Increasing Bisemigroups and Algebraic Routing

Timothy G. Griffin and Alexander J. T. Gurney

Computer Laboratory, University of Cambridge
Email:{Timothy.Griffin, Alexander.Gurney}@cl.cam.ac.uk

Abstract. The Internet protocol used today for global routing — the Border Gateway Protocol (BGP) — evolved in a rather organic manner without a clear theoretical foundation. This has stimulated a great deal of recent theoretical work in the networking community aimed at modeling BGP-like routing protocols. This paper attempts to make this work more accessible to a wider community by reformulating it in a purely algebraic setting. This leads to structures we call *increasing bisemigroups*, which are essentially non-distributive semirings with an additional order constraint. Solutions to path problems in graphs annotated over increasing bisemigroups represent locally optimal Nash-like equilibrium points rather than globally optimal paths as is the case with semiring routing.

1 Introduction

A software system can evolve organically while becoming an essential part of our infrastructure. This may even result in a system that is not well understood. Such is the case with the routing protocol that maintains global connectivity in the Internet — the Border Gateway Protocol (BGP). Although it may seem that routing is a well understood problem, we would argue that meeting the constraints of routing between autonomous systems in the Internet has actually given birth to a new class of routing protocols. This class can be characterized by the goal of finding paths that represent locally optimal Nash-like equilibrium points rather than paths that are optimal over all possible paths.

This paper is an attempt to present recent theoretical work on BGP in a purely algebraic setting. Section 2 describes BGP and presents an overview of some of the theoretical work modeling this protocol. Section 3 presents the quadrants model as a framework for discussing how this work relates to the literature on semiring routing. We define *increasing bisemigroups*, which are essentially non-distributive semirings with an additional order constraint. Solutions to path problems in graphs annotated over increasing bisemigroups represent locally optimal Nash-like equilibrium points rather than globally optimal paths as is the case with semiring routing. Section 4 reformulates the work described in Section 2 in terms of increasing bisemigroups. In particular, previous work on BGP modeling has involved reasoning about asynchronous protocols. Here we employ a more traditional approach based on simple matrix multiplication. Section 5 outlines several open problems.

2 Theory and practice of interdomain routing

We can think of routing protocols as being comprised of two distinct components,

$$\text{routing protocol} = \text{routing language} + \text{algorithm},$$

where the protocol's *routing language* is used to configure a network and the (often distributed) *algorithm* is for computing routing solutions to network configurations specified using the routing language. A routing language captures (1) how routes are described, (2) how best routes are selected, (3) how (low-level) policy is described, and (4) how policy is applied to routes.

This characterization of routing protocols may seem straightforward to those familiar with the literature on semiring routing [1–4], where we can consider a given semiring to be a routing language. However, the Internet Engineering Task Force (IETF) does not define or develop routing protocols to reflect this thinking. The IETF documents that define protocols (RFCs) tend to present all aspects of a routing protocol algorithmically, mostly due to the emphasis on system performance. The task of untangling the routing language from the routing algorithm for the purposes of analysis is often a very difficult challenge.

Perhaps the most difficult Internet routing protocol to untangle is the Border Gateway Protocol (BGP) [5–7]. This protocol is used to implement routing in the core of the Internet between Internet Service Providers (ISPs) and large organizations. (The vast majority of corporate and campus networks at the "edge" of the Internet are statically routed to their Internet provider and do not need to run BGP.) At the beginning of 2008 there were over 27,000 autonomous networks using BGP to implement routing in the public Internet[1]. An autonomous network can represent anywhere from one to thousands of routers each running BGP. Clearly this protocol is an essential part of the Internet's infrastructure.

The rather complex BGP route selection algorithm can be modeled abstractly as implementing a total pre-order $\leq$ so that if a and b are BGP routes and $a < b$, then a is preferred over b. BGP routes can be thought of as records containing multiple fields, and the order as a lexicographic order with respect to the orders associated with each field's domain. The most significant attribute tends to be used to implement economic relationships between networks, while the less significant tend to be used to implement local *traffic engineering* goals.

Network operators configure routing policies using low-level and vendor-specific languages. Abstractly, a policy can be modeled as a function f that transforms a route a to the route $f(a)$. Policy functions are applied when routes are *exported to* and *imported from* neighboring routers. An important thing to understand is that BGP standards have intentionally underspecified the language used for configuring policy functions. The actual policy languages used today have emerged over the last twenty years from a complex interaction between network operators, router vendors, and protocol engineers. This evolution has taken place with little or no theoretical guidance. This has been positive in

[1] Each network is associated with a unique identifier that can be found in BGP routing tables. See http://bgp.potaroo.net.

the sense that global routing was not overly constrained, allowing it to co-evolve along with a viable economic model of packet transport [8].

However, the negative side is that BGP can exhibit serious anomalies. Because of the unconstrained nature of policy functions, routing solutions are not guaranteed to exist and this can lead to protocol divergence [9, 10]. Another problem is that routing solutions are not guaranteed to be unique. In an interdomain setting routing policies are considered proprietary and not generally shared between competing ISPs. This can lead to situations where BGP falls into a local optimum that violates the intended policies of operators, yet no one set of operators has enough global knowledge to fix the problem [11].

If BGP policy functions could be constrained to always be *monotonic*, $a \leq b \rightarrow f(a) \leq f(b)$, then standard results might be applied to show that best routes are globally optimal routes and the above mentioned anomalies could not occur. However, it appears very unlikely that any fix imposing monotonicity requirements would be adopted by network operators. Sobrinho has shown that a very simple model of interdomain economic relationships can be implemented with monotonic functions [12, 13]. He also showed that more realistic models capturing common implementations of fail-over and load balancing [14] are not monotonic. Yet even if the interdomain world could agree on a monotonic model of interdomain economic relationships, combining this in a monotonic lexicographic order with other common traffic engineering metrics may be impossible. Recent work has shown that obtaining monotonicity with lexicographic products is fairly difficult [15].

