Towards a Unified Theory of Policy-Based Routing

Chi-kin Chau Richard Gibbens Timothy G. Griffin
Computer Laboratory
University of Cambridge
{chi-kin.chau, richard.gibbens, timothy.griffin}@cl.cam.ac.uk

***Abstract*— We use the term *policy-based routing* to refer collectively to the Stable Paths Problem, Sobrinho's Routing Algebras, and to classical Path Algebras (semi-rings used to generalise minimum-weight routing). These theories all contain sufficient conditions that ensure the existence of solutions (stable routings) for labelled graphs. We attempt to provide a unified theory from which all of these seemingly disparate sufficient conditions can be derived. Our theory is based purely on abstract relations and their properties and not on the syntactic or axiomatic details of the policy-based theories.**

I. Introduction

The Border Gateway Protocol (BGP) [1] has stimulated renewed interest in routing protocols and in formal methods to prove their convergence properties. The Stable Paths Problem (SPP) [2] grew out of an effort to formalise the underlying semantics of BGP in terms of a simple graph problem. Another approach was taken by Sobrinho with his definition of *routing algebras* [3], [4], which form the basis of metarouting [5]. Routing algebras can be seen as further generalisation of minimum-weight routing [6], [7] in the tradition of *path algebras* introduced over thirty years ago as the theoretical basis for generalised routing problems [8]–[10].

We will refer to path algebras, routing algebras, and the Stable Paths Problem as theories of *policy-based routing*. Each of these theories presents sufficient conditions on *labelled graphs* that ensure a solution to a given routing problem (stable routing) exists. We call such a condition an *instance condition*. For SPPs one instance condition is the lack of a structure called a *dispute wheel*, for routing algebras there is notion of a *free graph*, and for path algebras we have the *absorptive graph* condition (these and other conditions will be restated in the paper). For example, in classical minimum-weight routing, the instance condition is that there exists no negative-weight directed cycles.

In addition, path algebras and routing algebras have system-level sufficient conditions that ensure that the instance condition will hold in every labelled graph. We call this type of condition a *universal condition*. (The SPP formalism is restricted to instances, so it is not associated with a universal condition.) In path algebras the conditions *nilpotent* and *super-unitary* serve as universal conditions. In routing algebras, one universal condition is called *monotonicity*. For classical minimum-weight routing the constraint that all link-weights be non-negative serves as a universal condition. Note that in this case no cycles can have negative weight, so the instance condition holds for all labelled graphs. However, a particular labelled graphs could satisfy the instance condition even though negative link-weights are allowed in general.

In this paper we attempt to unify these three policy-based routing theories into a single theory. How this might be done was first hinted at in [11], where the instance condition on SPPs was stated in terms of properties of a certain derived *relation on paths*. Two instance-specific relations were defined on paths — one a sub-path relationship and the other a preference relationship — and the derived relation was then defined as the transitive closure of a combination of the instance-specific relations. The instance condition was stated in terms of the derived relation being "almost" a partial order (see [11] for the technical details).

Here we take a similar, although more general, approach. We start with *policy structures*, which are sets together with two relations and another relation derived from them. Our universal condition is that the derived relation be *anti-reflexive*. For each labelled graph (paths labelled with values associated with a policy structure), we construct a *routing structure* made up of paths, two relations on sets of paths, and a derived relation. The instance condition is again that the derived relation be anti-reflexive. We then show that any anti-reflexive instance has a solution (stable routing) and that if the universal condition (anti-reflexivity) holds, then the instance condition (again, anti-reflexivity) holds in all labelled graphs.

In what sense does this provide a unified theory of policy-based routing? Take routing algebras as an example. For each routing algebra we construct a policy structure, and we show that if the universal condition holds on the routing algebra, then the policy structure has an anti-reflexive derived relation. We then show that if the routing algebra's instance condition on labelled graphs holds, then the corresponding routing structure has an anti-reflexive derived relation and so has a solution. Finally, we show that a solution to the routing structure must also contain a solution to the original labelled graph. These results are summarised in Table I.

Our unification is not all-encompassing — we are required to restrict routing algebras to those using total orders, rather than the more general preference order on paths defined by Sobrinho, and we require that our path algebras satisfy additional constraints that ensure they correspond to *distributive lattices*. We argue that these constraints are in some sense natural, and that most path algebra examples conform to these constraints. Finally, our unified theory requires a slight generalisation of the SPPs beyond those defined in [2].

TABLE I
SUMMARY OF SUFFICIENT CONDITIONS FOR POLICY-BASED THEORIES.

	Minimum-weight Routing	Path Algebra	Stable Paths Problem	Sobrinho's Routing Algebra	**Policy/Routing Relation**
Universal Condition	Non-negative Weights	Super-unitary/ Nilpotent Algebra	–	Monotonicity	**Anti-reflexivity**
Instance Condition	No Negative Cycle	Absorptiveness	No Dispute Wheel	Freeness	**Anti-reflexivity**

II. RELATIONS AND ORDERS

Given a set X, *a relation over* X is any subset of $X \times X$. If R is a relation, then we write $(x, y) \in \mathsf{R}$ (resp. $(x, y) \notin \mathsf{R}$) as $x\mathsf{R}y$ (resp. $x \not\mathsf{R} y$). We use the following terminology for relations R:

- *reflexive*, if $x\mathsf{R}x$ for all $x \in X$,
- *anti-reflexive*, if $x\not\mathsf{R}x$ for all $x \in X$,
- *total*, if $x\mathsf{R}y$ or $y\mathsf{R}x$ for all $x, y \in X$,
- *transitive*, if ($x\mathsf{R}y$ and $y\mathsf{R}z \Rightarrow x\mathsf{R}z$) for all $x, y, z \in X$,
- *anti-symmetric*, if ($x\mathsf{R}y$ and $y\mathsf{R}x \Rightarrow x = y$) for all $x, y \in X$.

The strict relation of R is $\mathsf{R}_S \triangleq \{(x, y) \in \mathsf{R} \mid y\not\mathsf{R}x\}$. We will be interested in relations that have various combinations of these properties.

- A *preorder* is a reflexive and transitive relation.
- A *partial order* is an anti-symmetric preorder.
- A *total order* is a total partial order.
- A *preference order* is total preorder.

Given two relations R_1 and R_2 over the same set X, we define the *join of* R_1 *with* R_2 to be the relation R where $x\mathsf{R}z$ if and only if there exists some $y \in X$ such that $x\mathsf{R}_1y$ and $y\mathsf{R}_2z$. We use the notation

$$\mathsf{R} \triangleq \mathsf{R}_1 \bowtie \mathsf{R}_2$$

to denote this relation.

If R is a preorder with a strict relation R_S over X and $A \subset X$, then we define

$$\min_{\mathsf{R}} A \triangleq \{x \in A \mid \text{there exists no } y \in A \text{ such that } y\mathsf{R}_S x\}.$$

Note that if R is total, then

$$\min_{\mathsf{R}} A = \{x \in A \mid \text{for all } y \in A, x\mathsf{R}y\}.$$

For a partial order $\leq$, the elements of $\min_{\leq} A$ will be mutually incomparable, and for preference orders $\leq$, the elements of $\min_{\leq} A$ will be equally preferred.

III. A BRIEF SURVEY OF ROUTING THEORIES

The simplest routing theory is minimum-weight routing, which is implemented by attaching a number to each connection in the network, and the task is to decide a path for each vertex to a specific origin with the smallest sum of weights on all the connections along the path. It is well-known that there are two conditions that guarantee the existence of routing solutions — non-negative weights and absence of a negative cycle. The condition of non-negative weights is universal to all instances of networks, whereas the absence of a negative cycle is instance-specific. Many routing theories are motivated to generalise minimum-weight routing by developing different formalisms. A common goal among them is to estabish generalised versions of non-negative weights and absence of a negative cycle.

A. Graphs

A network is represented as a rooted directed graph $\mathcal{G} = \langle \mathcal{V},\ \mathcal{E},\ v_0 \rangle$, with a designated vertex $v_0 \in \mathcal{V}$, called the *origin*, where $\mathcal{V}$ and $\mathcal{E}$ are finite sets, and every $v \in \mathcal{V}$ is connected to v_0. Let $\mathcal{P}(v_2,\ v_1)$ be the set of all the paths in $\mathcal{G}$ from v_2 to v_1 (including non-simple paths). Let $\mathcal{P}(v_0) = \bigcup_{v \in \mathcal{V}} \mathcal{P}(v,\ v_0)$.

We denote $v_1, v_2, ..., v_k$ as some vertices in $\mathcal{V}$. A (directed) path is a string $e_k e_{k-1} \cdots e_1$ where $e_k, e_{k-1}, ..., e_1 \in \mathcal{E}$, and e_i is an edge from v_{i+1} to v_i for some $v_1, v_2, ..., v_{k+1} \in \mathcal{V}$. We also write $v_k v_{k-1} \cdots v_1$ as a path that sequentially transverses from v_k to v_1. A simple path in $\mathcal{G}$ is a path with no repeated vertex. Sometimes, we write v as a path consisting no edge and a single vertex v.

If $P \in \mathcal{P}(v_3,\ v_2)$ and $Q \in \mathcal{P}(v_2,\ v_1)$, then PQ will denote the path in $\mathcal{P}(v_3,\ v_1)$ that corresponds to the concatenation of paths P and Q.

B. Path Algebra

The literature on path algebras (semi-rings) and routing is vast, and we cite only a representative sample [6], [8]–[10], [12]–[14].

A *path algebra* is a system of

$$\mathfrak{B} = \langle \mathcal{X},\ \oplus,\ \otimes,\ \bar{0},\ \bar{1} \rangle,$$

where $\mathcal{X}$ is a set, $\oplus$ and $\otimes$ are binary operations over $\mathcal{X}$, and $\bar{0}$ and $\bar{1}$ are distinguished elements of $\mathcal{X}$. A path algebra must conform to the following axioms. For all $a, b, c \in \mathcal{X}$,

- ($\oplus$-Commutivity) $a \oplus b = b \oplus a$,
- ($\oplus$-Associativity) $(a \oplus b) \oplus c = a \oplus (b \oplus c)$,
- ($\oplus$-Identity) there exists $\bar{0} \in \mathcal{X}$ such that $a \oplus \bar{0} = a$,
- ($\oplus$-Idempotency) $a \oplus a = a$,
- ($\otimes$-Associativity) $(a \otimes b) \otimes c = a \otimes (b \otimes c)$,
- ($\otimes$-Identity) there exists $\bar{1} \in \mathcal{X}$ such that $a \otimes \bar{1} = a$ and $\bar{1} \otimes a = a$,
- ($\otimes$-Annihilator) $a \otimes \bar{0} = \bar{0}$ where $\bar{0}$ is the $\oplus$-identity,
- (Distributivity) $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$.

