On Non-Commutative Routing

Zhaozhen Wang*, Xingang Shi†‡, Haijun Geng§, Zitong Jin‡, Han Zhang†‡, Xia Yin*‡, Zhiliang Wang†‡
*Department of Computer Science and Technology, Tsinghua University, Beijing, China
†Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China
‡Zhongguancun Laboratory, Beijing, China
§School of Automation and Software Engineering, Shanxi University, Taiyuan, China

IEEE INFOCOM 2025 - IEEE Conference on Computer Communications | 979-8-3315-4305-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOM55648.2025.11044753

***Abstract*—The complexity of routing requirements leads to increasingly intricate routing metrics. Existing routing algebra theories have demonstrated that convergent and optimal routing algorithms can be designed only when path metrics satisfy certain properties such as monotonicity and isotonicity. Furthermore, some non-isotonic metrics can be converted into isotonic forms on partial orders through reduction. However, practical scenarios often involve non-commutative algebraic properties, which are overlooked by existing theories. For these problems, there lacks a unified framework to study their solvability, a systematic method for their reduction, and an efficient algorithm to compute optimal routes. In this work, we extend routing algebra to accommodate non-commutative routing problems, propose general reduction methods for them, and explore their solvability. In addition, we design a link state algorithm that converge fast on a reduced partial order. All these discussions are supported by concrete examples, theoretical proofs, and simulations on various network topologies.**

***Index Terms*—Non-commutativity, routing algebra, routing algorithm, partial order.**

I. Introduction

With the continuous evolution of networks and applications, the requirements for routing have become increasingly diverse. Modern routing schemes now take various application-specific requirements into consideration to improve quality of service (QoS), moving beyond the traditional objective of selecting the shortest path. This has led to more complex designs for routing path metrics and routing algorithms. For example, routing for autonomous driving and online gaming prioritizes packet loss rates and network latency, while routing for wireless networks and the Internet of Things (IoT) often focuses on low energy consumption to ensure network stability [1]–[3].

Routing algebra can abstract path metrics and path selection criteria to address these diverse requirements. It formulates specific routing problems with general algebraic elements, allowing for a unified study of key issues such as algorithm convergence and optimality. Existing routing algebra theories have shown that convergent and optimal algorithms can be designed only when path metrics satisfy properties such as monotonicity and isotonicity [4], [5]. Another critical finding is that when metrics lack isotonicity, some can be converted into isotonic forms on partial orders via reduction [6], [7].

Existing theories have mainly focused on commutative metrics, where the path weight is unaffected by the order in which the constituent links are concatenated. However, path metrics can become non-commutative in scenarios where link order matters. For instance, in reliable communication, TCP end-to-end retransmission is required when packet loss occurs at an intermediate node but is always initiated by the source, leading to unbalanced times of packet traversal on each link. In networks running various upper or lower-layer protocols, packets need to be translated when crossing region boundaries. If translation costs between two protocols are asymmetric, path metrics may become non-commutative. Currently, such problems still lack a unified framework to study their solvability, a comprehensive method for proper reduction, and an efficient algorithm to compute optimal routes.

Our contribution can be summarized as follows.

- We extend routing algebra to formulate non-commutative routing problems and provide practical examples. We propose general methods to reduce non-isotonic metrics to isotonic forms on partial orders for these problems.
- Based on our reduction methods, we examine the solvability of non-commutative problems. In particular, we prove sufficient conditions for a non-commutative problem to be solvable or unsolvable, and whether different partial order routing algorithms can be applied to compute optimal routes.
- We design a general link state routing algorithm on partial orders for reduced problems. We provide rigorous proof of its correctness, including convergence, loop-freeness, and dominance, and conduct simulations on various network topologies. The results demonstrate that our algorithm is efficient and capable of quickly recomputing routes after link failures.

II. Related Work

A. Multi-Objective Routing

Multi-objective routing optimizes multiple metrics based on their product order to balance different objectives. Numerous routing algorithms have been proposed, including dynamic programming [8], branch and bound [9], adaptions of Dijkstra's and Bellman-Ford's algorithms [10], [11], and approximation algorithms [12]–[17]. There are also centralized solutions utilizing software-defined networking (SDN) to collect global network information and compute paths [18]–[22]. However, multi-objective routing on product orders is only a simple case of routing on partial orders, which is our focus in this paper.

B. Routing Algebras

Previous research has introduced various routing algebras for modeling specific routing problems such as shortest or

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Routing algebras for the shortest path (MINSUM) and widest path (MAXMIN) problems. $S = \mathbb{R}_0^+ \cup \{+\infty\}$.

Problem	$w(\varepsilon)$	$w(\varphi)$	$\preceq$	Path Weight Function
MINSUM	0	$\{+\infty\}$	$\leq$	$w(a \oplus b) = w(a) + w(b)$
MAXMIN	$+\infty$	$\{0\}$	$\geq$	$w(a \oplus b) = \min\{w(a), w(b)\}$

longest paths [23] and BGP routing [24]–[26], and for studying algorithm convergence and optimality [4]–[7], [27], [28]. However, most of them assume commutativity [7], [23]–[27]. A few works [4], [5] discussed the solvability of routing problems which could be non-commutative, but they were limited to total orders and did not address non-isotonic problems. Some studies [6], [28] explored how to reduce non-isotonic algebras to left isotonic forms on partial orders, and extended path vector algorithms to solve such problems. However, this approach is not applicable to other routing algorithms (e.g. link state) because the reduction does not address the impact on monotonicity in non-commutative algebras, which is critical for algorithm correctness. A comprehensive theory for non-commutative algebras on general partial orders is still lacking.

III. Algebraic Framework

A network is modeled as a finite directed graph $G = (V, E)$, where V represents the set of nodes and E is the set of directed links in the network. A path in the network is a sequence of links, denoted as $u_0 u_1 \cdots u_n$, where $u_i \in V$ are nodes on the path and $(u_i, u_{i+1}) \in E$ are links.

The routing algebra used in this paper is defined as a five-tuple $(S, \preceq, L, \oplus, w)$. It contains a set of paths L, and a set of weights S, which represent metrics such as bandwidth, latency, or hop count. The relation $\preceq$ is a partial order on S, indicating the preference between paths, where $w(a) \prec w(b)$ $(a, b \in L)$ means that path a is preferred over path b. The extension operation $\oplus$ is a function $L \times L \to L$ concatenating two paths, and w is a function $L \to S$ calculating weights of paths.