One reaction to this situation is to simply declare interdomain routing a "broken mess" and move on to something more tractable. Another is to conclude that there is actually something new emerging here, and that we need to better understand this type of routing and how it relates to more standard approaches.

2.1 The Stable Paths Problem (SPP)

The Stable Paths Problem (SPP) [16, 17] was proposed as a simple graph-theoretic model of BGP routing, and was applied to the analysis of several real-world routing problems [14, 18, 19].

Let $G = (V,\ E,\ v_0)$ be a graph with origin v_0. The set $\mathcal{P}(v,\ v_0)$ denotes all *simple* paths from node v to node v_0. For each $v \in V$, $\mathcal{P}^v \subseteq \mathcal{P}(v,\ v_0)$ denotes the set of *permitted paths* from v to the origin. Let $\mathcal{P}$ be the union of all sets $\mathcal{P}^v$.

For each $v \in V$, there is a non-negative, integer-valued *ranking function* λ^v, defined over $\mathcal{P}^v$, which represents how node v ranks its permitted paths. If $P_1, P_2 \in \mathcal{P}^v$ and $\lambda^v(P_1) < \lambda^v(P_2)$, then P_2 is said to be *preferred over* P_1. Let $\Lambda = \{\lambda^v \mid v \in V - \{v_0\}\}$.

An instance of the *Stable Paths Problem*, $S_{\text{spp}} = (G,\ \mathcal{P},\ \Lambda)$, is a graph together with the permitted paths at each node and the ranking functions for each node. In addition, we assume that $\mathcal{P}^0 = \{(v_0)\}$, and for all $v \in V - \{v_0\}$:

- **(empty path is permitted)** $\epsilon \in \mathcal{P}^v$,
- **(empty path is least preferred)** $\lambda^v(\epsilon) = 0,\ \lambda^v(P) > 0$ for $P \neq \epsilon$,

– **(strictness)** If $P_1, P_2 \in \mathcal{P}^v$, $P_1 \neq P_2$, and $\lambda^v(P_1) = \lambda^v(P_2)$, then there is a u such that $P_1 = (v\ u)P_1'$ and $P_2 = (v\ u)P_2'$ (paths P_1 and P_2 have the same next-hop),
– **(simplicity)** If path $P \in \mathcal{P}^v$, then P is a simple path (no repeated nodes),

A *path assignment* is a function π that maps each node $u \in V$ to a path $\pi(u) \in \mathcal{P}^u$. (Note, this means that $\pi(v_0) = (v_0)$.) We interpret $\pi(u) = \epsilon$ to mean that u is not assigned a path to the origin.

The SPP work defines an asynchronous protocol for computing solutions to instances of the stable paths problem. This protocol is in the family of distributed Bellman-Ford algorithms. A sufficient condition (that the *dispute digraph* is acyclic, described below), is shown to imply that this protocol terminates with a locally optimal solution.

The dispute digraph is a directed graph where the nodes are paths in the SPP instance. A *dispute arc* $(p,\ q)$ represents the situation where

1. $p = (u, v)t$ is a feasible path from u to v_0 with next-hop v,
2. q is a path from v to v_0,
3. either $(u, v)q$ is not feasible at u or p is more preferred than $(u, v)q)$ at u.
4. path q is more preferred at v than t.

A *transmission arc* $(p, (u, v)p)$ is defined when p is permitted at v, $(u, v) \in E$, and $(u, v)p$ is permitted at u. The dispute digraph is then the union of dispute and transmission arcs.

Another concept used in [16, 17] is the *dispute wheel*. Suppose that p_m ends in the initial node of path p_0 and that p is a cycle $p_0 p_2 \cdots p_{m-1} p_m$. Suppose that there are paths q_j, each terminating in v_0, and each sharing its initial node node with p_j. Then this configuration represents a dispute wheel if for each j the path $p_j q_{j+1}$ is more preferred than path q_j, where the subscripts are taken mod m. In [16] it is shown that every dispute wheel can be mapped to a cycle in the dispute digraph.

2.2 Sobrinho's Model

Sobrinho approached the problem from a more algebraic point of view and introduced his routing algebras [20, 12]. This work extended his earlier algebraic generalizations of shortest-path routing [21]. Sobrinho's routing algebras take the form $\mathcal{A} = (S,\ \leq,\ L,\ \otimes)$, where $\leq$ is a preference order over S, L is a set of *labels*, and the operator $\otimes$ maps $L \times S$ to S. The set S contains a special element $\infty \in S$ such that: $\sigma < \infty$, for all $\sigma \in S \backslash \{\infty\}$ and $l \otimes \infty = \infty$, for all $l \in L$. A routing algebra $\mathcal{A}$ is said to be *increasing* if $\sigma < l \otimes \sigma$ for each $l \in L$ and each $\sigma \in S - \{\infty\}$.

A (finite) graph $G = (V,\ E)$ is annotated with a function w which maps edges of E into L. If an initial weight σ_0 is associated with node v_0, then the weight of a path terminating in v_0, $p = v_j v_{j-1} \cdots v_1 v_0$, is defined to be $w(p) \equiv w(v_j,\ v_{j-1}) \otimes \cdots \otimes \sigma_0$.

Sobrinho defines an asynchronous protocol for computing solutions to such path problems. Again this protocol is in the family of distributed Bellman-Ford algorithms. The algorithm itself forces paths to be simple — no repetitions of nodes along a path is allowed. Sobrhinho develops a sufficient condition (that all cycles are *free*, described below), which guarantees that this protocol terminates with a locally optimal solution. He shows that if an algebra is increasing, then this sufficient condition always holds.