Table II presents a few familiar examples of path algebras (these and many others can be found in the literature cited above). Here $\mathbb{Z}$ denotes the set of all integers, $\mathbb{Z}^+$ denotes

$\mathfrak{B}$	$\mathcal{X}$	$\oplus$,	$\otimes$	$\bar{0}$	$\bar{1}$	description
$\mathfrak{E}$	$\{0,\ 1\}$	max	min	0	1	usable-path routing (the Boolean semi-ring)
$\mathfrak{M}$	$\mathbb{Z} \cup \{\infty\}$	min	$+$	∞	0	minimum-weight routing
$\mathfrak{M}^+$	$\mathbb{Z}^+ \cup \{\infty\}$	min	$+$	∞	0	minimum-weight routing, non-negative weights
$\mathfrak{R}$	$[0,\ 1]$	max	$\times$	0	1	most-reliable routing
$\mathfrak{C}$	$\{0,\ 1,\ 2, \ldots,\ k\} \cup \{\infty\}$	max	min	0	∞	greatest-capacity routing

TABLE II

PATH ALGEBRA EXAMPLES.

the set of non-negative integers, and $\mathbb{R}^+$ is the set of non-negative real numbers.

For any path algebra $\mathfrak{B}$, the *natural partial order*, $\preccurlyeq_{\mathfrak{B}}$, is defined as

$$b \preccurlyeq_{\mathfrak{B}} a \quad \text{iff} \quad b \oplus a = b.$$

The idempotency, commutivity, and associativity of $\oplus$ ensure that $\preccurlyeq_{\mathfrak{B}}$ is a partial order [9], [10]. Note that $a \preccurlyeq_{\mathfrak{B}} \bar{0}$, for $a \in \mathfrak{X}$. A path algebra $\mathfrak{B}$ is said to be *super-unitary*, if for every $a \in \mathcal{X}$,

$$\bar{1} \preccurlyeq_{\mathfrak{B}} a.$$

Note that for a super-unitary path algebra we have

$$\bar{1} \preccurlyeq_{\mathfrak{B}} a \preccurlyeq_{\mathfrak{B}} \bar{0},$$

and some authors (for example [8]) use the term *bounded* instead of super-unitary. We note one source of potential confusion in that some authors (for example [10]) define the natural partial order in the other direction, saying that $a \preccurlyeq_{\mathfrak{B}} b$ holds if $b \oplus a = b$.

In the path algebras $\mathfrak{M}$ and $\mathfrak{M}^+$ (Table II), we see that $a \preccurlyeq_{\mathfrak{B}} \bar{0}$ corresponds to $\min(a,\ \infty) = a$ and that $\bar{1} \preccurlyeq_{\mathfrak{B}} a$ corresponds to $\min(a,\ 0) = 0$, which holds only in $\mathfrak{M}^+$.

Denote $a^2 = a \otimes a$. $\mathfrak{B}$ is called a *nilpotent* path algebra if there exists a finite positive integer q such that

$$a^q = \bar{0}, \text{ for all } a \in \mathfrak{X} \backslash \{\bar{1}\}.$$

The conditions nilpotent and super-unitary are both universal conditions for path algebras.

For any path algebra $\mathfrak{B}$, given $\mathcal{G}$, let $\mathcal{L}_{\mathfrak{B}}$ be a labelling function mapping the edges of $\mathcal{E}$ into $\mathcal{X}$. For each path $P = e_k e_{k-1} \cdots e_1 \in \mathcal{P}(v_0)$ of $\mathcal{G}$, we define the *weight* of the P as

$$\mathcal{L}_{\mathfrak{B}}(e_k) \otimes \cdots \otimes \mathcal{L}_{\mathfrak{B}}(e_1),$$

which we denote as $\mathcal{L}_{\mathfrak{B}}(P)$, with a slight abuse of notation. For the trivial path $P = v_0$, we define its weight , $\mathcal{L}_{\mathfrak{B}}(P)$, to be $\bar{1}$. We also have a special set $B \subseteq \mathcal{X}$ of *bad*, or *banned*, values. The tuple $I = \langle \mathcal{G},\ \mathcal{L}_{\mathfrak{B}},\ B \rangle$ is called *a* $\mathfrak{B}$*-instance*. At times we write $I = \langle \mathcal{G},\ \mathcal{L}_{\mathfrak{B}} \rangle$ in contexts where B is not important. The intuition is that routing solutions cannot be associated with paths such that $\mathcal{L}_{\mathfrak{B}}(P) \in B$. Most often we will have $B = \{\bar{0}\}$, but not always — the choice of B may depend on the interpretation of a routed graph. For example, with the Boolean path algebra $\mathfrak{E}$, we could define $B = \{0\}$ if we want no route to have value 0 (false). However, we could chose $B = \{\}$ if we want routes of value 0 only when there are no routes of value 1.

A *solution* to a $\mathfrak{B}$-instance I is a function δ mapping vertices of $\mathcal{V}$ to elements of $\mathcal{X}$ such that $\delta(v_0,\ v_0) = \bar{1}$ and for all $v \neq v_0$,

$$\delta(v,\ v_0) = \bigoplus_{P \in \mathcal{P}(v,\ v_0),\ \mathcal{L}_{\mathfrak{B}}(P) \in B} \mathcal{L}_{\mathfrak{B}}(P).$$

Note that in the case of minimum-weight path algebras, (where $\oplus = \min$, $\bar{1} = 0$, and $\otimes = +$) this corresponds to the familiar equations $\delta(v_0,\ v_0) = 0$ and

$$\delta(v,\ v_0) = \min_{P = e_k e_{k-1} \cdots e_1 \in \mathcal{P}(v,\ v_0)} \left(\sum_i \mathcal{L}_{\mathfrak{B}}(e_i) \right).$$

A $\mathfrak{B}$-instance I is said to be *absorptive* if for every simple directed cycle $v_1 v_2 \cdots v_n v_1$ in $\mathcal{G}$, we have

$$\bar{1} \preccurlyeq_{\mathfrak{B}} \mathcal{L}_{\mathfrak{B}}(v_1 v_2 v_3 \cdots v_n v_1).$$

Absorptive graph is a generalisation of the absence of a negative cycle. Proofs of the following fundamental theorems can be found in [8]–[10]. We also prove these theorems within our unified framework, as indicated here. As explained in Section V, the proofs require that we restrict path algebras to ones that correspond to *distributive lattices*.

Theorem 1: If path algebra $\mathfrak{B}$ is super-unitary, then every $\mathfrak{B}$-instance I has a solution.

Proof: See Theorem 15. ■

Theorem 2: If a $\mathfrak{B}$-instance I is absorptive, then I has a solution.

Proof: See Theorem 16. ■

Note that a $\mathfrak{B}$-instance I may be absorptive even though the path algebra $\mathfrak{B}$ is not super-unitary. For example, $\mathfrak{M}$ is not super-unitary, but all graphs labelled with only non-negative integers are absorptive.

C. Sobrinho's Routing Algebra

Sobrinho's routing algebra is given in [3]–[5] as

$$\mathcal{A} = \langle \Sigma,\ L,\ \lesssim,\ \otimes \rangle$$

comprised of

- a set Σ of signatures,
- a set L of labels,
- a preference order $\lesssim$ over Σ,
- an extension operator $\otimes$, mapping $L \times \Sigma$ to Σ.

$$
\begin{array}{rcl}
(v,\ ()) \otimes p & = & \infty \\
(v,\ l) \otimes \infty & = & \infty \\
(v,\ (r_1, r_2, \ldots, r_k)) \otimes (x, s) & = & \begin{cases} \infty & \text{if } v \in s, \\ (n,\ \text{append}(s,\ v)) & \text{if } r_1 = s \rightarrow n, \\ (r_2, \ldots, r_k) \otimes (x, s) & \text{otherwise} \end{cases}
\end{array}
$$

Fig. 1. The $\oplus_u$ operator for the universal routing algebra A_u.

In this paper we will restrict our attention to routing algebras in which the preference order is a total order.

The algebra is motivated by the common practice of implementing policy-based routing by attaching signatures to each connection for neighbouring routers. Routers broadcast messages via a specified connection, carrying the signatures. Receiving routers, then, process and transform incoming signatures to out-going signatures, prescripted by certain policy. This process is captured by the extension operator as

$$
\underbrace{l}_{transforming\ label} \quad \otimes \quad \underbrace{\sigma_1}_{incoming\ signature} \quad \rightarrow \quad \underbrace{\sigma_2}_{outgoing\ signature}
$$

In this manner, the interactions of routers can manifest a variety of policy-driven behaviours. Such policy-based interaction can be well-captured by the above algebra. Since operator $\otimes$ has the same function as the $\otimes$ in path algebra, we use the same symbol and its meaning can be interpreted in the context.

We employ the preference order $\lesssim$ to specify the selection priority of signatures. For a pair of signatures σ_1, σ_2 we write $\sigma_1 < \sigma_2$ as a strict preference, and $\sigma_1 \approx \sigma_2$ as an indifferent preference. With multiple alternative routes available, routers will select the route (or a set of routes) with a minimum preference order (that is, preference corresponds to cost).

The set Σ may contain a special element $\infty \in \Sigma$ such that: $\sigma < \infty$, for all $\sigma \in \Sigma \backslash \{\infty\}$ and $l \otimes \infty = \infty$, for all $l \in L$.

Each of the path algebras in Table II can be easily translated to a routing algebra by letting $L = \Sigma = \mathcal{X}$, $\otimes$ be as in the table, and taking $\lesssim$ to be the natural partial order, which in each case is actually a total order. Section III-D below gives an example of a path algebra that cannot be encoded as a routing algebra. Here we present a routing algebra that cannot be represented as a path algebra.

