CMSC 451: Max-Flow Extensions

Slides By: Carl Kingsford

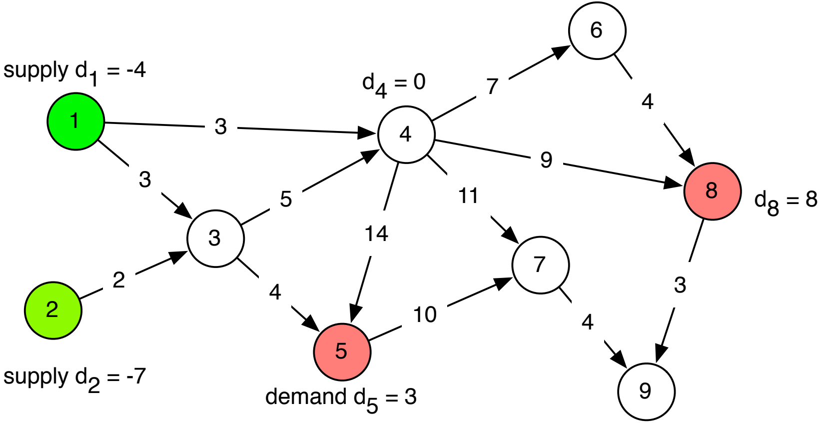
Department of Computer Science
University of Maryland, College Park

Based on Section 7.7 of *Algorithm Design* by Kleinberg & Tardos.

Circulations with Demands

- Suppose we have multiple sources and multiple sinks.
- Each sink wants to get a certain amount of flow (its **demand**).
- Each source has a certain amount of flow to give (its **supply**).
- We can represent supply as **negative demand**.

Demand Example



Constraints

Goal: find a flow f that satisfies:

1. Capacity constraints: For each $e \in E$, $0 \leq f(e) \leq c_e$.
2. Demand constraints: For each $v \in V$,

$$f^{\text{in}}(v) - f^{\text{out}}(v) = d_v.$$

The demand d_v is the excess flow that should come into node.

Sources and Sinks

Let S be the set of nodes with **negative** demands (supply).

Let T be the set of nodes with **positive** demands (demand).

In order for there to be a feasible flow, we must have:

$$\sum_{s \in S} -d_s = \sum_{t \in T} d_t$$

Let $D = \sum_{t \in T} d_t$.

Reduction

How can we turn the **circulation with demands** problem into the maximum flow problem?

Reduction

How can we turn the **circulation with demands** problem into the maximum flow problem?

1. Add a new source s^* with an edge (s^*, s) from s^* to every node $s \in S$.

2. Add a new sink t^* with an edge (t, t^*) from t^* to every node $t \in T$.

Reduction

How can we turn the **circulation with demands** problem into the maximum flow problem?

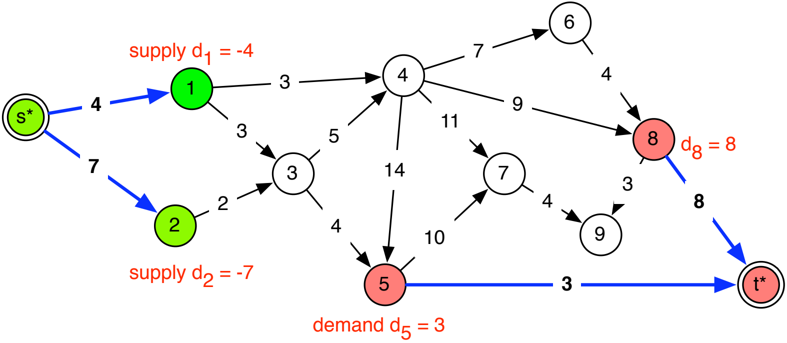
1. Add a new source s^* with an edge (s^*, s) from s^* to every node $s \in S$.

2. Add a new sink t^* with an edge (t, t^*) from t^* to every node $t \in T$.

The capacity of edges $(s^*, s) = -d_s$ (since $d_s < 0$, this is +ve)

The capacity of edges $(t, t^*) = d_t$.

Circulation Reduction Example



Feasible circulation if and only if there is a flow of value $D = \sum_{t \in T} d_t$.

Notes

Intuition:

- Capacity of edges (s^*, s) limit the supply for source nodes s.
- Capacity of edges (t, t^*) require that d_t flow reaches each t.

Hence, we can use max-flow to find these circulations.

Lower Bounds

Another extension: what if we want **lower** bounds on what flow goes through some edges?

In other words, we want to require that some edges are used.

Goal: find a flow f that satisfies:

1. Capacity constraints: For each $e \in E$, $\ell_e \leq f(e) \leq c_e$.
2. Demand constraints: For each $v \in V$,

$$f^{\text{in}}(v) - f^{\text{out}}(v) = d_v.$$

Lower Bounds

Suppose we defined an initial flow f_0 by setting the flow along each edge equal to the lower bound. In other words: $f_0(e) = \ell_e$.

This flow satisfies the capacity constraints, but not the demand constraints.

Define: $L_v = f_0^{\mathrm{in}}(v) - f_0^{\mathrm{out}}(v)$.

Recall that the demand constraints say that $f^{\mathrm{in}}(v) - f^{\mathrm{out}}(v) = d_v$. Hence, L_v is equal to the amount of the demand that f_0 satisfies at node v.

New Graph

For each node, our flow f_0 satisfies L_v of its demand, hence we have:

New demand constraints:

$$f^{\text{in}}(v) - f^{\text{out}}(v) = d_v - L_v$$

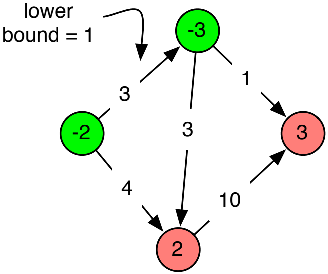
Also, f_0 uses some of the edge capacities already, so we have:

New capacity constraints:

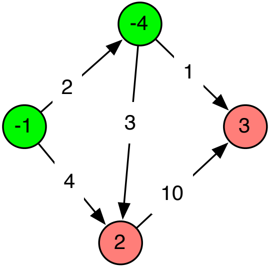
$$0 \leq f(e) \leq c_e - \ell_e$$

These constraints give a standard instance of the circulation problem.

Lower Bound Example



(a) Small instance where one edge has a lower bound. This makes the most obvious flow not feasible.



(b) After transformation, we have an equivalent instance with no lower bounds.

Reduction:

Given a circulation instance G with lower bounds, we:

1. subtract ℓ_e from the capacity of each edge e, and
2. subtract L_v from the demand of each node v.
   (This may create some new "sources" or "sinks".)

We then solve the circulation problem on this new graph to get a flow f'.

To find the flow that satisfies the original constraints, we add ℓ_e to every $f'(e)$.

Summary

We can efficiently find a feasible flow for the following general problem:

Circulations with demands and lower bounds

Given:

- a directed graph G
- a nonnegative lower bound ℓ_e for each edge $e \in G$
- a nonnegative upper bound $c_e \geq \ell_e$ for each edge $e \in G$
- and a demand d_v for every node

Find: a flow f such that

- $\ell_e \leq f(e) \leq c_e$ for every e, and
- $f^{\mathrm{in}}(v) - f^{\mathrm{out}}(v) = d_v$ for every v.

Serial Reductions…

We designed the algorithm for this general problem by reducing CIRCULATION WITH LOWER BOUNDS problem to the CIRCULATION WITHOUT LOWER BOUNDS problem. We in turn reduced that problem to the MAX FLOW problem.