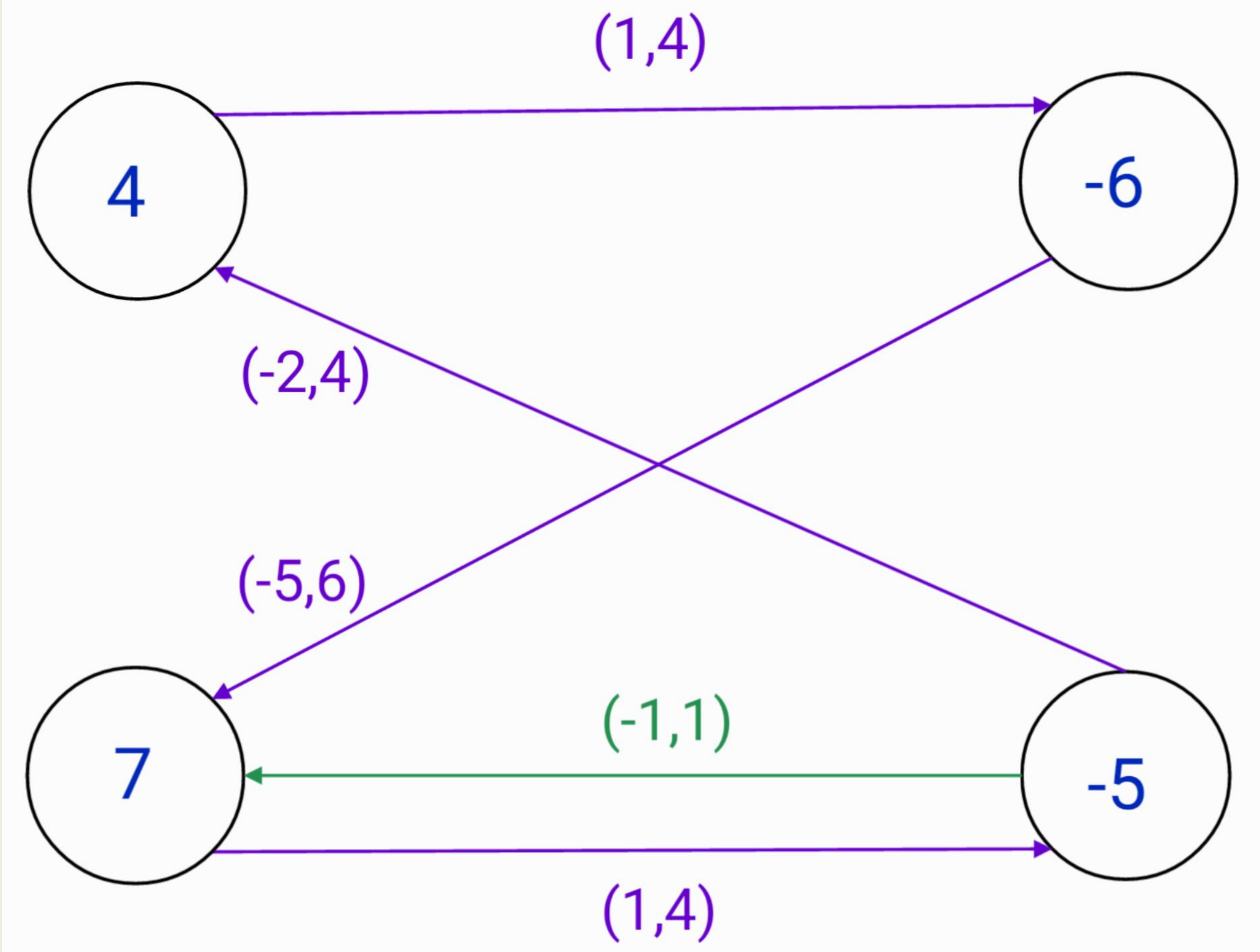


- Arcs:
  (Cost, Flow/Capacity)
- Nodes:
  Supply/Demand



Iteration 1: Residual Graph

- Arcs:
  (Cost, Capacity)
- Nodes:
  Supply/Demand
- Arcs:
  (Cost, Flow)