We define the *universal routing algebra* A_u to be

$$
\mathcal{A} = \langle \Sigma_u,\ L_u,\ \lesssim_u,\ \otimes_u \rangle.
$$

We let $\Sigma_u = (\mathbb{Z}^+ \times \mathbb{Z}^*) \cup \{\infty\}$. If $(x,\ s) \in \Sigma_u$, then x is a non-negative integer and s is a finite sequence of integers. We define $(x_1, s_1) \lesssim_u (x_2, s_2)$ to mean that $x_1 \leq x_2$, and $p \lesssim_u \infty$ for all $p \in \Sigma_u$. Labels L_u are pairs (v, l), where v is an integer and l is an ordered sequences of *rules*, $l = (r_1, r_2, \ldots, r_k)$, where $0 \leq k$ and each rule is of the form $s \rightarrow n$. Informally, this rules state that if the sequence component of a route is equal to s, then set the weight component to n. Rules are evaluated from first to last, and the $\oplus_u$ operator is defined in Figure 1. The idea is that in any network graph, all arcs into a node v have labels of the form $(v,\ l)$, so that the sequences s record the path traversed by a route. The rules simply provide a way for the policies to explicitly rank every path. With this universal routing algebra it is easy to encode an arbitrary Stable Paths Problem.

A routing algebra $\mathcal{A}$ is said to be *monotone* if

$$
\sigma \lesssim l \otimes \sigma \text{ for each } l \in L \text{ and for each } \sigma \in \Sigma.
$$

Given a $\mathcal{G}$, an *initial signature* σ_0, and $\mathcal{L}_\mathcal{A}$ is a labelling function mapping the edges of $\mathcal{E}$ into L. The triple $I = \langle \mathcal{G},\ \mathcal{L}_\mathcal{A},\ \sigma_0 \rangle$ is called *a $\mathcal{A}$-instance*.

For each path $P = e_k e_{k-1} \cdots e_1 \in \mathcal{P}(v_0)$ of $\mathcal{G}$, we define the *weight* of the P as

$$
\mathcal{L}_\mathcal{A}(P) \triangleq (\mathcal{L}_\mathcal{A}(e_k), ..., \mathcal{L}_\mathcal{A}(e_1)) \otimes \sigma_0.
$$

A *solution* to an $\mathcal{A}$-instance I is a function δ such that $\delta(v_0,\ v_0) = \{\sigma_0\}$ and for all $v \neq v_0$,

$$
\delta(v, v_0) = \min_{\lesssim} \{\mathcal{L}_\mathcal{A}(e) \otimes \sigma \mid e \in (v, w), \sigma \in \delta(w, v_0)\}.
$$

An $\mathcal{A}$-instance I is said to be *free* if every set of n signatures $\{\sigma_1,\ \sigma_2,\ \cdots,\ \sigma_n\}$ where no σ_i is ∞, and every directed cycle $W_1 W_2 \cdots W_n$, in $\mathcal{G}$, where the W_1 are paths such that the last node in W_i is the first node in W_{i+1}, for $1 \leq i \leq n-1$, and the last node in W_n is the first node in W_1, we have that there exists $i \in \{1, ..., n\}$ such that $\sigma_j < \mathcal{L}_\mathcal{A}(v_i, v_j) \otimes \sigma_i$, where $j = i{+}1$ if $1 \leq i < n$, otherwise 1. Such a cycle $v_1 v_2 \cdots v_n v_1$ is called a *free cycle*. A labelled graph is free if all such cycles are free. Note that this definition is slightly more general that that found in [4] in that we define the cycle in terms of paths W_i rather than edges.

Section VI applies our general framework to prove the following theorems.

Theorem 3: If routing algebra $\mathcal{A}$ is monotone, then every $\mathcal{A}$-instance I is free.

Proof: See Theorem 21. ■

Theorem 4: If an $\mathcal{A}$-instance I is free, then it has a solution.

Proof: See Theorem 22. ■

D. Path Algebras vs. Routing Algebras

The two policy-based systems give rise to very different ways of thinking about routes — Sobrinho's routing algebras seem closely tied to destination based forwarding, while path algebras are more general. We illustrate this with an example.

Let path descriptors (and labels) be of the form $\langle d,\ b \rangle$, where d is some measure of delay and b is some measure of bandwidth. We use these path descriptors as both labels and signatures in a routing algebra, and as the carrier set in the path algebra.

In both algebras we define

$$\langle d_1,\ b_1\rangle \otimes \langle d_2,\ b_2\rangle = \langle d_1 + d_2,\ \min(b_1,\ b_2)\rangle.$$

We can think of $\langle d_1,\ b_1\rangle$ as associated with an incoming link, while $\langle d_2,\ b_2\rangle$ is associated with a neighbor's route.

In the routing algebra, we define a total order on signatures using a lexicographic order:

$$\langle d_1,\ b_1\rangle \leq \langle d_2,\ b_2\rangle \Leftrightarrow d_1 < d_2 \text{ or } (d_1 = d_2 \text{ and } b_2 \leq b_1),$$

where shorter distances are preferred, with higher bandwidths breaking ties. For the path algebra we define

$$\langle d_1,\ b_1\rangle \oplus \langle d_2,\ b_2\rangle = \langle \min(d_1,\ d_2),\ \max(b1,\ b_2)\rangle.$$

Note that this operator gives us the natural partial order

$$\langle d_1,\ b_1\rangle \preceq \langle d_2,\ b_2\rangle \Leftrightarrow d_1 \leq d_2 \text{ and } b_2 \leq b_1.$$

That is, we obtain a "parallel" product.

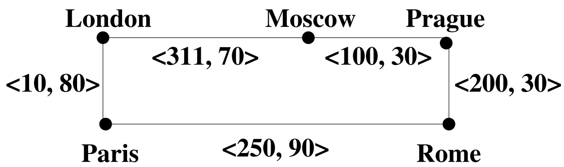

Fig. 2. A labelled graph.

Figure 2 illustrates a simple labelled graph for these algebras. Moscow is taken as the origin node, with $\sigma_0 = \langle 0,\ \infty\rangle$ for the routing algebra, and $\overline{1} = \langle 0,\ \infty\rangle$ for the path algebra. Let us look for the best path from Rome to Moscow. Note that path (Moscow, Prague, Rome) has value $\langle 300,\ 30\rangle$, while path (Moscow, London, Paris, Rome) has value $\langle 571,\ 70\rangle$.

In the routing algebra we find the solution to be

$$\min_{\leq}\{\langle 300,\ 30\rangle,\ \langle 571,\ 70\rangle\} = \langle 300,\ 30\rangle,$$

which is associated with the path (Moscow, Prague, Rome).

However, in the path algebra, the solution is

$$\langle 300,\ 30\rangle \oplus \langle 571,\ 70\rangle = \langle 300,\ 70\rangle,$$

which *is not associated with any single path* — In this case we have two paths associated with the best value. Various forwarding paradigms may be able to take advantage of this type of routing. For example, we could imagine a QoS forwarding paradigm that uses the path (Moscow, London, Paris, Rome) for delay-insensitive but bandwidth intensive flows, while using the (Moscow, Prague, Rome) path for delay-sensitive flows of small bandwidth.

In Section IV we define another notion of solution for the path algebra called the multi-value solution. In the example this will give the set of values

$$\min_{\preceq}\{\langle 300,\ 30\rangle,\ \langle 571,\ 70\rangle\} = \{\langle 300,\ 30\rangle,\ \langle 571,\ 70\rangle\}.$$

We then need a result (Theorem 14) which relates multi-valued solutions to the standard notion of a solution for a path algebra instance. In terms of this example, that relationship is expressed as

$$\langle 300,\ 70\rangle = \bigoplus\{\langle 300,\ 30\rangle,\ \langle 571,\ 70\rangle\}.$$

That is, a standard path algebra solution can be obtained from a multi-value solution by application of the $\oplus$ operator.

IV. Policy Structure and Routing Structure

A *policy structure* S is defined as:

$$S = \langle \mathfrak{X},\ \preccurlyeq,\ \sqsubseteq\rangle,$$

where $\preccurlyeq$ is a partial order over $\mathfrak{X}$ and $\sqsubseteq$ is a preorder over $\mathfrak{X}$. We write $\prec$ and $\sqsubset$ as the respective strict relations.

Informally, the interpretation of a policy structure is that the elements of $\mathfrak{X}$ represent *values* that will be associated with routes. The relation $x \preccurlyeq y$ will tell us that value x is at least as well-preferred as value y, while the relation $x \sqsubseteq y$ tells us that value y can be constructed from value x.

A. Generalised Routing Problem

Given a graph $\mathcal{G}$, an *S-instance* is a triple, $I = \langle \mathcal{G},\ \psi,\ B\rangle$, where ψ maps paths $P \in \mathcal{P}(v_0)$ to elements of $\mathfrak{X}$ such that for all $P \in \mathcal{P}(v,\ v_0)$ and all $Q \in \mathcal{P}(w,\ v)$ we have $\psi(P) \sqsubseteq \psi(QP)$. The set $B \subseteq \mathfrak{X}$ represent banned values that should not be used by any routing solution.

We can think of each S-instance $I = \langle \mathcal{G},\ \psi,\ B\rangle$ as a *generalised routing problem* that we wish to solve. We assume that $\psi(v_0)\} \notin B$. A *multi-value solution* to an S-instance I is a function δ such that $\delta(v_0,\ v_0) = \{\psi(v_0)\}$ and for all $v \neq v_0$,

$$\begin{aligned}\delta(v,\ v_0) = \min\nolimits_{\preccurlyeq}\{\psi(eP) | \psi(eP) &\notin B,\\ e &\in (v,\ w),\\ P &\in \mathcal{P}(w,\ v_0),\\ \psi(P) &\in \delta(w,\ v_0)\}.\end{aligned}$$

Note that a multi-value solution can be thought of as a *fixed-point* of a certain functional, $\mathscr{F}^{\text{multi}}$, where $\mathscr{F}^{\text{multi}}(\delta) = \delta$ iff δ is a multi-value solution.

B. Routing Structure of an S-Instance

Given an S-instance $I = \langle \mathcal{G},\ \psi,\ B\rangle$, we define a new policy structure, called *the routing structure for instance* I, as follows.

$$S_I = \langle \mathcal{P}^B_\psi,\ \preccurlyeq_I,\ \sqsubseteq_I\rangle,$$

where

$$\begin{array}{rcl}\mathcal{P}^B_\psi &=& \{P \in \mathcal{P}(v_0) \mid \psi(P) \notin B\},\\ P \preccurlyeq_I Q &\Leftrightarrow& \text{there is a path } W \text{ such that } Q = WP,\\ P \equiv_I Q &\Leftrightarrow& \text{head}(P) = \text{head}(Q) \text{ and } \psi(P) = \psi(Q),\\ P \sqsubset_I Q &\Leftrightarrow& \text{head}(P) = \text{head}(Q) \text{ and } \psi(P) \prec \psi(Q),\\ P \sqsubseteq_I Q &\Leftrightarrow& P \equiv_I Q \text{ or } P \sqsubset_I Q.\end{array}$$

It is easy to see that $\preccurlyeq_I$ is a partial order and that $\sqsubseteq_I$ is a preorder. Note that with respect to the policy structure S there is a type of reversal here — the relation $\preccurlyeq_I$ of S_I is related to relation $\sqsubseteq$ of S, while the relation $\sqsubseteq_I$ of S_I is related to relation $\preccurlyeq$ of S. In particular, we have

$$P \preccurlyeq_I Q \quad \Rightarrow \quad \psi(P) \sqsubseteq \psi(Q),$$
$$P \sqsubseteq_I Q \quad \Rightarrow \quad \psi(P) \preccurlyeq \psi(Q),$$
$$P \sqsubset_I Q \quad \Rightarrow \quad \psi(P) \prec \psi(Q).$$

C. Policy Relations and Routing Relations

First we define an instance condition for routing structures. The definition is motivated by dispute wheels in the Stable Paths Problem [2].

