Routing on Multiple Optimality Criteria: Theory and Protocols

Miguel Alves Ferreira
Instituto Superior Técnico

***Abstract*— The concept of optimal path in a network is bound to: (a) a set of path attributes; (b) a binary extension operation on attributes, which calculates path attributes from link attributes; and (c) a total order on attributes, which defines an optimality criterion by establishing the relative preference among path attributes. Standard vectoring protocols, such as EIGRP, BGP, DSDV or Babel, only solve the problem of routing on optimal paths if the binary extension operation is isotone for the total order, thus leaving out many optimality criteria of practical interest. Isotonicity expresses that the relative preference between two attributes is preserved when they are extended with a third attribute.**

We present a general solution to routing on optimal paths and, more broadly, to routing on multiple optimality criteria. A fundamental idea is the derivation of partial orders on attributes that satisfy isotonicity and respect every optimality criterion of a designated collection of such criteria. We design new routing protocols that compute on partial orders and have each node elect a set of attributes, rather than a single attribute, as standard vectoring protocols do. Our evaluation over realistic networks shows that the protocols devised require only a few elected attributes per destination and converge fast.

1. Introduction

The concept of optimal path is defined in general terms within an algebraic framework [1]–[5]. This framework is premised on: (a) a set of attributes, which represents arbitrary performance metrics; (b) a binary extension operation on attributes, which calculates path attributes from link attributes; and (c) a total order on attributes, which establishes the relative preference among path attributes and, thus, models an optimality criterion. The optimal attribute from source to destination in a network is the most preferred from among the path attributes from source to destination and an optimal path is one with such attribute.

Standard vectoring protocols, such as RIP [6], EIGRP [7], BGP [8], DSDV [9] and Babel [10], compute path attributes at nodes to reach destinations separately per destination and expedite data-packets based exclusively on their destinations. These protocols have each node of a network iterate: (a) a binary extension operation on attributes, which calculates path attributes to reach the destination from those advertised by neighbors; and (b) a selection operation in accordance with a total order on attributes, which elects a single attribute from among the path attributes to reach the destination via its neighbors. This approach only discovers optimal attributes and paths only if the binary extension operation is isotone for the total order. Isotonicity means that the relative preference between any two attributes is preserved when they are extended with any third attribute [1]–[3]. Without isotonicity, standard vectoring protocols do not route on optimal paths, in general [3], [11], [12].

The main contribution of this work is a solution to routing on optimal paths for multiple optimality criteria, which entails a solution to routing on optimal paths for an arbitrary optimality criterion as a special case. The solution to optimal path routing for an arbitrary optimality is based on two new ideas. The first idea is the reduction of total orders on attributes so that isotonicity is satisfied in the reduced orders [13]. Reducing means letting some pairs of attributes that were comparable become incomparable, none of the two being preferred to the other. This process leads to partial orders. In a partial order [14], one attribute is preferred to the other or the two are incomparable. The set of dominant attributes from source to destination in a network consists of the path attributes from source to destination that are not less preferred than any path attribute. This is a set of mutually incomparable attributes, in general, and it includes the optimal attribute for the original optimality criterion.

A second idea is to design partial-order vectoring protocols, which generalize standard vectoring protocols to operate according to partial orders on attributes. These protocols compute path attributes at nodes to reach destinations separately per destination, as standard vectoring protocols do. However, they have each node elect a set of incomparable attributes from among the path attributes to the reach the destination via its neighbors, as opposed to a single attribute. Since each node possibly elects multiple attributes per destination, data-packets can no longer be forwarded exclusively based on destination. Therefore, the node differentiates its elected attributes to reach the destination by assigning them unique identifiers, which are advertised alongside the respective attributes [15]. Data-packets are guided along computed paths through label-switching at each intermediate node. If the extension operation is isotone for the partial order, then partial-order vectoring protocols route on dominant paths, namely on the original optimal paths.

The solution to optimal path routing for multiple optimality criteria is further based on the reduction of total orders to partial orders. First, the total orders of a designated collection of optimality criteria are intersected. The intersection is a partial order that reduces each of the total orders. Second, the resulting partial order is once more reduced to satisfy isotonicity. The optimal attributes for all optimality criteria are included in the respective sets of dominant attributes. Then, a partial-order vectoring protocol provides routing optimally and concurrently for all optimality criteria.

This document is structured as follows. In Section 2, we illustrate the optimal path routing problem for an arbitrary optimality criterion and for multiple optimality criteria. Section 3 develops a procedure that starts with a generic collection of optimality criteria and ends with a partial order that respects each criterion and satisfies isotonicity. Section 4 designs partial-order vectoring protocols. In Section 5, we evaluate these protocols. Section 6 reviews previous work. Finally, in Section 7, we present our conclusions.

2. Routing based on Widths and Lengths

We illustrate how partial-order vectoring protocols support routing on multiple optimality criteria. In the upcoming examples, every link and path in a network is characterized by a pair *width-length* from the *Cartesian product* [14] between positive widths and finite lengths. A *width* is a non-negative real number or infinity. Widths extend with minimum: the width of a path is the minimum of the widths of its links. A width stands for a metric like capacity [16] or available bandwidth [16]. A *length* is a non-negative real number or infinity. Lengths extended with addition: the length of a path is the sum of the lengths of its links. A length stands for a metric such as delay [17] or transmission count [18]. Consequently, the *extension* of a width-length (w, l) with a width-length (w', l') is $(\min\{w, w'\}, l + l')$.

Section 2.1 explores routing on shortest-widest paths. Section 2.2 goes on to explore routing concurrently on shortest-widest and widest-shortest paths.

2.1. Routing on Shortest-Widest Paths

A *shortest-widest path* from source to destination is one of minimum length among those of maximum width [19]. The shortest-widest order is the total order underlying the selection of shortest-widest paths. In the *shortest-widest order* (lexicographic order) on width-lengths, a width-length (w, l) is *preferred* to a width-length (w', l') if its width is greater, $w > w'$, or if its width is the same and its length is smaller, $w = w'$ and $l < l'$.

In the network of Figure 1, each link is annotated with a width-length and all nodes want to route data-packets to destination x on shortest-widest paths. By inspection, we infer that the shortest-widest path from w to x is wvx with width-length $(20, 9) = (\min\{20, 20\}, 8 + 1)$. The shortest-widest path from u to x is uwx with width-length of uwx is $(10, 5) = (\min\{10, 20\}, 3 + 2)$.

With a *standard vectoring protocol*, nodes continually extend, elect and advertise to in-neighbors path width-lengths to reach the destination. The destination initiates the computation by advertising the width-length comprising infinite width and zero length to its in-neighbors. When a node receives a width-length from an out-neighbor, it first extends the width-length of the link to the out-neighbor with the width-length received from the out-neighbor, thus learning a candidate width-length to reach the destination via the out-neighbor. Then, the node elects the shortest-widest width-length from among its candidate width-lengths to reach the destination.

