What Have We Learned from Reverse-Engineering the Internet’s Inter-domain Routing Protocol?

Timothy G. Griffin

Computer Laboratory
University of Cambridge, UK
timothy.griffin@cl.cam.ac.uk

14èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications
Hérault, France
29 May, 2012
Internet routing has evolved organically, by the expedient hack.

... basic principles need to be uncovered by reverse engineering.

In the process, a new type of path problem is discovered!

This may have widespread applicability beyond routing — perhaps in operations research, combinatorics, and other branches of Computer Science.
Shortest paths example, $sp = (\mathbb{N}^{\infty}, \min, +)$

The adjacency matrix

$$A = \begin{bmatrix}
1 & \infty & 2 & 1 & 6 & \infty \\
2 & \infty & 5 & \infty & 4 \\
3 & 1 & 5 & \infty & 4 & 3 \\
4 & 6 & \infty & 4 & \infty & \infty \\
5 & \infty & 4 & 3 & \infty & \infty \\
\end{bmatrix}$$
Shortest paths example, continued

The routing matrix

\[
A^* = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & 0 & 2 & 1 & 5 & 4 \\
2 & 2 & 0 & 3 & 7 & 4 \\
3 & 1 & 3 & 0 & 4 & 3 \\
4 & 5 & 7 & 4 & 0 & 7 \\
5 & 4 & 4 & 3 & 7 & 0
\end{bmatrix}
\]

Matrix \( A^* \) solves this global optimality problem:

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

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

Bold arrows indicate the shortest-path tree rooted at 1.
Widest paths example, \((\mathbb{N}^\infty, \text{max}, \text{min})\)

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

The routing matrix

\[
A^* = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
\infty & 4 & 4 & 6 & 4 \\
2 & 4 & \infty & 5 & 4 & 4 \\
3 & 4 & 5 & \infty & 4 & 4 \\
4 & 6 & 4 & 4 & \infty & 4 \\
5 & 4 & 4 & 4 & 4 & \infty \\
\end{bmatrix}
\]

Matrix \(A^*\) solves this global optimality problem:

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

where \(w(p)\) is now the minimal edge weight in \(p\).
Fun example, \((2\{a, b, c\}, \cup, \cap)\)

We want a Matrix \(A^*\) to solve this global optimality problem:

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

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

For \(x \in \{a, b, c\}\), interpret \(x \in A^*(i, j)\) to mean that there is at least one path from \(i\) to \(j\) with \(x\) in every arc weight along the path.
Fun example, \((2\{a, b, c\}, \cup, \cap)\)

The matrix \(A^*\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{a, b, c}</td>
<td>{a, b, c}</td>
<td>{a, b, c}</td>
<td>{a, b}</td>
<td>{b, c}</td>
</tr>
<tr>
<td>2</td>
<td>{a, b, c}</td>
<td>{a, b, c}</td>
<td>{a, b, c}</td>
<td>{a, b}</td>
<td>{b, c}</td>
</tr>
<tr>
<td>3</td>
<td>{a, b, c}</td>
<td>{a, b, c}</td>
<td>{a, b, c}</td>
<td>{a, b}</td>
<td>{b, c}</td>
</tr>
<tr>
<td>4</td>
<td>{a, b}</td>
<td>{a, b}</td>
<td>{a, b}</td>
<td>{a, b, c}</td>
<td>{b}</td>
</tr>
<tr>
<td>5</td>
<td>{b, c}</td>
<td>{b, c}</td>
<td>{b, c}</td>
<td>{b}</td>
<td>{a, b, c}</td>
</tr>
</tbody>
</table>
## Semirings

### A few examples

<table>
<thead>
<tr>
<th>name</th>
<th>$S$</th>
<th>$\oplus$</th>
<th>$\otimes$</th>
<th>$\bar{0}$</th>
<th>$\bar{1}$</th>
<th>possible routing use</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp</td>
<td>$\mathbb{N}^\infty$</td>
<td>min</td>
<td>+</td>
<td>$\infty$</td>
<td>0</td>
<td>minimum-weight routing</td>
</tr>
<tr>
<td>bw</td>
<td>$\mathbb{N}^\infty$</td>
<td>max</td>
<td>min</td>
<td>0</td>
<td>$\infty$</td>
<td>greatest-capacity routing</td>
</tr>
<tr>
<td>rel</td>
<td>$[0,1]$</td>
<td>max</td>
<td>$\times$</td>
<td>0</td>
<td>1</td>
<td>most-reliable routing</td>
</tr>
<tr>
<td>use</td>
<td>${0,1}$</td>
<td>max</td>
<td>min</td>
<td>0</td>
<td>1</td>
<td>usable-path routing</td>
</tr>
<tr>
<td></td>
<td>$2^W$</td>
<td>$\cup$</td>
<td>$\cap$</td>
<td>${}$</td>
<td>$W$</td>
<td>shared link attributes?</td>
</tr>
<tr>
<td></td>
<td>$2^W$</td>
<td>$\cap$</td>
<td>$\cup$</td>
<td>$W$</td>
<td>${}$</td>
<td>shared path attributes?</td>
</tr>
</tbody>
</table>

