Max-Flow Extensions

Slides by Carl Kingsford

Apr. 9, 2014

Based on AD 7.7

Extensions to Flow Problem

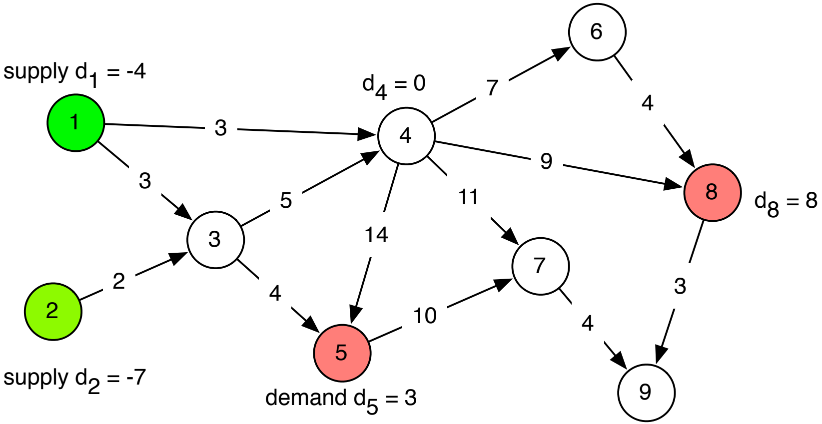
1. What if several nodes produce a given amount of material and other nodes want a certain amount. (E.g. factories produce a given amount and customers want a given amount.)

2. Requirement to send at least a given amount through each pipe.

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 $v \in V$,

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

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?

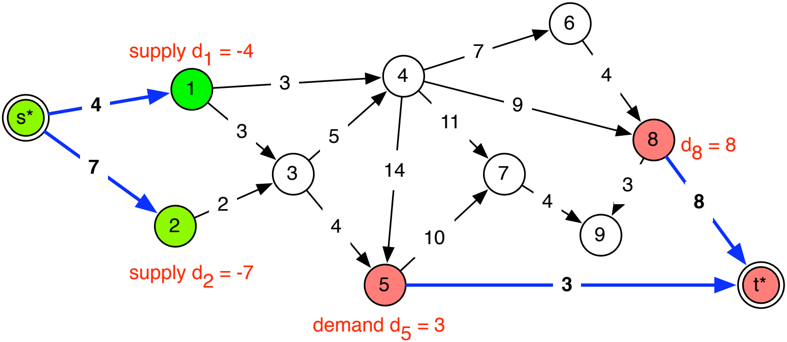
1. Add a new source s^* with an edge (s^*, s) from s^* to every node $s \in S$.

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

Recall that the demand constraints say that $f^{\text{in}}(v) - f^{\text{out}}(v) = d_v$. 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:

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

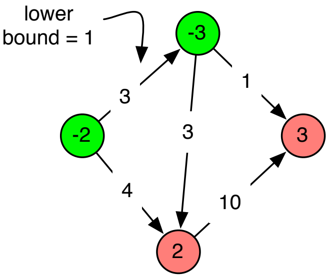
Also, f_0 uses some of the edge capacities already, so we have:

New capacity constraints:

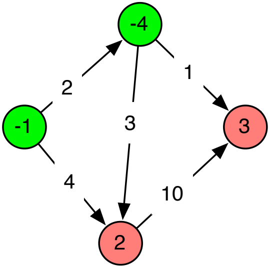
$$0 \leq f(e) \leq c_e - \ell_e$$

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:

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

Example Problem: Airline Scheduling

- We are given flight information of the form:

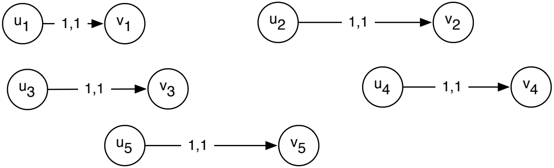
  Depart city C_1 at time T_1 and arrive D_1 at time L_1

- A single plane can fly from C_1 to C_2 to C_3 provided there is at least z hours between landing at C_2 and taking off at C_2 (to perform maintence).
- If you need to get a plane from C_1 to C_2 you can fly it there even if you don't have a scheduled flight.
- **Question:** Can you serve all flights using $\leq k$ planes?