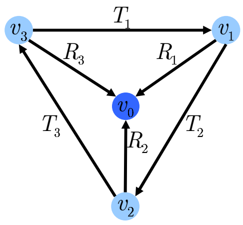

Fig. 3. A bad triangle.

Figure 3 shows a routing structure where

$$R_1 \preccurlyeq_I T_1R_1 \sqsubset_I R_3 \preccurlyeq_I T_3R_3 \sqsubset_I R_2 \preccurlyeq_I T_2R_2 \sqsubset_I R_1,$$

and it is easy to check that no solution can simultaneously satisfy all three vertices. Our instance condition must rule out such cases. We do this by defining *the routing relation of I* to be

$$\mathscr{R}_I \triangleq (\preccurlyeq_I \bowtie \sqsubset_I)^{tc},$$

where $(\cdot)^{tc}$ is the transitive closure of a relation. Our desired instance condition is that $\mathscr{R}_I$ be *anti-reflexive*.

For a policy structure $S = \langle \mathfrak{X}, \ \preccurlyeq, \ \sqsubseteq \rangle$, we define the *policy relation* in the "reverse" manner as

$$\mathscr{R}_S \triangleq (\sqsubseteq \bowtie \prec)^{tc}.$$

We now show that the anti-reflexivity of the policy relation (our universal condition) implies that all associated routing relations are also anti-reflexive (the instance condition always holds).

Theorem 5: Suppose that $S = \langle \mathfrak{X}, \ \preccurlyeq, \ \sqsubseteq \rangle$ is a policy structure and that $I = \langle \mathcal{G}, \ \psi, \ B \rangle$ is an S-instance. If $\mathscr{R}_S$ is anti-reflexive, then $\mathscr{R}_I$ is anti-reflexive.

Proof: Assume $\mathscr{R}_S$ is anti-reflexive. Suppose that $\mathscr{R}_I$ is not anti-reflexive. Then there must exist distinct $P_1, ..., P_m \in \mathcal{P}(v_0)$ where m is even such that

$$P_1 \preccurlyeq_I P_2 \sqsubset_I P_3 \cdots \preccurlyeq_I P_m \sqsubset_I P_1.$$

Therefore

$$\psi(P_1) \sqsubseteq \psi(P_2) \prec \psi(P_3) \cdots \sqsubseteq \psi(P_m) \prec \psi(P_1).$$

But this is telling us that that $\mathscr{R}_S$ is not anti-reflexive, which is a contradiction. ■

On the other hand, a routing relation may still be anti-reflexive even though the associated policy relation is not. In the next section we show that the anti-reflexivity of the routing relation is enough to imply that there exists a solution for the associated S-instance.

D. Relationship to the Stable Paths Problem (SPP)

We begin with a brief review the definitions of the Stable Paths Problem (SPP) taken directly from [2].

E. SPP Definitions

Let $G = (V, \ E, \ v_0)$ be a graph with origin 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 (node 0). 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 lowest ranked)** $\lambda^v(\epsilon) = 0$, $\lambda^v(P) > 0$ 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),

Let $S_{\text{spp}} = (G, \mathcal{P}, \Lambda)$ be an instance of the Stable Paths Problem. 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 set of paths choices(π, u) is defined to be

$$\text{choices}(\pi, u) = \begin{cases} \{(u \ v)\pi(v) \mid \{u, \ v\} \in E\} \cap \mathcal{P}^u & (u \neq 0) \\ \{(0)\} & \text{o.w.} \end{cases}$$

This set represents all possible permitted paths at u that can be formed by extending the paths assigned to the peers of u. Given a node u, suppose that W is a subset of the permitted paths $\mathcal{P}^u$ such that each path in W has a distinct next hop. Then the *best path in W* is defined to be

$$\text{best}(W, u) = \begin{cases} P \in W \text{ with maximal } \lambda^u(P) & (W \neq \emptyset) \\ \epsilon & \text{o.w.} \end{cases}$$

The path assignment π is *stable at node u* if

$$\pi(u) = \text{best}(\text{choices}(\pi, u), u).$$

Note that if π is stable at node u and $\pi(u) = \epsilon$, then the set of choices at u must be empty. The path assignment π is *a solution* if it is stable at each node u. We often write a path assignment as a vector, $(P_1, \ P_2, \ \cdots, \ P_n)$, where $\pi(u) = P_u$. Any stable path assignment implicitly defines a tree rooted at the origin. Note, however, that this is not always a spanning tree.

A *dispute wheel,* $\Pi = (\vec{U}, \vec{\mathcal{Q}}, \vec{\mathcal{R}})$*, of size* k, is a sequence of nodes $\vec{U} = u_0, u_1, \cdots u_{k-1}$, and sequences of non-empty paths $\vec{\mathcal{Q}} = Q_0, Q_1, \cdots Q_{k-1}$ and $\vec{\mathcal{R}} = R_0, R_1, \cdots R_{k-1}$, such that for each $0 \leq i \leq k-1$ we have (1) R_i is a path from u_i to u_{i+1}, (2) $Q_i \in \mathcal{P}^{u_i}$, (3) $R_i Q_{i+1} \in \mathcal{P}^{u_i}$, and (4) $\lambda^{u_i}(Q_i) \leq \lambda^{u_i}(R_i Q_{i+1})$. (All subscripts are to be interpreted modulo k.) See Figure 4 for an illustration of a dispute wheel. Since permitted paths are simple, it follows that the size of any dispute wheel is at least 2.

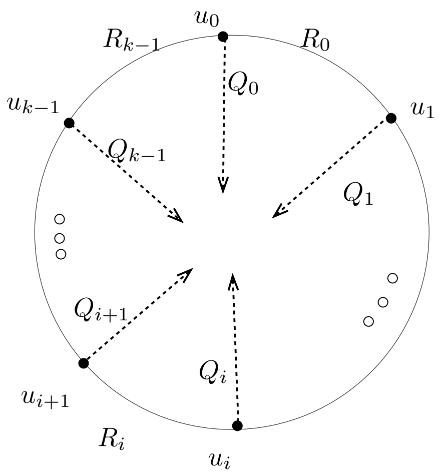

Fig. 4. A dispute wheel of size k.

F. *S-instances as Generalized SPPs*

Every S-instance $I = (\mathcal{G},\ \psi,\ Y)$, the routing structure

$$S_I = \langle \mathcal{P}_\psi,\ \ \preccurlyeq_I,\ \ \sqsubseteq_I \rangle,$$

can be viewed as a generalisation of the Stable Paths Problem (SPP).

In S_I, the relation $\sqsubseteq_I$, when restricted to the set of paths $\mathcal{P}(v,\ v_0)$, is a preorder. In an SPP, the ranking function for any node induces a special kind of preorder — one that is nearly a total order except for equally-ranked paths coming from the same neighbor (called the strictness condition).

The multi-value solution of an S-instance is defined in terms of the a mapping δ from nodes in $\mathcal{G}$ to values in $\mathfrak{X}$, while a solution for an SPP is a mapping from nodes to sets of paths. To model this we define a *multi-path solution* to an S-instance I is a function Δ such that $\Delta(v_0,\ v_0) = \{\psi(v_0)\}$ and for all $v \neq v_0$,

$$\begin{aligned}\Delta(v,\ v_0) = \min\nolimits_{\sqsubseteq_I}\{eP | e \in (v,\ w),\\ P \in \mathcal{P}(w,\ v_0),\ P \in \Delta(w,\ v_0)\}.\end{aligned}$$

Note that this solution is defined in terms of min with respect to $\sqsubseteq_I$ rather than $\preccurlyeq$.

As indicated in Section IV-C, the notion that $\mathscr{R}_I$ is anti-reflexive is a generalisation of lack of dispute wheels for SPPs, and so we take this as our instance condition on S-instances.

Theorem 6: If an $S[\mathcal{A}]$-instance is anti-reflexive, then it has a multi-path solution Δ.

Proof: (Proof Sketch.) Let I be an anti-reflexive $S[\mathcal{A}]$-instance. When I is viewed as an SPP we know that it can have no dispute wheel. The proof then proceeds very much like the proof of [2] showing that if there is no dispute wheel, there must be a solution. ■

Lemma 1: If $X \subseteq \mathcal{P}(v,\ v_0)$, then

$$\min\nolimits_{\preccurlyeq}\psi(X) = \psi(\min\nolimits_{\sqsubseteq_I} X),$$

where $\psi(W) = \{\psi(w) \mid w \in W\}$.

Proof:

$$\begin{aligned} & \psi(\min\nolimits_{\sqsubseteq_I} X) \\ = \ & \psi(\{x \in A \mid \text{there exists no } y \in X \text{ such that } y \sqsubset_I x\}) \\ = \ & \psi(\{x \in A \mid \text{there exists no } y \in X \text{ such that } y \prec x\}) \\ = \ & \{\psi(x) \in A \mid \text{there exists no } y \in X \text{ such that } y \prec x\} \\ = \ & \min\nolimits_{\preccurlyeq}\psi(X). \end{aligned}$$

■

Theorem 7: If Δ is a multi-path solution for S-instance $I = (G,\ \psi)$, then $\delta(v,\ v_0) = \psi(\Delta(v,\ v_0))$ is a multi-value solution for I.

Proof: Apply Lemma 1. ■

Theorem 8: If an $S[\mathcal{A}]$-instance is anti-reflexive, then it has a multi-value solution.

