

CSC373 Worksheet 5 Solution

August 9, 2020

1. *Proof.* Assume that a flow network $G = (V, E)$ violates the assumption that the network contains a path $s \rightsquigarrow v \rightsquigarrow t$ for all vertices $v \in V$. Let u be a vertex for which there is no path $s \rightsquigarrow u \rightsquigarrow t$.

I must show such that there is no flow at vertex u . That is, there exists a maximum flow f in G such that $f(u, v) = f(v, u) = 0$ for all vertices $v \in V$.

Assume for the sake of contradiction that there is some vertex u with flow f . That is, there exists some vertices $v \in V$ such that $f(u, v) > 0$ or $f(v, u) > 0$.

I see that three cases follows, and I will prove each separately.

1. **Cases 1:** $f(u, v) = 0$ and $f(v, u) > 0$

Here, assume that $f(u, v) = 0$ for all $v \in V$ and $f(v, u) > 0$ for some $v \in V$.

Then, we can write $\sum_{v \in V} f(u, v) = 0$ and $\sum_{v \in V} f(v, u) > 0$

But this violates the flow conservation property (i.e $\sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u)$)

Thus, by proof by contradiction, $f(u, v) = 0$ and $f(v, u) = 0$ for all $v \in V$ and all $u \in V$ with no path $s \rightsquigarrow u \rightsquigarrow t$.

2. **Cases 2:** $f(u, v) > 0$ and $f(v, u) = 0$

Here, assume that $f(u, v) > 0$ for some $v \in V$ and $f(v, u) = 0$ for all $v \in V$.

Then, by similar work as case 1, the same result follows.

3. Cases 3: $f(u, v) > 0$ and $f(v, u) > 0$

Here, assume that $f(u, v) > 0$ and $f(v, u) > 0$ for some $v \in V$.

Since $s \rightsquigarrow v \rightsquigarrow t$ and u is connected by some vertices v , we can write $s \rightsquigarrow u \rightsquigarrow t$.

Then, this violates the fact in header that the vertex u has no path $s \rightsquigarrow u \rightsquigarrow t$.

Thus, by proof by contradiction, $f(u, v) = 0$ and $f(v, u) = 0$ for all $v \in V$ and all $u \in V$ with no path $s \rightsquigarrow u \rightsquigarrow t$.

□

Notes

• Maximum Flow:

- Finds a flow of maximum value ^[1]

Example



Here, the maximum flow is $10 + 5 + 13 = 28$

• Flow Network:

- $G = (V, E)$ is a directed graph in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$.
- Two vertices must exist: **source** s and **sink** t
- **path** from source s to vertex v to sink t is represented by $s \rightsquigarrow v \rightsquigarrow t$



- **Capacity:**

- Is a non-negative function $f : V \times V \rightarrow \mathbb{R}_{\geq 0}$
- Has **capacity constraint** where for all $u, v \in V$ $0 \leq f(u, v) \leq c(u, v)$
 - * Means flow cannot be above capacity constraint

- **Flow:**

- Is a real valued function $f : V \times V \rightarrow \mathbb{R}$ in G
- Satisfies **capacity constraint** (i.e for all $u, v \in V$, $0 \leq f(u, v) \leq c(u, v)$)
- Satisfies **flow conservation**