Returning to the network of Figure 1, destination x elects $(\infty, 0)$. Node v elects the candidate $(20, 1)$ via x, the extension of $(20, 1)$ with $(\infty, 0)$. Node w has a candidate $(20, 9)$ via v, the extension of $(20, 8)$ with $(20, 1)$, and a candidate $(10, 2)$ via x, the extension of $(10, 2)$ with $(\infty, 0)$. It elects $(20, 9)$, the shortest-widest width-length in the set $\{(20, 9), (10, 1)\}$. Node u has a candidate $(5, 2)$ via v, the extension of $(5, 1)$ with $(20, 1)$, and a candidate $(10, 12)$ via w, the extension of $(10, 3)$ with $(20, 9)$. It elects $(10, 12)$, the shortest-widest width-length in the set $\{(5, 2), (10, 12)\}$.

Figure 1 shows the stable state of the protocol. Each line in the table next to each node consists of a candidate width-length and the out-neighbor from which this width-length was learned. Data-packets are guided along the paths computed with each intermediate node forwarding them to the out-neighbor from which the elected attribute was learned. Node u forwards data-packets to w, which forwards them to v, which delivers them to x. Data-packets sourced at u and destined for x travel along path $uwvx$, whereas the shortest-widest path from u to x is uwx. This is do to the failure of isotonicity of the extension of width-lenghts for the shortest-widest order: width-length $(20, 9)$ is preferred to $(10, 2)$, but $(10, 12)$, the extension of $(10, 3)$ with $(20, 9)$, is less preferred than $(10, 5)$, the extension of $(10, 3)$ with $(10, 2)$.

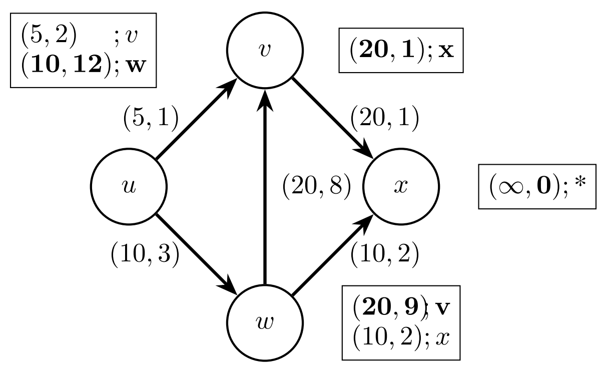


Figure 1: Stable state of a standard vectoring protocol operating on the shortest-widest order for destination x. Links are annotated with width-lengths. Elected width-lengths are in bold.

In the *product order* [14] on width-lengths, a width-length (w, l) is preferred to a width-length (w', l') if it is different, $(w, l) \neq (w', l')$, and both its width is greater or equal, $w \geq w'$, and its length is smaller or equal, $l \leq l'$. Two width-lengths are *incomparable*, none of them being preferred to the other, if one has greater width but the other smaller length. The product order is a partial order on width-lengths. A width-length in a set of width-lengths is *dominant* if no width-length in the set is preferred to it. A *dominant path* from a source to a destination in a network is one whose width-length is dominant in the set of all path width-lengths from the source to the destination. Figure 2 shows the same network as Figure 1. The dominant paths from w to x are wx and wvx. Their width-lengths, respectively, $(10, 2)$ and $(20, 9)$, are incomparable. The dominant paths from u to x are uvx and uwx, width-lengths $(10, 5)$ and $(5, 2)$, respectively.

With a *partial-order vectoring protocol*, nodes repeatedly extend, elect and advertise sets of incomparable width-lengths to reach the destination. When a node receives a set of width-lengths from an out-neighbor, it first extends the width-length of the link to the out-neighbor with every width-length received from the out-neighbor, thus obtaining a set of candidate width-lengths to reach the destination via the out-neighbor. Then, the node elects the set of dominant width-lengths from among its candidate width-lengths to reach the destination. It assigns a distinct label to each elected attribute to reach the destination. Labels are advertised to its in-neighbors alongside the respective attributes to enable the expedition of data-packets along the paths computed by switching labels at intermediate nodes [15].

Returning to the network of Figure 2, destination x elects $(\infty, 0)$ and assigns it label 1. Node v elects the candidate $(20, 1)$ learned from x with label 1 and assigns it label 2. Node w has a candidate $(20, 5)$ learned from v with label 2 and a candidate $(10, 2)$ learned from x with label 1. It elects both width-lengths, because the set of dominant width-lengths in set $\{(20, 9), (10, 2)\}$ is itself, assigning label 4 to $(20, 9)$ and label 6 to $(10, 2)$. Node u has a candidate $(5, 2)$ learned from v with label 3 and candidates $(10, 12)$ and $(10, 5)$ learned from w with labels 4 and 6, respectively. It elects $(5, 2)$ and $(10, 5)$, since the set of dominant width-lengths in set $\{(5, 2), (10, 5), (10, 12)\}$ is $\{(5, 2), (10, 5)\}$, assigning label 3 to $(5, 2)$ and label 5 to $(10, 5)$.

Figure 2 shows the stable state of the protocol. Each line in the table next to each node is of the form:

(width, length) **,** *label* **;** *next.hop* **,** *next.label*,

and reads as follows:

1) A data-packet sourced at the node and meant to travel along a path with width-length *(width, length)* is forwarded to out-neighbor *next.hop* carrying label *next.label*;
2) A data-packet arriving at the node from an in-neighbor carrying label *label* is forwarded to out-neighbor *next.hop* now carrying label *next.label*.

The partial-order vectoring protocol has the necessary routing and forwarding information to guide data-packets on dominant paths from u to x, namely on the shortest-widest path. Width-length $(10, 5)$ is the shortest-widest width-length from among the elected width-lengths at u to reach x. Therefore, node u forwards the data-packets to w carrying label 6. At w, incoming label 6 matches out-neighbor x and outgoing label 1. As a result, node w replaces label 6 with label 1 and delivers the data-packets to x.

2.2. Routing on Widest-Shortest Paths

A widest-shortest path from source to destination is one of maximum width among those of minimum length [19].

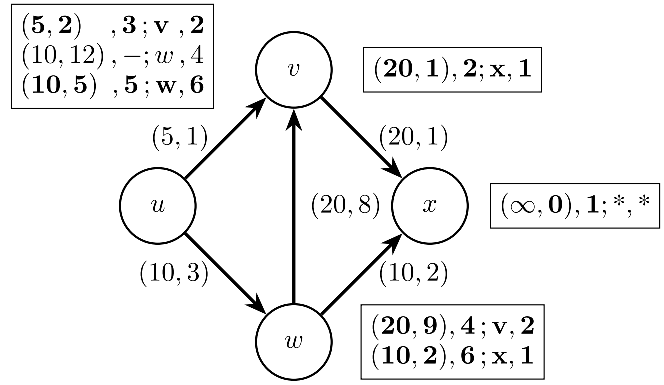


Figure 2: Stable state of a partial-order vectoring protocol operating on the product order on width-lengths for destination x.

