COMPSCI 330: Design and Analysis of Algorithms 09/25/2014

Lecture #10

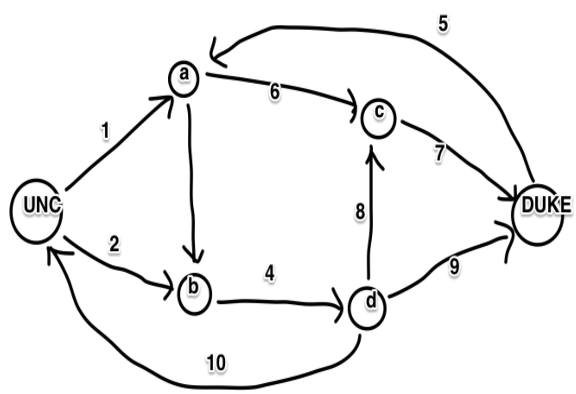
Lecturer: Debmalya Panigrahi *Scribe: Ang Li*

1 Overview

This lecture covers the basic definitions and properties of network flows. The maximum flow problem is also discussed and a simple algorithm for solving the maximum flow problem is presented.

2 Network Flow

Network flow problem is the problem of sending a flow from a source node to a target node in a directed graph while satisfying the properties of network flows. A real-life example will be to send a number of people from UNC to Duke to attend a basketball game while not overloading the traffic paths (edges) or causing accumulation of people at any traffic junctions (nodes). The network graph is illustrated in the diagram below.



2.1 Properties of Network Flows

In this subsection, we give the properties of Network Flows. The first property is capacity constraint. The second property is flow balance. Mathematically,

$$f_e \leq u_e, \ \forall e \in E \tag{1}$$
$$\Sigma_{e \in E_v^{in}} f_e = \Sigma_{e \in E_v^{out}} f_e, \ \forall v \in V \setminus \{s,t\} \tag{2}$$

2.2 Definition of Flow Value

$$Flow\ Value\ = \Sigma_{e\in E_s^{out}} - \Sigma_{e\in E_s^{in}} = \Sigma_{e\in E_t^{in}} - \Sigma_{e\in E_t^{out}} \tag{3}$$

Proof.

$$\begin{aligned}0 &= \Sigma_{e\in E}(f_e - f_e)\\ &= \Sigma_{e\in E} f_e - \Sigma_{e\in E} f_e\\ &= \Sigma_{v\in V}\Sigma_{e\in E_v^{out}} f_e - \Sigma_{v\in V}\Sigma_{e\in E_v^{in}} f_e\\ &= \Sigma_{v\in V}(\Sigma_{e\in E_v^{out}} f_e - \Sigma_{e\in E_v^{in}} f_e)\\ &= \Sigma_{v\in\{s,t\}}(\Sigma_{e\in E_v^{out}} f_e - \Sigma_{e\in E_v^{in}} f_e)\end{aligned}$$

Therefore,

$$\Sigma_{e\in E_s^{out}} f_e - \Sigma_{e\in E_s^{in}} f_e = \Sigma_{e\in E_t^{in}} f_e - \Sigma_{e\in E_t^{out}} f_e \tag{4}$$

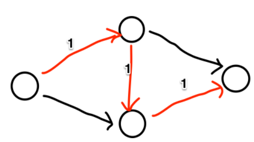
□

3 Maximum Flow Problem

Given $G = (V,E)$, whereby edge has capacity u_e, the source vertex is s, the sink vertex is t, the goal is to find the maximum flow $f : E \rightarrow \Re_+$ such that the flow value is maximized subject to capacity constraint and flow balance being satisfied.

3.1 Flow on a Path

An example of flow of magnitude 1 on a path is illustrated in the following diagram.



3.2 A Simple Algorithm

1. Find a path p from s to t.
2. Set flow on all edges of p to the minimum capacity of an edge in p.
3. Reduce u_e to $u_e - f_e$.
4. Repeat steps 1-3.

3.3 Combining Flows

1. Claim: if flows are added, flow balance is still satisfied.

Proof. Add the following equations.

$$f_e^1, f_e^2$$
$$\Sigma_{e \in E_v^{in}} f_e^1 = \Sigma_{e \in E_v^{out}} f_e^1$$
$$\Sigma_{e \in E_v^{in}} f_e^2 = \Sigma_{e \in E_v^{out}} f_e^2$$

□

2. Flow values get added.
3. The combined flow satisfies capacity constraints.

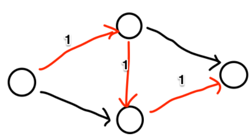
Proof.

$$u_e^1 = u_e$$
$$u_e^2 = u_e - f_e^1$$
$$f_e^2 \leq u_e^2 = u_e - f_e^1$$
$$f_e^1 + f_e^2 \leq u_e$$

□

3.4 Flaws in the Simple Algorithm

The algorithm described above fails to obtain the maximum flow on the example presented in the figure below. All edges have capacity 1 in the following graph. After the flow of magnitude 1 represented by the red edges is used, there is no other path from the source node on the left to the target node on the right. However, the optimum flow is of magnitude 2.



3.5 Residual Network

Introduce edges in the opposite directions of the assigned flow–gives us the ability to cancel assigned flows by using the residual network.
A residual network for flow f has the following capacities:
1. $u_e - f_e$ for edge e.
2. f_e for edge e^R(the reverse of e)

3.6 An improved algorithm

Repeat:
1. Find residual network $\bar{G}_f$ corresponding to current flow f.
2. Find a flow $\bar{f}$ in $\bar{G}_f$.
3. Combine f and $\bar{f}$ to obtain the new f.

3.7 More on combining flows using residual network

1. Add flows on graph edges (edges in the original graph).
2. Subtract flows on residual edges.
That is:

$$f_e = f_e^1 + f_e^2 - f_{e^R}^2 \tag{5}$$

Proof. Capacity constraint is still satisfied.

$$\begin{aligned} f_e^1 &\leq u_e \\ f_e^2 &\leq u_e - f_e^1 \\ f_e \leq f_e^1 + f_e^2 &\leq u_e \end{aligned}$$

□

Proof. Flow balance is still satisfied using the knowledge that f^1 and f^2 satisfy the flow balance.

$$\begin{aligned} \Sigma_{e \in E_v^{out}} f_e - \Sigma_{e \in E_v^{in}} f_e &= \Sigma_{e \in E_v^{out}} (f_e^1 + f_e^2 - f_{e^R}^2) - \Sigma_{e \in E_v^{in}} (f_e^1 + f_e^2 - f_{e^R}^2) \\ &= \Sigma_{e \in E_v^{out}} f_e^2 - \Sigma_{e \in E_v^{in}} f_e^2 \end{aligned}$$

□

Proof. Non-negativity

$$\begin{aligned} if\ f_e^1 + f_e^2 - f_{e^R}^2 &< 0 \\ then\ f_{e^R}^2 &> f_e^1 + f_e^2 \\ &\geq f_e^1 = u_{e^R}^2 \end{aligned}$$

□

i.e. capacity constraint is violated, which means the premise is always false.

4 Summary

This lecture covers the basic definitions and properties of network flows. An algorithm for solving the maximum flow problem is discussed with its properties and implementation details.