


Network Flow: Applications

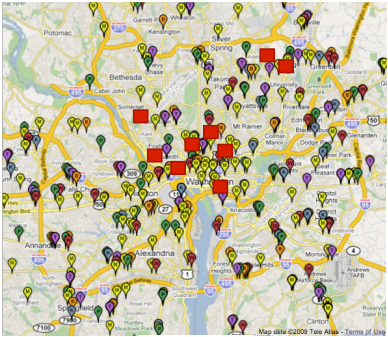
CMSC 451, Summer 2009



Some slides by Kevin Wayne.
Copyright © 2009 Pearson Education, Inc.
All rights reserved.

7.7 Extensions to Max Flow

Providing Flow to Multiple Customers: Fast Food Locations



We want:

- multiple sources
- multiple sinks

Screen Stats	
132	McDonald's
32	Burger King
47	Pizza Hut
41	Taco Bell
41	Wendy's
44	KFC
0	Jack
0	Hardee's
0	Carl's Jr.
0	In-N-Out
337	Total

fastfoodmaps.com

Circulation with Demands

Circulation with demands.

- Directed graph $G = (V, E)$.
- Edge capacities $c(e), e \in E$.
- Node supply and demands $d(v), v \in V$.

demand if $d(v) > 0$; supply if $d(v) < 0$; transshipment if $d(v) = 0$

Def. A **circulation** is a function that satisfies:

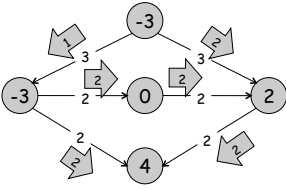
- For each $e \in E$: $0 \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem: given (V, E, c, d) , does there exist a circulation?

Example Circulation

Def. A **circulation** is a function that satisfies:

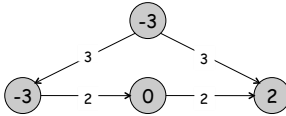
- For each $e \in E$: $0 \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)



Example With No Circulation

Def. A **circulation** is a function that satisfies:

- For each $e \in E$: $0 \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)



Why doesn't this example have a circulation?

Circulation with Demands

Necessary condition: sum of supplies = sum of demands.

$$\sum_{v: d(v) < 0} d(v) = \sum_{v: d(v) > 0} -d(v) =: D$$

Pf. Suppose a feasible circulation f exists.
 Then $\sum_v d(v) = \sum_v \left(\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) \right)$ by flow conservation.

Consider some edge $e=(u,v)$.
 The flow $f(e)$ into u is counted twice:
 Once into u (+)
 Once out of v (-)
 This holds for all $f(e)$.
 So $\sum_v d(v) = 0$.

Circulation with Demands

Max flow formulation.

Circulation with Demands

Max flow formulation.

- Add new source s and sink t .
- For each v with $d(v) < 0$, add edge (s, v) with capacity $-d(v)$.
- For each v with $d(v) > 0$, add edge (v, t) with capacity $d(v)$.
- Claim: G has circulation iff G' has max flow of value D .

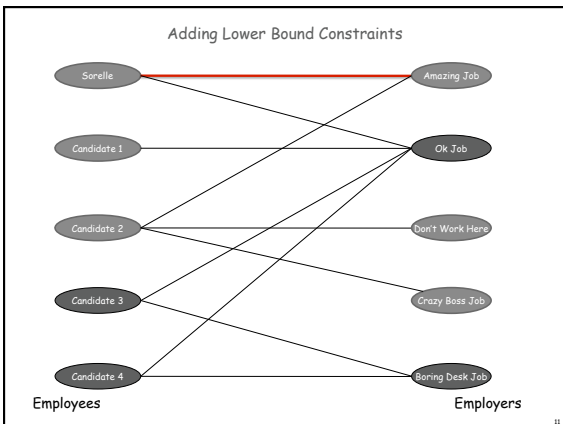
Circulation with Demands

Integrality theorem. If all capacities and demands are integers, and there exists a circulation, then there exists one that is integer-valued.

Pf. Follows from max flow formulation and integrality theorem for max flow.

Characterization. Given (V, E, c, d) , there does **not** exist a circulation iff there exists a node partition (A, B) such that $\sum_{v \in B} d_v \leq \text{cap}(A, B)$.

Pf idea. Look at min cut in G' .



Circulation with Demands and Lower Bounds

Feasible circulation.

- Directed graph $G = (V, E)$.
- Edge capacities $c(e)$ and lower bounds $l(e)$, $e \in E$.
- Node supply and demands $d(v)$, $v \in V$.

Def. A **circulation** is a function that satisfies:

- For each $e \in E$: $l(e) \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem with lower bounds. Given (V, E, l, c, d) , does there exist a circulation?

How can we reduce this to a previously solved problem?

Circulation with Demands and Lower Bounds

Idea. Model lower bounds with demands.