A cycle $v_n v_{n-1} \cdots v_1 v_0 = v_n$ is free if for every $\alpha_0, \alpha_1 \cdots \alpha_n = \alpha_0$, with $\alpha_j \in S - \infty$, there is an i, $1 \leq i \leq n$, such that $\alpha_i < w(u_i,\ u_{i-1}) \otimes \alpha_{i-1}$. Thus a cycle that is not free is closely related to a dispute wheel of the SPP framework.

3 The quadrants model

We first review how path problems are solved using semirings [1–4]. Let $S = (S,\ \oplus,\ \otimes,\ 0,\ 1)$ be a semiring with the additive identity 0, which is also a multiplicative annihilator, and with multiplicative identity 1. We will assume that $\oplus$ is commutative and idempotent. The operations $\oplus$ and $\otimes$ can be extended in the usual way to matrices over S. For example, the multiplicative identity matrix is defined as follows.

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

Given a finite directed graph $G = (V,\ E)$ and a function $w : E \to S$ we can define the *adjacency* matrix A as

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

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

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

where the empty path is usually give the weight 1. Define $A^{(k)}$ as

$$A^{(k)} \equiv I \oplus A \oplus A^2 \oplus \cdots \oplus A^k.$$

The following facts are well known. Let $P(i, j)$ be the set of all paths in G from i to j. The set of paths made up of exactly k arcs is denoted by $P^k(i, j) \subseteq P(i, j)$. Then

$$A^k(i, j) = \sum_{p \in P^k(i,j)} w(p).$$

Note that the proof of this fact relies on the (left) distribution rule $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$. The set of paths made up of at most k arcs is denoted by $P^{(k)}(i, j) \subseteq P(i, j)$, and

$$A^{(k)}(i, j) = \sum_{p \in P^{(k)}(i,j)} w(p).$$

In particular, if there exists a q such that $A^{(q)} = A^{(q+1)}$, then

$$A^{(q)}(i,j) = \sum_{p \in P(i,j)} w(p)$$

represents a "global optimum" over all possible paths from i to j.

3.1 Can iteration be used to obtain a "local" optimum?

The matrix $B = A^{(q)}$ is a fixed point of the equation

$$B = I \oplus (A \otimes B),$$

which suggests the following *iterative* method of computing $A^{(k)}$.

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

Of course, using distribution we can see that $A^{(k)} = A^{[k]}$.

However, if distribution does not hold in S we may in some cases still be able to use this iterative method to compute a fixed point! Note that in this case matrix multiplication is not associative.

But how could such a fixed point B be interpreted? For $i \neq j$ we can see that

$$B(i,\ j) = \sum_{s \in N(i)} w(i,s) \otimes B(s,\ j)$$

where $N(i)$ is the set of all nodes adjacent to i, $N(i) = \{s \mid (i,\ s) \in E\}$. Such a fixed point may not represent a "global optimum" yet it can be interpreted as a Nash-like equilibrium point in which each node i obtains "locally optimal" values — node i computes its optimal value associated with paths to node j given only the values adopted by its neighbors. This closely models the type of routing solution we expect for BGP-like protocols.

3.2 Relating routing models

We have described the *algebraic* method of computing path weights $w(p)$. The literature on routing also includes the *functional* method, where we have a set of transforms $F \subseteq S \to S$ and each directed arc $(i,\ j)$ is associated with a function $f_{(i,\ j)} \in F$. The weight of a path $p = i_1, i_2, i_3, \cdots, i_k$ is then calculated as

$$w(p) = f_{(i_1,\ i_2)}(f_{(i_2,\ i_3)}(\ldots f_{(i_{k-1},\ i_k)}(a)\ldots)).$$

where a is some value *originated by the node* i_k. BGP is perhaps the best example of a functional approach to path weight computation.

The literature also contains two methods for path weight summarization. We outlined the *algebraic* approach above using a commutative and idempotent semigroup. The *ordered* method uses an order $\leq$ on S, and we take 'best weights' to mean minimal with respect to $\leq$. These two approaches are closely related (more below), but they are at the same time quite distinct. For example, minimizing the set $S = \{\alpha,\ \beta\}$ with respect to an order $\leq$ will result in a subset of S, whereas $\alpha \oplus \beta$ may not be an element of S. If α and β are weights associated with network paths p and q, then the best weight $\alpha \oplus \beta$ in the algebraic approach need not be associated with any one network path.

weight computation	weight summarization: **algebraic**	weight summarization: **ordered**
algebraic	NW — Bisemigroups $(S,\ \oplus,\ \otimes)$ Semirings [1–3] Non-distributive semirings [22, 23]	NE — Order Semigroups $(S,\ \leq,\ \otimes)$ Ordered semirings [24–26] QoS algebras [21]
functional	SW — Semigroup Transforms $(S,\ \oplus,\ F)$ Monoid endomorphisms [1, 2]	SE — Order Transforms $(S,\ \leq,\ F)$ Sobrinho structures [12, 13].

Fig. 1. The Quadrants Model of Algebraic Routing.

Figure 1 presents the four ways we can combine the algebraic and ordered approaches to weight summarization with the algebraic and functional approaches to weight computation. We discuss each in more detail.

The northwest (NW) quadrant contains bisemigroups of the form $(S,\ \oplus,\ \otimes)$. Semirings [1–3] are included in this class, although we do not insist that bisemigroups satisfy the axioms of a semiring. For example, we do not require that $\otimes$ distributes over $\oplus$.

A semigroup $(S,\ \otimes)$ can be translated to a set of functions using Cayley's left- or right-representation.

$$(S,\ \otimes) \xmapsto{\mathsf{cayley}} (S,\ F)$$

For example, with the left representation we associate a function f_a with each element $a \in S$ and define $f_a(b) = a \otimes b$. The semigroup $(S,\ \otimes)$ then becomes the set of functions structure $F = \{f_a \mid a \in S\}$. We can then use a Cayley representation to translate a bisemigroup $(S,\ \oplus,\ \otimes)$ into a semigroup transform $(S,\ \oplus,\ F)$,

$$(S,\ \oplus,\ \otimes) \xmapsto{\mathsf{cayley}} (S,\ \oplus,\ F)$$

If we start with a *semiring*, then we arrive in the SW quadrant at what Gondran and Minoux call an *algebra of endomorphisms* [1]. However, it is important to note that not all semigroup transforms arise in this way from semirings, and we do not require the properties of monoid endomorphisms.