The widest-shortest order is the total order underlying the selection of widest-shortest paths. In the *widest-shortest order* (co-lexicographic order) on width-lengths, a width-length (w, l) is preferred to a width-length (w', l') if its length is smaller, $l < l'$, or if its length is the same and its width is greater, $l = l'$ and $w > w'$.

The partial-order vectoring protocol allows for routing a flow of data-packets on a shortest-widest path or on a widest-shortest path, as it is more convenient for each specific flow. Going back to Figure 2, suppose that u wants to route a flow of data-packets to x along the widest-shortest path. Width-length $(5, 2)$ is the widest-shortest width-length from among the elected width-lengths at u to reach x. Therefore, node u marks the data-packets with label 2 and forwards them to v. At v, label 2 is swapped for label 1 and the data-packets are delivered to x.

3. Multiple Optimality and Dominance

We formulate the problem of routing on multiple optimality criteria and develop new routing concepts and constructions that promote a solution to this problem. Section 3.1 introduces the basic algebraic routing concepts and presents an algebraic statement for the problem of routing on multiple optimality criteria. Section 3.2 introduces partial orders and the possibility of reducing them so that the algebraic property of isotonicity is satisfied. Section 3.3 transforms the problem of routing on multiple optimality criteria into that of routing based on a partial order that respects all criteria and satisfies isotonicity.

3.1. Multiple Optimality

Optimal path problems can be formulated with generality in algebraic terms [1]–[5]. A set S of attributes represents arbitrary performance metrics. Every link and path in a network is associated an attribute. The attribute of a path is obtained from the attributes of its constituent links through a binary extension operation, denoted by $\oplus$, that we assume to be associative and commutative with neutral attribute ϵ.

Letting $a(uv)$ denote the attribute of link uv, the attribute $a(P)$ of path $P = u_0u_1 \ldots u_n$ is given by

$$a(P) = a(u_0u_1) \oplus a(u_1u_2) \oplus \ldots \oplus a(u_{n-1}u_n). \quad (1)$$

The attribute of a trivial path, containing just one node, is ϵ.

We consider a collection I of optimality criteria. Optimality criterion $i \in I$ is modeled by a total order $\preceq_i$ on attributes, which is an antisymmetric, transitive, and connex binary relation on the set of attributes. Connexity means that $a \preceq_i b$ or $b \preceq_i a$ for all $a, b \in S$. We write $a \prec_i b$ for $a \preceq_i b$ and $a \neq b$. When $a \prec_i b$, we say that a is *i-preferred* to b and that b is *less i-preferred* than a. The null attribute $\bullet$ is the least preferred of all attributes and represents the absence of a valid path.

Given a network, the *i-optimal* attribute from source s to destination t in the network, which we denote by $a_i^*(s,t)$, is the i-optimal attribute in the set of all path attributes from s to t. A path from s to t is i-optimal if its attribute is $a_i^*(s,t)$.

Definition 3.1. Binary extension operation $\oplus$ on attributes is *inflationary* for total order $\preceq_i$ if:

$$b \preceq_i a \oplus b, \text{ for all attributes } a, b. \quad (2)$$

Inflation expresses that the attribute of a path does not become i-preferred when this path is prefixed by another path [2], [20]. This property is related to the termination of standard vectoring protocols to stable states that guide data-packets from sources to destinations. Inflation is usually satisfied by concrete performance metrics.

Definition 3.2. Binary extension operation $\oplus$ is *isotone* for total order $\preceq_i$ on attributes if:

$$b \preceq_i c \text{ implies } a \oplus b \preceq_i a \oplus c, \text{ for all attributes } a, b, c. \quad (3)$$

Isotonicity conveys that the relative preference between the attributes of two paths is preserved when these paths are prefixed by a third path [1], [2], [11]. This property is associated with the optimality of standard vectoring protocols. With isotonicity, standard vectoring protocols find the i-optimal attributes from sources to destination; without isotonicity, these attributes are not found, in general. Isotonicity is not met by most concrete performance metrics.

3.2. Partial Orders and Isotonic Reductions

A partial order $\preceq$ on attributes is a binary relation over S that satisfies antisymmetry, transitivity and *reflexivity*. Connexity implies reflexivity, so that a total order on attributes is a particular case of a partial order. We still write $a \prec b$ for $a \preceq b$ and $a \neq b$. When $a \prec b$, we still say a is preferred to b and b is less preferred than a. If $a \preceq b$ or $b \preceq a$, then a and b are comparable; otherwise, they are incomparable.

The *set of dominant attributes* in a set A of attributes, which we denote by $D(A)$, consists of the attributes of A with no attribute of A preferred to them:

$$D(A) = \{a \in A \,|\, \text{there is no } b \in A \text{ such that } b \prec a\}. \quad (4)$$

Given a network, the *set of dominant attributes* from source s to destination t, which we denote by $A^*(s,t)$, is the set of dominant attributes in the set of all path attributes from the source to the destination. A path from s to t is *dominant* if its attributes belongs to $A^*(s,t)$.

A fundamental idea of our approach is the possibility to reduce a partial order on attributes that does not satisfy isotonicity to one that does satisfy this property. Attaining isotonicity entails letting some pairs of attributes that were comparable in the original partial order become incomparable in the reduced one, namely those whose relative preference is inverted when extended with a third attribute. This idea is expressed precisely by the following definition [13].

Definition 3.3. An *isotonic reduction* of a partial order $\preceq$ on attributes for a binary extension operation $\oplus$ is a partial order that is contained in $\preceq$ and for which $\oplus$ is isotone.

The more the pairs of attributes that are comparable, the less the routing state required by the routing protocols devised in Section 4 and the more efficient these protocols are. We aspire to isotonic reductions with as many pairs of comparable attributes as possible. Theorem 3.1 shows that every partial order on attributes has a largest isotonic reduction by explicitly characterizing this reduction. This fact relies on the associativity of the binary extension operation.

Theorem 3.1 ([21]). *The largest isotonic reduction of a partial order $\preceq$ on attributes for a binary extension operation $\oplus$, which we denote by $\preceq_R$, consists of all pairs of comparable attributes whose relative preference is preserved when both are extended with any common third attribute:*

$$a \preceq_R b \text{ if } x \oplus a \preceq x \oplus b \text{ for all attributes } x, \quad \text{for all attributes } a, b. \quad (5)$$

Inflation remains a desirable algebraic property, as it is related to the termination of the routing protocols presented in Section 4 to stable states that guide data-packets from sources to destinations. Theorem 3.2 shows that the largest isotonic reduction of an inflationary partial order is itself inflationary. This result requires both associativity and commutativity of the binary extension operation. A corollary of this theorem is that the largest isotonic reduction of a partial order is non-trivial, in the sense that it includes other pairs of comparable attributes besides those consisting of an attribute and itself.