Proof: Let I be an anti-reflexive $S[\mathcal{A}]$-instance. By Theorem 6, I must have a multi-path solution Δ. But by Theorem 7, we have a multi-value solution with $\delta(v,\ v_0) = \psi(\Delta(v,\ v_0))$. ■

V. Path Algebras, Revisited

Given a path algebra $\mathfrak{B} = \langle \mathcal{X},\ \oplus,\ \otimes,\ \bar{0},\ \bar{1} \rangle$ we construct an associated policy structure

$$S[\mathfrak{B}] = \langle \mathcal{X},\ \ \preccurlyeq_{\mathfrak{B}},\ \ \sqsubseteq_{\mathfrak{B}} \rangle,$$

where, as in Section III, we define

$$b \preccurlyeq_{\mathfrak{B}} a \quad \text{iff} \quad b \oplus a = b,$$

and we define the new relation

$$b \sqsubseteq_{\mathfrak{B}} a \quad \text{iff} \quad \text{there exists } c \in \mathcal{X} \text{ such that } a = c \otimes b.$$

That is, $b \sqsubseteq_{\mathfrak{B}} a$ means that a can be generated from b using operator $\otimes$, and it is easy to check that is is a preorder.

The $\otimes$ operator is said to be *isotonic* with respect to $\preccurlyeq_{\mathfrak{B}}$ when for all $a, b, c \in \mathcal{X}$,

$$a \preccurlyeq_{\mathfrak{B}} b \Rightarrow c \otimes a \preccurlyeq_{\mathfrak{B}} c \otimes b.$$

Lemma 2: For any path algebra $\mathfrak{B}$, $\otimes$ is isotonic with respect to $\preccurlyeq_{\mathfrak{B}}$.

Proof: For all $a, b, c \in \mathcal{X}$, one obtains

$$\begin{aligned} & a \preccurlyeq_{\mathfrak{B}} b \Leftrightarrow a \oplus b = a \\ & \Rightarrow c \otimes a = c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b) \text{ [by distributivity]} \\ & \Leftrightarrow c \otimes a \preccurlyeq_{\mathfrak{B}} c \otimes b. \end{aligned}$$

■

Corollary 9: For any path algebra $\mathfrak{B}$, if $a \preccurlyeq_{\mathfrak{B}} b \sqsubseteq_{\mathfrak{B}} c$, then there always exists $d \in \mathcal{X}$ such that $a \sqsubseteq_{\mathfrak{B}} d \preccurlyeq_{\mathfrak{B}} c$.

Proof: If $b \sqsubseteq_{\mathfrak{B}} c$, then there exists $f \in \mathcal{X}$ such that

$$c = f \otimes b.$$

But $a \preccurlyeq_{\mathfrak{B}} b$, by Lemma 2, we have $f \otimes a \preccurlyeq_{\mathfrak{B}} f \otimes b = c$. Let $d = f \otimes a$, hence $a \sqsubseteq_{\mathfrak{B}} d \preccurlyeq_{\mathfrak{B}} c$. ■

Lemma 3: For any path algebra $\mathfrak{B}$, if $\mathscr{R}_{S[\mathfrak{B}]}$ is not anti-reflexive, then there exist some $a, b \in \mathcal{X}$ such that $b \otimes a \prec_{\mathfrak{B}} a$.

Proof: Since $\mathscr{R}_{S[\mathfrak{B}]}$ is not anti-reflexive, then there exist distinct $a_1, a_2, ..., a_{2k} \in \mathcal{X}$ such that

$$a_1 \sqsubset_{\mathfrak{B}} a_2 \prec_{\mathfrak{B}} a_3 \sqsubset_{\mathfrak{B}} \cdots \prec_{\mathfrak{B}} a_{2k-1} \sqsubset_{\mathfrak{B}} a_{2k} \prec_{\mathfrak{B}} a_1. \quad (**)$$

By Corollary 9, if $a_{2k-1} \sqsubset_{\mathfrak{B}} a_{2k} \prec_{\mathfrak{B}} a_1$, then there exists $a' \in \mathcal{X}$ such that $a_{2k-1} \preccurlyeq_{\mathfrak{B}} a' \sqsubseteq_{\mathfrak{B}} a_1$. Hence, by transitivity

$$a_{2k-2} \prec_{\mathfrak{B}} a_{2k-1} \preccurlyeq_{\mathfrak{B}} a' \sqsubseteq_{\mathfrak{B}} a_1 \sqsubset_{\mathfrak{B}} a_2$$

becomes

$$a_{2k-2} \prec_{\mathfrak{B}} a_{2k-1} \sqsubset_{\mathfrak{B}} a_2.$$

Inductively by Corollary 9 and transitivity, $(**)$ reduces to

$$a_2 \prec_{\mathfrak{B}} a_3 \sqsubset_{\mathfrak{B}} a_2.$$

Hence, for some $b \in \mathcal{X}$, we have $a_2 = b \otimes a_3 \prec_{\mathfrak{B}} a_3$. ■

Theorem 10: For any path algebra $\mathfrak{B}$, if $\mathfrak{B}$ is super-unitary, then $\mathscr{R}_{S[\mathfrak{B}]}$ is anti-reflexive.

Proof: Assume that $\mathfrak{B}$ is super-unitary but that $\mathscr{R}_{S[\mathfrak{B}]}$ is not anti-reflexive. By Lemma 3, there exist some $a, b \in \mathcal{X}$ such that $b \otimes a \prec_{\mathfrak{B}} a$. It contradicts to $\mathfrak{B}$ being super-unitary which implies

$$b \succcurlyeq_{\mathfrak{B}} \bar{1} \Rightarrow b \otimes a \succcurlyeq_{\mathfrak{B}} a.$$

Hence, $\mathfrak{B}$ is not super-unitary, which is a contradiction. ■

Note that the implication does not go in the other direction. That is, $\mathscr{R}_{S[\mathfrak{B}]}$ is may be anti-reflexive while $\mathfrak{B}$ is not super-unitary. This can happen when there exists $a \in \mathcal{X}$ such that $\bar{1} \not\succcurlyeq_{\mathfrak{B}} a$ and $\bar{1} \not\prec_{\mathfrak{B}} a$.

Theorem 11: For any path algebra $\mathfrak{B}$, if $\mathfrak{B}$ is nilpotent, then $\mathscr{R}_{S[\mathfrak{B}]}$ is anti-reflexive.

Proof: Assume that $\mathfrak{B}$ is nilpotent but that $\mathscr{R}_{S[\mathfrak{B}]}$ is not anti-reflexive. By Lemma 3, there exist some $a, b \in \mathcal{X}$such that $b \otimes a \prec_{\mathfrak{B}} a$. By Lemma 2 and nilpotency ($b^q = \bar{0}$ for some fixed q),

$$a \succ_{\mathfrak{B}} b \otimes a \succcurlyeq_{\mathfrak{B}} b^2 \otimes a \succcurlyeq_{\mathfrak{B}} \cdots \succcurlyeq_{\mathfrak{B}} b^q \otimes a = \bar{0}.$$

It is a contradiction, since $c \preccurlyeq_{\mathfrak{B}} \bar{0}$, for all $c \in \mathcal{X}$. Hence, $\mathfrak{B}$ is not nilpotent. ■

A. Instances of Path Algebra

Recall that given $\mathfrak{B}$, a $\mathfrak{B}$-instance is a tuple $I = \langle \mathcal{G},\ \mathcal{L}_{\mathfrak{B}}, B \rangle$, where $\mathcal{L}_{\mathfrak{B}}$ is a labelling function, mapping every $e \in \mathcal{E}$ to an element in $\mathcal{X}$. We define an induced $\mathfrak{B}$-instance in generalised routing problem as follows. For each $P \in \mathcal{P}(v_0)$, let $\psi(P) = \mathcal{L}_{\mathfrak{B}}(P)$. By abuse of notation, let the induced $\mathfrak{B}$-instance $I = \langle \mathcal{G},\ \psi,\ B \rangle$ and let $\mathscr{R}_I$ be the routing relation over the routing structure S_I.

Theorem 12: Given a path algebra $\mathfrak{B}$, if $\mathfrak{B}$-instance I is absorptive, then $\mathscr{R}_I$ is anti-reflexive.

Proof: Suppose that $\mathscr{R}_I$ as not anti-reflexive. Then, there exist distinct $R_1, ..., R_k, P_1, ..., P_k \in \mathcal{P}(v_0)$ such that

$$R_1 \preccurlyeq_I P_k \sqsubset_I R_{k-1} \preccurlyeq_I P_{k-1} \sqsubset_I \cdots \preccurlyeq_I P_1 \sqsubset_I R_1.$$

Note that $P_i \sqsubset_I R_i$ implies that $\mathcal{L}_{\mathfrak{B}}(P_i) \prec_{\mathfrak{B}} \mathcal{L}_{\mathfrak{B}}(R_i)$, and by that the definition of $\preccurlyeq_I$, implies that there exist $T_1, ..., T_k$ such that for each $i \in \{1, ..., k\}$, $P_i = T_i R_j$ and $T_i \in \mathcal{P}(v_i, v_j)$, where $j = i + 1$ if $1 \le i < k$, otherwise 1. Hence, $T_1 T_2 T_3 \cdots T_k$ is a directed cycle in $\mathcal{G}$.

Using isotonicity (Lemma 2) we have

$$\begin{aligned}
\mathcal{L}_{\mathfrak{B}}(R_1) &\succ_{\mathfrak{B}} \mathcal{L}_{\mathfrak{B}}(P_1) = \mathcal{L}_{\mathfrak{B}}(T_2) \otimes \mathcal{L}_{\mathfrak{B}}(R_2) \\
&\succcurlyeq_{\mathfrak{B}} \mathcal{L}_{\mathfrak{B}}(T_2) \otimes \mathcal{L}_{\mathfrak{B}}(P_2) = \mathcal{L}_{\mathfrak{B}}(T_2) \otimes \mathcal{L}_{\mathfrak{B}}(T_3 R_3) \\
&= \mathcal{L}_{\mathfrak{B}}(T_2 T_3) \otimes \mathcal{L}_{\mathfrak{B}}(R_3) \\
&\ \ \vdots \\
&\succcurlyeq_{\mathfrak{B}} \mathcal{L}_{\mathfrak{B}}(T_2 T_3 \cdots T_k) \otimes \mathcal{L}_{\mathfrak{B}}(P_k) \\
&= \mathcal{L}_{\mathfrak{B}}(T_2 T_3 \cdots T_k) \otimes \mathcal{L}_{\mathfrak{B}}(T_1 R_1) \\
&= \mathcal{L}_{\mathfrak{B}}(T_2 T_3 \cdots T_k T_1) \otimes \mathcal{L}_{\mathfrak{B}}(R_1).
\end{aligned}$$

But isotonicity and the fact that I is absortive imply that

$$\begin{aligned}
&\bar{1} \preccurlyeq_{\mathfrak{B}} \mathcal{L}_{\mathfrak{B}}(T_2 T_3 \cdots T_k T_1) \\
\Rightarrow\ &\mathcal{L}_{\mathfrak{B}}(R_1) \preccurlyeq_{\mathfrak{B}} \mathcal{L}_{\mathfrak{B}}(T_2 T_3 \cdots T_k T_1) \otimes \mathcal{L}_{\mathfrak{B}}(R_1).
\end{aligned}$$

So we arrive at a contradiction. ■

Note that $\mathcal{G}$ being not absorptive does not mean $\mathscr{R}_I$ being not anti-reflexive, because $\mathcal{G}$ may have a cycle such that $\mathcal{L}_{\mathfrak{B}}(v_1 v_2 \cdots v_k v_1) \not\prec_{\mathfrak{B}} \bar{1}$ and $\mathcal{L}_{\mathfrak{B}}(v_1 v_2 \cdots v_k v_1) \not\succcurlyeq_{\mathfrak{B}} \bar{1}$.