For all $u \in V - \{s, t\}$, we require

$$\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) \quad (1)$$

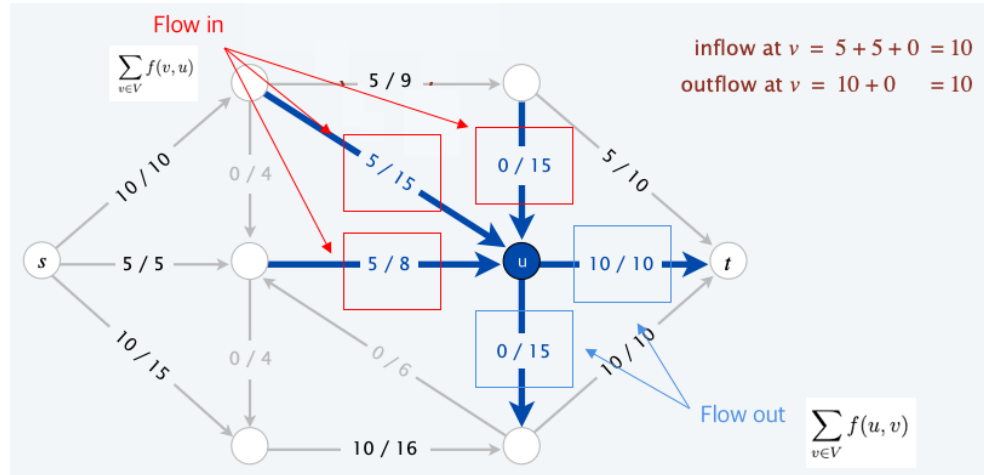
* Means flow into vertex u is the same as flow going out of vertex u . ^[1]

* $\sum_{v \in V} f(u, v)$ means flow out of vertex u

* $\sum_{v \in V} f(v, u)$ means flow into vertex u

* $v \in V$ in $\sum_{v \in V} f(u, v)$ means all vertices that are an edge away from vertex u

Example:



References

- 1) Princeton University, Network Flow 1, link
2. I need to formulate the problem of determining whether both of professor Adam's two children can go to the same school as maximum-flow problem.

The problem statement tells us the following:

1. There is 1 supersource (location of home)
2. There is 1 sink (location of school)
3. There are two sources (s_1 as child 1, s_2 as child 2)
4. Edge (u, v) has capacity of 0 or more (0 representing unavailable sidewalk, 1 for sidewalk with capacity of 1, 2 for street with capacity of 2 and so on)
5. Each vertex represents corner of intersection, and two children can have their paths crossing here.
6. Has flow of 2, 1 or 0 (1 is where one of the two children walking on the road. 0 is none.)

Here we are to find whether children must go on to a vertex and out to the same edge with the flow of 2, or determine whether there is only edge to school with capacity of 1 or less.

If none, then both children can safely go to school.



Notes:

- **Cross at a Corner**

- Means to walk across the street at a corner of the intersection.



- **Multiple Sources and Sinks**

- Has edges (s, s_i) where $i = 1 \dots n$ and (t_j, t) where $j = 1 \dots n$ with capacity of ∞

Example:

Lucky Puck Company having a set of m factories $\{s_1, s_2, \dots, s_m\}$, and a set of n warehouses and n warehouses $\{t_1, t_2, \dots, t_n\}$



3. I need to show how to transform a flow network $G = (V, E)$ with vertex capacities into an equivalent flow network $G' = (V', E')$ without vertex capacities.

For each vertex capacities, change as follows.



After transformation, there will be m more edges and vertices, where m represents the number of vertex capacities in G .

Notes:

- **Vertex Capacities**

- Each vertex v has limit $l(v)$ on how much flow can pass through v

4. I need to show how to convert the problem of finding a flow f that obeys the constraints into the problem of finding a maximum flow in a single source, single-sink flow network

The steps are as follows:

- Combine all sources s_i into a single source s
- Combine all sinks t_j into a single sink t
- Connect source s to each adjacent vertex v with edge weight $\sum_i f(s_i, v) = p_i$
 - The total edge weight from s should be $\sum_i p_i$
- Connect each adjacent vertex v of t to t with edge weight $\sum_j f(v, t_j) = q_j$
 - The total edge weight to t should be $\sum_j q_j$
- Find a simple path from s to t with the maximum amount of total flow



Correct Solution:

I need to show how to convert the problem of finding a flow f that obeys the constraints into the problem of finding a maximum flow in a single source, single-sink flow network

The steps are as follows:

