# Communications Network Design

## lecture 05

Matthew Roughan
<matthew.roughan@adelaide.edu.au>

Discipline of Applied Mathematics
School of Mathematical Sciences
University of Adelaide

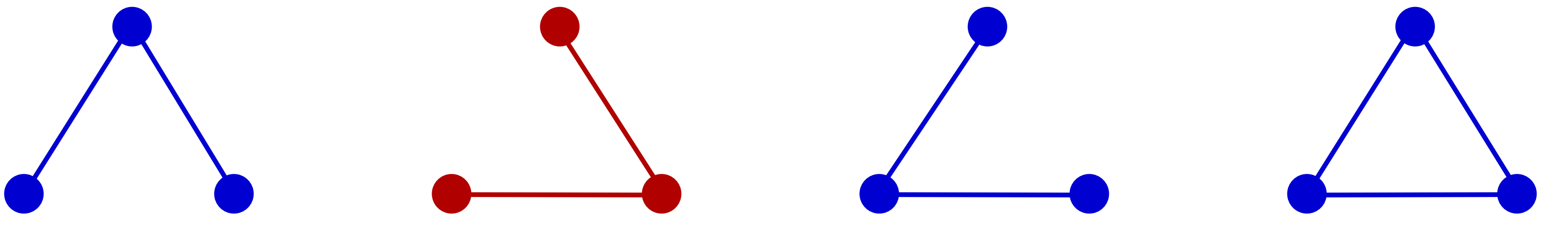
March 19, 2009

# Optimization: 1000 foot view

Its helpful for us to talk a little about optimization techniques before we start. We also presenta little notation.

# Simple example

Three node network has three acceptible designs:



| Cost | 2.2 | 2.0 | 2.4 | 3.1 |
| --- | --- | --- | --- | --- |

- 4 possible network designs
  - associated costs have been worked out for each
- easy to choose the second network as the cheapest

# Bigger problems

- Node with $N$ nodes
  - for small $N$ we can evaluate all designs, and choose the best
- But $2^{N(N-1)/2}$ possible network designs
  - some aren't practical
  - but we still have to check that
- Even for $N = 20$ we can't evaluate all of these
  - at least not in the life-time of the Universe

# Optimization

Optimization is about building automated methods for finding optima of such problems

- needs to work quickly (enough)
  - planning horizon
  - management requirements
  - size of the problem
- ideally attains provably best solution
  - can't always do this (in reasonable time)
  - our problems are often NP-hard
  - need heuristic (rule of thumb) methods
  - often this isn't a big issue:
    - look at all the approximation we already made

# High-level view of problems we have

| problem | planning horizon | typical goal | variables | common constraints |
|---|---|---|---|---|
| network design | months | min. cost | capacities | traffic |
| traffic engineering | days | min. congestion | weights | network design |
| routing | seconds | min. delay | routes | weights + network |

we'll start from back to front (with routing)

# Meta-heuristics

High-level view of heuristics

- Greedy
- Gradient Descent
- Branch and Bound
- Simulated Annealling
- Genetic Algorithms

We will often need to use specific properties of a problem in order to make the above practical.

# Simple Set Notation

| | | | |
|---|---|---|---|
| membership | $\omega \in U$ | $=$ | $\omega$ is in $U$ |
| subset | $L \subset U$ | $=$ | if $\omega \in L$, then $\omega \in U$ |
| intersection | $L \cap U$ | $=$ | $\{\omega \mid \omega \in L$ and $\omega \in U\}$ |
| union | $L \cup U$ | $=$ | $\{\omega \mid \omega \in L$ or $\omega \in U\}$ |
| set difference | $L \backslash U$ | $=$ | $\{\omega \mid \omega \in L$ and $\omega \notin U\}$ |
| empty set | $\phi$ | $=$ | $\{\}$ |
| for all | $\forall \omega$ | | do something for all $\omega$ |
| count | $\lvert U \rvert$ | $=$ | the number of elements of $U$ |

# Optimization Notation

We can write an optimization problem different ways

- (1) write it out in full

$$\begin{array}{lrcl} \text{maximize} & \mathbf{c}^T\mathbf{x} & & \\ \text{subject to} & A\mathbf{x} & \leq & \mathbf{b} \\ & \mathbf{x} & \geq & 0 \end{array}$$

- (2) shorter form

$$\max\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}$$

- (3) even shorter form

$$\underset{A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0}{\operatorname{argmax}} \; \mathbf{c}^T\mathbf{x}$$

# Other Notation

I usually use

- lower case for scalars, e.g., $x$
- lower-case boldface for (column) vectors, e.g., $\mathbf{x}$
- upper-case for matrices, e.g., $A$

When I write $\mathbf{x} < \mathbf{b}$ I mean every element of $\mathbf{x}$ is less than its corresponding element in $\mathbf{b}$, so

$$x_i < b_i, \ \ \forall i$$

and similarly for fo relational operators, e.g., $\leq$, $\geq$, ...

# References