- Send $\ell(e)$ units of flow along edge e .
- Update demands of both endpoints.

lower bound upper bound

$d(v)$ G $d(w)$

capacity

$d(v)+2$ G' $d(w)-2$

Circulation with Demands and Lower Bounds

Idea. Model lower bounds with demands.

- Send $\ell(e)$ units of flow along edge e .
- Update demands of both endpoints.

lower bound upper bound

$d(v)$ G $d(w)$

capacity

$d(v)+2$ G' $d(w)-2$

Theorem. There exists a circulation in G iff there exists a circulation in G' . (If all demands, capacities, and lower bounds in G are integers, then there is a circulation in G that is integer-valued.)

Pf sketch. $f(e)$ is a circulation in G iff $f'(e) = f(e) - \ell(e)$ is a circulation in G' .

7.9 Airline Scheduling

The Real World Problem

Airline carriers schedule flights so that the flights and routes are efficient in terms of...

- equipment usage
- crew allocation
- customer satisfaction

while maximizing profit

and remaining flexible enough to handle

- weather problems
- breakdowns
- other unpredictable last-minute events

Our Version of the Problem

Instead of the real-world version, we'll consider a cleaner, simplified version...

Let m be the number of flight segments we'd like to serve ($1 \leq j \leq m$):
The j th flight is specified as:

- Origin Airport j
- Destination Airport j
- Departure Time j
- Arrival Time j

For example:
Washington, DC to New York, NY
8am - 9am

Our Version of the Problem

Instead of the real-world version, we'll consider a cleaner, simplified version...

We can use the same plane for two flights if

- there is enough time to perform maintenance between the flights (1 hour needed)
- there is enough time for the plane to get from the destination of its previous flight to the origin of its next flight

For example:
Washington, DC to New York, NY 8am - 9am
-- move the plane back to DC (1 hour)
-- perform maintenance (1 hour)
Washington, DC to New York, NY 11am - noon

Modeling the Problem

Example input:

1. Boston to DC, 6am to 7am
2. Philadelphia to Pittsburgh, 7am to 8am
3. DC to Los Angeles, 8am to 11am
4. Philadelphia to San Francisco, 11am to 2pm
5. San Francisco to Seattle, 2:15pm to 3:15pm
6. Las Vegas to Seattle, 5pm to 6pm

Modeling the Problem

Definition:
A flight j is *reachable* from a flight i if it is possible to use the same plane for flight i and then for flight j .

The input to the problem consists of the **flight graph** with an edge between the nodes for flights i and j if j is reachable from i given the maintenance constraints.

Our Airline Scheduling Problem

Goal:
Given a problem instance:

- reachability graph
- m flights to schedule

 determine if it is possible to schedule all flights using at most k planes.

Our Airline Scheduling Problem: Example

Problem instance:
Can we use only $k=2$ planes to schedule the 6 flights below?

Our Airline Scheduling Problem: Example

Problem instance:
Can we use only $k=2$ planes to schedule the 6 flights below?

Yes!

Airline Scheduling: Developing an Algorithm

What do units of flow correspond to?
Airplanes

What are the edge capacities?
[0,1] for edges indicating reachability
[1,1] for edges of flights that need to be scheduled

What are the node demands?
- k for s , k for t , 0 for all other nodes

What does the final max flow/ min cut problem instance (graph) look like (let $k=2$)?
 $d(v) = \text{Grey nodes: } 0$

$c_e = \text{Grey, Red, and Green edges: } [0,1] \text{ Blue edges: } [1,1] \text{ Black edge: } [0,k]$

Airline Scheduling: Proof of Correctness

Theorem: There is a way to schedule all flights using at most k planes if and only if there is a feasible circulation in G .

Proof:

Schedule \Rightarrow Feasible circulation

Suppose there is some schedule using $K \leq k$ planes.

Then there are K paths with one unit of flow and one path s - t with k - K flow.

This results in a circulation satisfying the demand, capacity, and lower bound constraints.

Feasible circulation \Rightarrow Schedule

Suppose there is some feasible circulation in G .

Let K be the units of flow sent on edges other than (s,t) .

K is integral and capacity bounds are 1, so flows are 0 or 1.

Convert each collection of edges with flow 1 to a path. (conservation)

Assign one plane to each flight on the path.

25

Anonymous Review Survey

What do you need to review? What do you want us to make sure to talk about tomorrow?

Midterm Exam Topics:

Greedy Algorithms

- Intro, Interval Scheduling, Shortest Path, Minimum Spanning Tree, Union-Find, Clustering, and Huffman Coding

Divide and Conquer

- Merge Sort, Inversion Counting, Closest Pair of Points, Multiplication, and Matrix Multiplication

Dynamic Programming

- Weighted Interval Scheduling, Knapsack Problem, and Sequence Alignment

Network Flow

- Max Flow/ Min Cut, Capacity Scaling, Bipartite Matching, Extensions, and Airline Scheduling

26