Theorem 3.2 ([21]). *Let $\oplus$ be a binary extension operation on attributes and $\preceq$ be a partial order. If $\oplus$ is inflationary for $\preceq$, then $\oplus$ is also inflationary for the largest isotonic reduction of $\preceq$ for $\oplus$.*

3.3. From Multiple Optimality to Dominance

We provide a procedure that starts with a collection of optimality criteria and ends with a partial order on attributes that is contained in each of the total orders of the collection and that satisfies isotonicity.

First, we reduce the total orders that define the optimality criteria in the collection I to a common partial order that is contained in each of them. Define the binary relation $\preceq_I$ on attributes as the *intersection* of all total orders $\preceq_i$, $i \in I$:

$$a \preceq_I b \text{ if } a \preceq_i b \text{ for all } i \in I, \text{ for all attributes } a, b. \quad (6)$$

It is easy to show that the intersection of partial orders is a partial order. Therefore, binary relation $\preceq_I$ is a partial order on attributes. Two attributes are comparable in $\preceq_I$ only if one is i-preferred to the other, for all $i \in I$. Because we assume $\oplus$ is inflationary for $\preceq_i$, for all $i \in I$, $\oplus$ is inflationary for $\preceq_I$. Since we do not assume $\oplus$ is isotone for $\preceq_i$, for some $i \in I$, $\oplus$ is not isotone for $\preceq_I$, in general.

Second, we further reduce the common partial order to satisfy isotonicity. If $\oplus$ is not isotone for $\preceq_I$, then we retain only the largest isotonic reduction of $\preceq_I$, as detailed in Theorem 3.1, which we denote by $\preceq_{I,R}$. Theorem 3.2 guarantees that $\oplus$ is also inflationary for $\preceq_{I,R}$. In the special case the collection consist of a single optimality criterion, this procedure comes down to the isotonic reduction of the total order that defines the optimality criterion. This way, we obtain a partial order that expresses a trade-off between the connexity of the order and isotonicity.

Given a network, let $A^*_{I,R}(s,t)$ be the set of dominant attributes from source s to destination t according to $\preceq_{I,R}$. By construction, partial order $\preceq_{I,R}$ is contained in $\preceq_i$, for all $i \in I$. Therefore, the i-optimal attributes from s to t are in the set of dominant attributes from s to t, for all $i \in I$:

$$a^*_i(s,t) \in A^*_{I,R}(s,t), \text{ for all } i \in I. \quad (7)$$

Thus, the problem of computing i-optimal attributes concurrently for all $i \in I$ has become that of computing sets of dominant attributes according to a partial order that is contained in $\preceq_i$, for all $i \in I$, and for which $\oplus$ is both inflationary and isotone.

4. Protocols for Dominant Paths

We design *partial-order vectoring protocols*, which are routing protocols that generalize standard vectoring protocols to operate according to partial orders. If isotonicity is satisfied, then these protocols are able to route on dominant paths, namely on optimal paths for any optimality criterion defined by a total order that contains the partial order. Section 4.1 presents the class of partial-order non-restarting vectoring protocols and Section 4.2 presents the class of partial-order restarting vectoring protocols.

4.1. Non-restarting protocol

In a *standard non-restarting vectoring protocol*, a destination maintains a single computation instance all through time. A node stores a candidate attribute to reach the destination via each of its out-neighbors; at all times, it elects the most preferred from among its candidates attributes to reach the destination. The elected attribute improves and worsens throughout the computation instance. The election of a new attribute that is worse than the old one triggers the process of some nodes settling on worse attributes than those they started out with by persistently electing and advertising to in-neighbors the best from among the candidates learned from out-neighbors. This process is possibly long-lasting and, in the meantime, may see data-packets trapped in forwarding-loops. The stable state of the protocol delivers data-packets from sources to destinations, although not necessarily along optimal paths. Non-restarting protocols are traditional from wired networks, with RIP [6],EIGRP [7] and BGP [8] as prototypical examples.

Distributed path computation. Standard non-restarting vectoring protocols can be generalized to work with a partial order $\preceq$ on attributes that satisfies both inflation and isotonicity. In the canonical partial-order non-restarting vectoring protocol, destination t initiates the computation instance by advertising singleton $\{\epsilon\}$ to all its in-neighbors. Algorithm 1 presents the pseudo-code of the protocol for when node u receives the advertisement of set B of attributes from out-neighbor v concerning destination t, $u \neq t$. Variable $DomTab_u[t,v]$ stores the set of candidate attributes at u to reach t via v and variable $Dom_u[t]$ stores the set of elected attributes at u to reach t.

Upon receiving set B from out-neighbor v, node u updates its set of candidate attributes via v to the extension of the attribute of its link to v with every attribute of B (line 1). Then, it determines its own set of elected attributes as the set of dominant attributes in the set of all its candidate attributes (line 2). If the set of elected attributes has changed, then u advertises this set to all its in-neighbors (line 4-5).

Algorithm 1 Canonical partial-order non-restarting vectoring protocol. When node u receives the advertisement of set B of attributes from out-neighbor v concerning destination t, $u \neq t$.

1: $DomTab_u[t,v] := \{\, a(uv) \oplus b \mid b \in B \,\}$
2: $Dom_u[t] := D(\{\, DomTab_u[t,v], v \text{ out-neighbor of } u \,\})$
3: **if** $Dom_u[t]$ has changed **then**
4:     **for** all r in-neighbor of u **do**
5:         Send $Dom_u[t]$ to r

With multiple elected attributes at each node per destination, data-packets can no longer be forwarded exclusively based on their destinations. Each node assigns a different label to each elected attribute to reach the destination, which is advertised alongside the respective attribute to in-neighbors [15]. Therefore, each node maintains two tables:

- A *path selection* table with entries of the form:

  destination **;** *attribute* **;** *label*,

  where *destination* is a destination, *attribute* is an elected attribute to reach the destination and *label* is the label assigned to *attribute*.
- A *forwarding* table with entries of the form:

  destination **;** *label* **;** *next.hop* **,** *next.label*, where *next-hop* is the out-neighbor from which *attribute* was learned and *next.label* is the label advertised alongside the attribute that originated *attribute*.

Data-packets sourced at the node and meant to travel along a path with attribute *attribute* are first sent to the path selection table and then sent to the forwarding table. In the path selection table, data-packets are marked with label *label*. In the forwarding table, data-packets have label *label* replaced with label *next.label* and are forwarded to out-neighbor *next.hop*. Data-packets arriving at the node from an in-neighbor carrying label *label* are sent directly to the forwarding table.

A node may install multiple entries in the path selection table with the same value of *destination* and *attribute*, but different values of *label*. This action enables data-packets to travel from sources to destinations along multiple dominant paths with a common attribute, a possibility that in standard vectoring protocols is known as ECMP (Equal Cost Multi-Path).

