Algebraic Path Finding

Timothy G. Griffin
tgg22@cam.ac.uk

Computer Laboratory, University of Cambridge

Summer School on Formal Methods and Networks
June 10-14, 2013
Cornell University
Ithaca, NY, USA

Formal Methods *and* Networks? Seriously?

In networking, formal methods have often been associated with the Bellhead[1] tradition

Bellheads	Netheads
Smart Network	Dumb Network
Dumb terminals	Smart terminals
Ration scarcity	Liberate abundance
QoS guarantees	Best effort
Proprietary lock-in	Open standards
Premium service	Flat rates or free
Stalin	Bakunin

[1] See *Netheads vs Bellheads* by Steve G. Steinberg, 1996

Some approaches to bridge the gap

Build tools/libraries that

- enhance the ability to engage in the Internet culture
- encourage community-based development of open-source systems
- embrace the open-ended exploration of design spaces that are only partially understood.

Examples: Alloy, Frenetic, Isis2 ...

Domain-specific languages are a promising direction

- Raise level of abstraction
- Think of LEX and YACC : you can make productive use of such tools without knowing much about the underlying automata theory

THE routing problem does not exist

A large, ever growing, family of problems ...

- Flows in networks. Combinatorial Optimization, Linear Programming
  - *Network Flows: Theory, Algorithms, and Applications.* Ahuja, Magnanti, Orlin. 1993.
- Transportation/Road Networks
  - *Urban transportation networks: equilibrium analysis with mathematical programming methods.* Sheffi. 1985. (Available online at http://sheffi.mit.edu/urban-transportation.)
- Peer-to-Peer networks
  - *P2P Networking and Applications.* Buford, Yu, Lua. 2008.
- Wireless Networks
  - *Routing for Wireless Multi-Hop Networks.* Abdel Hamid, Hassanein, Takahara. 2013. (ad hoc, sensor, mesh, and vehicular networking)
- ...

There are similarities here, but also many problems and techniques that are domain-specific. The literature is vast.

Current (layer 3 infrastructure) Routing Protocols?

We will abstract away from the straightjackets of
$\{\text{RIP}, \text{EIGRP}, \text{OSPF}, \text{IS-IS}, \text{EIGRP}, \text{BGP}\}$ ...

... and see that these protocols are actually solving matrix equations!

- Not a mainstream point of view, to put it mildly
- Allows us to explore the *network-wide problem* being solved independent of the *algorithm(s) and protocols* used to solve it.
  - Algorithms can be distributed, centralized, hybrid, ...
  - ... and the algorithms are variants of familiar ones (Dijkstra's, Bellman-Ford, ...)

The Magic is in the Metric

- Imagine a world where a network engineer can tailor routing metrics to the unique needs of their network.
  - Problem : network engineers don't like to prove theorems ...
- *Metarouting*: towards a domain-specific language for the high-level specification/verification/implementation of path metrics.
  - Theorems are automatically derived, much like types in a programming language.
  - Warning : still very much work-in-progress

The Cisco appraoch to tailoring metrics (Part I)

Selecting EIGRP[2] parameters $K_1, \cdots, K_6$ allows an operator to define a new metric.

EIGRP path metric

$$\begin{aligned} K_1\mathrm{BW} &+ \left(\frac{K_2\mathrm{BW}}{256-\mathrm{LOAD}} + K_3\mathrm{DELAY}\right)\left[\frac{K_5}{\mathrm{RELIABILITY}+K_4}\right] \\ &+ K_6(\mathrm{JITTER}+\mathrm{ENERGY}) \end{aligned}$$

Here $a[0] = a$, and $a[b] = a \times b$ for $b \neq 0$.

[2] EIGRP is no longer proprietary! See `draft-savage-eigrp-00.txt` (Feb. 2013)

The Cisco appraoch to tailoring metrics (Part II)

Inter-Domain Routing
Internet-Draft
Intended status: Standards Track
Expires: November 21, 2013

A. Retana
Cisco Systems, Inc.
R. White
Verisign
May 20, 2013

BGP Custom Decision Process
draft-ietf-idr-custom-decision-03

Abstract

The BGP specification defines a Decision Process for installation of routes into the Loc-RIB. This process takes into account an extensive series of path attributes, which can be manipulated to indicate preference for specific paths. It is cumbersome (if at all possible) for the end user to define policies that will select, after partial comparison, a path based on subjective local (domain and/or node) criteria.

This document defines a new Extended Community, called the Cost Community, which may be used in tie breaking during the best path selection process. The end result is a local custom decision process.

Routing vs Forwarding

Routing

A *network wide* process of selecting paths compliant with policy and network conditions. Policy often expressed with *path metrics*.

Forwarding

The local, node-by-node (switch-by-switch) treatment of traffic. Implemented in *forwarding tables* populated by network-wide routing protocols.

Research Problem: Can we avoid *data plane verification* by automaically populating match tables correctly? Need to expand our notion of metric to capture the full richness of matching ...

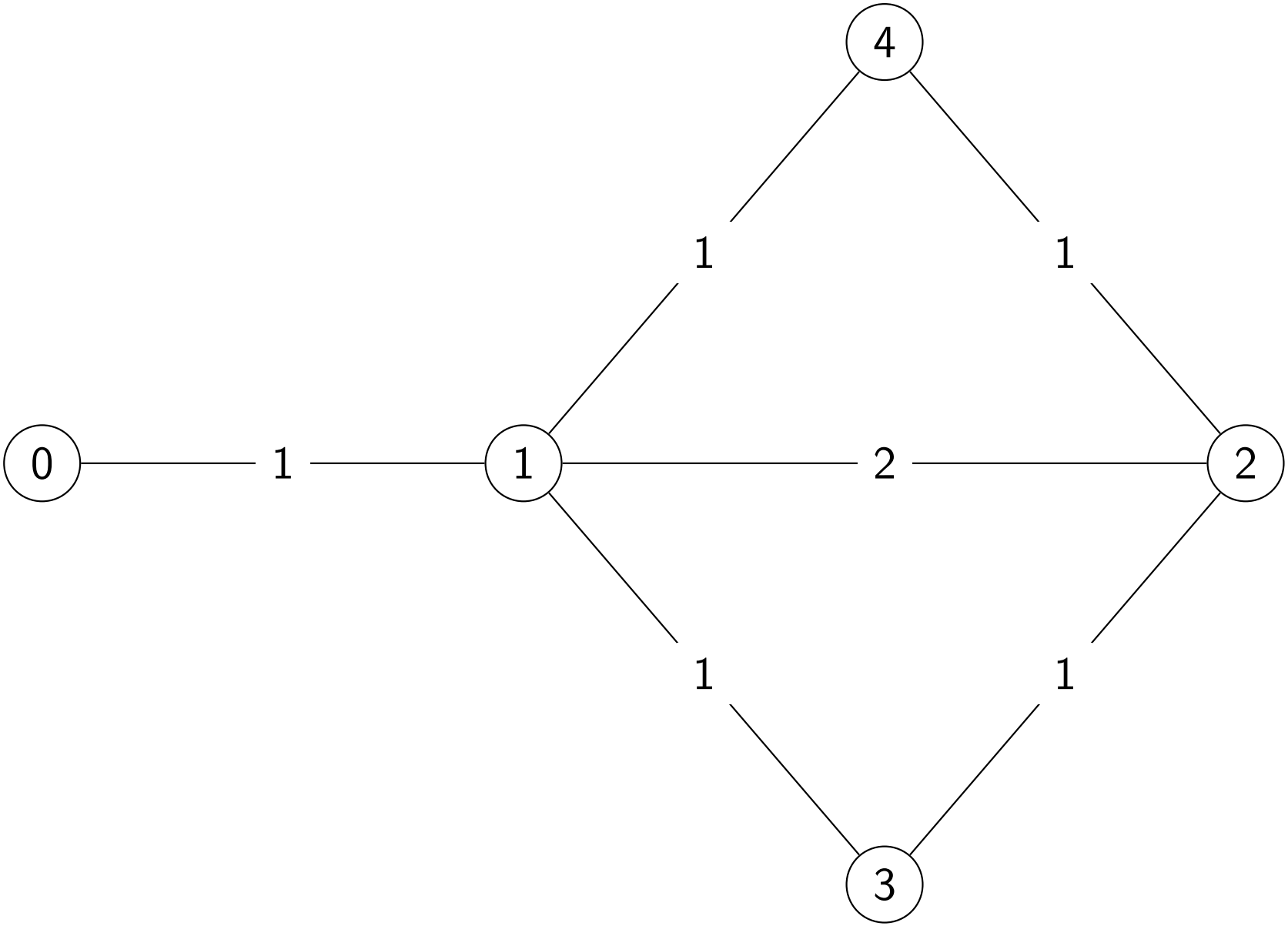
Outline of Topics

- I) A puzzle concerning the paths implicit in the Dijkstra's Algorithm and Bellman-Ford Algorithms[3]
- II) An algebraic framework for path metrics and matrix equations
- III) Current routing protocols only scratch the surface of what can be expressed!
- IV) Beyond Routing — we can do so much more within this algebraic fracework!
  - Example: calculate set of all shared risk groups whose failure could disconnect two given nodes[4].
- V) Metarouting progress report

[3] The puzzle is from current work with Seweryn Dynerowicz, University of Namur.

[4] Ongoing work with Paul Barford, University of Wisconsin, Madison.

Let's start with shortest paths



Can represent a problem instance with an adjacency matrix

$$\mathbf{A} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} \infty & 1 & \infty & \infty & \infty \\ 1 & \infty & 2 & 1 & 1 \\ \infty & 2 & \infty & 1 & 1 \\ \infty & 1 & 1 & \infty & \infty \\ \infty & 1 & 1 & \infty & \infty \end{bmatrix} \end{array}$$

But what problem are we solving?

Classic: globally optimal path weights

We want to find $\mathbf{A}^*$ such that

$$\mathbf{A}^*(i,\ j) = \min_{p \in P(i,\ j)} w(p),$$

where $P(i,\ j)$ is the set of all paths from i to j.