Path problems focus on **global optimality**

$$A^*(i, j) = \bigoplus_{p\in P(i, j)} w(p)$$
Recommended Reading

Graphs, Dioids and Semirings
New Models and Algorithms

Path Problems in Networks
John Baras
George Theodorakopoulos
What algebraic properties are needed for efficient computation of global optimality?

### Distributivity

**L.D**: \[ a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c), \]
**R.D**: \[ (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c). \]

### What is this in \( sp = (\mathbb{N}_\infty, \min, +) \)?

**L.DIST**: \[ a + (b \min c) = (a + b) \min (a + c), \]
**R.DIST**: \[ (a \min b) + c = (a + c) \min (b + c). \]

(I am ignoring all of the other semiring axioms here ... )
Lesson 1: Some realistic metrics are not distributive!

Two ways of forming “lexicographic” combination of shortest paths $sp$ and bandwidth $bw$.

**Widest shortest paths**

- Metric values of form $(d, \ b)$
- $d$ in $sp$
- $b$ in $bw$
- Consider $d$ first, break ties with $b$
- Is distributive (some details ignored ...)

**Shortest Widest paths**

- Metric values of form $(b, \ d)$
- $d$ in $sp$
- $b$ in $bw$
- Consider $b$ first, break ties with $d$
- Not distributive
Example

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

\[(5, 1) \otimes ((10, 100) \oplus (7, 1)) = (5, 1) \otimes (10, 100) = (5, 101)\]
\[((5, 1) \otimes (10, 101)) \oplus ((5, 1) \otimes (7, 1)) = (5, 101) \oplus (5, 2) = (5, 2)\]
Lesson 2: Left-Local Optimality

Say that $L$ is a left locally-optimal solution when

$$L = (A \otimes L) \oplus I.$$ 

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

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

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

Say that \( R \) is a right locally-optimal solution when

\[
R = (R \otimes A) \oplus I.
\]

That is, for \( i \neq j \) we have

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

- \( R(i, j) \) is the best possible value given the values \( R(q, j) \), for all in-neighbors \( q \) of destination \( j \).
- Rows \( L(i, _) \) represents **out-trees from** \( i \) (think Dijkstra).
- Columns \( L(_, i) \) represents **in-trees to** \( i \).
- Does not work well with hop-by-hop forwarding from \( i \).
With and Without Distributivity

With

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

\[ A^* = L = R \]

Without

Suppose that we drop distributivity and \( A^* \), \( L \), \( R \) exist. It may be the case they they are all distinct.

Health warning: matrix multiplication over structures lacking distributivity is not associative!
(bandwidth, distance) with lexicographic order (bandwidth first).
Global optima

\[ A^* = \begin{bmatrix}
  1 & 2 & 3 & 4 & 5 \\
  1 & (\infty, 0) & (5, 1) & (0, \infty) & (0, \infty) \\
  2 & (0, \infty) & (\infty, 0) & (0, \infty) & (0, \infty) \\
  3 & (5, 2) & (5, 3) & (\infty, 0) & (5, 1) & (5, 2) \\
  4 & (10, 6) & (5, 2) & (5, 2) & (\infty, 0) & (10, 1) \\
  5 & (10, 5) & (5, 4) & (5, 1) & (5, 2) & (\infty, 0)
\end{bmatrix} \]
Left local optima

\[
L = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & (\infty, 0) & (5, 1) & (0, \infty) & (0, \infty) & (0, \infty) \\
2 & (0, \infty) & (\infty, 0) & (0, \infty) & (0, \infty) & (0, \infty) \\
3 & (5, 7) & (5, 3) & (\infty, 0) & (5, 1) & (5, 2) \\
4 & (10, 6) & (5, 2) & (5, 2) & (\infty, 0) & (10, 1) \\
5 & (10, 5) & (5, 4) & (5, 1) & (5, 2) & (\infty, 0)
\end{bmatrix},
\]

