An Algebraic Approach to Internet Routing — Lectures 13, 14
Routing in Equilibrium
(presented at MTNS 2010, Budapest)

João Luís Sobrinho Timothy G. Griffin

Instituto de Telecomunicações
Instituto Superior Técnico, Lisbon, Portugal
joao.sobrinho@lx.it.pt

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

Michaelmas Term
2010
What algebraic properties are associated with global optimality?

Distributivity

\[
\begin{align*}
\text{L.D} & : \quad a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c), \\
\text{R.D} & : \quad (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c).
\end{align*}
\]

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

\[
\begin{align*}
\text{L.DIST} & : \quad a + (b \min c) = (a + b) \min (a + c), \\
\text{R.DIST} & : \quad (a \min b) + c = (a + c) \min (b + c).
\end{align*}
\]
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** $i$. 

Sobrinho, Griffin (Instituto de Telecomunicações Computer Laboratory Instituto Superior Técnico, Lisbon, Portugal University of Cambridge, UK joao.sobrinho@lx.it.pt timothy.griffin@cl.cam.ac.uk) 

An Algebraic Approach to Internet Routing — Lectures 13, 14 

Routing in Equilibrium (presented at MTNS 2010, Budapest) 

T.G.Griffin ©2010 3 / 29
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** $i$. 

Sobrinho, Griffin (Instituto de Telecomunicações Computer Laboratory Instituto Superior Técnico, Lisbon, Portugal University of Cambridge, UK joao.sobrinho@lx.it.pt timothy.griffin@cl.cam.ac.uk **An Algebraic Approach to Internet Routing — Lectures 13, 14** Routing in Equilibrium (presented at MTNS 2010, Budapest) T.G.Griffin©2010 4 / 29
With and Without Distributivity

**With**

For (well behaved) 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.
Global Optimality

This has been studied, for example [?, ?] in the context of circuit layout. See Chapter 5 of [?]. This approach does not play well with (loop-free) hop-by-hop forwarding (need tunnels!)

Left Local Optimality

At a very high level, this is the type of problem that BGP attempts to solve!!

Right Local Optimality

This approach does not play well with (loop-free) hop-by-hop forwarding (need tunnels!)
(bandwidth, distance) with lexicographic order (bandwidth first).
Example’s adjacency matrix

\[
A = \begin{bmatrix}
(0, \infty) & (5, 1) & (0, \infty) & (0, \infty) & (0, \infty) \\
(0, \infty) & (0, \infty) & (0, \infty) & (0, \infty) & (0, \infty) \\
(0, \infty) & (5, 4) & (0, \infty) & (5, 1) & (0, \infty) \\
(5, 1) & (0, \infty) & (0, \infty) & (0, \infty) & (10, 1) \\
(10, 5) & (0, \infty) & (5, 1) & (0, \infty) & (0, \infty)
\end{bmatrix}
\]
Global optima

\[ A^* = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & (\infty, 0) & (5, 1) & (0, \infty) & (0, \infty) \\
2 & (0, \infty) & (\infty, 0) & (0, \infty) & (0, \infty) \\
3 & (5, 2) & (5, 3) & (\infty, 0) & (5, 1) & (5, 2) \\
4 & (10, 6) & (5, 2) & (5, 2) & (\infty, 0) & (10, 1) \\
5 & (10, 5) & (5, 4) & (5, 1) & (5, 2) & (\infty, 0)
\end{bmatrix}, \]
Entries marked in **bold** indicate those values which are not globally optimal.
Left-locally optimal paths to node 2
Right local optima

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

Note: the \((5, 6)\) is \((5, 7)\) in the paper, which appears to be a bug!
Right-locally optimal paths to node 2

1, 3, 4 → 2

3 → 2

4 → 2

5 → 2

4 → 2

1

2

3

4

5
What are the conditions needed to guarantee existence of local optima?

For a non-distributed structure $S = (S, \oplus, \otimes, \overline{0}, \overline{1})$, can be used to find local optima when the following property holds.

**Strictly Inflationary**

$S.\text{INFL} : \forall a, b \in S : a \neq \overline{0} \implies a < b \otimes a$

where $a \leq b$ means $a = a \oplus b$.