The NE quadrant includes ordered semigroups, which have been studied extensively [24–26]. Such structures have the form $(S,\ \leq,\ \otimes)$, where $\otimes$ is *monotonic* with respect to $\leq$. That is, if $a \leq b$, then $c \otimes a \leq c \otimes b$ and $a \otimes c \leq b \otimes c$. Sobrinho [21] studied such structures (with total orders) in the context of Internet routing. In our framework, we require only that $\leq$ be a *pre-order* (reflexive and transitive), and we do not require monotonicity but infer it instead (which is why we call these structures *order semigroups* rather than *ordered semigroups*).

Turning to the SE quadrant of Figure 1, we have structures of the form $(S,\ \leq,\ F)$, which include Sobrinho's routing algebras [12] as a special case. Sobrinho algebras (as defined in [13]) have the form $(S, \preceq,\ L,\ \otimes)$, where $\preceq$ is a *preference relation* over signatures (that is, a total pre-order), L is a set of *labels*, and $\otimes$ is a function mapping $L \times S$ to S. We can map this to an order transform $(S, \preceq,\ F_L)$ with $F_L = \{g_\lambda \mid \lambda \in L\}$, where $g_\lambda(a) = \lambda \otimes a$. Thus we can think of the pair $(L,\ \otimes)$ as a means of indexing the set of transforms F_L. In addition to this slightly higher level of abstraction, we do not insist that $\preceq$ be total.

Commutative, idempotent monoids can be translated into orders,

$$(S,\ \oplus) \xmapsto{\mathsf{natord}} (S,\ \leq)$$

in two ways, either $a \leq^{\oplus}_{R} b \equiv b = a \oplus b$, or $a \leq^{\oplus}_{L} b \equiv a = a \oplus b$. These orders are clearly dual, with $a \leq_L b$ iff $b \leq_R a$. If 1 is also an additive annihilator, then we have for all $a \in S$, $0 \leq^{\oplus}_{R} a \leq^{\oplus}_{R} 1$ and $1 \leq^{\oplus}_{L} a \leq^{\oplus}_{L} 0$, and the orders are *bounded*.

Using the natord and cayley translations we can move from the NW to the SE quadrants of Figure 1,

$$\begin{tikzcd}
(S,\ \oplus,\ \otimes) \arrow[r, mapsto, "\mathsf{natord}"] \arrow[d, mapsto, "\mathsf{cayley}"] & (S,\ \leq,\ \otimes) \arrow[d, mapsto, "\mathsf{cayley}"] \\
(S,\ \oplus,\ F) \arrow[r, mapsto, "\mathsf{natord}"] & (S,\ \leq,\ F)
\end{tikzcd}$$

We can use these translations to investigate how properties appropriate to each quadrant are related. For example, an order transform is *increasing* when for all a and f we have

$$a \neq \top \implies a < f(a),$$

where $\top$ is the top element of the order. Pushing this property through the above translations yields a definition of increasing for each quadrant.

$$\begin{tikzcd}
\begin{array}{c}(a \neq 0 \implies a = a \oplus (b \otimes a)) \wedge \\ (b \otimes a = a \oplus (b \otimes a) \implies a = 0)\end{array} \arrow[r, mapsto, "\text{left-natord}"] \arrow[d, mapsto, "\text{left-cayley}"] & a \neq \top \implies a < b \otimes a \arrow[d, mapsto, "\text{left-cayley}"] \\
\begin{array}{c}(a \neq 0 \implies a = a \oplus f(a)) \wedge \\ (f(a) = a \oplus f(a) \implies a = 0)\end{array} \arrow[r, mapsto, "\text{left-natord}"] & a \neq \top \implies a < f(a)
\end{tikzcd}$$

For example, a *left increasing bisemigroup* is a bisemigroup where for all a and b we have $a \neq 0 \implies a = a \oplus (b \otimes a))$ and $b \otimes a = a \oplus (b \otimes a) \implies a = 0$. In other words, where $a \neq 0 \implies a <_L^{\oplus} b \otimes a$. In this paper we will use the term *increasing bisemigroup* to mean left increasing bisemigroup.

3.3 Quadrants model and metarouting

Griffin and Sobrinho [13] proposed *metarouting* as a means of defining routing protocols in a high-level and declarative manner. Metarouting is based on using a *metalanguage* to specify routing languages. Algebraic properties required by algorithms are derived automatically from a metalanguage specification, in much the same way that types are derived in modern programming languages.

It is envisioned that metarouting will be used to specify (and implement) new routing protocols as follows. Assume that a fixed menu of generic routing algorithms has been implemented, each associated with a specific set of correctness requirements. First, the algebraic component is defined using the metalanguage, resulting in a set of automatically inferred properties. Next, the routing language can then be associated with any algorithm whose requirements set is contained in the set of inferred properties. This checking could be done at protocol design time or later at network configuration time. A metarouting implementation must then compile the specification and algorithm choices into efficient code for representing routing tables, calculating best routes, parsing and packing binary on-the-wire representations and so on. Protocol compilation is a topic of ongoing research.

The quadrants model of Figure 1 has been adopted as the algebraic basis for metarouting. Rather than confining metarouting to the SE quadrant, as was done in [13], the metarouting project is now attempting to capture structures and operations in each of the four quadrants, as well as operations between quadrants. In this model, properties are not required but inferred.

