Algorithms in networking

#0  Three example algorithmic problems in networking

#1  Linear programming: a powerful, generic tool

#2  Exploiting randomness for networking

#3  Distributed decision-making
Linear programming
Clue Words

Clue words can help you figure out how to solve problems.

Here's the problem.

In one amusement park, 21 of the rides are water rides and 16 are not water rides. How many rides are there combined?

Addition

Problems that combine values, such as rides, are addition problems.

In this problem the clue word “combined” tells you to add the water rides and the non-water rides. Some other clue words that tell you a problem might use addition are: add, sum, total, plus, more, together, increase, and both.

[Amusement Park Word Problems Starring Pre-algebra, Rebecca Wingard-Nelson]
What is a linear program?

Variables:  flows on edges, distance from start, …

Objective:  maximize flows_{s \rightarrow t}, minimize distances_{s \rightarrow t}, …

Constraints:  flow on edge \leq capacity of edge, …
What is a linear program?

Variables: must be real numbers, \( x \in \mathbb{R} \)

Objective: max \( \Theta \), min \( \Theta \), \( \Theta \) = linear combination of variables

Constraints: linear combination of variables \( \leq 0 \)
Many problems can be expressed as LPs

#1 Traffic engineering: managing traffic on your network

#2 Matchings: inside a circuit switch

#3 Shortest paths: finding friends and lower latency

Note: specialized algorithms will often be better!
Max-flow as a linear program
Max-flow as a linear program

Objective:
Maximize flow from s to t

Constraint:
Obey edge capacities, $c_{uv}$
Max-flow LP: what are the variables?

Variables:

- Flow \( s \rightarrow t \) : \( f \)
- Flow on edge \( u \rightarrow v \) : \( f_{uv} \)
Max-flow LP: what are the variables?

Variables:
- Flow $s \rightarrow t : f$
- Flow on edge $u \rightarrow v : f_{uv}$

Objective:
maximize $f$
Max-flow LP: what are the constraints?

Capacity: \( f_{uv} \leq c_{uv} \)

Flow from \( s \), and into \( t \):

\[ \sum_{s \to v} f_{sv} = f \]

\[ \sum_{v \to t} f_{vt} = f \]
Max-flow LP: what are the constraints?

Capacity: \( f_{uv} \leq c_{uv} \)

Conservation: in-flow = out-flow

\[
\sum_{w \to u} f_{wu} = \sum_{u \to v} f_{uv} \quad \forall \ u \in \{s, t\}
\]

\[
\sum_{s \to v} f_{sv} = f
\]

\[
\sum_{v \to t} f_{vt} = f
\]

\( f_{uv} \geq 0 \)
Max-flow LP

Variables:
- Flow $s \to t$: $f$
- Flow on edge $u \to v$: $f_{uv}$

Capacity: $f_{uv} \leq c_{uv}$

Conservation: in-flow = out-flow

Objective:
- $\sum_{s \to v} f_{sv} = f$
- $\sum_{v \to t} f_{vt} = f$
- $f_{uv} \geq 0$

Much slower than Edmonds-Karp!
MCF: multiple commodities
MCF: multiple commodities

Variables:

- Flows: $f_x$
- Flow $f_x$ on edge $u \rightarrow v$: $f_{x,uv}$

Objective:

- maximize $\sum_x f_x$  
  [ or $\min_x f_x$ ]

Simplification: each source $x$ only sends to one destination $d(x)$
MCF: multiple commodities

Capacity: \( \sum_x f_{x,uv} \leq c_{uv} \)

Conservation: in-flow = out-flow

\[ \sum_{w \rightarrow u} f_{x,wu} = \sum_{u \rightarrow v} f_{x,uv} \quad \forall \; u \in \{x, d(x)\} \]

\[ \sum_{x \rightarrow v} f_{x,xv} = f_x \]

\[ \sum_{v \rightarrow d(x)} f_{x,vd(x)} = f_x \]

Flows don’t mix!

Simplification: each source \( x \) only sends to one destination \( d(x) \)
OK, let’s try shortest path
Shortest path as a linear program

Objective:
Minimize s-t path length
(given edge lengths $w_{uv}$)

Constraint:
“Path” = connected edges
Shortest path LP: what are the variables?

Variables:

Is edge $u \rightarrow v$ on SP? $x_{uv}$

Objective:

Minimize $\sum_{u \rightarrow v} x_{uv}w_{uv}$
Shortest path LP: what are the constraints?

Path $s \rightarrow t$ is connected:

\[ \sum_{u \rightarrow v} x_{uv} - \sum_{v \rightarrow w} x_{vw} = 0 \quad \forall \{s, t\} \]

\[ = 1 \quad \forall \{v = t\} \]

\[ = -1 \quad \forall \{v = s\} \]

Also: $x_{uv} \in \{0, 1\}$?

$x_{uv} \in [0, 1]$ is enough!

Lucky in this case.
Shortest path LP: what are the constraints?

Path $s \rightarrow t$ is connected:

$$
\sum_{u \rightarrow v} x_{uv} - \sum_{v \rightarrow w} x_{vw} = 0 \forall v \neq \{s, t\} \\
= 1 \ [v = t] \\
= -1 \ [v = s]
$$

Also: $x_{uv} \in \{0, 1\}$?

$x_{uv} \in [0, 1]$ is enough!

Lucky in this case.
Shortest path LP: alternative formulation

Define $d_u$ as $s \rightarrow u$ distance

$$d_s = 0$$

for each $u \rightarrow v$: $d_v \leq d_u + w_{uv}$

$$d_c \leq 2$$

$$d_g \leq d_c + 8$$

$$d_g \leq d_a + 2$$

...
Shortest path LP: alternative formulation

Define $d_u$ as $s \to u$ distance

$d_s = 0$

for each $u \to v$: $d_v \leq d_u + w_{uv}$

minimize $d_t$

Incorrect! We have to maximize.
This has a neat physical interpretation!

Now pull S and T apart as far as possible
This has a neat physical interpretation!

SP algorithms and protocols with Prof. Perrig!
Linear programming: a nice, generic tool

Efficient solvers exist, so your task is simply writing LPs

- CPLEX, Gurobi

Framing “word problems” is non-trivial

- Even correctness can be difficult to see
- Some formulations are faster than others
- Think of it as a (highly restricted) programming language

Theory for LPs, efficient algorithms not in this course