B. Solutions of Path Algebra

Recall that a solution to a $\mathfrak{B}$-instance I is a function δ mapping vertices of $\mathcal{V}$ to elements of $\mathcal{X}$ such that $\delta(v_0,\ v_0) = \bar{1}$ and for all $v \ne v_0$,

$$\delta(v,\ v_0) = \bigoplus_{P \in \mathcal{P}(v,\ v_0)} \mathcal{L}_{\mathfrak{B}}(P).$$

By distributivity between $\oplus$ and $\otimes$, it corresponds to

$$\delta(v,\ v_0) = \bigoplus_{e \in (v,\ w)} \mathcal{L}_{\mathfrak{B}}(e) \otimes \delta(w,\ v_0). \quad (1)$$

Path algebra is an abstraction of many problems [15], and solving routing problems is one of its many applications. In this paper we focus on a suitable class of path algebra whose solutions can be reduced to solutions in our generalised routing problem. We argue that this particular class of path algebra (DL-path algebra) suffices to capture all practically useful routing problems, to our best knowledge, such as minimum-weight routing, most-reliable routing, and greatest-capacity routing.

The $\oplus$ operation yields a partial order $\preccurlyeq_{\mathfrak{B}}$ on $\mathcal{X}$, as defined earlier in section III. Given a partial order on $\mathcal{X}$, we say that $\mathcal{X}$ is a lattice with respect to $\preccurlyeq_{\mathfrak{B}}$ if both $x \vee y = \sup\{x, y\}$ and $x \wedge y = \inf\{x, y\}$ are well-defined. Further, we say that a lattice is a distributive lattice if $\vee$ and $\wedge$ obey the distributive law such that for $x, y, z \in \mathcal{X}$,

$$(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z) \text{ and } (x \wedge y) \vee z = (x \vee z) \wedge (y \vee z)$$

A collection $X \subseteq P(K)$ is a lattice of sets if it is closed under finite unions and intersections. If X is a lattice of sets then $\langle X, \ \subseteq \rangle$ forms a lattice where $A \vee B = A \cup B$ and $A \wedge B = A \cap B$, for $A, B \in X$. A well-known result from lattice theory is that a distributive lattice is isomorphic to a lattice of sets ([16] Chap. 4).

Let *DL-path algebra* be the subclass of path algebras that form distrbutive lattices. This class includes each of the examples in Table II, and appears to enable practical techniques of distributed policy-based routing.

1) Lattices of Sets: Suppose $\mathfrak{L} = \langle X \subseteq \mathscr{P}(K), \ \cup, \ \cap \rangle$ is a lattice of sets for some set K. We call $U \in X$ an *unsplittable* set w.r.t. $\mathfrak{L}$, if

$$\text{there exists no } A, B \in X \text{ such that } A \neq B \neq U, A \cup B = U.$$

Let the family of non-empty unsplittable sets be $\mathscr{U}_{\mathfrak{L}}$:

$$\mathscr{U}_{\mathfrak{L}} \triangleq \{U \in X \mid U \text{ is unsplittable w.r.t. } \mathfrak{L} \text{ and } U \neq \varnothing\}.$$

Since an unsplittable set cannot be represented as a union of other unsplittable sets, they are atomic elements in X (see Fig.5). From the definition of unsplittable sets, every $A \in X$ can be represented as $A = U_1 \cup \cdots \cup U_k$ for some set $\{U_1, ..., U_k\} \in \mathscr{P}(\mathscr{U}_{\mathfrak{L}})$. Note that set inclusion of $\mathscr{U}_{\mathfrak{L}}$ gives a natural partial order, $\langle \mathscr{U}_{\mathfrak{L}}, \ \subseteq \rangle$.

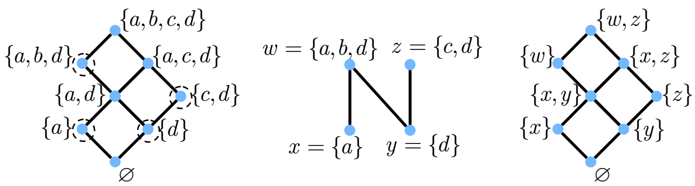

Fig. 5. The left figure is a lattice of sets $\langle X, \ \cup, \ \cap \rangle$, with dotted circles denote the unsplittable sets. The unsplittable sets gives a natural partial order in the middle figure. The right figure shows the isomorphic $\langle \mathscr{X}_{\mathfrak{L}}, \ \vee \rangle$.

Lemma 4: Suppose $A = U_1 \cup \cdots \cup U_k$, where $U_1, ..., U_k \in \mathscr{U}_{\mathfrak{L}}$ such that $U_1 \not\subseteq U_i$ for $1 < i \leq k$. Then there exist no $U'_1, ..., U'_n \in \mathscr{U}_{\mathfrak{L}} \backslash \{U_1\}$ such that $A = U'_1 \cup \cdots \cup U'_n$.

Proof: Suppose $U_1 \cup \cdots \cup U_k = U'_1 \cup \cdots \cup U'_n$. U_1 can be split as $(U'_1 \cap U_1) \cup \cdots \cup (U'_n \cap U_1)$, unless there exists $U'_j \cap U_1 = U_1$. That is, $U'_j \supsetneq U_1$. Then U'_j can be split as $(U'_j \cap U_1) \cup (U'_j \cap (U_2 \cup \cdots \cup U_k))$. Since $U_2 \cup \cdots \cup U_k \neq A$ as $U_1 \not\subseteq U_i$ for $1 < i \leq k$, so $(U'_j \cap (U_2 \cup \cdots \cup U_k)) \neq U'_j$ and $(U'_j \cap U_1) = U_1 \neq U'_j$. Hence, this is a contradiction that $U_1, ..., U_k$ and $U'_1, ..., U'_n$ are unsplittable. ■

For $\mathscr{V} = \{U_1, ..., U_k\}$, we can write $\max_{\subseteq} \mathscr{V}$ for the maximum set in $\mathscr{V}$ under partial order $\subseteq$. Define $\langle \mathscr{X}_{\mathfrak{L}}, \ \vee \rangle$ where

$$\mathscr{X}_{\mathfrak{L}} \triangleq \{\mathscr{V} \in \mathscr{P}(\mathscr{U}_{\mathfrak{L}}) \mid \mathscr{V} = \max_{\subseteq} \mathscr{V}\},$$
$$\mathscr{V} \vee \mathscr{W} \triangleq \max_{\subseteq}(\mathscr{V} \cup \mathscr{W}).$$

Lemma 5: $\langle X, \ \cup \rangle$ is isomorphic to $\langle \mathscr{X}_{\mathfrak{L}}, \ \vee \rangle$.

Proof: Define $f : X \to \mathscr{X}_{\mathfrak{L}}$ such that

$$f(A) = \cap \{\mathscr{U} \in \mathscr{P}(\mathscr{U}_{\mathfrak{L}}) \mid \cup \mathscr{U} = A\}.$$

$f(A) \in \mathscr{X}_{\mathfrak{L}}$, since for every $\cup \mathscr{U} = A$, if $U_1, U_2 \in \mathscr{U}$ and $U_1 \subsetneq U_2$, then $\cup(\mathscr{U} \backslash \{U_1\}) = A$. Hence, there is no $U_1, U_2 \in f(A)$ such that $U_1 \subsetneq U_2$ and therefore, $f(A) = \max_{\subseteq} f(A)$.

Now we prove f is bijective. Note that $\cup f(A) = A$, because if $x \in A$, then $x \in U$ for some $U \in \mathscr{U}$ whenever $\cup \mathscr{U} = A$. Hence, $f(A) = f(B) \Rightarrow A = \cup f(A) = \cup f(B) = B$. Thus, f is injective.

For any $\mathscr{V} \in \mathscr{X}_{\mathfrak{L}}$, let $\cup \mathscr{V} = A$. Since $\mathscr{V} = \max_{\subseteq} \mathscr{V}$ (no $U_1, U_2 \in \mathscr{V}$ such that $U_1 \subsetneq U_2$) and Lemma 4, $f(A) = \mathscr{V}$. Thus, f is surjective. Hence, f is bijective.

To prove f is an isomorphism, it suffices to show $f(U_1) \vee f(U_2) = f(U_1 \cup U_2)$ only for unsplittable sets $U_1, U_2 \in \mathscr{U}_{\mathfrak{L}}$. Since U_1, U_2 are unsplittable sets, $f(U_1) = \{U_1\}$, $f(U_2) = \{U_2\}$. If $U_1 \subseteq U_2$ or $U_2 \subseteq U_1$, then

$$\begin{aligned} f(U_1 \cup U_2) &= \cap \{\mathscr{U} \in \mathscr{P}(\mathscr{U}_{\mathfrak{L}}) \mid \cup \mathscr{U} = U_1 \cup U_2\} \\ &= \cap \{\mathscr{U} \in \mathscr{P}(\mathscr{U}_{\mathfrak{L}}) \mid \cup \mathscr{U} = \max_{\subseteq}\{U_1, U_2\}\} \\ &= \max_{\subseteq}\{U_1, U_2\} = f(U_1) \vee f(U_2), \end{aligned}$$

where $\max_{\subseteq}\{U_1, U_2\} \in \{U_1, U_2\}$.