In the example:

$$\mathbf{A}^* = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 3 & 2 & 2 \\ 1 & 0 & 2 & 1 & 1 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 2 & 1 & 1 & 2 & 0 \end{bmatrix} \end{array}$$

An Algorithm: Dijkstra's

Input : adjacency matrix $\mathbf{A}$ and source vertex $i \in V$,
Output : the i-th row of $\mathbf{R}$, where $\mathbf{R}(i,\ j)$ is the shortest distance from i to j in the graph represented by $\mathbf{A}$.

(1) **for each** $q \in V$ **do** $\mathbf{R}(i,\ q) \leftarrow \infty$
(2) $S \leftarrow \{\}$; $\mathbf{R}(i,\ i) \leftarrow 0$
(3) **while** $S \neq V$ **do**
(4) find $q \in V - S$ such that $\mathbf{R}(i,\ q)$ is minimal
(5) $S \leftarrow S \cup \{q\}$
(6) **for each** $j \in V - S$ **do**
(7) $\mathbf{R}(i,\ j) \leftarrow \mathbf{R}(i,\ j)\ \min\ (\mathbf{R}(i,\ q) + \mathbf{A}(q,\ j))$

Run this $|V|$ times to get $\mathbf{R} = \mathbf{A}^*$.

But wait! What about the PATHS???

A bit of notation

Assume X and Y are sets of paths over E.

$$X \diamond Y \quad \equiv \quad \{pq \mid p \in X,\ q \in Y\}$$

Dijkstra's Algorithm Augmented With Paths

Input : adjacency matrix $\mathbf{A}$ and source vertex $i \in V$,
Output : the i-th row of $\mathbf{R}$ as before. Now with $\mathbf{P}(i,\ j)$ the set of **all** paths from i to j of distance $\mathbf{R}(i,j)$

(1) **for each** $q \in V$ **do** $\mathbf{R}(i,\ q) \leftarrow \infty$; $\mathbf{P}(i,\ q) \leftarrow \{\}$
(2) $S \leftarrow \{\}$; $\mathbf{R}(i,\ i) \leftarrow 0$; $\mathbf{P}(i,\ i) \leftarrow \{\epsilon\}$
(3) **while** $S \neq V$ **do**
(4) find $q \in V - S$ such that $\mathbf{R}(i,\ q)$ is minimal
(5) $S \leftarrow S \cup \{q\}$
(6) **for each** $j \in V - S$ **do**
(7) **if** $\mathbf{R}(i,\ j) = \mathbf{R}(i,\ q) + \mathbf{A}(q,\ j)$
(8) **then** $\mathbf{P}(i,\ j) \leftarrow \mathbf{P}(i,\ j) \cup (\mathbf{P}(i,\ q) \diamond \{(q,\ j)\})$
(9) **else if** $\mathbf{R}(i,\ j) > \mathbf{R}(i,\ q) + \mathbf{A}(q,\ j)$
(10) **then** $\mathbf{R}(i,\ j) \leftarrow \mathbf{R}(i,\ q) + \mathbf{A}(q,\ j)$;
(11) $\mathbf{P}(i,\ j) \leftarrow \mathbf{P}(i,\ q) \diamond \{(q,\ j)\}$

Solution(s)

$$\mathbf{R} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 3 & 2 & 2 \\ 1 & 0 & 2 & 1 & 1 \\ 3 & 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 2 & 1 & 1 & 2 & 0 \end{bmatrix} \end{array}$$

$$\begin{aligned} \mathbf{P}(0,0) &= \{\epsilon\} \\ \mathbf{P}(0,1) &= \{(0,1)\} \\ \mathbf{P}(0,2) &= \{(0,1,2),\ (0,1,3,2),\ (0,1,4,2)\} \\ \mathbf{P}(2,1) &= \{(2,1),\ (2,3,1),\ (2,4,1)\} \\ \mathbf{P}(2,0) &= \{(2,1,0),\ (2,3,1,0),\ (2,4,1,0)\} \\ &\vdots \quad \vdots \quad \vdots \end{aligned}$$

Note : could implement hop-by-hop ECMP.

Let's enrich the metric to *Widest Shortest-Paths*

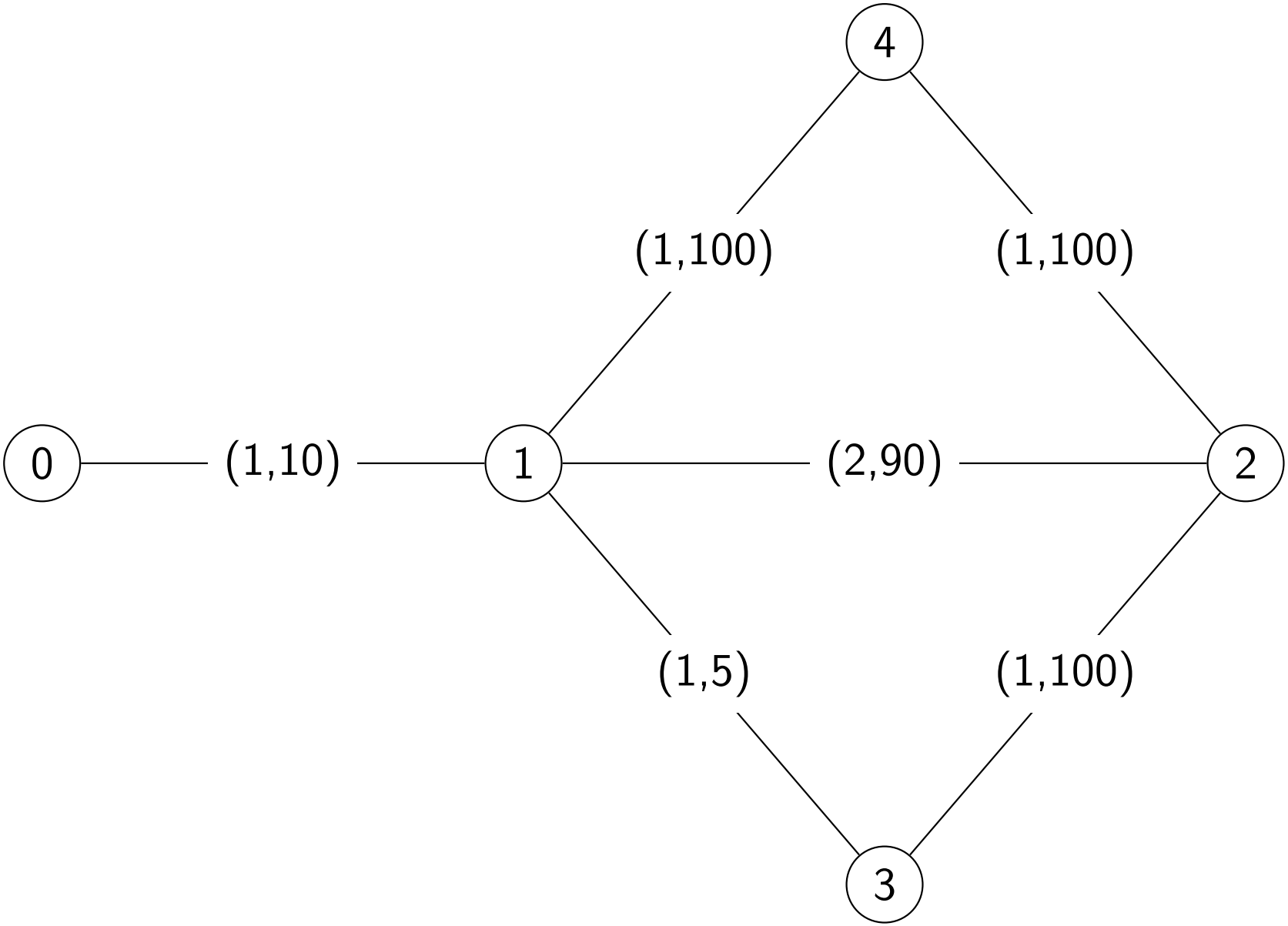
shortest paths	widest shortest paths
$\mathbb{N} \cup \{\infty\}$	$S_{\text{wsp}} \equiv (\mathbb{N} \times \{1, \ldots, \top\}) \cup \{\infty\}$
min	$\circ$
$+$	$\bullet$
0	$(0, \top)$

Can replace $+$ by $\bullet$ and min by $\circ$ in both Dijkstra and Bellman-Ford.

$$(a,\ b) \circ (c,\ d) = \begin{cases} (a,\ \ b \text{ max } d) & (a = c) \\ (a,\ b) & (a < c) \\ (c,\ d) & (c < a) \end{cases}$$

$$(a,\ b) \bullet (c,\ d) = (a + c,\ b \text{ min } d)$$

Add bandwith to link weights



Weights are globally optimal

Widest shortest-path weights computed by Dijkstra and Bellman-Ford

$$\mathbf{R} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} (0,\top) & (1,10) & (3,10) & (2,5) & (2,10) \\ (1,10) & (0,\top) & (2,100) & (1,5) & (1,100) \\ (3,10) & (2,100) & (0,\top) & (1,100) & (1,100) \\ (2,5) & (1,5) & (1,100) & (0,\top) & (2,100) \\ (2,10) & (1,100) & (1,100) & (2,100) & (0,\top) \end{bmatrix} \end{array}$$

Four optimal paths of weight $(3, 10)$. Do our algorithms find all of them?

$$\begin{aligned} \mathbf{P}_{\text{optimal}}(0,2) &= \{(0,1,2),\ (0,1,4,2)\} \\ \mathbf{P}_{\text{optimal}}(2,0) &= \{(2,1,0),\ (2,4,1,0)\} \end{aligned}$$

Surprise!