Termination and dominance. The *state* of a partial-order non-restarting vectoring protocol at a given instant corresponds to the candidate and elected attributes at every node, plus the advertisements of attributes in transit across links. This state is *stable* if there are no advertisements in transit. A partial-order non-restarting vectoring protocol *terminates* if, in the absence of changes in the network, it reaches a stable state from any initial state.

Inflation alone does not guarantee termination. In addition, the network must satisfy two algebraic properties. First, the set of all path attributes in the network must be finite, to prevent count-to-infinity. Second, all circuits in the network must be strictly inflationary, to avoid oscillatory behaviors. An attribute a is *strictly inflationary* if $b \prec_i a \oplus b$, for all non-null attributes b, and a circuit in a network is strictly inflationary if its attribute is strictly inflationary.

Whether or not isotonicity holds, in stable state the partial-order non-restarting vectoring protocol routes data-packets via label-switching on paths whose attributes are those elected at the nodes. If isotonicity holds, then these attributes are dominant.

4.2. Restarting protocol

In the canonical partial-order restarting vectoring protocol, a destination t initiates a fresh computation instance by advertising attribute ϵ to all its in-neighbors. Algorithm 2 presents the pseudo-code of the protocol for when node u receives the advertisement of pair (b, n) from out-neighbor v concerning destination t, $u \neq t$. Again, pair (b, n) comprises attribute b and the sequence number n of the instance that produces this attribute. Variable $Dom_u[t]$ stores the set of elected attributes at u to reach t and variable $Seq_u[t]$ stores the sequence number of the computation instance that produces $Dom_u[t]$.

Upon receiving (b, n) from v, node u first calculates the extension of the attribute of its link to v with b, $a(uv) \oplus b$ (line 1). If b is from a more recent computation instance than the current set of elected attributes at u, then this set is replaced with the singleton consisting of $a(uv) \oplus b$ and the current sequence number at u is substituted by n (line 2-4). Otherwise, in case $a(uv) \oplus b$ is from the same computation instance as the current set of elected attributes at u, this set is updated to account for $a(uv) \oplus b$, that is, if $a(uv) \oplus b$ is not less preferred than any attribute of the set, then $a(uv) \oplus b$ is included in the set and all attributes less preferred than it are withdrawn (line 5-6). If there has been a change in the set of elected attributes or in the sequence number, then u send both to all its in-neighbors (lines 7-9).

Algorithm 2 Canonical partial-order restarting vectoring protocol. When node u receives pair (b, n) of attribute b and sequence number n from out-neighbor v concerning a destination t, $u \neq t$.

1: $a := a(uv) \oplus b$
2: **if** $Seq_u[t] < n$ **then**
3: $\quad Dom_u[t] := a$
4: $\quad Seq_u[t] := n$
5: **else if** $Seq_u[t] = n$ **then**
6: $\quad Dom_u[t] := D(\, Dom_u[t] \cup \{a\}\,)$
7: **if** $Dom_u[t]$ has changed or $Seq_u[t]$ has changed **then**
8: $\quad$ **for** all r in-neighbor of u **do**
9: $\quad\quad$ Send $(a, Seq_u[t])$ to r

As in a partial-order non-restarting vectoring protocol, nodes assign distinct labels to elected attributes, with labels and respective attributes being advertised together to in-neighbors. Nodes maintain path selection and forwarding tables with entries of the same meaning and usage as before.

We opt for a version of the protocol with a single entry per destination, whose value of *next.hop* corresponds to the out-neighbor from which the elected attribute was learned for the first time. This option prevents data-packets from traveling along multiple dominant paths, but, on the other hand, it ensures that the algebraic conditions for the termination and dominance are weaker in comparison to partial-order non-restarting protocols.

Termination and dominance. The concept of termination applies here to a single computation instance, which is initiated by a destination when it advertises attribute ϵ to all its in-neighbors. This concept is operationally relevant when the period with which the destination initiates fresh computation instances is large compared to the time it takes for them to terminate.

Contrary to a partial-order non-restarting protocol, in a partial-order restarting protocol, inflation by itself implies the termination of the protocol to a stable state. Strict inflation of circuits and finiteness of path attributes is not necessary for termination and dominance; on the other hand, if isotonicity does not hold, then the stable state does not deliver data-packets from sources to destinations, in general. If isotonicity holds, data-packets are delivered from sources to destinations and travel along dominant paths.

5. Evaluation

Our solution to routing on multiple optimality criteria required the design of partial-order vectoring protocols that route on dominant paths for partial orders that satisfy isotonicity. We evaluate the practicality and efficiency of these protocols. We built a simulator of partial-order vectoring protocols to answer two questions:

(a) What are the sizes of sets of dominant attributes from sources to destinations?
(b) How do partial-order vectoring protocols behave during periods of convergence following a network event?

We based our evaluation on publicly available topologies and attributes. Section 5.1 presents the test networks and the simulator used in this evaluation. Section 5.2 studies the number of dominant attributes from sources to destinations. Section 5.3 investigates the transient behaviour of partial-order non-restarting vectoring protocols, respectively.

5.1. Networks and simulator

The networks used for testing are based on the topologies of ISPs (ISP) inferred by the Rocketfuel project [22]. These topologies have each link annotated with an OSPF weight and with the propagation delay across the link. The test networks consist of the largest 2-edge-connected component of each topology. We assign a width and a length to each link of a test network. The width of a link is set to the inverse of the OSPF weight. The length of a link is set to the propagation delay across the link. Table 1 characterizes the networks by their number of nodes and links.

We constructed a simulator of partial-order vectoring protocols. The source code of the simulator and the accompanying documentation are publicly available on GitHub [1]. The simulator supports four instantiations of attributes: pairs hops-length, width-hops and width-length, and triples width-hops-length. Widths extend with minimum: the width of a path is the minimum of the widths of its links. Hops and lengths extend with addition: the hops of a path in the number of links of the path and the length of a path is the sum of the lengths of its links. The product orders on the pairs and on the triple were considered, as well as several total orders on width-lengths. Advertisements traverse links in first in, first out order and are subject to a random delay taken from a uniform distribution. We set the range of this distribution from 0.01 to 1 ms. With the purpose of avoiding count-to-infinity, advertisements of the non-restarting protocol that travel more than a prespecified maximum number of hops are invalidated. As in RIP, we set that maximum number to 15.

5.2. Sets of dominant attributes

The amount of routing states maintained by partial-order vectoring protocols is proportional to the size of the sets of elected attributes at nodes to reach destinations. With isotonicity, the sets of elected attributes in stable state are precisely the sets of dominant attributes from nodes to destinations. Partial-order vectoring protocols are feasible to the extend that these sets are small.

Table 1 shows the average number of dominant width-lengths over all source-destination pairs for all test networks. The average number of dominant width-lengths ranges from 1.9, in AS 1221, to 3.7, in AS 3967. The number of dominant width-lengths is upper bounded by the number of distinct path widths, which equals the number of distinct link widths, because the width of a path is the width of one of its links. The table also shows the number of distinct link widths for all test networks. The average number of dominant width-lengths is considerably lower than the number of distinct link widths. For example, in AS 1239, the average number of dominant width-lengths is 2.5, while there are 19 distinct link widths.