Entries marked in bold indicate those values which are not globally optimal.
Right local optima

\[
R = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & (\infty, 0) & (5, 1) & (0, \infty) & (0, \infty) \\
2 & (0, \infty) & (\infty, 0) & (0, \infty) & (0, \infty) \\
3 & (5, 2) & (5, 3) & (\infty, 0) & (5, 1) & (5, 2) \\
4 & (10, 6) & (5, 6) & (5, 2) & (\infty, 0) & (10, 1) \\
5 & (10, 5) & (5, 5) & (5, 1) & (5, 2) & (\infty, 0) \\
\end{pmatrix}
\]
Left-locally optimal paths to node 2
Right-locally optimal paths to node 2

1, 3, 4 → 2
3 → 2
4 → 2
5 → 2

1 → 2, 3, 4 → 2

What Have We Learned from Reverse-Engineering the Internet's Inter-domain Routing Protocol?

29-05-2012 21 / 36
Lesson 3: Bellman-Ford can compute left-local solutions

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

- Bellman-ford algorithm must be modified to ensure only loop-free paths are inspected.
- \((S, \oplus, 0)\) is a commutative, idempotent, and selective monoid,
- \((S, \otimes, 1)\) is a monoid,
- 0 is the annihilator for \(\otimes\),
- 1 is the annihilator for \(\oplus\),
- Left strictly inflationarity, \(L.S.\text{INF}: \forall a, b : a \neq 0 \implies a < a \otimes b\)
- 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.
Lesson 4: Dijkstra’s algorithm can work for right-local optima!

Input: adjacency matrix $A$ and source vertex $i \in V$,
Output: the $i$-th row of $R$, $R(i, \_)$.

begin
  $S \leftarrow \{i\}$
  $R(i, i) \leftarrow 1$
  for each $q \in V - \{i\}$ : $R(i, q) \leftarrow A(i, q)$
  while $S \neq V$
    begin
      find $q \in V - S$ such that $R(i, q)$ is $\leq^L$-minimal
      $S \leftarrow S \cup \{q\}$
      for each $j \in V - S$
        $R(i, j) \leftarrow R(i, j) \oplus (R(i, q) \otimes A(q, j))$
    end
end
The goal

Given adjacency matrix $A$ and source vertex $i \in V$, Dijkstra’s algorithm will compute $R(i, \_)$ such that

$$\forall j \in V : R(i, j) = I(i, j) \oplus \bigoplus_{q \in V} R(i, q) \otimes A(q, j).$$

Main invariant

$$\forall k : 1 \leq k \leq |V| \implies \forall j \in S_k : R_k(i, j) = I(i, j) \oplus \bigoplus_{q \in S_k} R_k(i, q) \otimes A(q, j)$$

Minimal subset of semiring axioms needed right-local Dijkstra

**Semiring Axioms**

<table>
<thead>
<tr>
<th>Axiom Type</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADD. ASSOCIATIVE</strong></td>
<td>( a \oplus (b \oplus c) = (a \oplus b) \oplus c )</td>
</tr>
<tr>
<td><strong>ADD. COMMUTATIVE</strong></td>
<td>( a \oplus b = b \oplus a )</td>
</tr>
<tr>
<td><strong>ADD. LEFT. ID</strong></td>
<td>( 0 \oplus a = a )</td>
</tr>
<tr>
<td><strong>MULT. ASSOCIATIVE</strong></td>
<td>( a \otimes (b \otimes c) \uplus (a \otimes b) \otimes c )</td>
</tr>
<tr>
<td><strong>MULT. LEFT. ID</strong></td>
<td>( 1 \otimes a = a )</td>
</tr>
<tr>
<td><strong>MULT. RIGHT. ID</strong></td>
<td>( a \otimes 1 = a )</td>
</tr>
<tr>
<td><strong>MULT. LEFT. ANN</strong></td>
<td>( 0 \otimes a \uplus 0 )</td>
</tr>
<tr>
<td><strong>MULT. RIGHT. ANN</strong></td>
<td>( a \otimes 0 \uplus 0 )</td>
</tr>
<tr>
<td><strong>L. DISTRIBUTIVE</strong></td>
<td>( a \otimes (b \oplus c) \uplus (a \otimes b) \oplus (a \otimes c) )</td>
</tr>
<tr>
<td><strong>R. DISTRIBUTIVE</strong></td>
<td>( (a \oplus b) \otimes c \uplus (a \otimes c) \otimes (b \otimes c) )</td>
</tr>
</tbody>
</table>
Additional axioms needed right-local Dijkstra