Four **optimal** paths of weight $(3, 10)$

$$\begin{aligned} \mathbf{P}_{\text{optimal}}(0,2) &= \{(0,1,2),\ (0,1,4,2)\} \\ \mathbf{P}_{\text{optimal}}(2,0) &= \{(2,1,0),\ (2,4,1,0)\} \end{aligned}$$

Paths computed by **Dijkstra**

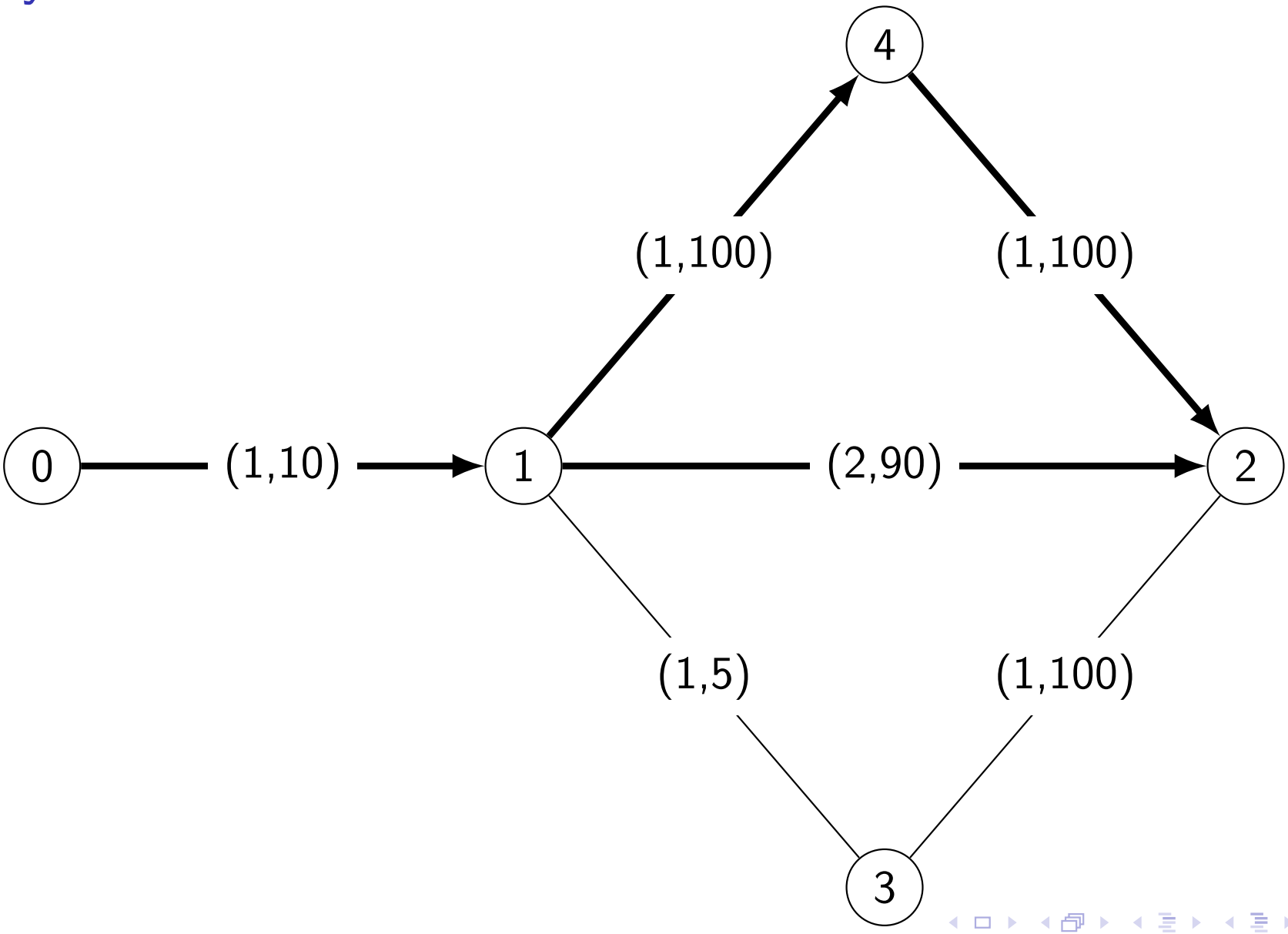
$$\begin{aligned} \mathbf{P}_{\text{Dijkstra}}(0,2) &= \{(0,1,2),\ (0,1,4,2)\} \\ \mathbf{P}_{\text{Dijkstra}}(2,0) &= \{(2,4,1,0)\} \end{aligned}$$

Notice that 0's paths cannot both be implemented with next-hop forwarding since $\mathbf{P}_{\text{Dijkstra}}(1,2) = \{(1,4,2)\}$.

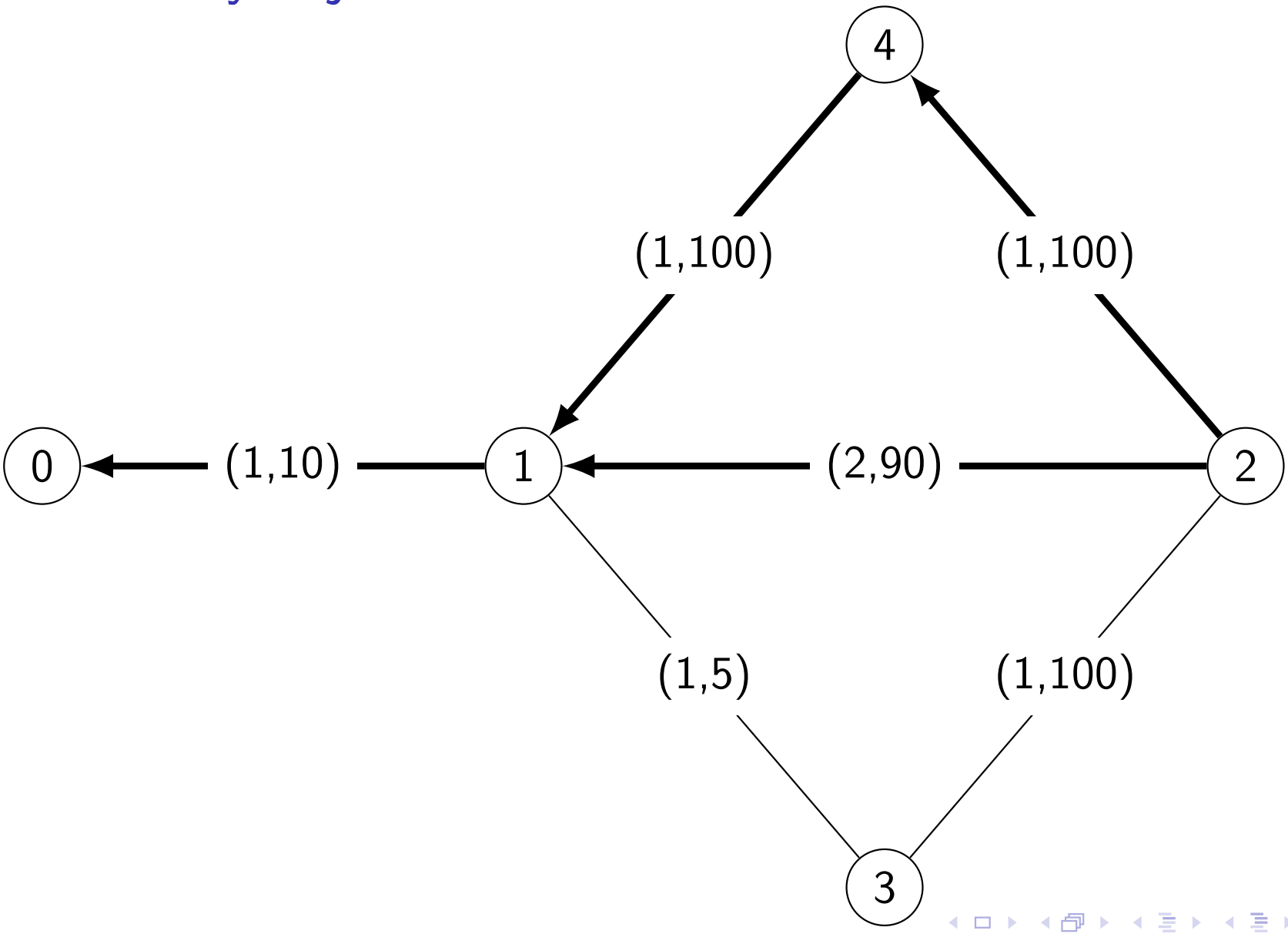
Paths computed by **Bellman-Ford**

$$\begin{aligned} \mathbf{P}_{\text{Bellman}}(0,2) &= \{(0,1,4,2)\} \\ \mathbf{P}_{\text{Bellman}}(2,0) &= \{(2,1,0),\ (2,4,1,0)\} \end{aligned}$$

Optimal paths from 0 to 2. Computed by Dijkstra but not by Bellman-Ford



Optimal paths from 2 to 1. Computed by Bellman-Ford but not by Dijkstra



What is going on here???

Help!