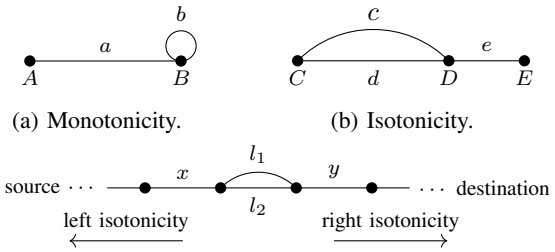
Let $L(s,t)$ denote the set of paths from source s to destination t, and l_{st} denote some path in $L(s,t)$. In routing problems, a path is selected from $L(s,t)$ based on the preference criterion $\preceq$. If $\preceq$ is a total order, we seek the optimal path with the least weight in the poset $(w(L(s,t)), \preceq)$. If $\preceq$ is a partial order, we look for the dominant path, whose weight is a minimal element in $(w(L(s,t)), \preceq)$. The dominant path is typically not unique.

The trivial path ε has no links and satisfies $l \oplus \varepsilon = \varepsilon \oplus l = l$ for all $l \in L$. Its weight, $w(\varepsilon)$, is the least (most preferred) weight of $w(L)$. The invalid path, denoted by φ, represents all the paths that cannot transmit packets. It satisfies $l \oplus \varphi = \varphi \oplus l = \varphi$ for all $l \in L$. The weight set $w(\varphi)$ is the maximal (least preferred) weights of $w(L)$.

Table I illustrates examples of routing algebras for the shortest path problem and the widest path problem.

Monotonicity and isotonicity are fundamental concepts in routing algebras [4]–[7], [27].

Definition 1. A path $l \in L$ is monotonic if it is both left monotonic and right monotonic.



(a) Monotonicity. (b) Isotonicity.

(c) Commutativity. $w(l_1) \preceq w(l_2)$.

Fig. 1: Example of algebraic properties.

TABLE II: Algebraic properties required for routing algorithms to satisfy convergence, loop-freeness, and optimality. S denotes source routing, and H denotes hop-by-hop routing. ✓* means that a strict property is needed.

Algorithm	Path Vector [4]		Flooding [5]		Link State [5]	
Routing Scheme	S	H	S	H	S	H
L. M.		✓				
R. M.					✓	✓
L. I.	✓	✓		✓*		✓*
R. I.			✓	✓	✓	✓

Left monotonic: $\forall a \in L,\ w(a) \preceq w(l \oplus a)$.
Right monotonic: $\forall a \in L,\ w(a) \preceq w(a \oplus l)$.

Definition 2. An algebra $(S, \preceq, L, \oplus, w)$ is monotonic if it is both left monotonic and right monotonic.

Left monotonic (L. M.): $\forall a, b \in L,\ w(a) \preceq w(b \oplus a)$.
Right monotonic (R. M.): $\forall a, b \in L,\ w(a) \preceq w(a \oplus b)$.

Definition 3. An algebra $(S, \preceq, L, \oplus, w)$ is isotonic if it is both left isotonic and right isotonic.

Left isotonic (L. I.): $\forall a, b \in L,\ w(a) \preceq w(b) \iff \forall c \in L,\ w(c \oplus a) \preceq w(c \oplus b)$.

Right isotonic (R. I.): $\forall a, b \in L,\ w(a) \preceq w(b) \iff \forall c \in L,\ w(a \oplus c) \preceq w(b \oplus c)$.

If the inequality sign is strict, then we obtain the definition of strict monotonicity and strict isotonicity.

Monotonicity is often associated with the convergence and loop-freeness of routing algorithms. Figure 1a provides an example. If b is not right monotonic, and $\forall l \in L(A, B),\ w(a) \succeq w(l) \succeq w(a \oplus b)$, then the optimal path from A to B contains a loop. Moreover, if $w(a) \succ w(a \oplus b) \succ w(a \oplus b \oplus b) \succ \cdots$, the optimal path does not exist, and routing algorithms do not converge. In addition, isotonicity is typically related to the optimality of routing algorithms. In Figure 1b, if isotonicity is violated, such as $w(c) \prec w(d)$ and $w(c \oplus e) \succ w(d \oplus e)$, then the optimal path from C to D is not a subpath of the optimal path from C to E, which may cause algorithms to fail in finding the optimal path. Table II summarizes the properties needed for three common routing algorithms [4], [5].

Commutativity is a simple but often overlooked property.

Definition 4. An algebra $(S, \preceq, L, \oplus, w)$ is commutative if $\forall a, b \in L,\ w(a \oplus b) = w(b \oplus a)$.

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.

A common misunderstanding is that unequal metrics (e.g. bandwidth, latency) on the uplink and downlink between two nodes imply non-commutativity. However, the uplink and downlink should be modeled as two different paths. The metric calculation method remains commutative: $w(a \oplus b) = w(b \oplus a) = \min\{w(a), w(b)\}$.

Figure 1c illustrates the impact of commutativity. If commutativity holds, the distinction between left and right algebraic properties is no longer necessary, and isotonicity, if present, can be extended in both directions. However, if commutativity is violated, it becomes crucial to differentiate between left and right properties. In cases where isotonicity holds on only one side or needs to be reduced from non-isotonic algebras, the solvability of routing problems and the applicability of existing routing algorithms become complicated.

IV. Examples of Non-Commutative Scenarios

In this section, we present two non-commutative examples: routing with end-to-end retransmission in wireless networks, and routing with inter-protocol translation in IoT networks.

A. Routing with End-to-End Retransmission

Multi-hop wireless networks suffer from high packet loss rates and significant transmission energy consumption. Meanwhile, mobile nodes, such as phones and satellites, typically have limited battery capacity and available power. Transport layer protocols such as TCP are commonly used to ensure end-to-end reliable communication through acknowledgments and retransmissions, which add to the energy overhead [29]–[33].

Typically, we can use $e(uv) = c\beta n d_{uv}^{\alpha}$ to denote the energy consumption of link uv, which represents the minimum transmission power required for good quality [30], [32], where c is a constant, β is the signal-to-noise ratio (SNR), n is the strength of ambient noise, d_{uv} is the distance between u and v, and α is the coefficient of attenuation ($2 \le \alpha \le 4$). Let $Er(uv)$ be the error rate of packet delivery, then $N(uv) = \frac{1}{1-Er(uv)}$ represents the expected number of transmissions until a successful delivery. For path p and link l, the path weight $w(p \oplus l)$ measuring energy consumption can be defined as follows.

$$w(p \oplus l) = N(l)(w(p) + e(l)) \tag{1}$$

Based on (1), we consider the metric of Minimizing Energy of One-Way Communication (MEOW1). As shown in Table III, $w_1(u_0 \cdots u_n)$ represents the energy consumption of delivering a packet successfully from u_0 to u_n. While some routing algorithms have been developed for this problem [32], they do not always guarantee optimality and do not address its fundamental nature. We will demonstrate that its isotonicity can only be satisfied on one side. We further define the metric of MEOW2 for the reverse direction and the metric of Minimizing Energy of Round-Trip Communication (MERT) (see Table III), and systematically analyze these problems from the perspective of routing algebra.