TABLE 1: ASN, number of nodes, number of links, number of distinct link widths and average number of dominant width-lengths for all test networks.

ASN	Nodes	Links	Distinct Widths	Dominant Width-Lengths
1221	50	194	8	1.9
1239	284	1882	19	2.5
1755	73	292	18	2.2
3257	113	558	21	3.5
3967	72	180	19	3.7
6461	129	726	19	2.8

Figure 3 shows the Complementary Cumulative Distribution Function (CCDF) of the number of dominant attributes in the largest test network, AS 1239. The product order on the different instantiations of attributes is presented: (a) hops-lengths, in black; (b) width-hops, in red; (c) width-lengths, in green; and (d) width-hops-lengths, in blue. The average number of dominant hops-lengths is 1.1 and the maximum number of dominant hops-lengths is three. We deduce that path hops and path lengths are highly correlated, that is to say, a path with the minimum number of links from source to destination is very often a shortest path. The average number of dominant attributes for width-hops, width-lengths and width-hops-lengths is 2.2, 2.5 and 2.9, respectively. The percentages of source-destination pairs for which the number of dominant attributes is greater than three are 11.6%, 21.2% and 31.9% respectively for width-hops, width-lengths and width-hops-lengths. As opposed to path hops and path lengths, we infer that path widths and path hops are many times at odds with one another: a widest path from source to destination is usually not a path with the minimum number of links. Furthermore, there is more diversity in path lengths than in path hops, which justifies that there are more dominant width-lengths than dominant

[1] See https://github.com/miferrei/rmoc-sigcomm2020-artifact.

width-hops than dominant hops-lengths. Finally, it is easy to show that if a triple width-hops-length is dominant, then the pairs hops-length, width-hops, and width-length are dominant as well. Thus, sets of dominant width-hops-lengths are necessarily larger than the other sets of dominant attributes.

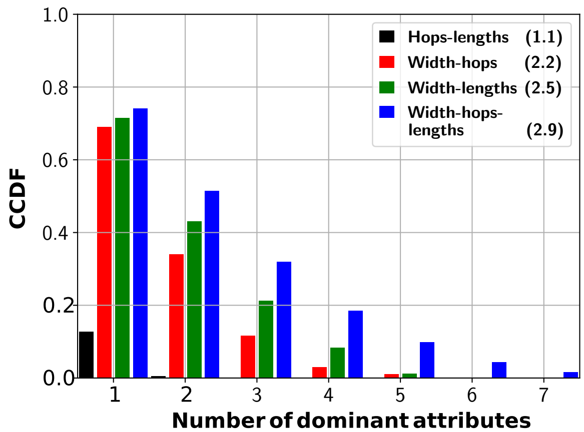


Figure 3: CCDF of the number of dominant attributes in AS 1239 for the product order on hops-lengths, width-hops, width-lengths and width-hops-length. Averages are given inside parenthesis.

5.3. Transient behavior of non-restarting protocols

Network-wide announcement of a destination. Figure 4a shows the CCDF of the termination times in AS 1239 following the network-wide announcement of a destination, over all possible destination nodes and all 25 independent trials. The product order on the different instantiations of attributes is presented. The curves of termination times are rather smooth and steep. The average termination times are 6.2 ms, 7.7 ms, 8.3 ms and 8.4 ms for length-hops, width-hops, width-lengths and width-hops-lengths, respectively.

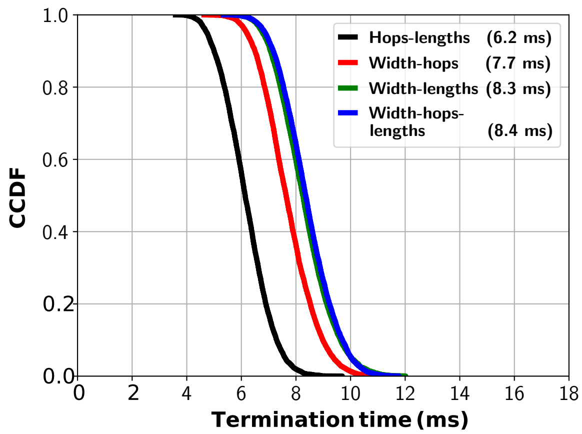
The product order on all four instantiations of attributes satisfies isotonicity. In a partial-order non-restarting vectoring protocol, nodes continually elect and advertise to its neighbors the best from among the attributes learned from its neighbors and isotonicity entails that an elected attribute can only be replaced by a better attribute during a trial. Therefore, the termination time is equal to the maximum delay to propagate an advertisement all the way up a dominant path and is roughly proportional to the number of links in a dominant path with the largest such number. As discussed, path widths and path hops are often not correlated, which means that a widest path from source to destination typically traverses more than the minimum number of links required to reach the destination from the source. This fact explains why attributes involving width lead to longer termination times.

Failure of a link. Figure 4b shows the CCDF of the termination times in AS 1239 following the failure of a link, over all possible failures and all 25 independent trials. The product order on the different instantiations of attributes is presented. The curves of termination time have a sharp drop before 1 ms. In particular, 45.3%, 37.4%, 36.7%, and 36.6% of the failures have termination times less than 1 ms, for hops-lengths, width-hops, width-lengths and width-hops-lengths, respectively. The nodes of AS 1239 are clustered around geographical areas, with nodes inside each cluster densely connected with links of unit length. Therefore, the failure of a link within the cluster has only a localized impact in the state of the protocol.

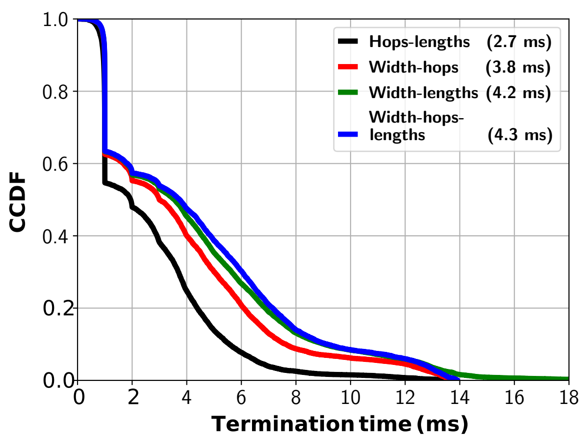
The curves of termination time also have a heavy tail, which translates that some links failures are associated with long termination times. Concretely, 45.3%, 37.5%, 36.7% and 36.6% of the failures lead to termination times in excess of 10 ms, for width-hops, width-lengths and width-hops-lengths, respectively. The failure of a link may require that some nodes have to settle on worse attributes than those they started out with. Trying to stabilize on worse attributes by always electing the best attributes from among those learned from its neighbors takes many iterations, corresponding to as many paths being explored and, hence, to long termination times.