Iteration 1: Residual Graph

- Arcs in Neg. Cycle: (Cost, Capacity)
- Nodes: Supply/Demand

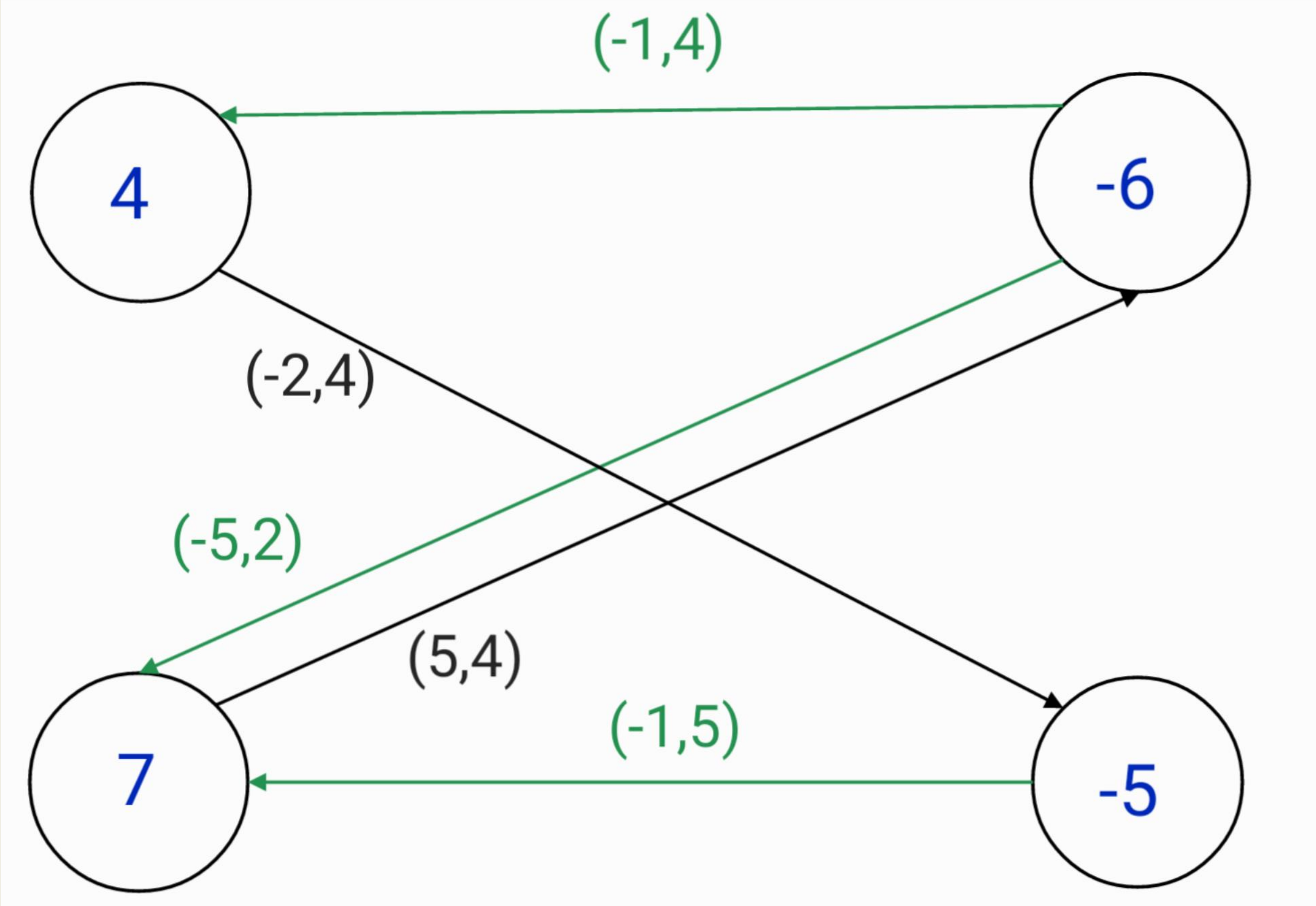
<u>Neg. cycle cost:</u>