We know that (a modified) Bellman-Ford iteration will converge, but we currently have no bound on the number of iterations needed!
**Dijkstra’s algorithm**

**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
end
Dijkstra’s algorithm, annotated version

Subscripts make proofs by induction easier ....

begin
    \[ S_1 \leftarrow \{i\} \]
    \[ R_1(i, i) \leftarrow 1 \]
    for each \( q \in V - S_1 : R_1(i, q) \leftarrow A(i, q) \]
    for each \( k = 2, 3, \ldots, |V| \)
        begin
            find \( q_k \in V - S_{k-1} \) such that \( R(i, q) \) is \( \leq_L \) -minimal
            \[ S_k \leftarrow S_{k-1} \cup \{q_k\} \]
            for each \( j \in V - S_k \)
                \[ R_k(i, j) \leftarrow R_{k-1}(i, j) \oplus (R_{k-1}(i, q_k) \otimes A(q_k, j)) \]
        end
    end
Assumptions on \((S, \oplus, \otimes, \bar{0}, \bar{1})\)

- \((S, \oplus, \bar{0})\) is a commutative, idempotent, and selective monoid,
- \((S, \otimes, \bar{1})\) is a monoid,
- \(\bar{0}\) is the annihilator for \(\otimes\),
- \(\bar{1}\) is the annihilator for \(\oplus\),
- \(\text{RINF} : \forall a, b : a \leq a \otimes b\)

Recall that \(a \leq b \equiv a \leq^L b \equiv a = a \oplus b\).
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 Claim

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

Observation 1

$$\forall k : 1 \leq k < |V| \implies \forall j \in S_{k+1} : R_k(i, j) = R_{k+1}(i, j)$$

This is easy to see — once a node is put into $S$ its weight never changes.
Observation 2

∀k : 1 ≤ k ≤ |V| → ∀q ∈ S_k : ∀w ∈ V − S_k : R_k(i, q) ≤ R_k(i, w)

By induction.
Base: Need \( 1 \leq A(i, w) \). OK
Induction. Assume

\[ \forall q \in S_k : \forall w \in V - S_k : R_k(i, q) \leq R_k(i, w) \]

and show

\[ \forall q \in S_{k+1} : \forall w \in V - S_{k+1} : R_{k+1}(i, q) \leq R_{k+1}(i, w) \]

Since \( S_{k+1} = S_k \cup \{q_{k+1}\} \), this is means showing

1. \[ \forall q \in S_k : \forall w \in V - S_{k+1} : R_{k+1}(i, q) \leq R_{k+1}(i, w) \]
2. \[ \forall w \in V - S_{k+1} : R_{k+1}(i, q_{k+1}) \leq R_{k+1}(i, w) \]
By Observation 1, showing (1) is the same as

\[ \forall q \in S_k : \forall w \in V - S_{k+1} : R_k(i, q) \leq R_{k+1}(i, w) \]

which expands to (by definition of \( R_{k+1}(i, w) \))

\[ \forall q \in S_k : \forall w \in V - S_{k+1} : R_k(i, q) \leq R_k(i, w) \oplus (R_k(i, q_{k+1}) \otimes A(q_{k+1}, w)) \]

But \( R_k(i, q) \leq R_k(i, w) \) by the induction hypothesis, and \( R_k(i, q) \leq (R_k(i, q_{k+1}) \otimes A(q_{k+1}, w)) \) by the induction hypothesis and RINF.

Since \( a \leq^{L} b \wedge a \leq^{L} c \implies a \leq^{L} (b \oplus c) \), we are done.
By Observation 1, showing (2) is the same as showing

\[ \forall w \in V - S_{k+1} : R_k(i, q_{k+1}) \leq R_{k+1}(i, w) \]

which expands to

\[ \forall w \in V - S_{k+1} : R_k(i, q_{k+1}) \leq R_k(i, w) \oplus (R_k(i, q_{k+1}) \otimes A(q_{k+1}, w)) \]

But \( R_k(i, q_{k+1}) \leq R_k(i, w) \) since \( q_{k+1} \) was chosen to be minimal, and \( R_k(i, q_{k+1}) \leq (R_k(i, q_{k+1}) \otimes A(q_{k+1}, w)) \) by \( RINF \).