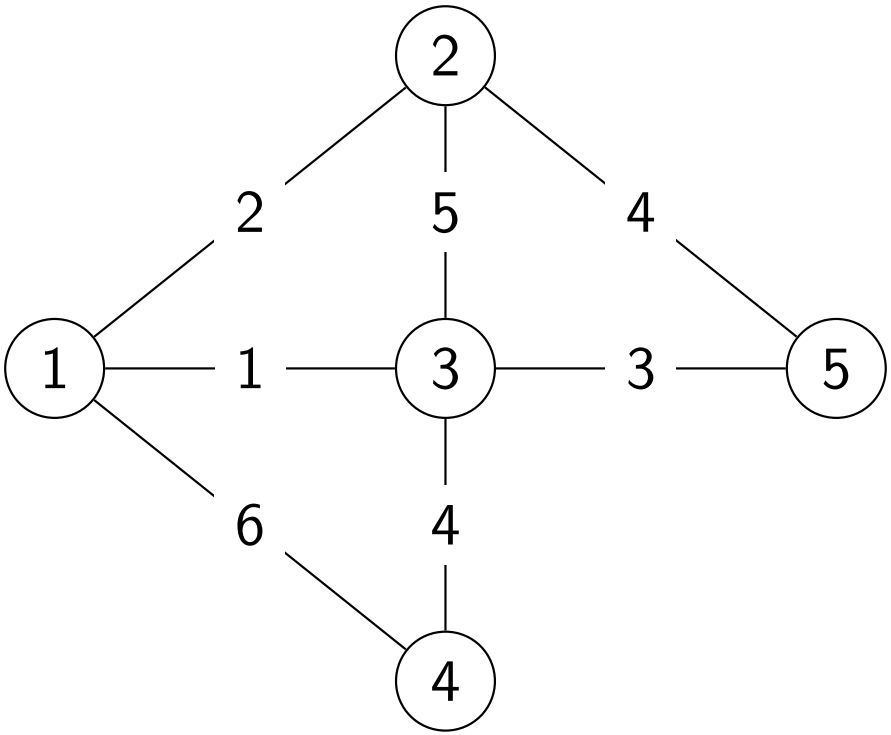
- Are the algorithms broken?
- Is the new metric broken?

Hint

We will see that we are actually solving three *distict* problems here.

We will see this clearly once we *fold the paths into the metric* and understand *what problems our algorithms can actually solve*.

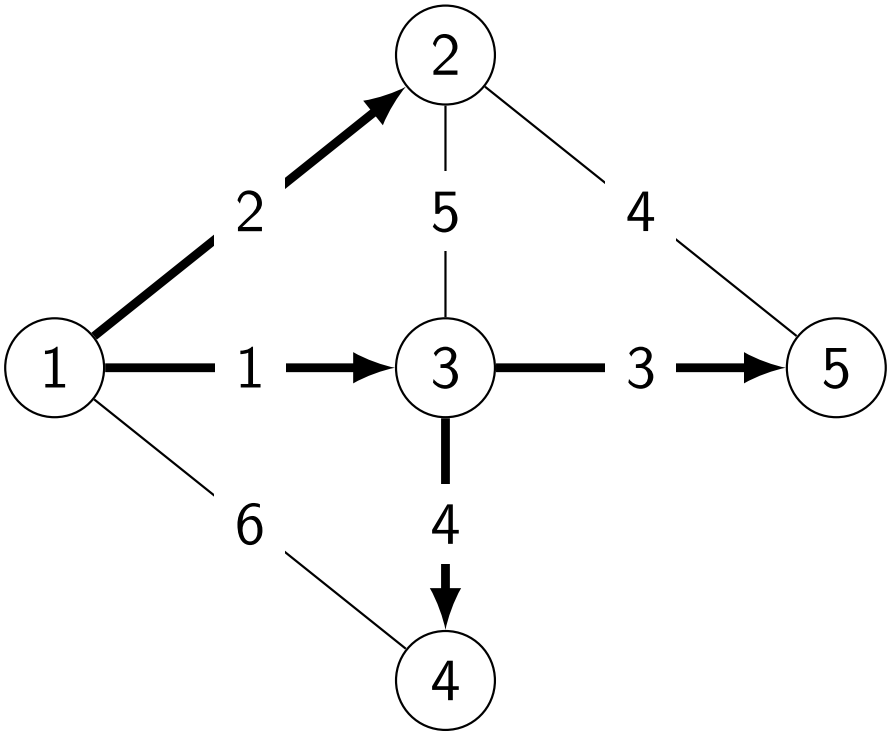
Shortest paths example, $(\mathbb{N}^\infty,\ \min,\ +)$



The adjacency matrix

$$\mathbf{A} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \infty & 2 & 1 & 6 & \infty \\ 2 & \infty & 5 & \infty & 4 \\ 1 & 5 & \infty & 4 & 3 \\ 6 & \infty & 4 & \infty & \infty \\ \infty & 4 & 3 & \infty & \infty \end{bmatrix} \end{array}$$

Shortest paths example, $(\mathbb{N}^\infty,\ \min,\ +)$



Bold arrows indicate the shortest-path tree rooted at 1.

The routing matrix

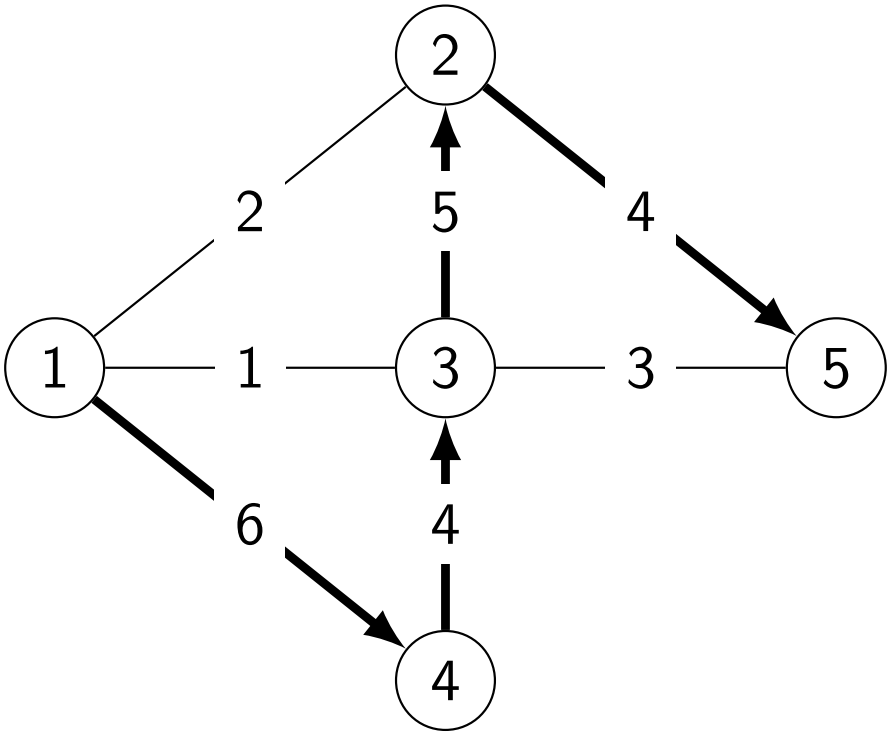
$$\mathbf{A}^* = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 5 & 4 \\ 2 & 0 & 3 & 7 & 4 \\ 1 & 3 & 0 & 4 & 3 \\ 5 & 7 & 4 & 0 & 7 \\ 4 & 4 & 3 & 7 & 0 \end{bmatrix} \end{array}$$

Matrix $\mathbf{A}^*$ solves this global optimality problem:

$$\mathbf{A}^*(i,\ j) = \min_{p \in P(i,\ j)} w(p),$$

where $P(i,\ j)$ is the set of all paths from i to j.

Widest paths example, $(\mathbb{N}^\infty,\ \max,\ \min)$



Bold arrows indicate the widest-path tree rooted at 1.

The routing matrix

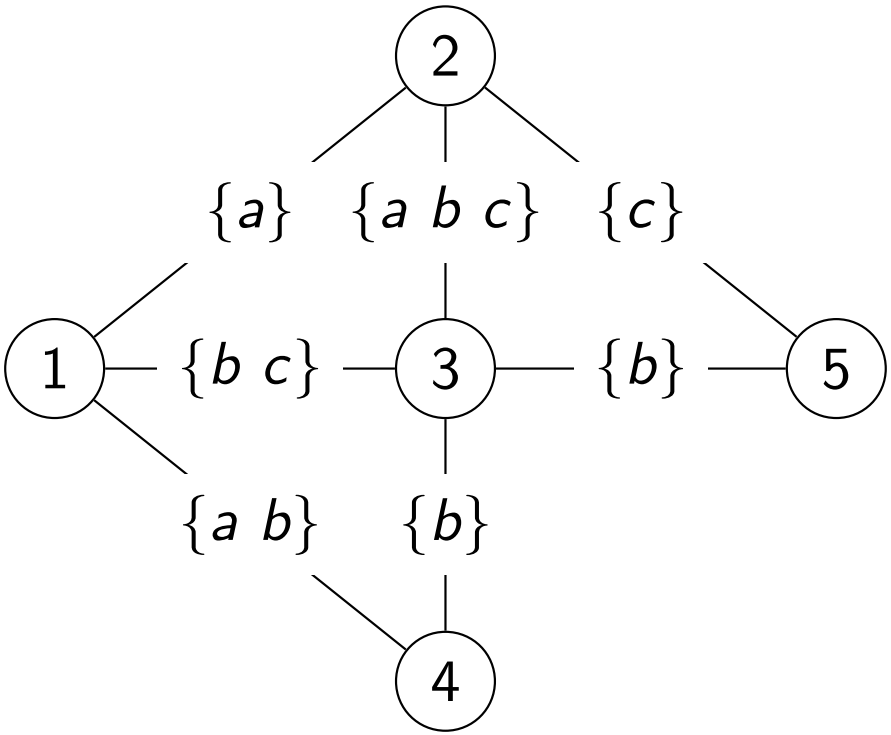
$$\mathbf{A}^* = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \infty & 4 & 4 & 6 & 4 \\ 4 & \infty & 5 & 4 & 4 \\ 4 & 5 & \infty & 4 & 4 \\ 6 & 4 & 4 & \infty & 4 \\ 4 & 4 & 4 & 4 & \infty \end{bmatrix} \end{array}$$

Matrix $\mathbf{A}^*$ solves this global optimality problem:

$$\mathbf{A}^*(i,\ j) = \max_{p \in P(i,\ j)} w(p),$$

where $w(p)$ is now the minimal edge weight in p.

Unfamiliar example, $(2^{\{a,\ b,\ c\}},\ \cup,\ \cap)$



We want a Matrix $\mathbf{A}^*$ to solve this global optimality problem:

$$\mathbf{A}^*(i,\ j) = \bigcup_{p \in P(i,\ j)} w(p),$$

where $w(p)$ is now the intersection of all edge weights in p.