Theorem 5. *The algebra of MEOW1 is monotonic and right isotonic, but not left isotonic.*

TABLE III: Routing algebras for measuring energy consumption in reliable communication. $S = \mathbb{R}_0^+ \cup \{+\infty\}$, $w(\varepsilon) = 0$, $w(\varphi) = +\infty$, $\preceq = \le$, $N_i = N(u_i u_{i+1})$, and $e_i = e(u_i u_{i+1})$.

	Path Weight Function
MEOW1	$w_1(u_0 \cdots u_n) = N_{n-1}(w_1(u_0 \cdots u_{n-1}) + e_{n-1})$
MEOW2	$w_2(u_0 \cdots u_n) = N_0(e_0 + w_2(u_1 \cdots u_n))$
MERT	$w_3(u_0 \cdots u_n) = w_1(u_0 \cdots u_n) + w_2(u_0 \cdots u_n)$

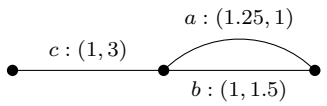


Fig. 2: Counterexample of MEOW1 not satisfying left isotonicity, where $1.25 = w_1(a) \preceq w_1(b) = 1.5$, but $5 = w_1(c \oplus a) \not\preceq w_1(c \oplus b) = 4.5$. Each path is labeled with (N, e).

Proof: Let $l = u_0 u_1 \cdots u_n$ be a path, then the expected number of transmissions until a successful delivery through l is denoted as $N(l) = \prod_{j=0}^{n-1} N_j = \prod_{j=0}^{n-1} \frac{1}{1-Er(u_j u_{j+1})}$. $\forall k = 0, 1, \cdots, n$, we have

$$\begin{aligned} &w_1(u_0 u_1 \cdots u_n) = N_{n-1}(w_1(u_0 u_1 \cdots u_{n-1}) + e_{n-1}) \\ &= \sum_{i=0}^{n-1} e_i \prod_{j=i}^{n-1} N_j = \sum_{i=k}^{n-1} e_i \prod_{j=i}^{n-1} N_j + \sum_{i=0}^{k-1} e_i \prod_{j=i}^{k-1} N_j \prod_{j=k}^{n-1} N_j \\ &= w_1(u_k u_{k+1} \cdots u_n) + w_1(u_0 u_1 \cdots u_k) N(u_k u_{k+1} \cdots u_n) \end{aligned}$$

We obtain that

$$\forall a, b \in L,\ w_1(a \oplus b) = w_1(b) + w_1(a) N(b) \tag{2}$$

Therefore, $w_1(b) \preceq w_1(a \oplus b)$, i.e., left monotonicity holds for MEOW1. Since $N(b) \ge 1$ we know that $w_1(a) \preceq w_1(a \oplus b)$ and right monotonicity holds.

From (2), we can obtain $w_1(a \oplus c) = w_1(c) + w_1(a)N(c)$ and $w_1(b \oplus c) = w_1(c) + w_1(b)N(c)$. $w_1(a) \preceq w_1(b)$ implies $w_1(a \oplus c) \preceq w_1(b \oplus c)$ and therefore right isotonicity holds for MEOW1. However, left isotonicity does not hold and a counterexample is shown in Figure 2. ■

Theorem 6. *The algebra of MEOW2 is monotonic and left isotonic, but not right isotonic.*

Proof: We can similarly obtain

$$\forall a, b \in L,\ w_2(a \oplus b) = w_2(a) + w_2(b) N(a) \tag{3}$$

The proof of Theorem 6 is similar to that of Theorem 5. ■

Theorem 7. *The algebra of MERT is monotonic, but not left or right isotonic.*

Proof: Since MEOW1 and MEOW2 are monotonic, it is obvious that MERT is also monotonic. We temporarily skip the proof of left (right) isotonicity not holding. The remaining proof is given in Section VI. ■

Theorem 8. *The algebras of MEOW1, MEOW2, and MERT are non-commutative.*

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.

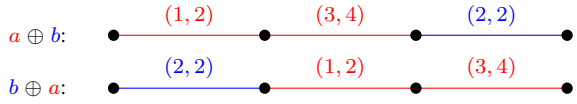


Fig. 3: Counterexample of MERT not satisfying commutativity, where $w_3(a \oplus b) = 40 + 26 = 66$, but $w_3(b \oplus a) = 30 + 32 = 62$. Path a consists of two links and path b consists of one link. Each link is labeled with (N, e).

Proof: MEOW1 and MEOW2 are not commutative, otherwise left isotonicity is equivalent to right isotonicity and this contradicts Theorems 5 and 6. MERT is not commutative either, as shown by the example in Figure 3. ■

In end-to-end retransmissions, packet loss triggers retransmissions from the source, leading to unbalanced times of packet traversal across each link. This lack of commutativity necessitates distinguishing between left and right isotonicity, impacting the applicability of routing algorithms. Due to the unidirectional isotonicity of MEOW1, Table II shows that flooding and link state algorithms with source routing are applicable to MEOW1, while the absence of left isotonicity renders path vector algorithms ineffective. Additionally, none of the three algorithms in Table II work for MERT. Figure 4 demonstrates the execution of the flooding algorithm. While the optimal weight from source s to destination t is $w(suvt) = 18$, the flooding algorithm either fails to set the correct forwarding path (Figure 4a) or cannot find the optimal path (Figure 4b). Similarly, Figure 5 shows that the link state algorithm also fails to identify the optimal path.

B. Routing with Inter-Protocol Translation

Numerous protocols and APIs have been developed to support diverse IoT applications. However, these independently created protocols often lack interoperability. When data need to cross regions with different protocols, translation through middleware or protocol proxies becomes necessary, resulting in additional resource consumption and delays [34]–[38].

The translation overhead is typically asymmetric. For example, SLP [39] and UPnP [40] are application layer protocols facilitating the discovery and interaction of services and devices in IoT networks. Translating from UPnP to SLP incurs higher overhead than translating in the reverse direction, as UPnP is a more general protocol and requires adaptation to the more specific SLP. The average translation time from UPnP to SLP using the Starlink middleware is 6208 ms, while the reverse translation takes only 319 ms [35]. Note that routing with asymmetric protocol translation costs is a widespread issue in various types of networks and not limited to IoT networks.