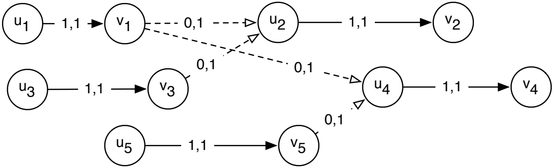
FROM
AMSTERDAM, NETHERLANDS (AMS)

ATLANTA, GA (ATL)							4395mi
May10		9 15a	1 03p	DL0175	0	X3	76W
May2		10 55a	2 38p	DL0239	0	Daily	333
	Apr16	10 55a	2 42p	DL0239	0	Daily	333
Apr18	May1	10 55a	2 42p	DL0239	0	Daily	333
Apr17	Apr17	10 55a	2 53p	DL0239	0	4	76W
Apr17	Apr17	1 05p	4 49p	DL0603	0	4	333
May1	May13	1 05p	4 56p	DL0603	0	X3	76W
May14		1 05p	4 56p	DL0603	0	345	76W
	Apr15	1 05p	5 00p	DL0049	0	X3	76W
Apr18	Apr29	1 05p	5 00p	DL0049	0	X3	76W
Apr2	Apr20	4 50p	8 20p	DL9374	0	X245	333
		Operated By KLM					
	Apr18	5 00p	8 20p	DL9374	0	245	777
		Operated By KLM					
Apr21		5 00p	8 20p	DL9374	0	Daily	777
		Operated By KLM					
BOSTON, MA (BOS)							3451mi
May2		11 10a	12 59p	DL0231	0	Daily	333
	May1	11 10a	1 03p	DL0231	0	Daily	333
May16		2 55p	4 44p	DL0267	0	5	332
CALGARY, CANADA (YYC)							4459mi
		12 30p	1 30p	DL9397	0	Daily	332
		Operated By KLM					

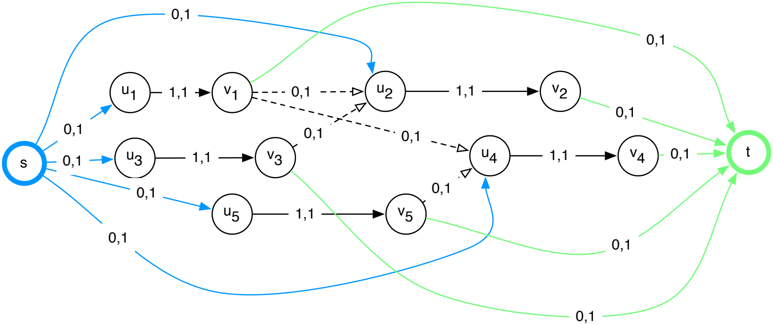
A DAG model



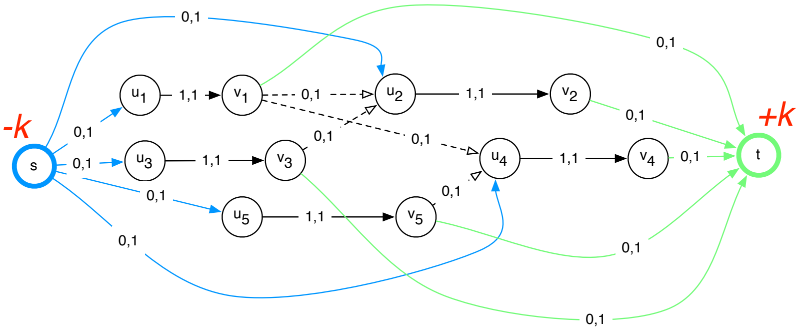
A DAG model



A DAG model



A DAG model



-k
+k
s
t
u1
v1
u2
v2
u3
v3
u4
v4
u5
v5
1,1
1,1
1,1
1,1
1,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1

A DAG model