Comparison against various optimality criteria. Figure 5 shows the CCDF of the termination times in AS 1239 following the network-wide announcement of a destination, over all possible destination nodes and all 25 independent trials. Three orders on width-lengths are presented: (a) the widest-shortest order, in lime; (b) the shortest-widest order, in purple; and (c) the product order, in green. The curves of termination time of the standard vectoring protocol for shortest-widest paths are heavy tailed, with 37.8% of the announcements leading to termination times in excess of 10 ms. The average termination time of the partial-order vectoring protocol is lower than that of the total-order vectoring protocol for shortest-widest paths. The respective averages are 8.3 ms and 9.6 ms. The worse termination time of the standard vectoring protocol is justified by the absence of isotonicity in the shortest-widest order. Without isotonicity, the improvement of an elected attribute at a node may entail the worsen of an elected attribute at an in-neighbor of its. As mentioned before in the case of a link failure, the process whereby a non-restarting vectoring protocol has some nodes settle on worse attributes than those they started out with may take a long time. We emphasize that the standard non-restarting vectoring protocol for shortest-widest paths not only takes longer to reach a stable state, as the stable state does not route on shortest-widest paths, in general. Specifically, 30.3% of the source-destination pairs fail to route data-packets on a shortest-widest path from source to destination.

The curves of termination times of standard vectoring protocol for widest-shortest paths and of the partial-order vectoring protocol are both smooth and steep. This is due to the fact that both these orders satisfy isotonicity. The partial-order vectoring protocol has worse termination time than the standard vectoring protocol for widest-shortest paths, the respective averages being 6.3 ms and 8.3 ms. With



(a) Network-wide announcement of a destination



(b) Failure of a link

Figure 4: CCDF of the termination times in AS1239 for a partial-order non-restarting vectoring protocol operating on the product orders on hops-lengths, width-hops, width-lengths and width-hops-length. Averages of the distributions are given inside parenthesis.

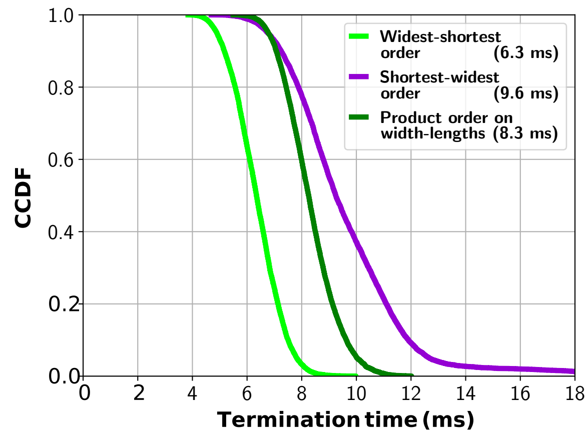


Figure 5: CCDF of the termination times in AS 1239 following the network-wide announcement of a destination for a partial-order vectoring protocol operating on the product order on width-lengths and a standard vectoring protocol operating on the shortest-widest and the widest-shortest order.

the partial-order vectoring protocol, nodes have to settle on an average of 2.5 width-lengths per destination, such that 21.2% of the nodes electing more than three width-lengths per destination, whereas, with the standard vectoring protocol, they have to settle on a single width-length. However, the termination times of the partial-order vectoring protocol are only slightly worse, since the computation of the multiple attributes in a set of dominant attributes is done simultaneously during the execution of the protocol.

6. Previous Work

Algebraic conceptualization of routing. The algebraic framework proposed in References [2], [3] provides means to reason about routing problems and protocols with generality, by abstracting away the specificity of performance metrics and protocol parameters. The framework is predicated on a set of attributes, a binary extension operation on attributes, and a total order on attributes.

The present work generalizes the framework by accepting partial orders on attributes and proposing routing protocols that compute on them. Furthermore, it presents a procedure that reduces a collection of total orders on attributes to a common partial order that respects all of them and satisfies isotonicity.

Multi-objective path problem. Multi-objective path problems are well-known in the operations research community. These problems can be formulated within the same algebraic framework considered in this work. Attributes are tuples of elementary metrics, each of which either extends with addition and is totally ordered by the less-than-or-equal order or extends with minimum and is totally ordered by the greater-than-or-equal order. Tuples extend coordinate-wise and are partially ordered by the product order of their coordinate-wise total orders. Because each coordinate extends with addition or minimum and is, therefore, individually isotone, the extension of tuples is isotone for the product order. The goal of multi-objective path problems is to find sets of dominant attributes [23]–[26].

In most cases, multi-objective path problems are solved by extensions of Dijkstra's algorithm [23], [24], [27] and Bellman-Ford algorithm [25]–[28] that compute on sets of incomparable attributes, rather than on a single attribute. If attributes are tuples of several additive components, then the number of dominant attributes from source to destination can be exponential in the number of nodes in the network [23]. Approximate algorithms have been devised to deal with the explosion in the number of dominant attributes [29].

The setting considered in work is more general and the problem addressed is different. Attributes are assumed arbitrary, not necessarily tuples of minimal or additive components. Even when they are, a partial order on them is derived, rather than assumed a priori, and does not necessarily coincide with the product order. The sets of dominant attributes are to be found by a routing protocol, that is, a distributed algorithm, rather than a sequential one, that can expedite data-packets along the paths computed.

Multi-path routing protocols. Multi-path routing protocols select multiple paths from a source to destination. They have been mostly suggested as extensions to BGP [8], the prevailing standard vectoring protocol for routing on the Internet, with one of the following goals in mind. A first goal is to ensure the termination of BGP [30]–[34]. A second goal is to improve the capability of BGP to deliver data-packets during periods of convergence of the protocol following link failures [35]–[38]. And a third goal is to increase the path diversity available [32], [39], [40].

Since partial-order vectoring protocols possibly find multiple paths from a source to a destination, they are a type of multipath routing protocol. However, these protocols target a different goal. We seek to route data-packets on optimal paths for a variety of optimality criteria, some of which do not lend themselves to a solution by a standard vectoring protocol. Besides, partial-order vectoring protocols are formulated with generality rather than being specific to BGP.

7. Conclusion

We presented solutions to two routing problems: (a) optimal path routing problem for an arbitrary optimality criterion; and (b) optimal path routing problem for multiple optimality criteria. These problems can be formulated within the algebraic framework for routing. The first problem is well-known, but it did not have a solution. The second problem is a new one and it is a superset of the first problem.

In the process of developing solutions to these problems, we introduced two new routing concepts: (1) the intersection of a collection of total orders; and (2) the isotonic reduction of a partial order (total order). Equipped with these concepts, we presented a procedure that starts with a collection of total orders and ends with a partial order that is contained in each of the total orders and satisfies isotonicity. Thus, the original optimal path problems are reduced to that computing sets of dominant attributes for an isotone partial order. We devised new partial-order vectoring protocols that find those sets of dominant attributes. The assignment of unique labels to elected attributes enables the expedition of data-packets along any dominant paths. While our running examples emphasized widths and lengths, the solution was developed for arbitrary performance metrics that satisfy the algebraic properties of associativity, commutativity, and inflation.