We consider the routing algebra of Minimizing Delay (MD). For two paths $a, b \in L$, let $t(a, b)$ be the translation time from the protocol used by the last link of a to the protocol used by the first link of b. The path weight w is defined as

$$w(a \oplus b) = w(a) + w(b) + t(a, b)$$

Since $w \geq 0$, $c \geq 0$, and $t(a, b) \not\equiv t(b, a)$, we derive the following theorem.

Theorem 9. *The algebra of MD is monotonic, neither left nor right isotonic, and non-commutative.*

According to Table II, none of the three routing algorithms work for MD.

V. General Reduction Methods

To address the non-commutative issues in the previous section, we propose general reduction methods to transform non-isotonic routing metrics into isotonic forms on partial orders. We then examine the solvability of non-commutative routing problems that lack monotonicity or isotonicity.

A. Left and Right Isotonic Reduction

The concept of isotonic reduction was first introduced in [6], [7] to address the use of path vector algorithms for commutative but non-isotonic routing algebras. For general routing algebras, we propose the concept of left and right isotonic reduction and explore their properties.

Definition 10. For an algebra $(S, \preceq, L, \oplus, w)$, a left isotonic reduction $\preceq_{LR}$ is a subrelation of $\preceq$ that makes $(S, \preceq_{LR}, L, \oplus, w)$ left isotonic.

Definition 11. For an algebra $(S, \preceq, L, \oplus, w)$, a right isotonic reduction $\preceq_{RR}$ is a subrelation of $\preceq$ that makes $(S, \preceq_{RR}, L, \oplus, w)$ right isotonic.

An isotonic reduction yields a partial order, and a routing algebra can have multiple left or right isotonic reductions. The greatest left or right isotonic reduction (GLR or GRR) is the one where all other reductions are its subrelations. GLR and GRR can be calculated through the following theorems. The proofs are analogous to those in commutative algebras [7] and are therefore omitted.

Theorem 12. *The greatest left isotonic reduction $\preceq_{GLR}$ of the partial order $\preceq$ exists:* $\forall a, b \in L,\ w(a) \preceq_{GLR} w(b) \iff \forall x \in L,\ w(x \oplus a) \preceq w(x \oplus b)$.

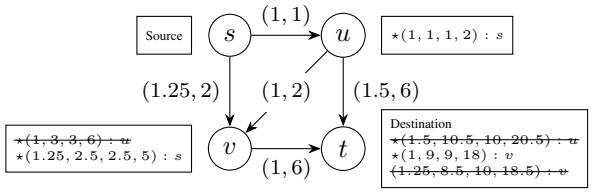
Theorem 13. *The greatest right isotonic reduction $\preceq_{GRR}$ of the partial order $\preceq$ exists:* $\forall a, b \in L,\ w(a) \preceq_{GRR} w(b) \iff \forall x \in L,\ w(a \oplus x) \preceq w(b \oplus x)$.

Isotonic reduction is essentially the selection of a subrelation $\preceq_R$ of the original partial order $\preceq$ to achieve isotonicity. By making more path weights incomparable, isotonicity becomes easier to establish. However, there is no free lunch. While gaining isotonicity, there may be a loss of monotonicity. It can be proved that any reduction does not facilitate establishing monotonicity. This explains why isotonic reductions cannot be performed without restriction and why we prefer the greatest isotonic reduction.

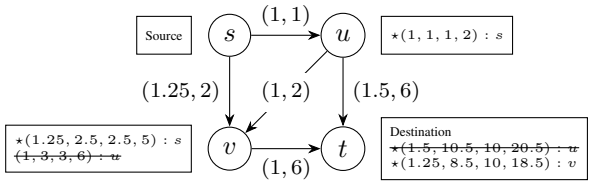
Lemma 14. *If an algebra $(S, \preceq, L, \oplus, w)$ is not left (right) monotonic, then for any arbitrary subrelation of $\preceq$, denoted as $\preceq'$, $(S, \preceq', L, \oplus, w)$ is not left (right) monotonic.*

Proof: If $(S, \preceq', L, \oplus, w)$ is left monotonic, then $\forall a, b \in L$, $w(a) \preceq' w(b \oplus a) \implies w(a) \preceq w(b \oplus a)$. We obtain that $(S, \preceq$

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.



(a) $(1,3,3,6)$ arrives first at v



(b) $(1.25,2.5,2.5,5)$ arrives first at v

Fig. 4: Stable state of the flooding algorithm for MERT with source routing. Each link is labeled with (N,e). Nodes transfer N,w_1,w_2 to calculate a new value for w_3 using (2) (3). Each node is labeled with its calculated quadruples (N,w_1,w_2,w_3), with the elected quadruple marked by stars.

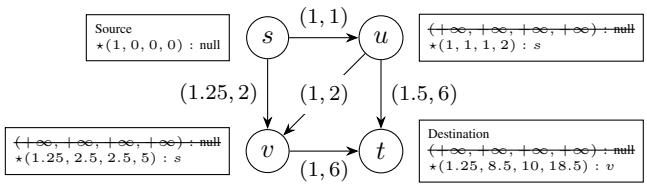


Fig. 5: Stable state of the link state algorithm for MERT with source routing. Each link is labeled with (N,e). Each node is labeled with its calculated quadruples (N,w_1,w_2,w_3), with the elected quadruple marked by stars.

$, L, \oplus, w)$ is left monotonic, which leads to a contradiction. The proof for right monotonic case is similar. ∎

It has been proved that the greatest isotonic reduction preserves monotonicity in commutative algebras [7]. However, this proof depends on commutativity and does not apply to general algebras. Therefore, it is necessary to examine the impact of GLR and GRR on monotonicity in general algebras. We propose the following theorems, with the properties of GLR and GRR summarized in Table IV.

Lemma 15. *(i) Right isotonicity implies left monotonicity, and (ii) left isotonicity implies right monotonicity.*

Proof: We only prove (i). From $\forall b \in L,\ w(\varepsilon) \preceq w(b)$ and right isotonicity, we obtain $\forall a,b \in L,\ w(a) = w(\varepsilon \oplus a) \preceq w(b \oplus a)$, and therefore left monotonicity holds. ∎

Theorem 16. *(i) GLR preserves right monotonicity and right isotonicity. Generally, GLR does not preserve left monotonicity. (ii) For a right isotonic algebra, GLR preserves left monotonicity if and only if the algebra is right monotonic.*

Theorem 17. *(i) GRR preserves left monotonicity and left isotonicity. Generally, GRR does not preserve right monotonicity. (ii) For a left isotonic algebra, GRR preserves right monotonicity if and only if the algebra is left monotonic.*

TABLE IV: Properties of GLR and GRR. O is for preserving the corresponding properties, X means not preserving the properties, and — is a meaningless entry. For example, for a left isotonic algebra, its GLR is identical to itself.