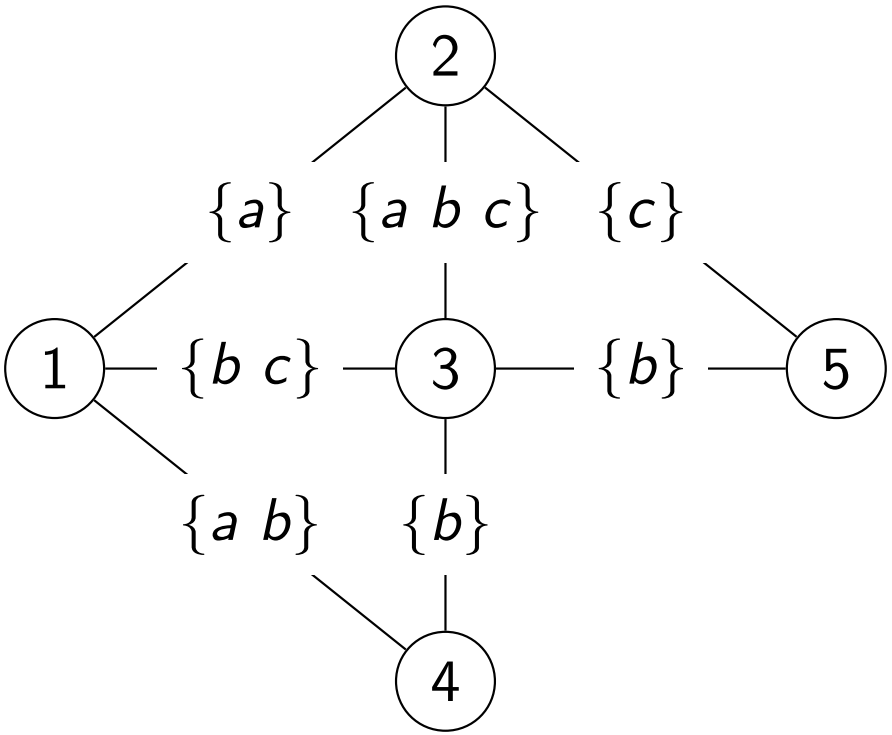
For $x \in \{a,\ b,\ c\}$, interpret $x \in \mathbf{A}^*(i,\ j)$ to mean that there is at least one path from i to j with x in every arc weight along the path.

Unfamiliar example, $(2^{\{a,\ b,\ c\}},\ \cup,\ \cap)$

The matrix $\mathbf{A}^*$

	1	2	3	4	5
1	$\{a\ b\ c\}$	$\{a\ b\ c\}$	$\{a\ b\ c\}$	$\{a\ b\}$	$\{b\ c\}$
2	$\{a\ b\ c\}$	$\{a\ b\ c\}$	$\{a\ b\ c\}$	$\{a\ b\}$	$\{b\ c\}$
3	$\{a\ b\ c\}$	$\{a\ b\ c\}$	$\{a\ b\ c\}$	$\{a\ b\}$	$\{b\ c\}$
4	$\{a\ b\}$	$\{a\ b\}$	$\{a\ b\}$	$\{a\ b\ c\}$	$\{b\}$
5	$\{b\ c\}$	$\{b\ c\}$	$\{b\ c\}$	$\{b\}$	$\{a\ b\ c\}$

Another unfamiliar example, $(2^{\{a,\ b,\ c\}},\ \cap,\ \cup)$



We want matrix $\mathbf{A}^*$ to solve this global optimality problem:

$$\mathbf{A}^*(i,\ j) = \bigcap_{p \in P(i,\ j)} w(p),$$

where $w(p)$ is now the union of all edge weights in p.

For $x \in \{a,\ b,\ c\}$, interpret $x \in \mathbf{A}^*(i,\ j)$ to mean that every path from i to j has at least one arc with weight containing x.

Another unfamiliar example, $(2^{\{a,\ b,\ c\}},\ \cap,\ \cup)$

The matrix $\mathbf{A}^*$

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\left[\begin{array}{ccccc}
\{\} & \{\} & \{b\} & \{b\} & \{\} \\
\{\} & \{\} & \{b\} & \{b\} & \{\} \\
\{b\} & \{b\} & \{\} & \{b\} & \{b\} \\
\{b\} & \{b\} & \{b\} & \{\} & \{b\} \\
\{\} & \{\} & \{b\} & \{b\} & \{\}
\end{array}\right]
\end{array}
$$

Semirings

A generalization of your favorite ring $(\mathbb{R}, +, \times, 0, 1)$ from linear algebra.

name	S	$\oplus$	$\otimes$	$\overline{0}$	$\overline{1}$	possible routing use
sp	$\mathbb{N}^\infty$	min	$+$	∞	0	minimum-weight routing
bw	$\mathbb{N}^\infty$	max	min	0	∞	greatest-capacity routing
rel	$[0,\ 1]$	max	$\times$	0	1	most-reliable routing
use	$\{0,\ 1\}$	max	min	0	1	usable-path routing
	2^W	$\cup$	$\cap$	$\{\}$	W	shared link attributes?
	2^W	$\cap$	$\cup$	W	$\{\}$	shared path attributes?

A wee bit of notation

Symbol	Interpretation
$\mathbb{N}$	Natural numbers (starting with zero)
$\mathbb{N}^\infty$	$\mathbb{N} \cup \{\infty\}$
$\overline{0}$	Identity for $\oplus$
$\overline{1}$	Identity for $\otimes$

Those metrics are all **Semirings**

- $\oplus$ and $\otimes$ are associative
- $\oplus$ is commutative
- $\overline{0}$ is the identity for $\oplus$
- $\overline{1}$ is the identity for $\otimes$
- $\overline{0}$ is an annihilator for $\otimes$, $a \otimes \overline{0} = \overline{0} \otimes a = \overline{0}$

and left- and right-distributivity hold,

$$\begin{aligned} \text{LD} &: a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c) \\ \text{RD} &: (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \end{aligned}$$

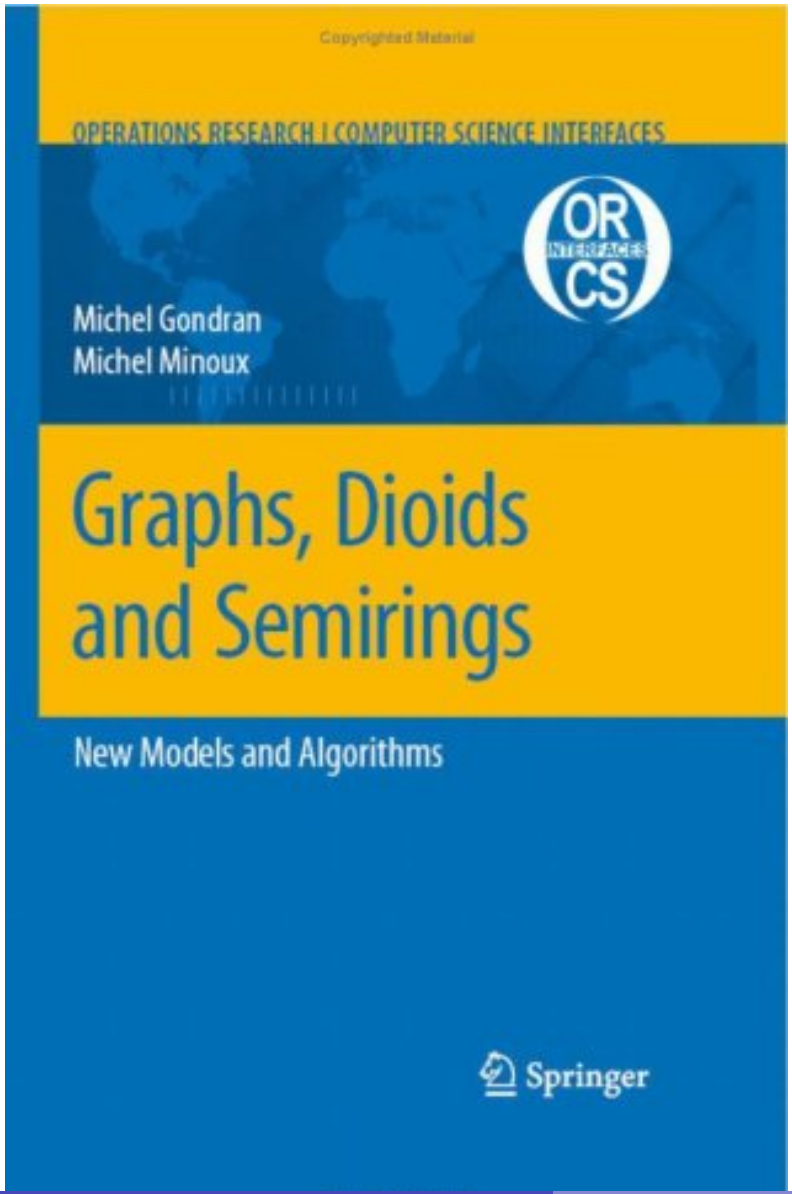
Our examples so far also have

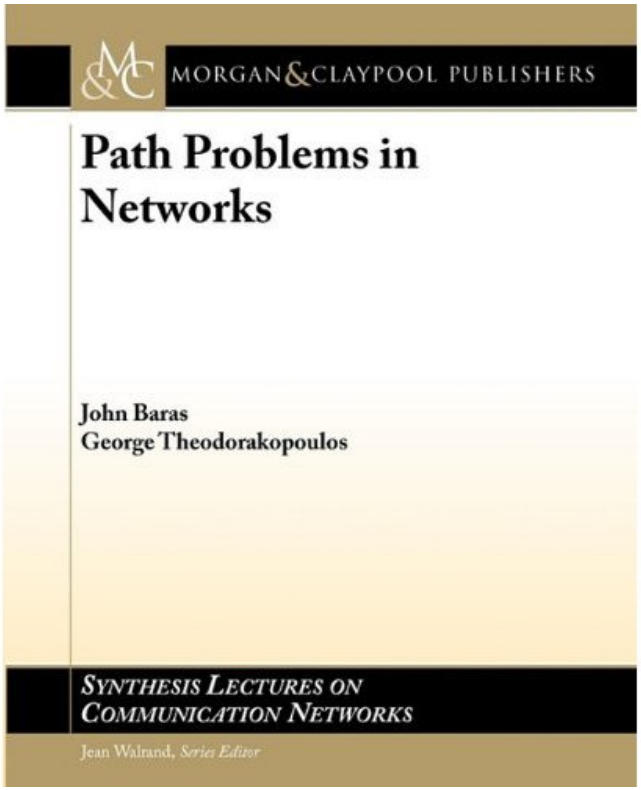
$$\begin{aligned} \text{IDEM} &: \overline{a} \oplus a = a \\ \text{ADD.ANN} &: \overline{1} \oplus a = \overline{1} \end{aligned}$$