A preliminary evaluation shows that this approach is promising. The sets of dominant attributes computed on the Rocketfuel networks are rather small. The average number of dominant attributes is below four for networks with hundreds of nodes and thousands of links. Furthermore, our simulations on the Rocketfuel networks revealed that: (1) the computation of the multiple attributes in a set of dominant attributes is done simultaneously; and that (2) isotonicity promotes fast computations. The convergence time of a partial-order vectoring protocol operating on an isotone partial order is only marginally worse than that of a standard vectoring protocol operating on an isotone total order and sometimes much better than that of a standard vectoring protocol operating on a non-isotone total order.

References

[1] B. Carré, "An algebra for network routing problems," *IMA Journal of Applied Mathematics*, vol. 7, no. 3, pp. 273–294, 1971.

[2] J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the internet," in *Proc. of the 16th IEEE International Conference on Computer Communications (INFOCOM 2001)*, vol. 2, 2001, pp. 727–735.

[3] J. L. Sobrinho, "An algebraic theory of dynamic network routing," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1160–1173, 2005.

[4] T. Griffin and J. L. Sobrinho, "Metarouting," in *Proc. of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM 2005)*, 2005, pp. 1–12.

[5] M. Gondran and M. Minoux, *Graphs, Dioids and Semirings*, 1st ed. Springer, 2008.

[6] G. Malkin, "RIP version 2," RFC 2453, 1998.

[7] D. Savage, D. Slice, R. White, J. Ng, P. Paluch, and S. Moore, "Cisco's enhanced interior gateway routing protocol (EIGRP)," RFC 7868, 2016.

[8] Y. Rehkter, T. Li, and S. Hares, "A border gateway protocol 4," RFC 4271, 2002.

[9] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[10] J. Chroboczek, "The babel routing protocol," RFC 6126, 2011.

[11] M. Gouda and M. Schneider, "Maximizable routing metrics," in *Proc. of the 6th International Conference on Network Protocols (ICNP 1998)*. IEEE, 1998, pp. 71–68.

[12] J. L. Sobrinho, "Fundamental differences among vectoring routing protocols on non-isotonic metrics," *IEEE Networking Letters*, vol. 1, no. 3, pp. 95–98, 2019.

[13] T. Lengauer and D. Theune, "Efficient algorithms for path problems with general cost criteria," in *Proc. of the 18th International Colloquium on Automata, Languages, and Programming (ICALP 1991)*, 1991, pp. 314 – 326.

[14] E. Harzheim, "Ordered sets," *SIAM Review*, vol. 48, pp. 160–163, 2006.

[15] G. Chandranmenon and G. Varghese, "Trading packet headers for packet processing," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 141–152, 1996.

[16] R. Carter and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance Evaluation*, vol. 27, pp. 297–318, 2001.

[17] A. Khanna and J. Zinky, "The revised (arpanet) routing metric," *ACM SIGCOMM Computer Communication Review*, vol. 19, no. 4, pp. 45–56, 1989.

[18] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom 2003)*, 2003, pp. 134–146.

[19] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.

[20] T. Griffin, "The stratified shortest-paths problem," in *Proc. of the 2nd International Conference on Communication Systems and Networks (COMSNETS 2010)*, 2010, pp. 1–10.

[21] J. L. Sobrinho and M. A. Ferreira, "Routing on multiple optimality criteria," in *Proc. of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM 2020)*, 2020, pp. 211–225.

[22] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 133–145, 2002.

[23] P. Hansen, "Bicriterion path problems," in *Multiple Criteria Decision Making: Theory and Application*, 1980, pp. 109–127.

[24] E. Martins, "On a multicriteria shortest path problem," *European Journal of Operational Research*, vol. 16, no. 2, pp. 236–245, 1984.

[25] J. Brumbaugh-Smith and D. Shier, "An empirical investigation of some bicriterion shortest path algorithms," *European Journal of Operational Research*, vol. 43, no. 2, pp. 216–224, 1989.

[26] F. Guerriero and R. Musmanno, "Label correcting methods to solve multicriteria shortest path problems," *Journal of Optimization Theory and Applications*, vol. 111, no. 3, pp. 589–613, 2001.

[27] R. Loui, "Optimal paths in graphs with stochastic or multidimensional weights," *Communications of the ACM*, vol. 26, no. 9, pp. 670–676, 1983.

[28] H. Daellenbach and C. De Kluyver, "Note on multiple objective dynamic programming," *Journal of the Operational Research Society*, vol. 31, no. 7, pp. 591–594, 1980.

[29] Z. Tarapata, "Selected multicriteria shortest path problems: An analysis of complexity, models and adaptation of standard algorithms," *International Journal of Applied Mathematics and Computer Science*, vol. 17, no. 2, pp. 269–287, 2007.

[30] A. Basu, C. Ong, A. Rasala, F. Shepherd, and G. Wilfong, "Route oscillations in I-BGP with route reflection," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 235–247, 2002.

[31] A. Flavel and M. Roughan, "Stable and flexible iBGP," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 183–194, 2009.

[32] Y. Wang, M. Schapira, and J. Rexford, "Neighbor-specific BGP: More flexible routing policies while improving global stability," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, pp. 217–228, 2009.

[33] R. Agarwal, V. Jalaparti, M. Caesar, and P. Godfrey, "Guaranteeing BGP stability with a few extra paths," in *Proc. of the 30th International Conference on Distributed Computing Systems (ICDCS 2010)*. IEEE, 2010, pp. 221–230.

[34] J. Scudder, A. Retana, D. Walton, and E. Chen, "Advertisement of multiple paths in BGP," RFC 7911, 2016.

[35] N. Kushman, S. Kandula, D. Katabi, and B. Maggs, "R-BGP: Staying connected in a connected world," in *Proc. of the 4th Symposium on Networked Systems Design and Implementation (NSDI 2007)*, 2007.

[36] F. Wang and L. Gao, "A backup route aware routing protocol - fast recovery from transient routing failures," in *Proc. of the 27th IEEE International Conference on Computer Communications (INFOCOM 2008)*, 2008, pp. 2333–2341.

[37] Y. Liao, L. Gao, R. Guérin, and Z. Zhang, "Reliable interdomain routing through multiple complementary routing processes," in *Proc. of the 4th International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2008)*, 2008, pp. 235–247.

[38] I. Ganichev, B. Dai, P. Godfrey, and S. Shenker, "YAMR: Yet another multipath routing protocol," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 5, pp. 13–19, 2010.

[39] W. Xu and J. Rexford, "MIRO: Multi-path interdomain routing," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 171–182, 2006.

[40] V. Van den Schrieck, P. Francois, and O. Bonaventure, "BGP add-paths: The scaling/performance tradeoffs," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1299–1307, 2010.