Preserving the property of	L. M.[a]	R. M.[a]	L. I.	R. I.
GLR	X	O	—	O[b]
GRR	O	X	O[c]	—

[a] can be strict.
[b] R. M. before reduction $\iff$ preserving (strict) L. M.
[c] L. M. before reduction $\iff$ preserving (strict) R. M.

Proof: We only prove Theorem 16. (i) From Theorem 12,

$$\begin{aligned}
&(S,\preceq_{GLR},L,\oplus,w) \text{ is right monotonic}\\
\iff &\forall a,b\in L,\ w(a)\preceq_{GLR} w(a\oplus b)\\
\iff &\forall a,b,x\in L,\ w(x\oplus a)\preceq w(x\oplus(a\oplus b))\\
\iff &\forall a,b,x\in L,\ w(x\oplus a)\preceq w((x\oplus a)\oplus b)\\
\iff &(S,\preceq,L,\oplus,w) \text{ is right monotonic}
\end{aligned}$$

If $(S,\preceq,L,\oplus,w)$ is right isotonic, we show that $(S,\preceq_{GLR}, L,\oplus,w)$ is right isotonic.

$$\begin{aligned}
&\forall a,b\in L,\ w(a)\preceq_{GLR} w(b)\\
\iff &\forall a,b,x\in L,\ w(x\oplus a)\preceq w(x\oplus b)\\
\iff &\forall a,b,x,c\in L,\ w((x\oplus a)\oplus c)\preceq w((x\oplus b)\oplus c)\\
\iff &\forall a,b,x,c\in L,\ w(x\oplus(a\oplus c))\preceq w(x\oplus(b\oplus c))\\
\iff &\forall a,b,c\in L,\ w(a\oplus c)\preceq_{GLR} w(b\oplus c)
\end{aligned}$$

The following is a counterexample of GLR without preserving left monotonicity: $w(u_0u_1\cdots u_n)=\sum_{i=0}^{n-1}2^{i-(n-1)}$ $w(u_iu_{i+1})$. This algebra is left monotonic and right isotonic. Consider three links x,a,b such that $w(x)=8$, $w(b)=2$, $w(a)=1$. We have $5=w(x\oplus a)\succ w(x\oplus b\oplus a)=4$. Therefore, $w(a)\not\preceq_{GLR} w(b\oplus a)$, and $(S,\preceq_{GLR},L,\oplus,w)$ is not left monotonic.

(ii) If $(S,\preceq,L,\oplus,w)$ is right isotonic, it is also left monotonic according to Lemma 15. We show that GLR preserves left monotonicity if and only if the algebra is right monotonic.

$$\begin{aligned}
&\forall a,b\in L,\ w(a)\preceq_{GLR} w(b\oplus a)\\
\iff &\forall a,b,x\in L,\ w(x\oplus a)\preceq w(x\oplus(b\oplus a))\\
\iff &\forall a,b,x\in L,\ w(x\oplus a)\preceq w((x\oplus b)\oplus a)\\
\iff &\forall b,x\in L,\ w(x)\preceq w(x\oplus b)
\end{aligned}$$

∎

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.

TABLE V: Solvability of routing problems with various algebraic properties. Source routing is assumed because it demands less stringent algebraic properties than hop-by-hop routing. O indicates that the algorithm can directly solve the problem. X→O (or X→?) means that the algorithm cannot directly solve the problem, but its partial order version is (or may be) applicable after GLR or GRR. X→X means that the algorithm cannot work, and its partial order version cannot work either after any reduction.

Properties	Path Vector	Flooding	Link State
None	X $\xrightarrow{GLR}$ O	X $\xrightarrow{GRR}$ O	X $\xrightarrow{GRR}$ X
L. M.	X $\xrightarrow{GLR}$ O	X $\xrightarrow{GRR}$ O	X $\xrightarrow{GRR}$ X
R. M.	X $\xrightarrow{GLR}$ O	X $\xrightarrow{GRR}$ O	X $\xrightarrow{GRR}$?
M.	X $\xrightarrow{GLR}$ O	X $\xrightarrow{GRR}$ O	X $\xrightarrow{GRR}$?
M. & L. I.	O	X $\xrightarrow{GRR}$ O	X $\xrightarrow{GRR}$ O
L. M. & R. I.	X $\xrightarrow{GLR}$ O	O	X ⟶ X
R. M. & L. I.	O	X $\xrightarrow{GRR}$ O	X ⟶ X
M. & R. I.	X $\xrightarrow{GLR}$ O	O	O
M. & I.	O	O	O

B. Solvability of Routing Problems

Based on Lemma 14, Theorem 16, Theorem 17, Table II, and Table IV, the solvability of routing problems in general algebras can be determined. Table V provides sufficient conditions for routing problems with various algebraic properties to be solvable or unsolvable.

According to Lemma 15, only the properties listed in Table V need to be considered. We focus on three common algorithms with source routing: path vector, flooding, and link state, along with their partial order versions. The path vector algorithm on partial orders is detailed in [7], while the other partial order algorithms still lack relevant research. In Section VII, we propose a partial order version of the link state algorithm.

Path vector and flooding algorithms do not require monotonicity, so GLR or GRR can be applied when they fail due to the lack of left or right isotonicity (see X→O entries in Table V). However, for link state algorithm, right monotonicity must be ensured after GRR. There are some X→X and X→? entries because the reduction is ineffective for establishing right monotonicity (see Lemma 14 and Theorem 17 for details). For X→? entries, the right monotonicity after GRR is uncertain and depends on the specific algebra.

When addressing practical problems, a detailed analysis is necessary. While GLR or GRR is a feasible option, it may not always be the best choice. An illustrative example is provided in Section VI-A.

VI. Examples Revisited

In this section, we revisit the examples in Section IV based on the reduction methods proposed in the previous section.

A. GLR and GRR of MERT

Based on the row of M. in Table V, we can calculate GLR (or GRR) to enable the path vector (or flooding, link state) algorithm for MERT. Additionally, the proof of Theorem 7 is completed in this section. The GLR and GRR results for MERT differ from the original partial order before reduction, indicating that MERT is neither left nor right isotonic.