If $U_1 \not\subseteq U_2$ and $U_2 \not\subseteq U_1$, by Lemma 4, there exist no $U_3, U_4 \in \mathscr{U}_{\mathfrak{L}} \backslash \{U_1, U_2\}$ such that $U_1 \cup U_2 = U_3 \cup U_4$. Hence, $f(U_1 \cup U_2) = \{U_1, U_2\} = \max_{\subseteq}\{U_1, U_2\} = f(U_1) \vee f(U_2)$.

f is an isomorphism; $\langle X, \ \cup \rangle$ is isomorphic to $\langle \mathscr{X}_{\mathfrak{L}}, \ \vee \rangle$. ■

2) Mapping of DL-Path Algebra:

Corollary 13: Given a DL-path algebra $\mathfrak{B}$, $\langle \mathcal{X}, \ \oplus \rangle$ is isomorphic to $\langle \mathscr{X}_{\mathfrak{B}}, \ \vee \rangle$, defined as:

$$\mathscr{X}_{\mathfrak{B}} \triangleq \{A \in \mathscr{P}(K) \mid A = \min_{\preccurlyeq_{\mathfrak{B}}} A\}, \quad A \vee B \triangleq \min_{\preccurlyeq_{\mathfrak{B}}}(A \cup B),$$

for some set K.

Proof: Since $\mathfrak{B}$ is a DL-path algebra, $\langle \mathcal{X}, \ \oplus \rangle$ is isomorphic to a lattice of sets $\langle X \subseteq \mathscr{P}(K), \ \cup, \ \cap \rangle$. Note that it is always possible that $\oplus$ is isomorphic to $\cap$ or to $\cup$ by inverting the isomorphic lattice. Choose $\oplus$ be isomorphic to $\cup$. For $a, b \in \mathcal{X}$, and their isomorphic counterparts $A, B \in X$:

$$b \preccurlyeq_{\mathfrak{B}} a \Leftrightarrow b = b \oplus a \Leftrightarrow B = B \cup A \Leftrightarrow B \supseteq A.$$

Hence, $\min_{\preccurlyeq_{\mathfrak{B}}}$ is isomorphic to $\max_{\subseteq}$. By Lemma 5, it follows that $\langle \mathcal{X}, \ \oplus \rangle$ is isomorphic to $\langle \mathscr{X}_{\mathfrak{B}}, \ \vee \rangle$. ■

Lattices of sets have the property that we can represent $\langle \mathcal{X}, \ \oplus \rangle$ isomorphically as the structure $\langle \mathscr{X}_{\mathfrak{B}}, \ \vee \rangle$. Thus, $a \oplus b = b$ is isomorphically represented as $\min_{\preccurlyeq_{\mathfrak{B}}}\{a, b\} = \{b\}$, and $a \oplus b = c$ where $c \neq a, b$ is isomorphically represented as $\min_{\preccurlyeq_{\mathfrak{B}}}\{a, b\} = \{a, b\}$, such that c in $\mathcal{X}$ is uniquely determined by $\{a, b\}$ in $\mathscr{X}_{\mathfrak{B}}$.

C. Theorems for Path Algebras

Theorem 14: If $\mathfrak{B}$ is a DL-path algebra, then the path algebra solution, $\delta(v, \ v_0) = \bar{1}$ for $v = v_0$, and for all $v \neq v_0$,

$$\delta(v, \ v_0) = \bigoplus_{e \in (v, \ w)} \mathcal{L}_{\mathfrak{B}}(e) \otimes \delta(w, \ v_0).$$

is isomorphic to the multi-value solution to S-instance I given by $\delta(v,\ v_0) = \{\psi(v_0)\}$ for $v = v_0$, and for all $v \neq v_0$,

$$\delta(v,\ v_0) \in \min_{\preccurlyeq_{\mathfrak{B}}}\{\psi(eP)|e \in (v,\ w), \\ P \in \mathcal{P}(w,\ v_0),\ \psi(P) = \delta(w,\ v_0)\}.$$

Proof: Note by Corollary 13, given a DL-path algebra $\mathfrak{B}$, $\langle \mathcal{X},\ \oplus\rangle$ is isomorphic to $\langle \mathscr{X}_{\mathfrak{B}},\ \vee\rangle$, which allows us to map $\oplus$ to $\min_{\preccurlyeq_{\mathfrak{B}}}$ and we have that

$$\mathcal{L}_{\mathfrak{B}}(e) \otimes \delta(w,\ v_0)$$

is isomorphic to

$$\psi(eP),$$

for $e \in (v,\ w), P \in \mathcal{P}(w,\ v_0)$ and $\psi(P) \in \delta(w,\ v_0)$. ■

Theorem 15: If $\mathfrak{B}$ is a super-unitary DL-path algebra, then every $\mathfrak{B}$-instance has a solution.

Proof: If $\mathfrak{B}$ is a DL-path algebra, by Theorem 14, we can rewrite the solution of path algebra in terms of the solution to generalised routing problem.

Theorem 10 state that if $\mathfrak{B}$ is super-unitary, then $\mathscr{R}_{S[\mathfrak{B}]}$ is anti-reflexive. Theorem 5 shows that if $\mathscr{R}_{S[\mathfrak{B}]}$ is anti-reflexive, then every $S[\mathfrak{B}]$-instance is anti-reflexive. Theorem 12 states that if an $S[\mathfrak{B}]$-instance is anti-reflexive, then the corresponding $S[\mathfrak{B}]$-instance has a solution. ■

Theorem 16: If a DL-path algebra $\mathfrak{B}$-instance I is absorptive, then I has a solution.

Proof: If $\mathfrak{B}$ is a DL-path algebra, by Theorem 14, we can rewrite the solution of path algebra in terms of the solution to generalised routing problem.

Theorem 12 states that if $\mathfrak{B}$-instance I is absorptive, then the corresponding $S[\mathfrak{B}]$-instance is anti-reflexive.

Theorem 8 states that if an $S[\mathfrak{B}]$-instance is anti-reflexive, then it has a fixed-point solution. Finally, Theorem 20 shows that A fixed point solution for an $S[\mathfrak{B}]$-instance corresponds to an $S[\mathfrak{B}]$-instance solution. ■

VI. Sobrinho's Routing Algebra

Let

$$\mathcal{A} = \langle \Sigma,\ L,\ \lesssim,\ \otimes\rangle$$

be a Sobrinho routing algebra. Let L^* be the set of all finite sequences of labels in L. We usually write a sequence $(l_n, l_{n-1}, ..., l_1)$ as $\vec{l} \in L^*$. Define $\vec{l} \otimes \sigma \triangleq l_n \otimes \cdots \otimes l_1 \otimes \sigma$, and $l \cdot \vec{l} \triangleq (l, l_n, l_{n-1}, ..., l_1)$.

We assume that $\lesssim$ is a total order. We construct an associated policy structure

$$S[\mathcal{A}] = \langle \Sigma,\ \preccurlyeq_{\mathcal{A}},\ \sqsubseteq_{\mathcal{A}}\rangle,$$

where $\preccurlyeq_{\mathcal{A}} = \lesssim$ and

$$\sigma \sqsubseteq_{\mathcal{A}} \beta \quad \text{if} \quad \text{there exists } \vec{l} \in L^* \text{ such that } \beta = \vec{l} \otimes \sigma.$$

Clearly $\preccurlyeq_{\mathcal{A}}$ is a partial order and it is easy to check that $\sqsubseteq_{\mathcal{A}}$ is a preorder.

Lemma 6: If $\mathcal{A}$ is monotone and $(\sigma,\ \beta) \in \mathscr{R}_{S[\mathcal{A}]}$, then there exists a label sequence $\vec{l}$ such that $\vec{l} \otimes \sigma \prec_{\mathcal{A}} \beta$.

Proof: If $(\sigma,\ \beta) \in \mathscr{R}_{S[\mathcal{A}]}$, then there must exist a k, $1 \leq k$, and σ_i, $1 \leq i \leq 2k+1$ such that

$$\sigma_1 \sqsubseteq_{\mathcal{A}} \sigma_2 \prec_{\mathcal{A}} \sigma_3 \ldots \sigma_{2k-1} \sqsubseteq_{\mathcal{A}} \sigma_{2k} \prec_{\mathcal{A}} \sigma_{2k+1}$$

We prove the claim by induction on k. If $n = 1$ then we have

$$\sigma_1 \sqsubseteq_{\mathcal{A}} \sigma_2 \prec_{\mathcal{A}} \sigma_3.$$

Then there exists a label sequence $\vec{l}$ such that $\sigma_2 = \vec{l} \otimes \sigma_1$. So we have $\vec{l} \otimes \sigma_1 \prec_{\mathcal{A}} \sigma_3$. Now we show that if the claim is true for some $1 \leq i-1 < k$, then it is true for i. If the claim is true for $i-1$, then there must exist a sequence $\vec{l_1}$ such that

$$\vec{l_1} \otimes \sigma_1 \prec_{\mathcal{A}} \sigma_{2i-1} \sqsubseteq_{\mathcal{A}} \sigma_{2i} \prec_{\mathcal{A}} \sigma_{2i+1}.$$

But this means that there is a a sequence $\vec{l_2}$ such that $\sigma_{2i} = \vec{l_2} \otimes \sigma_{2i-1}$. By monotonicity we have $\sigma_{2i-1} \preccurlyeq_{\mathcal{A}} \vec{l_2} \otimes \sigma_{2i-1} = \sigma_{2i}$. Therefore

$$\vec{l_1} \otimes \sigma_1 \prec_{\mathcal{A}} \sigma_{2i-1} \preccurlyeq_{\mathcal{A}} \sigma_{2i} \prec_{\mathcal{A}} \sigma_{2i+1},$$

which gives us

$$\vec{l_1} \otimes \sigma_1 \prec_{\mathcal{A}} \sigma_{2i+1}.$$

So the claim must be true for k. ■

Theorem 17: For any Sobrinho's routing algebra $\mathcal{A}$, $\mathscr{R}_{S[\mathcal{A}]}$ is anti-reflexive if and only if $\mathcal{A}$ is monotone.

Proof: (If -) Suppose that $\mathcal{A}$ is monotone but that $\mathscr{R}_{S[\mathcal{A}]}$ as not anti-reflexive. So there must exist a σ such that $(\sigma,\ \sigma) \in \mathscr{R}_{S[\mathcal{A}]}$. But by Lemma 6 this means that there exists a label sequence $\vec{l}$ such that $\vec{l} \otimes \sigma \prec_{\mathcal{A}} \sigma$, which contradicts monotonicity.