1 + (-5) + 1 + (-2) = -5

<u>Neg. cycle capacity:</u>

Min(4,6,4,4) = 4

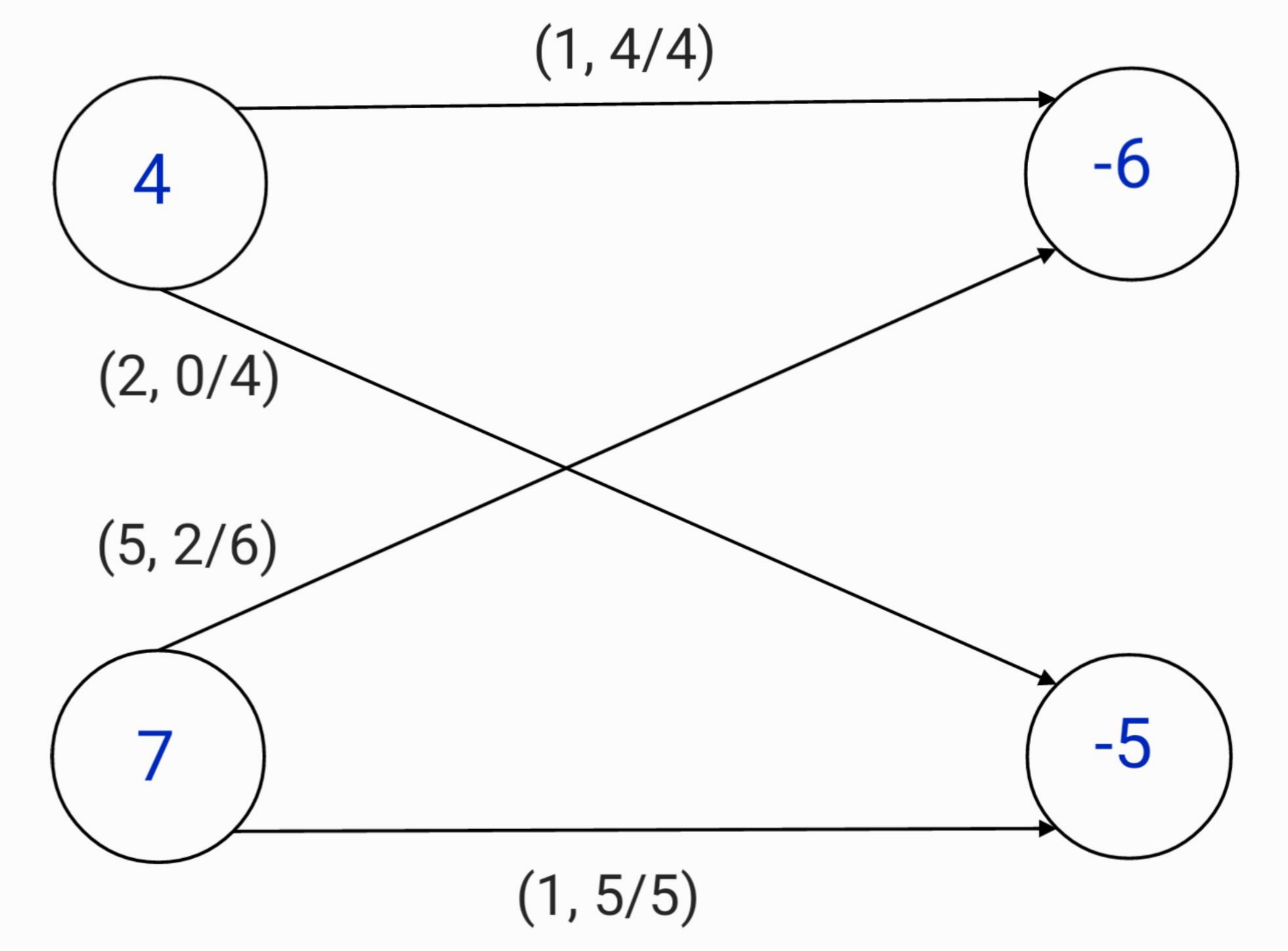
We “cancel” the cycle by augmenting it with 4 units of flow.