And some of them have *selectivity*

$$\text{SEL} : a \oplus b \in \{a,\ b\}$$

Recommended Reading





Natural Orders

Definition (Natural orders)

Let $(S,\ \oplus)$ be a semigroup.

$$a \leq^{L}_{\oplus} b \quad \equiv \quad a = a \oplus b$$
$$a \leq^{R}_{\oplus} b \quad \equiv \quad b = a \oplus b$$

S	$\oplus$	α	ω	$\leq^{\mathrm{L}}_{\oplus}$	$\leq^{\mathrm{R}}_{\oplus}$
$\mathbb{N} \cup \{\infty\}$	min	∞	0	$\leq$	$\geq$
$\mathbb{N} \cup \{\infty\}$	max	0	∞	$\geq$	$\leq$
$\mathcal{P}(W)$	$\cup$	$\{\}$	W	$\supseteq$	$\subseteq$
$\mathcal{P}(W)$	$\cap$	W	$\{\}$	$\subseteq$	$\supseteq$

When S is a ring (when $(S,\ \oplus,\ \overline{0})$ is a group), the orders are trivial. This is why we are using SEMIrings, not rings.

Matrix Semirings

- $(S,\ \oplus,\ \otimes,\ \overline{0},\ \overline{1})$ a semiring
- Define the semiring of $n \times n$-matrices over S : $(\mathbb{M}_n(S),\ \oplus,\ \otimes,\ \mathbf{J},\ \mathbf{I})$

$\oplus$ and $\otimes$

$$(\mathbf{A} \oplus \mathbf{B})(i,\ j) \quad = \quad \mathbf{A}(i,\ j) \oplus \mathbf{B}(i,\ j)$$

$$(\mathbf{A} \otimes \mathbf{B})(i,\ j) \quad = \quad \bigoplus_{1 \leq q \leq n} \mathbf{A}(i,\ q) \otimes \mathbf{B}(q,\ j)$$

J and I

$$\mathbf{J}(i,\ j) \quad = \quad \overline{0}$$

$$\mathbf{I}(i,\ j) \quad = \quad \begin{cases} \overline{1} & (\text{if } i = j) \\ \overline{0} & (\text{otherwise}) \end{cases}$$

$\mathbb{M}_n(S)$ is a semiring!

For example, here is left distribution

$$\mathbf{A} \otimes (\mathbf{B} \oplus \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) \oplus (\mathbf{A} \otimes \mathbf{C})$$

$$\begin{aligned}
& (\mathbf{A} \otimes (\mathbf{B} \oplus \mathbf{C}))(i,\ j) \\
= & \bigoplus_{1 \le q \le n} \mathbf{A}(i,\ q) \otimes (\mathbf{B} \oplus \mathbf{C})(q,\ j) \\
= & \bigoplus_{1 \le q \le n} \mathbf{A}(i,\ q) \otimes (\mathbf{B}(q,\ j) \oplus \mathbf{C}(q,\ j)) \\
= & \bigoplus_{1 \le q \le n} (\mathbf{A}(i,\ q) \otimes \mathbf{B}(q,\ j)) \oplus (\mathbf{A}(i,\ q) \otimes \mathbf{C}(q,\ j)) \\
= & \left(\bigoplus_{1 \le q \le n} \mathbf{A}(i,\ q) \otimes \mathbf{B}(q,\ j)\right) \oplus \left(\bigoplus_{1 \le q \le n} \mathbf{A}(i,\ q) \otimes \mathbf{C}(q,\ j)\right) \\
= & ((\mathbf{A} \otimes \mathbf{B}) \oplus (\mathbf{A} \otimes \mathbf{C}))(i,\ j)
\end{aligned}$$

Note : we only needed left-distributivity on S.

Matrix representation of a weighted graph

- $(S,\ \oplus,\ \otimes,\ \overline{0},\ \overline{1})$ a semiring
- $G = (V,\ E)$ a directed graph
- $w \in E \to S$ a weight function

Path weight

The *weight* of a path $p = i_1, i_2, i_3, \cdots, i_k$ is

$$w(p) = w(i_1,\ i_2) \otimes w(i_2,\ i_3) \otimes \cdots \otimes w(i_{k-1},\ i_k).$$

The empty path ϵ is given the weight $\overline{1}$.

Adjacency matrix **A**

$$\mathbf{A}(i,\ j) = \begin{cases} w(i,\ j) & \text{if } (i,\ j) \in E, \\ \overline{0} & \text{otherwise} \end{cases}$$

Matrix methods

Matrix powers, $\mathbf{A}^k$

$$\mathbf{A}^0 = \mathbf{I}$$

$$\mathbf{A}^{k+1} = \mathbf{A} \otimes \mathbf{A}^k$$

Closure, $\mathbf{A}^*$

$$\mathbf{A}^{(k)} = \mathbf{I} \oplus \mathbf{A}^1 \oplus \mathbf{A}^2 \oplus \cdots \oplus \mathbf{A}^k$$

$$\mathbf{A}^* = \mathbf{I} \oplus \mathbf{A}^1 \oplus \mathbf{A}^2 \oplus \cdots \oplus \mathbf{A}^k \oplus \cdots$$

Note: $\mathbf{A}^*$ might not exist. Why?

Solving (some) equations

If $\mathbf{A}^*$ exists , then $\mathbf{L} = \mathbf{A}^*$ solves the equation

$$\mathbf{L} = \mathbf{AL} \oplus \mathbf{I}$$

and $\mathbf{R} = \mathbf{A}^*$ solves the equation

$$\mathbf{R} = \mathbf{RA} \oplus \mathbf{I}.$$

Hmmmm ....

If we weaken the axioms of the semiring, could it be that we can find examples where $\mathbf{A}^*$, $\mathbf{L}$, and $\mathbf{R}$ exist, but are all distinct?

Left-Local Optimality

Say that $\mathbf{L}$ is a left locally-optimal solution when

$$\mathbf{L} = (\mathbf{A} \otimes \mathbf{L}) \oplus \mathbf{I}.$$

That is, for $i \neq j$ we have

$$\mathbf{L}(i,\ j) = \bigoplus_{q \in V} \mathbf{A}(i,\ q) \otimes \mathbf{L}(q,\ j)$$

- $\mathbf{L}(i,\ j)$ is the best possible value given the values $\mathbf{L}(q,\ j)$, for all out-neighbors q of source i.
- Rows $\mathbf{L}(i,\ \_)$ represents **out-trees <u>from</u>** i (think Bellman-Ford).
- Columns $\mathbf{L}(\_,\ i)$ represents **in-trees <u>to</u>** i.
- Works well with hop-by-hop forwarding from i.

Right-Local Optimality

Say that $\mathbf{R}$ is a right locally-optimal solution when

$$\mathbf{R} = (\mathbf{R} \otimes \mathbf{A}) \oplus \mathbf{I}.$$

That is, for $i \neq j$ we have

$$\mathbf{R}(i,\ j) = \bigoplus_{q \in V} \mathbf{R}(i,\ q) \otimes \mathbf{A}(q,\ j)$$

- $\mathbf{R}(i,\ j)$ is the best possible value given the values $\mathbf{R}(q,\ j)$, for all in-neighbors q of destination j.
- Rows $\mathbf{L}(i,\ \_)$ represents **out-trees <u>from</u>** i (think Dijkstra).
- Columns $\mathbf{L}(\_,\ i)$ represents **in-trees <u>to</u>** i.

With and Without Distributivity

With distributivity

For (bounded) semirings, the three optimality problems are essentially the same — locally optimal solutions are globally optimal solutions.

$$\mathbf{A}^* = \mathbf{L} = \mathbf{R}$$

Without distributivity

It may be that $\mathbf{A}^*$, $\mathbf{L}$, and $\mathbf{R}$ exists but are all distinct.

Health warning : matrix multiplication over structures lacking distributivity is not associative!

A useful method

$$(\text{metric} + \text{complex algorithm}) \rightarrow (\text{complex metrix} + \text{generic algorithm})$$

Can help us understand

- relationship between routing algorithms and forwarding paths.
- best match forwarding.
- areas in OSPF and Levels in ISIS.
- eBGP vs iBGP.
- route redistribution.
- administrative distance.
- ...

Direct Product of Semigroups

Let $(S, \oplus_S)$ and $(T, \oplus_T)$ be semigroups.