4 A relational reformulation in terms of bisemigroups

We reformulate the theories described in Section 2 in terms of bisemigroups. This is not meant to be completely faithful in every detail, rather it represents an attempt to recast the essential ideas in a purely algebraic setting.

Let $S = (S,\ \oplus,\ \otimes)$ be a bisemigroup. Throughout this section we will assume that $\oplus$ is idempotent, commutative, and selective ($a \oplus b = a \vee a \oplus b = b$), and that both 0 and 1 exist and that 0 is a multiplicative annihilator. Note that since $\oplus$ is idempotent, commutative, and selective it follows that $\leq_L^{\oplus}$ is a total order.

Let A be an adjacency matrix over S. Since $\oplus$ is selective, for each $i \neq j$ there exists $s^k_{(i,j)} \in N(i) \equiv \{s \mid (i, s) \in E\}$ such that

$$A^{[k+1]}(i,\ j) = \sum_{s \in N(i)} w(i, s) \otimes B(s,\ j) = w(i, s^k_{(i,j)}) \otimes A^{[k]}(s^k_{(i,j)},\ j)$$

We assume that we have a deterministic method of selecting a unique $s^k_{(i,j)}$.

For the iterative algorithm we define a particular sequence of values that is called *the history of* $A^{[k]}(i,\ j)$. Histories are inspired by constructs of the same name in [27] that record causal chains of events in an asynchronous protocol. Here, the history of $A^{[k]}(i,\ j)$, denoted $H^{[k]}(i,\ j)$, will in some sense explain how the value $A^{[k]}(i,\ j)$ came to be adopted at step k of the iteration.

$$H^{[0]}(i,\ j) = (1)$$

$$H^{[k+1]}(i,\ j) = \begin{cases} H^{[k]}(i,\ j) & \text{if } A^{[k]}(i,\ j) = A^{[k+1]}(i,\ j), \\ H^{[k]}(s^k_{(i,j)},\ j),\ A^{[k+1]}(i,\ j) & \text{if } A^{[k+1]}(i,\ j) <_L^{\oplus} A^{[k]}(i,\ j), \\ H^{[k]}(s^{k-1}_{(i,j)},\ j),\ A^{[k]}(i,\ j) & \text{if } A^{[k]}(i,\ j) <_L^{\oplus} A^{[k+1]}(i,\ j). \end{cases}$$

Note that if $A^{[k+1]}(i,\ j) <_L^{\oplus} A^{[k]}(i,\ j)$, then node i obtained a more preferred value at step $k+1$. In this case the history $H^{[k+1]}(i,\ j)$ is the sequence $H^{[k]}(s^k_{(i,j)},\ j),\ A^{[k+1]}(i,\ j)$, where $H^{[k]}(s^k_{(i,j)},\ j)$ is a history explaining how value $A^{[k]}(s^k_{(i,j)},\ j)$ was adopted at state k. Since $A^{[k+1]}(i,\ j) = w(i, s^k_{(i,j)}) \otimes A^{[k]}(s^k_{(i,j)},\ j)$, the complete history explains how $A^{[k+1]}(i,\ j)$ was adopted at step $k+1$.

On the other hand, when $A^{[k]}(i,\ j) <_L^{\oplus} A^{[k+1]}(i,\ j)$, then node i lost a more preferred value at step $k+1$. In this case the history $H^{[k+1]}(i,\ j)$ is the sequence $H^{[k]}(s^{k-1}_{(i,j)},\ j),\ A^{[k]}(i,\ j)$, which ends in the value lost at step $k+1$. Since this lost value is $A^{[k]}(i,\ j) = w(i, s^{k-1}_{(i,j)}) \otimes A^{[k-1]}(s^{k-1}_{(i,j)},\ j)$, the sequence $H^{[k]}(s^{k-1}_{(i,j)},\ j)$ explains how node $s^{k-1}_{(i,j)}$ came to adopt $A^{[k]}(s^{k-1}_{(i,j)},\ j)$ at step k, thus forcing node i to abandon $A^{[k]}(i,\ j)$ at step $k+1$.

Of course this last type of history depends on *violations* of monotonicity,

$$\forall a, b, c \in S : a \leq_L^{\oplus} b \rightarrow c \otimes a \leq_L^{\oplus} c \otimes b.$$

We define the *dispute relation* D_S to record such violations,

$$D_S \equiv \{(a,\ c \otimes b) \mid a, b, c \in S,\ a \leq_L^{\oplus} b \wedge c \otimes b <_L^{\oplus} c \otimes a\}$$

Of course, in the case that S is monotonic, then is D_S is empty. In addition we define a relation

$$T_S \equiv \{(a,\ b \otimes a) \mid a,\ b \in S,\ b \neq 1\}.$$

Note that T_S is the anti-reflexive sub-relation of $\leq_R^{\otimes}$, (using $\otimes$!) where

$$a \leq_R^{\otimes} b \equiv \exists c \in S : b = c \otimes a.$$

The *generalized dispute digraph* is then defined as the relation

$$\mathfrak{D}_S = (T_S \cup D_S)^{tc},$$

where tc denotes the transitive closure.

Note that if $(a,\ b \otimes a) \in T_S$, then if S is increasing we have $a <_L^{\oplus} b \otimes a$. If $(a,\ c \otimes b) \in D_S$, then $a \leq_L^{\oplus} b$, and if S is increasing then $b <_L^{\oplus} c \otimes b$, so $a <_L^{\oplus} c \otimes b$. Thus we have proved the following.

Lemma 1. *If S is increasing, then $\mathfrak{D}_S \subseteq <_L^{\oplus}$.*

A $\mathfrak{D}_S$ *sequence* σ is any non-empty sequence of values over S such that if $\sigma = a_1, a_2, \ldots, a_k$, for $2 \leq k$, then for each $1 \leq i < k$ we have $(a_i,\ a_{i+1}) \in \mathfrak{D}_S$.