Iteration 2: Residual Graph

- Arcs: (Cost, Capacity)
- Nodes: Supply/Demand
- Arcs: (Cost, Flow)

There are no more negative cycles, so we have reached the optimal solution!



Iteration 2: Graph

- Arcs:
  (Cost, Flow/Capacity)
- Nodes:
  Supply/Demand

Optimal solution cost:
4(1)+2(5)+5(1) = 19

Optimality Condition

- We will know we have an optimal solution when no negative cycles remain.
- A negative cost cycle existing implies that an alternate path with a lower cost still exists and/or has not been fully utilized yet.
- With both maximum flow and no negative cycles the solution is optimal!

Complexity

There are three components to the complexity:

1. The initial alteration to a max flow problem: **$O(n)$**
2. Solving the max flow problem:
   – ***$O(nm^2)$*** *Edmonds-Karp*
   – ***$O(mF)$*** *Ford-Fulkerson*
   – ***$O(n^3)$*** *Preflow-Push (using FIFO selection rule)*
3. Finding negative cycles (for mCU “max” iterations):
   – ***$O(n^3 mCU)$*** *Floyd-Warshall*
   – ***$O(nm^2 CU)$*** *FIFO Label-correcting algorithm*

Application

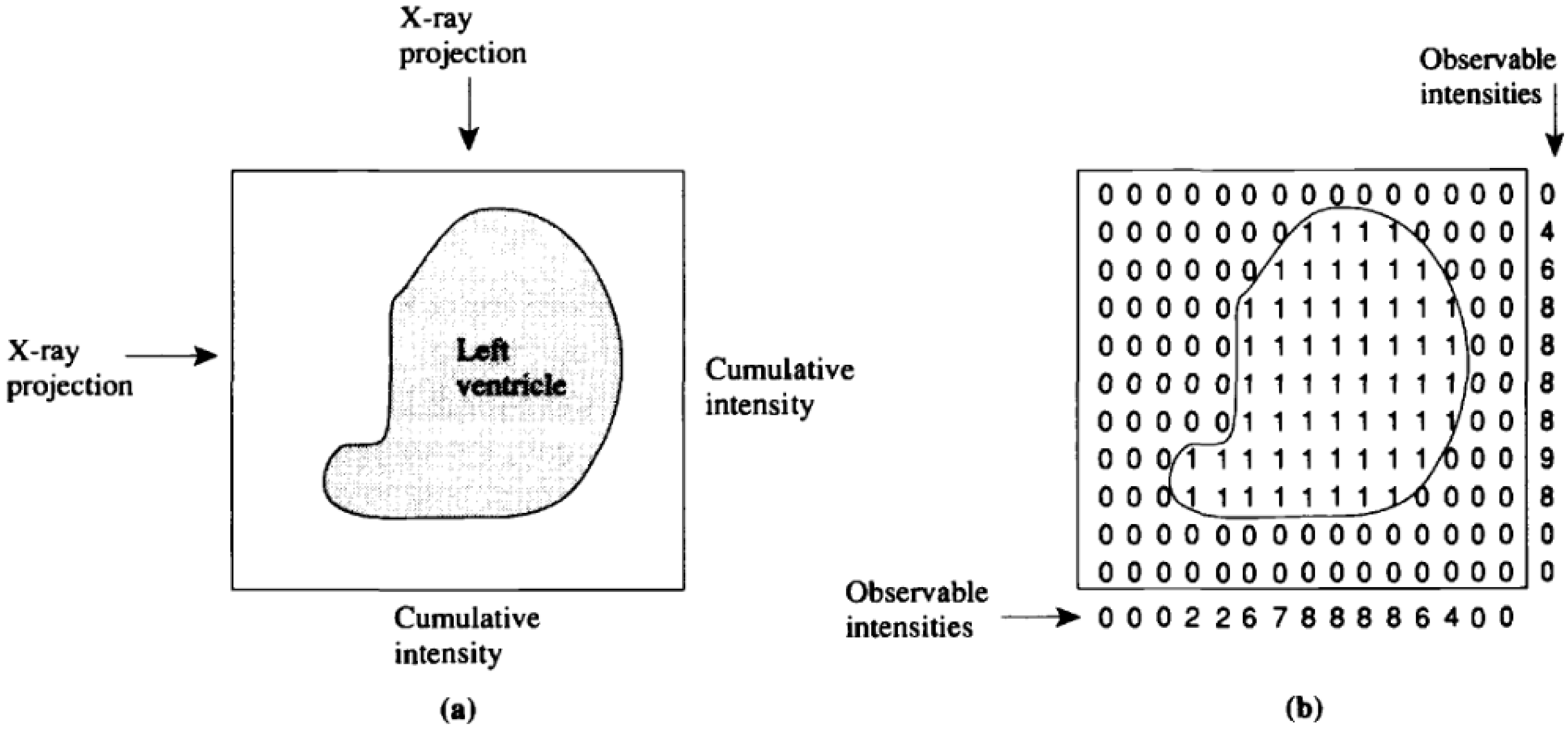
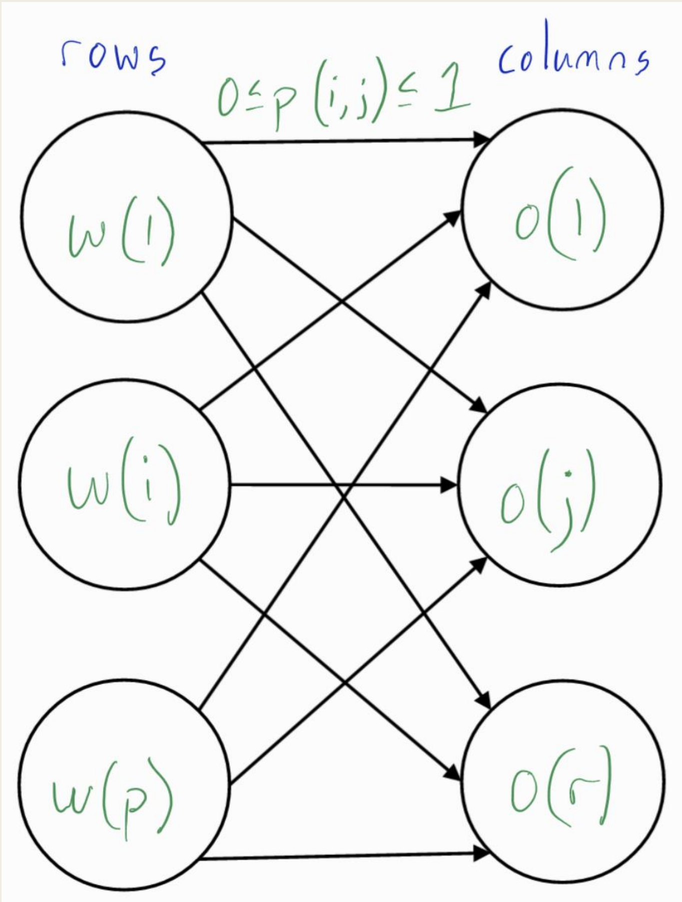
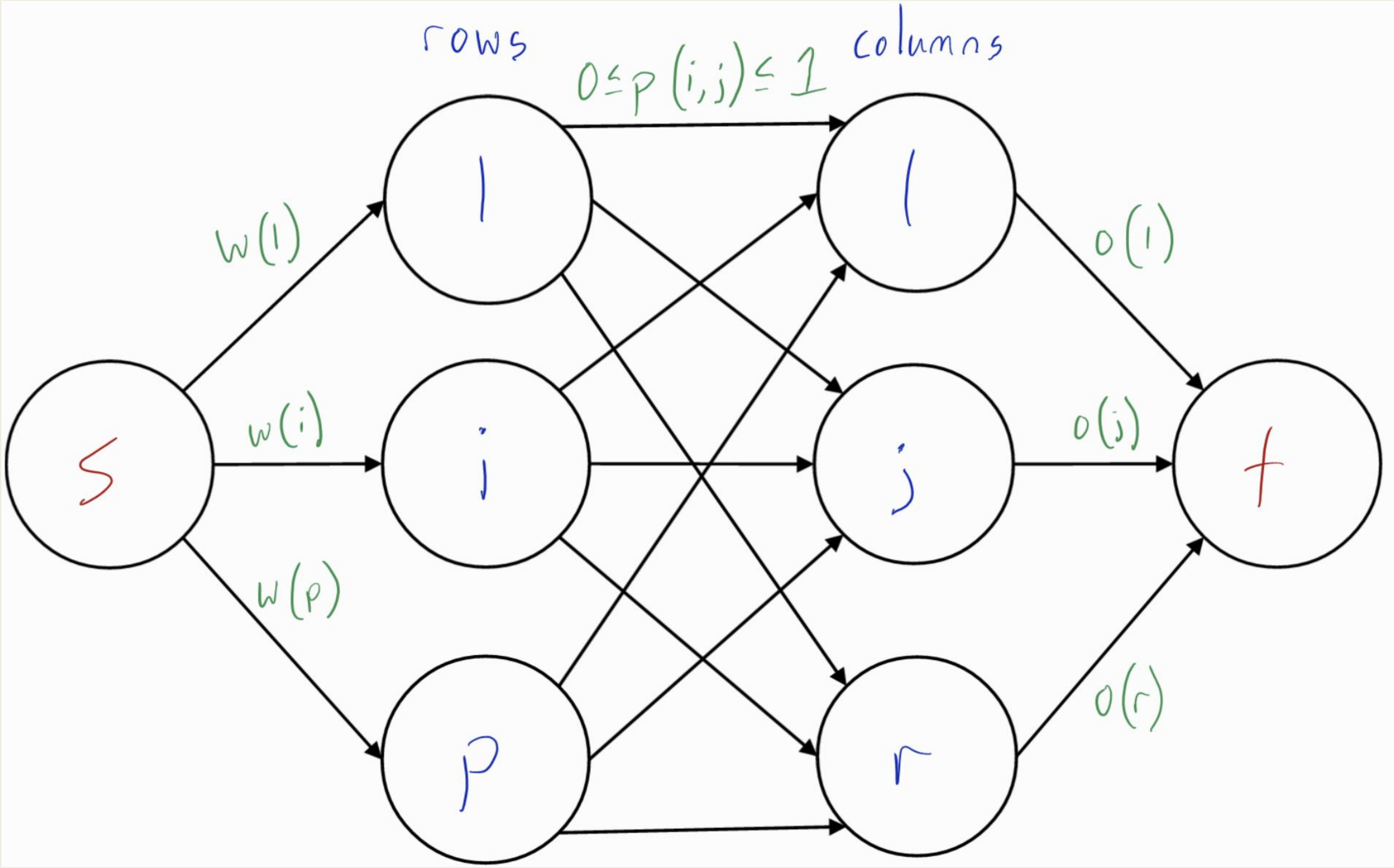


Figure 9.2 Using x-ray projections to measure a left ventricle.

Application II





Questions?

Sources

1. Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). Network flows: Theory, algorithms, and applications. Prentice-Hall.
2. Korte, B., & Vygen, J. (2018). Combinatorial optimization: Theory and algorithms. Springer.