First, we calculate GLR $\preceq_{RTGLR}$ of MERT.

$$
\begin{aligned}
& w_3(a) \preceq_{RTGLR} w_3(b) \\
\iff & \forall x \in L,\ w_3(x \oplus a) \preceq w_3(x \oplus b) \\
\iff & \forall x \in L,\ w_1(x \oplus a) + w_2(x \oplus a) \le w_1(x \oplus b) + w_2(x \oplus b) \\
\iff & \forall x \in L,\ w_1(x)(N(a) - N(b)) \\
& \qquad \le w_1(b) - w_1(a) + (w_2(b) - w_2(a))N(x) \\
\iff & \sup_{x \in L}(w_1(x)(N(a) - N(b))) \\
& \quad \le w_1(b) - w_1(a) + \inf_{x \in L}((w_2(b) - w_2(a))N(x)) \\
\iff & (w_2(b) - w_2(a) \ge 0) \wedge \\
& (N(a) \le N(b) \longrightarrow w_3(b) - w_3(a) \ge 0) \wedge \\
& (N(a) > N(b) \longrightarrow \\
& \qquad w_3(b) - w_3(a) \ge (N(a) - N(b)) \sup_{x \in L} w_1(x))
\end{aligned}
$$

Note that $\inf_{x \in L} N(x) = 1$ and $\inf_{x \in L}((w_2(b) - w_2(a))N(x)) = -\infty$ when $w_2(b) - w_2(a) < 0$. However, it is related to $\sup_{x \in L} w_1(x)$, which requires global network information. We can avoid it and use a simpler partial order $\preceq_{RTLR}$ for MERT.

Definition 18. $\forall a, b \in L,\ w_3(a) \preceq_{RTLR} w_3(b) \iff (N(a) \le N(b)) \wedge (w_3(a) \preceq w_3(b)) \wedge (w_2(a) \preceq w_2(b))$.

Theorem 19. *The algebra $(S, \preceq_{RTLR}, L, \oplus, w)$ is monotonic and left isotonic.*

Proof: $\forall a, b \in L$, it is trivial that $N(a) \le N(b \oplus a)$. From left monotonicity of MEOW2 and MERT, we have $w_2(a) \preceq w_2(b \oplus a)$, and $w_3(a) \preceq w_3(b \oplus a)$. According to Definition 18, $w_3(a) \preceq_{RTLR} w_3(b \oplus a)$. The proof of $w_3(a) \preceq_{RTLR} w_3(a \oplus b)$ is similar.

If $w_3(a) \preceq_{RTLR} w_3(b)$, we have $N(a) \le N(b)$, $w_3(a) \preceq w_3(b)$ and $w_2(a) \preceq w_2(b)$, and then obtain that $\forall c \in L$,

$$
\begin{aligned}
N(c \oplus a) &= N(c)N(a) \le N(c)N(b) = N(c \oplus b) \\
w_3(c \oplus a) &= w_1(c \oplus a) + w_2(c \oplus a) \\
&= w_1(a) + w_1(c)N(a) + w_2(c) + w_2(a)N(c) \\
&= w_3(a) + w_1(c)N(a) + w_2(c) + w_2(a)(N(c) - 1) \\
&\le w_3(b) + w_1(c)N(b) + w_2(c) + w_2(b)(N(c) - 1) \\
&= w_1(b) + w_1(c)N(b) + w_2(c) + w_2(b)N(c) \\
&= w_1(c \oplus b) + w_2(c \oplus b) = w_3(c \oplus b) \\
w_2(c \oplus a) &= w_2(c) + w_2(a)N(c) \\
&\le w_2(c) + w_2(b)N(c) = w_2(c \oplus b)
\end{aligned}
$$

That is, $w_3(c \oplus a) \preceq_{RTLR} w_3(c \oplus b)$. Therefore, $\preceq_{RTLR}$ is a left isotonic reduction of MERT. ■

A solution for MERT with path vector algorithm involves using $\preceq_{RTLR}$ as the path selection criterion, comparing N, w_3 and w_2 of the paths and employing multi-objective routing.

Similarly, we can also calculate GRR $\preceq_{RTGRR}$ of MERT, and get a simpler right isotonic reduction $\preceq_{RTRR}$.

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.

Definition 20. $\forall a, b \in L,\ w_3(a) \preceq_{RTRR} w_3(b) \iff (N(a) \leq N(b)) \wedge (w_3(a) \preceq w_3(b)) \wedge (w_1(a) \preceq w_1(b))$.

Theorem 21. *The algebra $(S, \preceq_{RTRR}, L, \oplus, w)$ is monotonic and right isotonic.*

For flooding algorithm, we can use $\preceq_{RTRR}$ as the path selection criterion. For link state algorithm, in addition to the previous steps, it is crucial to verify the right monotonicity after reduction. Fortunately, according to Theorem 21, $\preceq_{RTRR}$ is indeed right monotonic.

B. GLR and GRR of MD

According to the row of M. in Table V, we can calculate GLR (or GRR) to enable the path vector (or flooding, link state) algorithm for MD.

$$\begin{aligned} & w(a) \preceq_{MDGLR} w(b) \\ \iff & \forall x \in L,\ w(x \oplus a) \preceq w(x \oplus b) \\ \iff & \forall x \in L,\ w(x) + w(a) + t(x, a) \leq w(x) + w(b) + t(x, b) \\ \iff & w(b) - w(a) \geq \sup_{x \in L} (t(x, a) - t(x, b)) \end{aligned}$$

The calculation of $\sup_{x \in L} (t(x, a) - t(x, b))$ is easy, since it is only related to the protocol translation cost matrix. A path vector solution for MD can be obtained by using $\preceq_{MDGLR}$ as the path selection criterion. Similarly, we can derive

$$w(a) \preceq_{MDGRR} w(b) \iff w(b) - w(a) \geq \sup_{x \in L} (t(a, x) - t(b, x))$$

A flooding solution for MD can be obtained by using $\preceq_{MDGRR}$ as the path selection criterion. To apply the link state algorithm using $\preceq_{MDGRR}$, it is necessary to verify the right monotonicity of the reduced algebra.