Since \( a \leq_L b \land a \leq_L c \implies a \leq_L (b \oplus c) \), we are done.
Observation 3

∀ \( k : 1 \leq k \leq |V| \) \implies \forall w \in V - S_k : \( R_k(i, w) = \bigoplus_{q \in S_k} R_k(i, q) \otimes A(q, w) \)

Proof: By induction:
Base : easy, since
\[
\bigoplus_{q \in S_1} R_1(i, q) \otimes A(q, w) = \bar{1} \otimes A(i, w) = A(i, w) = R_1(i, w)
\]
Induction step. Assume
\[
\forall w \in V - S_k : R_k(i, w) = \bigoplus_{q \in S_k} R_k(i, q) \otimes A(q, w)
\]
and show
\[
\forall w \in V - S_{k+1} : R_{k+1}(i, w) = \bigoplus_{q \in S_{k+1}} R_{k+1}(i, q) \otimes A(q, w)
\]
By Observation 1, and a bit of rewriting, this means we must show

\[ \forall w \in V - S_{k+1} : R_{k+1}(i, w) = R_k(i, q_{k+1}) \otimes A(q_{k+1}, w) \oplus \bigoplus_{q \in S_k} R_k(i, q) \otimes A(q, w) \]

Using the induction hypothesis, this becomes

\[ \forall w \in V - S_{k+1} : R_{k+1}(i, w) = R_k(i, q_{k+1}) \otimes A(q_{k+1}, w) \oplus R_k(i, w) \]

But this is exactly how \( R_{k+1}(i, w) \) is computed in the algorithm.
Proof of Main Claim

Main Claim

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

Proof: By induction on \( k \).
Base case: \( S_1 = \{i\} \) and the claim is easy.
Induction: Assume that

\[ \forall j \in S_k : R_k(i, j) = I(i, j) \oplus \bigoplus_{q \in S_k} R_k(i, q) \otimes A(q, j) \]

We must show that

\[ \forall j \in S_{k+1} : R_{k+1}(i, j) = I(i, j) \oplus \bigoplus_{q \in S_{k+1}} R_{k+1}(i, q) \otimes A(q, j) \]
Since $S_{k+1} = S_k \cup \{q_{k+1}\}$, this means we must show

\begin{align*}
(1) \quad & \forall j \in S_k : R_{k+1}(i, j) = I(i, j) \oplus \bigoplus_{q \in S_{k+1}} R_{k+1}(i, q) \otimes A(q, j) \\
(2) \quad & R_{k+1}(i, q_{k+1}) = I(i, q_{k+1}) \oplus \bigoplus_{q \in S_{k+1}} R_{k+1}(i, q) \otimes A(q, q_{k+1})
\end{align*}

By use Observation 1, showing (1) is the same as showing

\[
\forall j \in S_k : R_k(i, j) = I(i, j) \oplus \bigoplus_{q \in S_{k+1}} R_k(i, q) \otimes A(q, j),
\]

which is equivalent to

\[
\forall j \in S_k : R_k(i, j) = R_k(i, j) \oplus (R_k(i, q_{k+1}) \otimes A(q_{k+1}, j)) \oplus \bigoplus_{q \in S_k} R_k(i, q) \otimes A(q, j),
\]

By the induction hypothesis, this is equivalent to

\[
\forall j \in S_k : R_k(i, j) = R_k(i, j) \oplus (R_k(i, q_{k+1}) \otimes A(q_{k+1}, j)),
\]
Put another way,

\[ \forall j \in S_k : R_k(i, j) \leq R_k(i, q_{k+1}) \otimes A(q_{k+1}, j) \]

By observation 2 we know \( R_k(i, j) \leq R_k(i, q_{k+1}) \), and so

\[ R_k(i, j) \leq R_k(i, q_{k+1}) \leq R_k(i, q_{k+1}) \otimes A(q_{k+1}, j) \]

by RINF.
To show (2), we use Observation 1 and $I(i, q_{k+1}) = 0$ to obtain

$$R_k(i, q_{k+1}) = \bigoplus_{q \in S_{k+1}} R_k(i, q) \otimes A(q, q_{k+1})$$

which, since $A(q_{k+1}, q_{k+1}) = 0$, is the same as

$$R_k(i, q_{k+1}) = \bigoplus_{q \in S_k} R_k(i, q) \otimes A(q, q_{k+1})$$

This then follows directly from Observation 3.
Finding Left Local Solutions?

\[
L = (A \otimes L) \oplus I \quad \iff \quad L^T = (L^T \otimes^T A^T) \oplus I
\]

\[
R^T = (A^T \otimes^T R^T) \oplus I \quad \iff \quad R = (R \otimes A) \oplus I
\]

where

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

Notice that this exchanges RINF for LINF!

\[
\text{LINF} : \forall a, b : a \leq b \otimes a
\]
Conclusion

- Complexity of solving for left local optima?
  - Previous work has shown that Bellman-Ford will find a solution as long as only simple paths are explored — but no time bounds are known.
  - But, now we know that $O(V^3)$ will due with Dijkstra’s greedy algorithm.
  - Could do better in sparse graphs using Fibonacci heaps ...