(Only if -) Suppose that $\mathscr{R}_{S[\mathcal{A}]}$ as anti-reflexive, but that $\mathcal{A}$ is not monotone. There must exist a label l and a signature σ such that $\lambda \otimes \sigma \prec \sigma$. But this means that

$$\sigma \sqsubseteq_{\mathcal{A}} \lambda \otimes \sigma \prec_{\mathcal{A}} \sigma,$$

and so $\mathscr{R}_{S[\mathcal{A}]}$ is not anti-reflexive. This contradiction tells us that $\mathcal{A}$ must be monotone. ■

Recall that $\mathcal{A}$ is *strictly monotone* if for all labels l and all signatures $\sigma \neq \infty$ we have $\sigma \prec_{\mathcal{A}} l \otimes \sigma$. The next theorem shows that strict monotonicity is also a universal condition for routing algebras.

Theorem 18: For any Sobrinho's routing algebra $\mathcal{A}$, if $\mathcal{A}$ is strictly monotone, then $\mathscr{R}_{S[\mathcal{A}]}$ is anti-reflexive

Proof: Suppose that $\mathcal{A}$ is strictly monotone, then it is monotone, and the proof proceeds as in the proof of Theorem 17. ■

For routing algebras $\mathcal{A}$, all $\mathcal{A}$-instances must have the form $I = (\mathcal{G},\ \psi,\ B)$ where $\infty \in B$. Normally we will have $B = \{\infty\}$. That is, paths associated with ∞ cannot be used in a solution.

Theorem 19: If an $\mathcal{A}$-instance I is free, then the corresponding $S[\mathcal{A}]$-instance is has an anti-reflexive routing relation.

Proof: Suppose tht an $\mathcal{A}$-instance I is free, but the corresponding $S[\mathcal{A}]$-instance $\langle \mathcal{P}_{\psi},\ \preccurlyeq_I,\ \sqsubseteq_I\rangle$ does not have an anti-reflexive routing relation. Then there must exist distinct

$P_1, ..., P_m \in \mathcal{P}(v_0)$ where $m = 2k$ is even, and $\psi(P_i) \neq \infty$, such that

$$P_1 \preccurlyeq_I P_2 \sqsubset_I P_3 \cdots \preccurlyeq_I P_m \sqsubset_I P_1.$$

Since $P_{2i-1} \preccurlyeq_I P_{2i}$, for $1 \leq i \leq k$, there must exists paths W_i so that $P_{2i} = W_i P_{2i-1}$. Let $\sigma_j = \psi(P_{2i-1}$, for $1 \leq j \leq k$.

$$\sigma_1 \sqsubseteq_{\mathcal{A}} \psi(W_1 P_1) \prec_{\mathcal{A}} \sigma_2 \cdots \sqsubseteq_{\mathcal{A}} \psi(W_k P_{2k-1}) \prec_{\mathcal{A}} \sigma_1.$$

Therefore, the directed $W_1 W_2 \cdots W_k$cycle is not free, and we have a contradiction. ■

Theorem 20: Let $\mathcal{A}$ be a routing algebra and an $\mathcal{A}$-instance $I = (G,\ \mathcal{L}_{\mathcal{A}},\ \sigma_0)$. and let I' be the corresponding instance for the policy structure $S[\mathcal{A}]$. Every multi-value solution I' corresponds to an $\mathcal{A}$-instance solution.

Proof: Recall that a multi-value solution for the corresponding I' is a function δ such that $\delta(v_0,\ v_0) = \{\psi(v_0)\}$ and for all $v \neq v_0$,

$$\begin{aligned}\delta(v,\ v_0) = \min\nolimits_{\preccurlyeq}\{\psi(eP) | \psi(eP) \notin B,\\ e \in (v,\ w),\\ P \in \mathcal{P}(w,\ v_0),\\ \psi(P) \in \delta(w,\ v_0)\}.\end{aligned}$$

Note that $\psi(v_0) = \sigma_0$, so we have $\delta(v_0,\ v_0) = \{\sigma_0\}$. Since ψ is derived from $\mathcal{L}_{\mathcal{A}}$, we have for all $v \neq v_0$,

$$\delta(v, v_0) = \min\nolimits_{\preccurlyeq}\{\mathcal{L}_{\mathcal{A}}(e) \otimes \sigma \mid e \in (v, w), \sigma \in \delta(w, v_0)\}.$$

This is exactly the definition of an $\mathcal{A}$-instance solution. ■

Theorem 21: If routing algebra $\mathcal{A}$ is monotone, then every $\mathcal{A}$-instance I is free.

Proof: Theorem 17 states that if $\mathcal{A}$ is monotone, then $\mathscr{R}_{S[\mathcal{A}]}$ is anti-reflexive. Theorem 5 shows that every $S[\mathcal{A}]$-instance is anti-reflexive. Theorem 19 states that if an $S[\mathcal{A}]$-instance is anti-reflexive, then the corresponding $\mathcal{A}$-instance I is free. ■

Theorem 22: If an $\mathcal{A}$-instance I is free, then it has a solution.

Proof: Theorem 19 states that if an $\mathcal{A}$-instance I is free, then the corresponding $S[\mathcal{A}]$-instance is anti-reflexive. Theorem 8 states that if an $S[\mathcal{A}]$-instance is anti-reflexive, then it has a multi-value solution. Finally, Theorem 20 shows that A multi-value solution for an $S[\mathcal{A}]$-instance corresponds to an $\mathcal{A}$-instance solution. ■

VII. Discussions

Our theory helps to clarify some of the rather confusing issues involved with the distinction between the value associated with a stable routing, the values associated with the "best paths," and the best paths themselves. For example, if $\mathfrak{B}$ is a path algebra and $I = (G,\ \psi)$ is a $\mathfrak{B}$-instance, then we can view the routing structure S_I as a generalised SPP, and

$$\delta(v,\ v_0) = \bigoplus \psi(\Delta(v,\ v_0)),$$

where Δ is a multi-path solution to the generalised SPP. Returning to the example presented in Section III-D, we have

$$\begin{aligned}\Delta(\text{Rome},\ \text{Moscow}) \;=\; \{&(\text{Moscow},\ \text{Prague},\ \text{Rome}),\\ &(\text{Moscow},\ \text{London},\ \text{Paris},\ \text{Rome})\}\end{aligned}$$

as the "best paths" from Moscow to Rome, while

$$\psi(\Delta(\text{Rome},\ \text{Moscow}) = \{\langle 300,\ 30\rangle,\ \langle 571,\ 70\rangle\}$$

represents the "best values" associated with those paths. Finally, we have

$$\begin{aligned}\langle 300,\ 70\rangle &= \bigoplus\{\langle 300,\ 30\rangle,\ \langle 571,\ 70\rangle\}\\ &= \bigoplus \psi(\Delta(\text{Rome},\ \text{Moscow}).\end{aligned}$$

as the single value associated with the solution at the level of the path algebra $\mathfrak{B}$. In the case of this example, the value $\langle 300,\ 70\rangle$ is not associated with any single path. It is hoped that this clarification will prove useful in future work.

We mention a few questions that may give rise to interesting research. First, we ask if the framework presented here can be extended to include routing algebras with preference orders rather than the more restricted total order. Simply "moding out" by equivalence classes does not seem to work since labels in routing algebras can actually take on the form of small programs that inspect the syntactic details of a signature (see metarouting work of [5]).

Even though anti-reflexive policy relations are implied by both super-unitary/nilpotent path algebras, the multi-path solution concept in generalised routing problem is not sufficient to capture the general solution in path algebra with non-distributive lattices. The implication of non-distributive lattices to application in routing problem is unclear so far. Either such systems are of no interest in routing, or some type of generalisation to the theory presented here is needed.

Our theory of policy structures and routing structures is based purely on abstract relations and their relational properties and not on the syntactic or axiomatic details of the policy-based theories. This leads us to suspect that there are many "algebraic" theories that may be of interest in network routing that might also fall within the scope of our framework. That, is, perhaps there is a spectrum of algebraic routing theories that includes path algebras and routing algebras as interesting instances of something more general. In terms of metarouting, it would then be interesting to think about meta-languages that can define a broader spectrum of algebraic structures, not just routing algebras as is currently the case.

As mentioned in Section III-D, there seems to be an interesting relationship between algebraic paradigms and forwarding paradigms. Can this connection can be explored formally with some kind of "algebraic" theory of forwarding? Is network coding theory [17] a candidate theory?

Acknowledgement

The authors would like to thank João Luíz Sobrinho, Alex Gurney, and the anonymous reviewers for helpful comments and corrections. Any remaining errors are of course our own. Chi-kin Chau is grateful to The Croucher Foundation for financial support. Richard Gibbens acknowledges support from EPSRC grant GR/S86266/01.

REFERENCES

[1] Y. Rekhter and T. Li, "A Border Gateway Protocol," March 1995, rFC 1771 (BGP version 4).

[2] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 232–243, April 2002.

[3] J. Sobrinho, "An algebraic theory of dynamic network routing," 2005, to appear in IEEE/ACM Transactions on Networking.

[4] ——, "Network routing with path vector protocols: Theory and applications," in *Proc. ACM SIGCOMM*, September 2003.

[5] T. G. Griffin and J. Sobrinho, "Metarouting," in *Proc. ACM SIGCOMM*, September 2005.

[6] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms (First Edition)*. The MIT Press, 1992.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1989.

[8] M. Mohri, "Semiring frameworks and algorithms for shortest-distance problems," *Journal of Automata, Languages and Combinatorics*, vol. 7, no. 3, pp. 321–350, 2002.

[9] M. Gondran and M. Minoux, *Graphs and Algorithms*. Addison Welsey, 1984.

[10] B. Carré, *Graphs and Networks*. Oxford University Press, 1979.

[11] T. G. Griffin, A. D. Jaggard, and V. Ramachandran, "Design principles of policy languages for path vector protocols," in *Proc. ACM SIGCOMM*, September 2003.

[12] G. Rote, "Path problem in graphs," in *Computational Graph Theory*, G. Tinhofer, E. Mayr, H. Noltemeier, and M. Syslo, Eds. Springer-Verlag, 1990, pp. 155–189.

[13] D. Lehmann, "Algebraic structures for transitive closure," *Theoretical Computer Science*, vol. 4, pp. 57–66, 1977.

[14] A. V. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*. Addison Welsey, 1974.

[15] S. Gaubert and Max Plus, "Methods and applications of (max, +) linear algebra," in *Proc. Symposium on Theoretical Aspects of Computer Science*, 1997.

[16] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order (Second Edition)*. Cambridge University Press, 2002.

[17] S.-Y. R. L. R. Ahlswede, N. Cai and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, pp. 1204–1216, 2000.