Lemma 2. *For each k, i, and j, $H^{[k]}(i,\ j)$ is a $\mathfrak{D}_S$ sequence.*

Lemma 3. *Suppose that $A^{[k]}(i,\ j) \neq A^{[k+1]}(i,\ j)$, then $\mid H^{[k+1]}(i,\ j) \mid = k + 1$.*

Theorem 1. *If S is an increasing bisemigroup and only simple paths are allowed, then there must exist a k such that $A^{[k]} = A^{[k+1]}$. Thus $B = A^{[k]}$ is a solution to the equation $B = I \oplus (A \otimes B)$.*

As mentioned in Section 2, the SPP theory also used the concept of *dispute wheels* while Sobrinho's theory used the related concept of *non-free cycles*. We now show how these concepts are related to generalized dispute digraphs.

Dispute wheels and non-free cycles can both be captured relationally [28]. Let

$$\mathfrak{R}_S \equiv (\leq_R^{\otimes} \circ <_L^{\oplus})^{tc}.$$

Lemma 4. *Suppose that $a_1\ \mathfrak{R}_S\ a_2\ \mathfrak{R}_S\ a_3$. That is, there exists b_1 and b_2 such that*

$$a_1 \leq_R^{\otimes} b_1 \otimes a_1 <_L^{\oplus} a_2 \leq_R^{\otimes} b_2 \otimes a_2 <_L^{\oplus} a_3.$$

Then either $a_1 \leq_R^{\otimes} a_3$ or $(b_1 \otimes a_1,\ b_2 \otimes a_2) \in \mathfrak{D}_S$.

Corollary 1. *If $(a,\ a) \in \mathfrak{R}_S$, then $(a,\ a) \in \mathfrak{D}_S$.*

In particular, if S is an increasing bisemigroup, then we know that all cycles are free and that dispute wheels cannot exist.

5 Open problems and discussion

We do not mean to suggest that the only possible application of increasing bisemigroups is in network routing. Non-distributive semirings have been considered in other types of path optimization problems such as circuit layout [22, 23], and there may be problems in areas such as operations research to which increasing bisemigroups could be applied. This suggests several open problems.

5.1 Problem 1: dropping selectivity

To what extent can the results of the previous section be extended to non-selective bisemigroups?

The assumption that $\leq_L^{\oplus}$ is a total order pervades the proof techniques we use. However, there is good motivation for relaxing the totality condition and allowing for a non-selective $\oplus$. This is important for the metarouting effort [13], since many of the translations going from eastern to western quadrants of Figure 1 involve a *min-set construction*, which does not, in general, result in an additive semigroup that is selective.

Min-set constructions are a type of *reduction* defined by Wongseelashote [29]. For any finite subset $A \subseteq S$, let $\min_{\leq}(A) \equiv \{x \in A \mid \forall y \in A : \neg(y < x)\}$, be the *minimal subset of* A. Here $y < x$ means $y \leq x \wedge \neg(x \leq y)$ and so the operation is well defined even for pre-orders. The set of all minimal sets is denoted as $\min_{\leq}(S) \equiv \{A \subseteq S \mid A \text{ is finite and } \min_{\leq}(A) = A\}$. If $A, B \in \min_{\leq}(S)$, then define $A \oplus B \equiv \min_{\leq}(A \cup B)$. Thus we can construct a commutative and idempotent semigroup $(\min_{\leq}(S),\ \oplus)$ from a pre-ordered set $(S,\ \leq)$.

If $a \neq b$ and both are in a minimal set $A = \min_{\leq}(A)$, then either they are equivalent $a \sim b$ ($a \leq b$ and $b \leq a$), or they are incomparable $a \sharp b$ ($\neg(a \leq b)$ and $\neg(b \leq a)$). We believe that min-set semigroups closely model the way Internet routing protocols compute *equal cost multi-paths* and they way they can partition routes into distinct *service classes*. Equal cost multi-paths arise when the weights of at least two distinct paths are equivalent, $w(p) \sim w(q)$. *Load balancing* can then be implemented by forwarding traffic along both paths p and q (today this is usually accomplished with a function that selects paths by hashing on information such as IP addresses and port numbers). In the case that $w(p) \sharp w(q)$, then we can interpret this as meaning that the *data traffic itself* must contain information that can be used to select path p or path q. As a simple example, suppose that weights $w(p)$ somehow contain a destination address and that $w(p) \sharp w(q)$ arises only when these addresses differ. In this case the destination address carried in a data packet is used to select a path. For another example, suppose that weights $w(p)$ contain a *type of service* and that $w(p) \sharp w(q)$ means the associated paths support different types of service. In this case the data traffic would be expected to contain a type-of-service field used to select an appropriate path.

5.2 Problem 2: complexity bounds

What is the computational complexity (number of steps required) of the iterative algorithm for increasing bisemigroups? We suspect that the worst case complexity will involve an exponential in the number of nodes in the graph. However, this may not be the case for all (non-distributive) increasing bisemigroups.

As mentioned, previous complexity analysis of BGP has invariably involved distributed (asynchronous) algorithms. Yet an asynchronous version of our iterative algorithm can have exponential worst-case complexity even in the case of shortest-paths routing due to the non-deterministic interleaving of routing messages (see for example [30]). Here we are asking instead for the inherent complexity associated with an increasing bisemigroups, in terms of the complexity of our iterative algorithm alone.

Acknowledgments

This paper benefited greatly from discussions with Gordon Wilfong and João Luís Sobrinho. We also thank John Billings, Martin Hyland, Philip Taylor, and Barney Stratford for their helpful comments. A. Gurney is supported by a Doctoral Training Account from the Engineering and Physical Sciences Research Council (EPSRC). T. Griffin is grateful for support under the the Cisco Collaborative Research Initiative.