Definition (Direct product semigroup)

The direct product is denoted $(S, \oplus_S) \times (T, \oplus_T) = (S \times T, \oplus)$, where $\oplus = \oplus_S \times \oplus_T$ is defined as

$$(s_1, t_1) \oplus (s_2, t_2) = (s_1 \oplus_S s_2,\ \ t_1 \oplus_T t_2).$$

Lexicographic Product of Semigroups

Definition (Lexicographic product semigroup)

Suppose that semigroup $(S, \oplus_S)$ is commutative, idempotent, and selective and that $(T, \oplus_T)$ is a semigroup. The lexicographic product is denoted $(S, \oplus_S) \vec{\times} (T, \oplus_T) = (S \times T, \vec{\oplus})$, where $\vec{\oplus} = \oplus_S \vec{\times} \oplus_T$ is defined as

$$(s_1, t_1) \vec{\oplus} (s_2, t_2) = \begin{cases} (s_1 \oplus_S s_2, t_1 \oplus_T t_2) & s_1 = s_1 \oplus_S s_2 = s_2 \\ (s_1 \oplus_S s_2, t_1) & s_1 = s_1 \oplus_S s_2 \neq s_2 \\ (s_1 \oplus_S s_2, t_2) & s_1 \neq s_1 \oplus_S s_2 = s_2 \end{cases}$$

Lexicographic product of Bi-semigroups

$$(S,\ \oplus_S,\ \otimes_S) \vec{\times} (T,\ \oplus_T,\ \otimes_T) = (S \times T,\ \oplus_S \vec{\times} \oplus_T,\ \otimes_S \times \otimes_T)$$

Theorem

If $\oplus_S$ is commutative, idempotent, and selective, then

$$\mathrm{LD}(S \vec{\times} T) \iff \mathrm{LD}(S) \wedge \mathrm{LD}(T) \wedge (\mathrm{LC}(S) \vee \mathrm{LK}(T))$$

Where

Property	Definition
LD	$\forall a, b, c : c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$
LC	$\forall a, b, c : c \otimes a = c \otimes b \implies a = b$
LK	$\forall a, b, c : c \otimes a = c \otimes b$

Examples

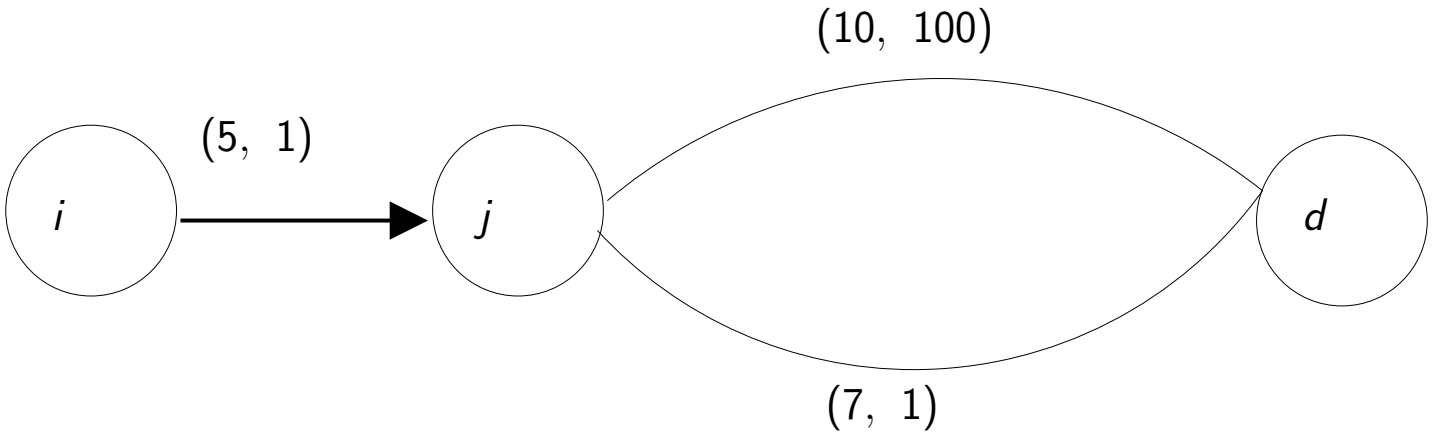
name	S	$\oplus$,	$\otimes$	$\overline{0}$	$\overline{1}$
min_plus	$\mathbb{N}$	min	$+$		0
max_min	$\mathbb{N}$	max	min	0	
PA	2^{E^*}	$\cup$	$\diamond$	$\{\}$	

name	LD	LC	LK
min_plus	Yes	Yes	No
max_min	Yes	No	No
PA	Yes	Yes	No

$\text{LD}(\text{min\_plus} \vec{\times} \text{max\_min})$
$\neg\text{LC}(\text{min\_plus} \vec{\times} \text{max\_min})$
$\text{LD}(\text{min\_plus} \vec{\times} \text{PA})$

$\neg\text{LD}(\text{max\_min} \vec{\times} \text{min\_plus})$
$\neg\text{LD}(\text{min\_plus} \vec{\times} \text{max\_min} \vec{\times} \text{PA})$

Shorest widest paths

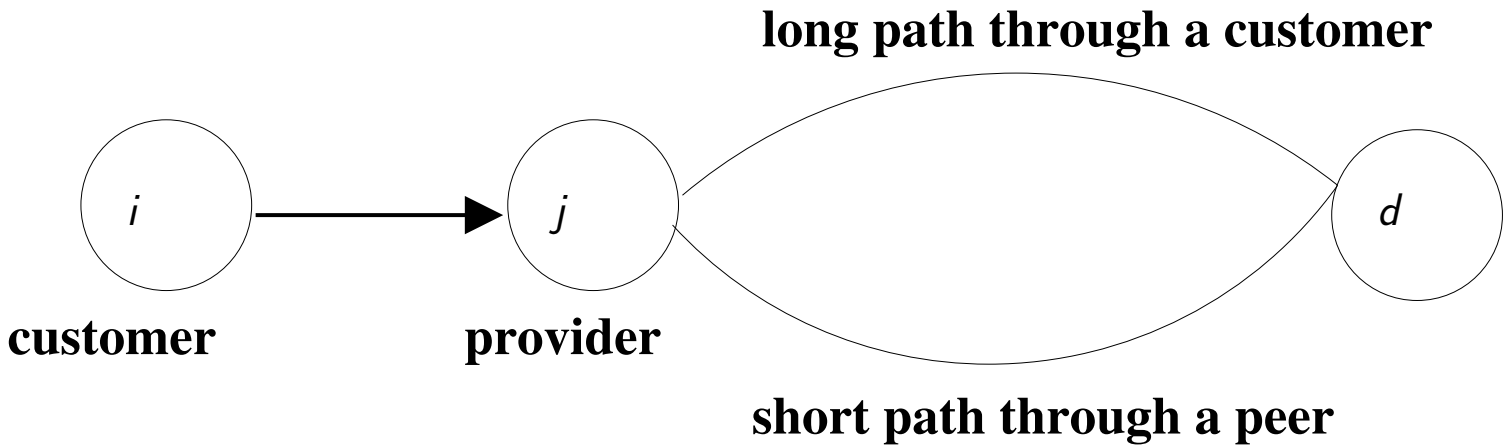


- node j prefers $(10,\ 100)$ over $(7,\ 1)$.
- node i prefers $(5,\ 2)$ over $(5,\ 101)$.

$$(5,\ 1) \otimes ((10,\ 100) \oplus (7,\ 1)) = (5,\ 1) \otimes (10,\ 100) = (5,\ 101)$$

$$((5,\ 1) \otimes (10,\ 101)) \oplus ((5,\ 1) \otimes (7,\ 1)) = (5,\ 101) \oplus (5,\ 2) = (5,\ 2)$$

Something similar from inter-domain routing in the global Internet



- j prefers long path though one of its customers
- i prefers the shorter path

Bellman-Ford can compute left-local solutions

$$\begin{aligned} \mathbf{A}^{[0]} &= \mathbf{I} \\ \mathbf{A}^{[k+1]} &= (\mathbf{A} \otimes \mathbf{A}^{k}) \oplus \mathbf{I}, \end{aligned}$$

- Bellman-ford algorithm must be modified to ensure only loop-free paths are inspected.
- $(S,\ \oplus,\ \overline{0})$ is a commutative, idempotent, and selective monoid,
- $(S,\ \otimes,\ \overline{1})$ is a monoid,
- $\overline{0}$ is the annihilator for $\otimes$,
- $\overline{1}$ is the annihilator for $\oplus$,
- Left strictly inflationarity, $\text{L.S.INF} : \forall a, b : a \neq \overline{0} \implies a < a \otimes b$
- Here $a \leq b \equiv a = a \oplus b$.

Convergence to a unique left-local solution is guaranteed. Currently no polynomial bound is known on the number of iterations required.

Minimal subset of semiring axioms needed right-local Dijkstra : Eliminate all underlined

Semiring Axioms

$$
\begin{array}{rcrcl}
\text{ADD.ASSOCIATIVE} & : & a \oplus (b \oplus c) & = & (a \oplus b) \oplus c \\
\text{ADD.COMMUTATIVE} & : & a \oplus b & = & b \oplus a \\
\text{ADD.LEFT.ID} & : & \overline{0} \oplus a & = & a \\
\underline{\text{MULT.ASSOCIATIVE}} & : & \underline{a \otimes (b \otimes c)} & = & \underline{(a \otimes b) \otimes c} \\
\text{MULT.LEFT.ID} & : & \overline{1} \otimes a & = & a \\
\underline{\text{MULT.RIGHT.ID}} & : & \underline{a \otimes \overline{1}} & = & \underline{a} \\
\underline{\text{MULT.LEFT.ANN}} & : & \underline{\overline{0} \otimes a} & = & \underline{\overline{0}} \\
\underline{\text{MULT.RIGHT.ANN}} & : & \underline{a \otimes \overline{0}} & = & \underline{\overline{0}} \\
\underline{\text{L.DISTRIBUTIVE}} & : & \underline{a \otimes (b \oplus c)} & = & \underline{(a \otimes b) \oplus (a \otimes c)} \\
\underline{\text{R.DISTRIBUTIVE}} & : & \underline{(a \oplus b) \otimes c} & = & \underline{(a \otimes c) \oplus (b \otimes c)}
\end{array}
$$

Additional axioms needed right-local Dijkstra

$$
\begin{array}{rcrcl}
\textsc{add.selective} & : & a \oplus b & \in & \{a,\ b\} \\
\textsc{add.left.ann} & : & \overline{1} \oplus a & = & \overline{1} \\
\textsc{add.right.ann} & : & a \oplus \overline{1} & = & \overline{1} \\
\textsc{right.absorbtion} & : & a \oplus (a \otimes b) & = & a
\end{array}
$$

$\textsc{right.absorbtion}$ gives inflationarity, $\forall a, b : a \leq a \otimes b$.