- Combine all sources s_i into a single source s
- Combine all sinks t_j into a single sink t
- Connect source s to each adjacent vertex v with edge weight $\sum_i f(s_i, v) = p_i$
 - The total edge weight from s should be $\sum_i p_i$
- Connect each adjacent vertex v of t to t with edge weight $\sum_j f(v, t_j) = q_j$
 - The total edge weight to t should be $\sum_j q_j$
- Find a simple path from s to t with the maximum amount of total flow

Example



Notes:

- **Ford-Fulkerson Method**
 - Is a greedy algorithm that solves the maximum-flow problem
 - * Determines maximum flow from start vertex to sink vertex in a graph
 - Called method (not algorithm) because several different implementations with different running time is used

FORD-FULKERSON-METHOD(G, s, t)

- 1 initialize flow f to 0
- 2 **while** there exists an augmenting path p in the residual network G_f
- 3 augment flow f along p
- 4 **return** f

• Residual Network

- Indicates how much more flow is allowed in each edge in the network graph ^[1]
- Consists of edges with capacities that represents how we can change the flow on edges of G .
- Provides roadmap for adding flow to the original flow network



Steps

- 1) $Flow = Capacity$: Opposite arrow



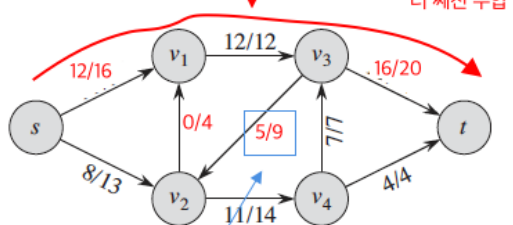
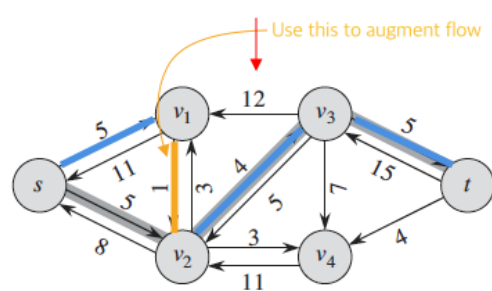
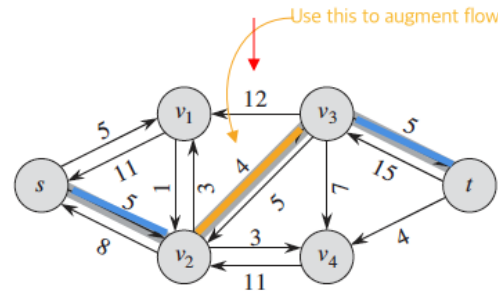
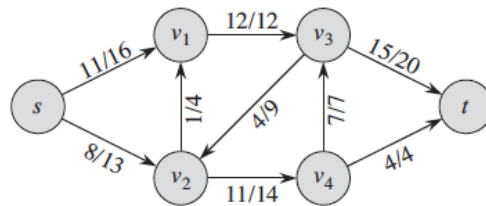
- 2) $Flow < Capacity$:

- $Flow$: Opposite Arrow
- $Capacity - Flow$: Current Arrow



• Augmenting Path

- Is a path from source S to sink T where you can increase the amount of flow
- Is a path that doesn't contain cycle (simple path) [2]



A good augmentation

An augmentation but not good
(decrease 가 필요한 곳에 쓰이면 더 좋았을 걸)

- Edge (u, v) on simple path on an augmenting path can be increased by upto $c_f(u, v)$ without violating the capacity constraint

• Augmentation

- 한국어로 '불필요한 수압 decrease 해서 앞으로 가는 수압 더 쉐게 만들기'
- Is symbolized by $f \uparrow f'$

- * f is a flow in G
- * f' is a flow in the residual network G_f

References

- 1) Hacker Earth, Maximum Flow, [link](#)
- 2) Stack Overflow, What Exactly Is Augmentation Path, [link](#)

5. **Rough Works:**

I need to show if the augmented flow of f and $f' \in G$ and satisfy the flow conservation property and capacity constraint.

- Proving the flow conservation property
- Proving the capacity constraint