References

1. Gondran, M., Minoux, M.: Graphes, dioïdes et semi-anneaux: Nouveaux modéles et algorithmes. Tec & Doc (2001)
2. Gondran, M., Minoux, M.: Graphs and Algorithms. Wiley (1984)
3. Carré, B.: Graphs and Networks. Oxford University Press (1979)
4. Backhouse, R., Carr, B.: Regular algebra applied to path-finding problems. J. Inst. Math. Appl. **15** (1975) 161–18
5. Rekhter, Y., Li, T.: A Border Gateway Protocol. RFC 1771 (BGP version 4) (March 1995)
6. Stewart, J.W.: BGP4: Inter-Domain Routing in the Internet. Addison-Wesley (1999)
7. Halabi, S., McPherson, D.: Internet Routing Architectures. Second edn. Cisco Press (2001)
8. Huston, G.: Interconnection, peering and settlements: Parts I and II. Internet Protocol Journal **2**(1 and 2) (March, June 1999)
9. K.Varadhan, R.Govindan, Estrin., D.: Persistent route oscillations in inter-domain routing. Computer Networks **32** (2000) 1–16 based on a 1996 technical report.
10. Systems, C.: Endless BGP convergence problem in Cisco IOS software releases. Field Note, October 10 2001, `http://www.cisco.com/warp/public/770/fn12942.html`
11. Griffin, T.G., Huston, G.: RFC 4264: BGP Wedgies (November 2005) IETF.
12. Sobrinho, J.L.: An algebraic theory of dynamic network routing. IEEE/ACM Transactions on Networking **13**(5) (October 2005) 1160–1173
13. Griffin, T.G., Sobrinho, J.L.: Metarouting. In: Proc. ACM SIGCOMM. (August 2005)
14. Griffin, T.G., Gao, L., Rexford, J.: Inherently safe backup routing with BGP. In: Proc. IEEE INFOCOM. (April 2001)
15. Gurney, A., Griffin, T.G.: Lexicographic products in metarouting. In: Proc. Inter. Conf. on Network Protocols. (October 2007)
16. Griffin, T.G., Shepherd, F.B., Wilfong, G.: Policy disputes in path-vector protocols. In: Proc. Inter. Conf. on Network Protocols. (November 1999)
17. Griffin, T.G., Shepherd, F.B., Wilfong, G.: The stable paths problem and interdomain routing. IEEE/ACM Transactions on Networking **10**(2) (April 2002) 232–243

18. Griffin, T.G., Wilfong, G.: On the correctness of IBGP configuration. In: Proc. ACM SIGCOMM. (September 2002)
19. Griffin, T.G., Wilfong, G.: An analysis of the MED oscillation problem in BGP. In: Proc. Inter. Conf. on Network Protocols. (2002)
20. Sobrinho, J.L.: Network routing with path vector protocols: Theory and applications. In: Proc. ACM SIGCOMM. (September 2003)
21. Sobrinho, J.L.: Algebra and algorithms for QoS path computation and hop-by-hop. IEEE/ACM Transactions on Networking **10**(4) (August 2002) 541–550
22. Lengauer, T., Theune, D.: Unstructured path problems and the making of semirings. Lecture Notes in Computer Science **519** (1991) 189–200
23. Lengauer, T., Theune, D.: Efficient algorithms for path problems with general cost criteria. Lecture Notes in Computer Science **510** (1991) 314–326
24. Fuchs, L.: Partially Ordered Algebraic Systems. Addison-Wesley (1963)
25. Birkhoff, G.: Lattice Theory, 3rd edition. Amer. Math. Soc. (1967)
26. Johnson, R.E.: Free products of ordered semigroups. Proceedings of the American Mathematical Society **19**(3) (1968) 697–700
27. Griffin, T., Wilfong, G.: A safe path vector protocol. In: Proc. IEEE INFOCOM. (March 2000)
28. Chau, C., Gibbens, R., G.Griffin, T.: Towards a unified theory of policy-based routing. In: Proc. IEEE INFOCOM. (April 2006)
29. Wongseelashote, A.: Semirings and path spaces. Discrete Mathematics **26**(1) (1979) 55–78
30. Karloff, H.: On the convergence time of a path-vector protocol. In: ACM-SIAM Symposium on Discrete Algorithms (SODA). (2004)

A Proofs

Lemma 3 The proof is by induction on k. The base case is clear. Suppose every entry of $H^{[k]}$ is a $\mathfrak{D}_S$ sequence. The analysis of $H^{[k+1]}(i,\ j)$ is in three cases.
Case 1 : $A^{[k]}(i,\ j) = A^{[k+1]}(i,\ j)$. Then $H^{[k+1]}(i,\ j) = H^{[k]}(i,\ j)$ and the claim holds.
Case 2 : $A^{[k+1]}(i,\ j) <_L^{\oplus} A^{[k]}(i,\ j)$, so we have

$$\begin{aligned} w(i, s^k_{(i,j)}) \otimes A^{[k]}(s^k_{(i,j)},\ j) &<_L^{\oplus} w(i, s^{k-1}_{(i,j)}) \otimes A^{[k-1]}(s^{k-1}_{(i,j)},\ j) \\ &\leq_L^{\oplus} w(i, s^k_{(i,j)}) \otimes A^{[k-1]}(s^k_{(i,j)},\ j). \end{aligned}$$