Routing in Equilibrium. João Luís Sobrinho and Timothy G. Griffin. The 19th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2010).

Using a Link-State approach with hop-by-hop forwarding ...

Need left-local optima!

$$\mathbf{L} = (\mathbf{A} \otimes \mathbf{L}) \oplus \mathbf{I} \qquad \Longleftrightarrow \qquad \mathbf{L}^T = (\mathbf{L}^T \hat{\otimes}^T \mathbf{A}^T) \oplus \mathbf{I}$$

where $\otimes^T$ is matrix multiplication defined with as

$$a \otimes^T b = b \otimes a$$

and we assume left-inflationarity holds, $\text{L.INF} : \forall a, b : a \leq b \otimes a$.

Each node would have to solve the entire "all pairs" problem.

The *metarouting* idea

Defining a language of combinators for algebraic structures.

Starting with an initial set of properties $\mathcal{P}_0$ ...

- Define a language $\mathcal{L}$ of combinators,
- a well-formedness condition $\mathrm{WF}(E)$, for $E \in \mathcal{L}$,
- and a set of properties $\mathcal{P}$, with $\mathcal{P}_0 \subseteq \mathcal{P}$

so that properties are decidable for well-formed expressions:

$$\forall \mathrm{Q} \in P : \forall E \in \mathcal{L} : \mathrm{WF}(E) \implies (\mathrm{Q}([\![E]\!]) \vee \neg \mathrm{Q}([\![E]\!]))$$

(The logic is constructive!) These rules can be turned into bottom-up inference rules, much like typing rules in a programming language.

Difficulty: increase expressive power while preserving decidability ...

The language design methodology

For every combinator C and every property P

find $\mathrm{WF}_{P,C}$ and $\beta_{P,C}$ such that

$$\mathrm{WF}_{P,C}(\vec{a}) \Rightarrow (P(C(\vec{a})) \Leftrightarrow \beta_{P,C}(\vec{a}))$$

... which is then turned into two "bottom-up rules" ...

$$\begin{aligned} \mathrm{WF}_{P,C}(\vec{a}) \wedge \beta_{P,C}(\vec{a}) &\Rightarrow P(C(\vec{a})) \\ \mathrm{WF}_{P,C}(\vec{a}) \wedge \neg\beta_{P,C}(\vec{a}) &\Rightarrow \neg P(C(\vec{a})), \end{aligned}$$

Starting point : Semiring properties, and a few more

These properties are called $\mathcal{P}_0$

Property	Definition
EZ	$\exists \overline{0} : \forall a : a \oplus \overline{0} = \overline{0} \oplus a = a$
EO	$\exists \overline{1} : \forall a : a \otimes \overline{1} = \overline{1} \otimes a = a$
ZA	$\forall a : a \otimes \overline{0} = \overline{0} \otimes a = \overline{0}$
LD	$\forall a, b, c : c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$
RD	$\forall a, b, c : (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$
IA	$\forall a : a \oplus a = a$
SA	$\forall a, b : a \oplus b = a \vee a \oplus b = b$
OA	$\forall a : a \oplus \overline{1} = \overline{1} \oplus a = \overline{1}$
LINF	$\forall a, b : a = a \oplus (b \otimes a)$ (that is, $a \leq^L_\oplus b \otimes a$)
SLINF	$\forall a, b : a = a \oplus (b \otimes a) \neq b \otimes a$ (that is, $a <^L_\oplus b \otimes a$)

Marching towards closure ...

Once we have fixed $\mathcal{P}_0$, grammar for expressions E, definitions of $[\![E]\!]$ and $\mathrm{WF}(E)$.

- $\mathcal{P} := \mathcal{P}_0$
- For each $\mathrm{Q} \in \mathcal{P}$ and each construction $\mathrm{op}(E_1, \cdots E_k)$ in the language, attempt to construct a boolean formula F such that

$$\mathrm{WF}(\mathrm{op}(E_1, \cdots E_k)) \implies (\mathrm{Q}([\![\mathrm{op}(E_1, \cdots E_k)]\!]) \iff F([\![E_1]\!], \cdots [\![E_k]\!]).)$$

- if no new properties are required, then stop.
- otherwise, add the new properties to $\mathcal{P}$ and continue.

Good Luck!

Lexicographic example: closing $\mathcal{P}_0$ introduces auxiliary properties $\mathcal{P}_1$

Properties $\mathcal{P}_1$

Property	Definition
LC	$\forall a, b, c : c \otimes a = c \otimes b \implies a = b$
RC	$\forall a, b, c : a \otimes c = b \otimes c \implies a = b$
LK	$\forall a, b, c : c \otimes a = c \otimes b$
RK	$\forall a, b, c : a \otimes c = b \otimes c$

And closing $\mathcal{P}_1$ introduces auxiliary properties $\mathcal{P}_2$.

$\mathcal{P}_2$

Property	Definition
ANTILEFT	$\forall a, b, : a \otimes b \neq a$
ANTIRIGHT	$\forall a, b, : a \otimes b \neq b$

And closing $\mathcal{P}_2$ introduces auxiliary properties $\mathcal{P}_3 \cdots$

Current development snapshot

name	signature	(positive) properties	constructors
Sets	(S)	3	9
Semigroups	$(S, \oplus)$	14	17
Preorders	$(S, \leq)$	4	5
Bisemigroups	$(S, \oplus, \otimes)$	22	20
Order semigroups	$(S, \leq, \oplus)$	17	6
Transforms	$(S, L, \triangleright)$	2	8
Order transforms	$(S, L, \leq, \triangleright)$	3	2
Semigroup transforms	$(S, L, \oplus, \triangleright)$	4	10

where $\triangleright \in L \to S \to S$.

This represents over 1700 bottom-up rules. For this reason we are using Coq ...

Advanced Example

min-set definitions

$(S, \lesssim)$ is a pre-ordered set, $A \subseteq S$ finite.

$$\begin{aligned} \min_{\lesssim}(A) &\equiv \{a \in A \mid \forall b \in A : \neg(b < a)\} \\ \mathcal{P}(S, \lesssim) &\equiv \{A \subseteq S \mid A \text{ is finite and } \min_{\lesssim}(A) = A\} \end{aligned}$$

Operations over $\mathcal{P}(S, \lesssim)$:

$$\begin{aligned} A \oplus^{\lesssim}_{\min} B &= \min_{\lesssim}(A \cup B) \\ A \otimes^{\lesssim}_{\min} B &= \min_{\lesssim}(\{a \otimes b \mid a \in A,\ b \in B\}) \\ F(S, \oplus, \otimes) &= (\mathcal{P}(S, \lesssim), \otimes^{\leq^R_{\oplus}}_{\min}, \oplus^{\leq^R_{\oplus}}_{\min}) \\ M &= F(2^E, \cup, \cup) \end{aligned}$$

M is Martelli's Semiring for computing miminal cut sets in a graph (1976). We can automatically infer all of the properties (like types) to show that this is a well-behaved semiring. (The "typing" rules are rather complex ...)

Recommended Reading : Classic semiring theory and related topics

- Note on the lexicographic product of ordered semigroups. Saitô. Proceedings of the Japan Academy. v46.5, 1970.
- Regular Algebra Applied to Path-Finding Problems. Backhouse and Carr. J.Inst.Maths.Applics, v15, 1975
- A Gaussian elimination algorithm for the enumeration of cut sets in a graph. Martelli. Journal of the ACM. v23.1, 1976.
- Algebraic structures for transitive closure. Lehmann. Theoretical Computer Science. v4, 1977.
- Semirings and path spaces. Wongseelashote. Discrete Mathematics. v26.1, 1979.
- Path Problems in Graphs. Rote. In *Computational Graph Theory*, 1990.

Recommended Reading : Algebraic Internet Routing

- Algebra and algorithms for QoS path computation and hop-by-hop Routing in the Internet. Sobrinho. ToN v10.4, 2002.
- An Algebraic Theory of Dynamic Network Routing. Sobrinho. ToN v13.5, 2005.
- Towards a Unified Theory of Policy-Based Routing. Chau and Gibbens and Griffin. INFOCOM 2006.
- Increasing Bisemigroups and Algebraic Routing. Griffin and Gurney. RelMiCS10, April 2008.
  - Shows Bellman-Ford algorithm can solve left-local equations (without distributivity)
- Routing in Equilibrium. Joo Lus Sobrinho and Timothy G. Griffin. The 19th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2010).
  - Shows that Dijkstra's algorithm can solve right-local equations (without distributivity)

Metarouting

- Metarouting. Griffin and Sobrinho. SIGCOMM 2005
- An Implementation of Metarouting using Coq. Naudžiūnas and Griffin. W-RiPE workshop, sponsored by ICNP 2011.
- A Domain-Specific Language for the Specification of Path Algebras. Naudžiūnas and Griffin. Proceedings of the First Workshop on Automated Theory Engineering.

Do we have the right operators in the metalanguage? Answer with reverse engineering.

- Lexicographic Products in Metarouting. Gurney and Griffin. ICNP, October 2007.
- A model of Internet routing using semi-modules. Billings and Griffin. RelMiCS11/AKA6, November 2009.
  - Algebraic model of route redistribution
- Hybrid Link-State, Path-Vector Routing. Alim and Griffin. AINTEC, 2010.
- Neighbor-specific BGP: An algebraic exploration. Gurney and Griffin. ICNP, 2010.
- Pathfinding through Congruences. Gurney and Griffin. RAMiCS 12, 2011
- On the interaction of multiple routing algorithms. M. Abdul Alim, Timothy G. Griffin. ACM CoNEXT 2011, December 2011.
  - More on route redistribution, plus administrative distance.

Reverse Engineering Can be Fun

Taylor, Billings, Alim, Naudžiūnas, Griffin, Singh, Gurney.