Theorem 22. *The algebra $(S, \preceq_{MDGRR}, L, \oplus, w)$ is right monotonic.*

Proof: For arbitrary path a and link b,

$$\begin{aligned} & w(a) \preceq_{MDGRR} w(a \oplus b) \\ \iff & w(a \oplus b) - w(a) \geq \sup_{x \in L} (t(a, x) - t(a \oplus b, x)) \\ \iff & w(b) \geq \sup_{x \in L} (t(a, x) - t(b, x)) - t(a, b) \end{aligned}$$

To prove this, it is sufficient to show that the right side of the inequality is non-positive. According to the practical meaning of t, the triangle inequality $t(a, x) \leq t(a, b) + t(b, x)$ holds. If this were not true, we could use the protocol of b as a transition to complete the translation from the protocol of a to that of x at a lower cost. Hence, the inequality holds. If b is a path containing multiple links, we can break down b into links and repeat the above process to conclude the proof. ■

VII. Link State Algorithm on Partial Orders

Routing algorithms on partial orders are needed to solve problems after reduction. Although path vector algorithms on partial orders have been explored [7], [28], [41], the link state approach is still indispensable when fast convergence is required. It is a classic method for calculating the shortest path based on Dijkstra's algorithm [42]. Its core concept involves *current nodes*, which are not updated once visited. In the case of partial order, however, the dominant path is not necessarily unique. Nodes must retain multiple paths that are not locally optimal but may generate a global optimal path. For example, in MERT routing (Figure 5), the optimal path from s to t is $suvt$, but link state algorithm fails to find it because node v only retains the locally optimal path sv and discards suv.

Algorithm 1 Dominant Link State Algorithm

Input: Network $G(V, E)$, source s, path selection criterion $\preceq$.
Output: Set of dominant weights $E(s, v)$ and dominant paths $L^*(s, v)$ from source s to node v for all $v \in V$.

1: **for all** $v \in V$ **do**
2: $\quad \Pi(v) \leftarrow \{w(\varphi)\}$, $E(s, v) \leftarrow \varnothing$, $L^*(s, v) \leftarrow \{\varphi\}$
3: $\Pi(s) = \{w(\varepsilon)\}$, $L^*(s, s) \leftarrow \{\varepsilon\}$
4: **while** $\bigcup_{v \in V} \Pi(v) \neq \varnothing$ **do**
5: $\quad w(l_{si}) \leftarrow$ arbitrary minimal element in $\bigcup_{v \in V} \Pi(v)$
6: $\quad$ **if** $w(l_{si}) \neq w(\varphi)$ **then**
7: $\quad\quad E(s, i) \leftarrow E(s, i) \cup \{w(l_{si})\}$, $L^*(s, i) \leftarrow L^*(s, i) \cup \{l_{si}\}$
8: $\quad\quad \Pi(i) \leftarrow \Pi(i) - \{w(l_{si})\}$
9: $\quad\quad$ **for all** j an out-neighbor of i **do**
10: $\quad\quad\quad \Pi(j) \leftarrow \mathcal{D}_{\preceq}(\Pi(j) \cup E(s, j) \cup \{w(l_{si} \oplus ij)\}) \setminus E(s, j)$
11: **return** $E(s, v)$ and $L^*(s, v)$ for all $v \in V$

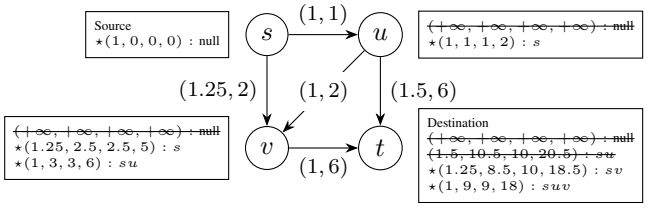


Fig. 6: Stable state of dominant link state algorithm for MERT with source routing and path selection criterion $\preceq_{RTRR}$. Each link is labeled with (N, e). w_3 is calculated using (2) (3). Each node is labeled with its calculated quadruples (N, w_1, w_2, w_3), where the ones with stars are elected by the node.

Let $\mathcal{D}_{\preceq}(A) = \{a \in A \mid \forall b \in A,\ b \not\prec a\}$ represent the dominant weight set of A. Algorithm 1 is a link state algorithm that can operate on a partial order. The main modification is using *current weights* instead of *current nodes*. In each iteration, a dominant weight $w(l_{si})$ is elected from the union of candidate weight sets $\bigcup_{v \in V} \Pi(v)$ that have not yet been chosen, and it becomes the *current weight*. Even with multiple dominant weights, electing any one of them will not affect the algorithm's correctness. It is added into the elected weight set $E(s, i)$ of node i, and the candidate weight sets of i's out-neighbors are updated. When an out-neighbor j receives the weight $w(l_{si} \oplus ij)$, it checks for more preferred weights in $\Pi(j)$ or $E(s, j)$. If none are found, the received weight is added into $\Pi(j)$, and less preferred weights in $\Pi(j)$ are eliminated (Line 10).

Algorithm 1 can repeatedly update nodes and compute all dominant paths, but this process may be time-consuming. To expedite convergence, we introduce an incremental recovery strategy. When a link failure occurs, we identify the affected nodes and recalculate only their paths, similar to the incremental Shortest Path First (SPF) [43].

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.

The stable state of Algorithm 1 for MERT routing is shown in Figure 6. The optimal path from s to t of MERT (path $suvt$) can be obtained from the dominant path set of $\preceq_{RTRR}$ (path set $\{suvt, svt\}$) using $\preceq_{MERT}$, i.e., choosing the path with the minimum w_3.

The following properties are essential to ensure the proper execution of Algorithm 1.

- Convergence and loop-freeness: strict right monotonicity of all loops in the network (Theorem 23). If source routing is used and loops are excluded during routing, this property is no longer needed and the convergence condition is automatically met.
- Dominance: right monotonicity and right isotonicity of the routing algebra (Theorem 27).

A. Proof of Convergence and Loop-Freeness

Theorem 23. *If all loops in the algebra $(S, \preceq, L, \oplus, w)$ are strictly right monotonic, Algorithm 1 terminates and the routing is loop-free.*

Proof: Consider a path $u_0u_1 \cdots u_n$ where $n \geq |V|$. It must contain a loop, i.e., there exist indices $i < j$ such that $u_i = u_j$. The loop $u_iu_{i+1} \cdots u_j$ is strictly right monotonic, meaning that $w(u_0u_1 \cdots u_i) \prec w(u_0u_1 \cdots u_iu_{i+1} \cdots u_j)$. When attempting to add $w(u_0u_1 \cdots u_iu_{i+1} \cdots u_j)$ into $\Pi(u_i)$ (Line 10), $w(u_0u_1 \cdots u_i)$ must have been elected as a *current weight* before and is present in $E(s, u_i)$, which prevents this addition. Thus, $\bigcup_{v \in V} \Pi(v)$ only contains weights of paths with fewer than $|V|$ links, ensuring that it is finite. Consequently, Algorithm 1 terminates in finite steps. The above proof also implies that the routing is loop-free. ∎

B. Proof of Dominance

Denote $A^*(s, u) = \mathcal{D}_{\preceq}(w(L(s, u))) = \{w(l_{su}) \mid \nexists l'_{su}$ s.t. $w(l'_{su}) \prec w(l_{su})\}$ as the set of dominant weights from node s to node u.