**ADD.SELECTIVE**: \( a \oplus b \in \{a, b\} \)

**ADD.LEFT.ANN**: \( \overline{1} \oplus a = \overline{1} \)

**ADD.RIGHT.ANN**: \( a \oplus \overline{1} = \overline{1} \)

**RIGHT.ABSORBTION**: \( a \oplus (a \otimes b) = a \)

RIGHT.ABSORBTION gives inflationarity, \( \forall a, b : a \leq a \otimes b. \)
Expressed in Coq

```
Variable plus_associative : ∀ x y z, x ⊕ (y ⊕ z) = (x ⊕ y) ⊕ z.
Variable plus_commutative : ∀ x y, x ⊕ y = y ⊕ x.
Variable plus_selective  : ∀ x y, (x ⊕ y == x) || (x ⊕ y == y).

(* identities *)
Variable zero_is_left_plus_id : ∀ x, zero ⊕ x = x.
Variable one_is_left_times_id  : ∀ x, one ⊗ x = x.

(* one is additive annihilator *)
Variable one_is_left_plus_ann : ∀ x, one ⊕ x = one.
Variable one_is_right_plus_ann : ∀ x, x ⊕ one = one.

(* right absorption *)
Variable right_absorption : ∀ a b : T, a ⊕ (a ⊗ b) == a.

Definition lno (a b : T) := a ⊕ b == a.
Notation "A ⊆ B" := (lno A B) (at level 60).

Lemma lno_right_increasing : ∀ a b : T, a ⊆ a ⊕ b.
```
Using a Link-State approach with hop-by-hop forwarding ...

Need left-local optima!

\[
L = (A \otimes L) \oplus I \iff L^T = (L^T \hat{\otimes} A^T) \oplus I
\]

where \( \otimes^T \) is matrix multiplication defined with as

\[
a \otimes^T b = b \otimes a
\]

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

Each node would have to solve the entire “all pairs” problem.
Inter-domain routing in the Internet

The Border Gateway Protocol (BGP)

- In the distributed Bellman-Ford family.
- Hard-state (not refresh based).
- Complex policy and metrics.
- Primary requirement: connectivity should not violate the economic relationships between autonomous networks.
- At a very high-level, the metric combines economics and traffic engineering.
- This is implemented using a lexicographic product, where economics is most significant.
Simplified model (Gao and Rexford)

- **customer route**: from somebody paying you for transit services.
- **provider route**: from somebody you are paying for transit services.
- **peer route**: from a competitor.
  - If you are at top of food chain you are forced to do this.
  - Smaller networks do this to reduce their provider charges.
- **customer < peer < provider**
Example

- node $j$ prefers long path though one of its customers
- node $i$ prefers the shorter path through its provider
These restrictions are another source for violations of distributivity.
As a result ...

- Protocol will diverge when no solution exists.
- Protocol may diverge even when a solution exists.
- BGP Wedgies, RFC 4264.
  - Multiple stable states may exist.
  - No guarantee that each state implements intended policy.
  - Manual intervention required when system gets stuck in unintended local optima.
  - Debugging nearly impossible when policy is not shared between networks.
How to fix?
First, allow functions on arcs.

\((S, \oplus, F \subseteq S \rightarrow S, \overline{0})\)

General conditions
- \((S, \oplus, \overline{0})\) is a commutative, idempotent, and selective monoid,
- \(\forall f \in F : f(\overline{0}) = \overline{0}\)
- For local-optima need \(\text{INF} : \forall a, f : a \leq f(a)\)

Simplest model for “fixed” interdomain routing
- metrics of the form \((c, d)\) or \(\infty\), where \(c \in \{0, 1, 2\}\) and \(d\) is a path length,
- metrics compared lexicographically.
- 0 is for downstream routes (towards paying customers),
- 1 is for peer routes (towards competitor’s customers),
- 2 is for upstream routes (towards charging providers),
The inflationary policy functions

- Gao/Rexford rules in red.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>1</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>2</td>
<td>∞</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>∞</td>
<td>2</td>
</tr>
<tr>
<td>f</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>g</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>h</td>
<td>1</td>
<td>1</td>
<td>∞</td>
</tr>
<tr>
<td>i</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>j</td>
<td>1</td>
<td>2</td>
<td>∞</td>
</tr>
<tr>
<td>k</td>
<td>1</td>
<td>∞</td>
<td>2</td>
</tr>
<tr>
<td>l</td>
<td>1</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>
Lessons

- Some non-distributive algebras make are useful.
- Local optimality is a useful notion for non-distributive algebras.
- Bellman-Ford (path vectoring) can compute left-local optima ...
- ... and so can Dijkstra’s algorithm!