In this case $H^{[k+1]}(i,\ j) = H^{[k]}(s^k_{(i,j)},\ j),\ A^{[k+1]}(i,\ j)$. There are three sub-cases to consider.
Case 2.1 : $A^{[k-1]}(s^k_{(i,j)},\ j) = A^{[k]}(s^k_{(i,j)},\ j)$. This is not possible.
Case 2.2 : $A^{[k]}(s^k_{(i,j)},\ j) <_L^{\oplus} A^{[k-1]}(s^k_{(i,j)},\ j)$. Then $(A^{[k]}(s^k_{(i,j)},\ j),\ w(i, s^k_{(i,j)}) \otimes A^{[k]}(s^k_{(i,j)},\ j))$ is in T_S, and since $H^{[k]}(s^k_{(i,j)},\ j)$ ends in $A^{[k]}(s^k_{(i,j)},\ j)$, it follows that $H^{[k+1]}(i,\ j)$ is a $\mathfrak{D}_S$ sequence.
Case 2.3 : $A^{[k-1]}(s^k_{(i,j)},\ j) <_L^{\oplus} A^{[k]}(s^k_{(i,j)},\ j)$. Then $(A^{[k-1]}(s^k_{(i,j)},\ j),\ A^{[k+1]}(i,\ j))$ is in D_S, and since $H^{[k]}(s^k_{(i,j)},\ j)$ ends in the value $A^{[k-1]}(s^k_{(i,j)},\ j)$, it follows that $H^{[k+1]}(i,\ j)$ is a $\mathfrak{D}_S$ sequence.

Case 3 : $A^{[k]}(i,\ j) <_L^{\oplus} A^{[k+1]}(i,\ j)$, so we have

$$\begin{aligned} w(i,\ s_{(i,j)}^{k-1}) \otimes A^{[k-1]}(s_{(i,j)}^{k-1},\ j) &<_L^{\oplus} w(i,\ s_{(i,j)}^{k}) \otimes A^{[k]}(s_{(i,j)}^{k},\ j) \\ &\leq_L^{\oplus} w(i,\ s_{(i,j)}^{k-1}) \otimes A^{[k]}(s_{(i,j)}^{k-1},\ j). \end{aligned}$$

In this case $H^{[k+1]}(i,\ j) = H^{[k]}(s_{(i,j)}^{k-1},\ j),\ A^{[k]}(i,\ j)$. There are three sub-cases to consider.
Case 3.1 : $A^{[k-1]}(s_{(i,j)}^{k-1},\ j) = A^{[k]}(s_{(i,j)}^{k-1},\ j)$. This is not possible.
Case 3.2 : $A^{[k]}(s_{(i,j)}^{k-1},\ j) <_L^{\oplus} A^{[k-1]}(s_{(i,j)}^{k-1},\ j)$. Then

$$(A^{[k]}(s_{(i,j)}^{k-1},\ j),\ w(i,\ s_{(i,j)}^{k-1}) \otimes A^{[k-1]}(s_{(i,j)}^{k-1},\ j)) \in D_S,$$

and since $H^{[k]}(s_{(i,j)}^{k-1},\ j)$ ends in $A^{[k]}(s_{(i,j)}^{k-1},\ j)$, $H^{[k+1]}(i,\ j)$ is a $\mathfrak{D}_S$ sequence.
Case 3.3 : $A^{[k-1]}(s_{(i,j)}^{k-1},\ j) <_L^{\oplus} A^{[k]}(s_{(i,j)}^{k-1},\ j)$. Then $H^{[k]}(s_{(i,j)}^{k-1},\ j)$ ends in the value $A^{[k-1]}(s_{(i,j)}^{k-1},\ j)$, and

$$(A^{[k-1]}(s_{(i,j)}^{k-1},\ j),\ w(i,\ s_{(i,j)}^{k-1}) \otimes A^{[k-1]}(s_{(i,j)}^{k-1},\ j)) \in T_S,$$

so $H^{[k+1]}(i,\ j)$ is a $\mathfrak{D}_S$ sequence.
Lemma 3 The proof is by induction on k. For $k = 0$, suppose $A^{[0]}(i,\ j) \neq A^{[1]}(i,\ j)$. Since $A^{[1]}(i,\ j) = w(i, s_{(i,j)}^0) \otimes A^{[0]}(s_{(i,j)}^0,\ j) = w(i, s_{(i,j)}^0) \otimes I(s_{(i,j)}^0,\ j)$ it must be that $s_{(i,j)}^0 = j$ and $A^{[1]}(i,\ j) = w(i,\ j)$. Therefore $H^{[1]}(i,\ j) = 1,\ w(i,\ j)$, and $\mid H^{[1]}(i,\ j) \mid = k+1$.

Next, suppose that $A^{[k]}(i,\ j) \neq A^{[k+1]}(i,\ j)$. There are two cases to consider.
Case 1 : $A^{[k+1]}(i,\ j) <_L^{\oplus} A^{[k]}(i,\ j)$. In this case

$$H^{[k+1]}(i,\ j) = H^{[k]}(s_{(i,j)}^{k},\ j),\ A^{[k+1]}(i,\ j).$$

As in the proof of Lemma 3, it must be that $A^{[k-1]}(s_{(i,j)}^{k},\ j) \neq A^{[k]}(s_{(i,j)}^{k},\ j)$. By induction, $\mid H^{[k]}(s_{(i,j)}^{k} \mid = k$, so $\mid H^{[k+1]}(i,\ j) \mid = k+1$.
Case 2 : $A^{[k]}(i,\ j) <_L^{\oplus} A^{[k+1]}(i,\ j)$, so we have

$$H^{[k+1]}(i,\ j) = H^{[k]}(s_{(i,j)}^{k-1},\ j),\ A^{[k]}(i,\ j).$$

As in the proof of Lemma 3, it must be that $A^{[k-1]}(s_{(i,j)}^{k-1},\ j) \neq A^{[k]}(s_{(i,j)}^{k-1},\ j)$. By induction, $\mid H^{[k]}(s_{(i,j)}^{k-1},\ j) \mid = k$, so $\mid H^{[k+1]}(i,\ j) \mid = k+1$.
Theorem 1 Suppose that k does not exist. Since only simple paths are allowed, the set of values $w(p)$ for all paths p is finite. Since histories must grow without bound there must at some point be an a such that $(a,\ a) \in \mathfrak{D}_S$, which contradicts Lemma 1.