Lemma 24. *$\forall l \in L(s, u)$, $\exists a \in A^*(s, u)$ s.t. $a \preceq w(l)$.*

Proof: If $w(l) \in A^*(s, u)$, we are done. Otherwise, $\exists l'_{su}$ s.t. $w(l'_{su}) \prec w(l)$. If $w(l'_{su}) \in A^*(s, u)$, we are done. Otherwise, $\exists l''_{su}$ s.t. $w(l''_{su}) \prec w(l'_{su}) \prec w(l)$. Since $A^*(s, u) \neq \varnothing$ and $L(s, u)$ is finite, this process must terminate. Eventually, we can find a dominant weight $a \in A^*(s, u)$ s.t. $a \preceq w(l)$. ∎

Denote $W(s, v) = \{w(l_{su} \oplus uv) \mid w(l_{su}) \in E(s, u), u$ is an in-neighbor of $v\}$ as the set of weights that extend from $E(s, u)$ to node v for all in-neighbors u of v. It is evident that $E(s, v) \subseteq W(s, v)$. The following lemma further demonstrates the dominance of $E(s, v)$.

Lemma 25. *If the algebra $(S, \preceq, L, \oplus, w)$ is right monotonic, in the stable state of Algorithm 1, $E(s, v) = \mathcal{D}_{\preceq}(W(s, v))$.*

Proof: First, we prove $E(s, v) \subseteq \mathcal{D}_{\preceq}(W(s, v))$. For any $w(l_{sv}) \in E(s, v)$, if $w(l_{sv}) \notin \mathcal{D}_{\preceq}(W(s, v))$, then there exists $w(l'_{sv}) \in W(s, v)$ such that $w(l'_{sv}) \prec w(l_{sv})$. When $w(l_{sv})$ is elected as a *current weight*, $w(l'_{sv})$ has not yet been generated. Otherwise, $w(l_{sv})$ would have been eliminated by $w(l'_{sv})$ or some other more preferred weight in Line 10. Now, $w(l'_{sv})$ corresponds to a weight $w(l'_{sx})$ in $\Pi(x)$, which has not yet been extended to v, i.e., $l'_{sv} = l'_{sx} \oplus l'_{xv}$. Since $w(l_{sv})$ is the dominant weight among the candidate weights, we have $w(l'_{sx}) \not\prec w(l_{sv})$. From right monotonicity, we get $w(l'_{sx}) \preceq w(l'_{sx} \oplus l'_{xv}) = w(l'_{sv}) \prec w(l_{sv})$, which leads to a contradiction. Therefore, $w(l_{sv}) \in \mathcal{D}_{\preceq}(W(s, v))$.

Then, we prove $\mathcal{D}_{\preceq}(W(s, v)) \subseteq E(s, v)$. For any $w(l_{su} \oplus uv) \in \mathcal{D}_{\preceq}(W(s, v))$, since $l_{su} \in E(s, u)$, $w(l_{su} \oplus uv)$ will update $\Pi(v)$ in Line 10 after $w(l_{su})$ is elected as a *current weight*. As $w(l_{su} \oplus uv)$ is a dominant weight in $W(s, v)$, it cannot be eliminated by any other weight. Thus, it will be added into $\Pi(v)$ and eventually included in $E(s, v)$. ∎

Lemma 26. *If the algebra $(S, \preceq, L, \oplus, w)$ is right monotonic and right isotonic, in the stable state of Algorithm 1, $\forall l \in L(s, u)$, $\exists a \in E(s, u)$ s.t. $a \preceq w(l)$.*

Proof: We use induction on the link count of the path from source $s = u_0$ to destination u_n.

Base case: If the link count is 0, it is a trivial path ε. The elected weight at u_0 is $\{w(\varepsilon)\}$. The lemma holds.

Inductive hypothesis: For arbitrary $k \geq 1$, assume that the lemma holds for a link count less than or equal to $k - 1$.

Inductive step: Let $u_0u_1 \cdots u_k$ denote a path from source u_0 to destination u_k. According to the induction hypothesis, $\exists w(l_{u_0u_{k-1}}) \in E(u_0, u_{k-1})$ s.t. $w(l_{u_0u_{k-1}}) \preceq w(u_0u_1 \cdots u_{k-1})$. From right isotonicity, $w(l_{u_0u_{k-1}} \oplus u_{k-1}u_k) \preceq w(u_0u_1 \cdots u_k)$. Since $w(l_{u_0u_{k-1}} \oplus u_{k-1}u_k) \in W(u_0, u_k)$, according to Lemma 25, $\exists a \in E(u_0, u_k) = \mathcal{D}_{\preceq}(W(u_0, u_k))$ s.t. $a \preceq w(l_{u_0u_{k-1}} \oplus u_{k-1}u_k) \preceq w(u_0u_1 \cdots u_k)$. Therefore, the lemma holds for the case of link count k as well. ∎

Theorem 27. *If the algebra $(S, \preceq, L, \oplus, w)$ is right monotonic and right isotonic, Algorithm 1 satisfies dominance, which means that in the stable state, the weight set $E(s, u)$ elected by any destination u is equal to the set of dominant weights $A^*(s, u)$ from s to u.*

Proof: First, we prove $E(s, u) \subseteq A^*(s, u)$. $\forall w(l_{su}) \in E(s, u)$, from Lemma 24, $\exists w(l'_{su}) \in A^*(s, u)$ s.t. $w(l'_{su}) \preceq w(l_{su})$. According to Lemma 26, $\exists w(l''_{su}) \in E(s, u)$ s.t. $w(l''_{su}) \preceq w(l'_{su})$. Therefore, $w(l''_{su}) \preceq w(l'_{su}) \preceq w(l_{su})$, where $w(l''_{su})$, $w(l_{su}) \in E(s, u)$. However, according to Lemma 25, the weights in $E(s, u)$ are incomparable, and thus they are the same weights. Hence, $w(l_{su}) = w(l'_{su}) \in A^*(s, u)$.

Then, we prove $A^*(s, u) \subseteq E(s, u)$. $\forall w(l_{su}) \in A^*(s, u)$, according to Lemma 26, $\exists w(l'_{su}) \in E(s, u)$ s.t. $w(l'_{su}) \preceq w(l_{su})$. According to Lemma 24, $\exists w(l''_{su}) \in A^*(s, u)$ s.t. $w(l''_{su}) \preceq w(l'_{su})$. Therefore, $w(l''_{su}) \preceq w(l'_{su}) \preceq w(l_{su})$, where $w(l''_{su})$, $w(l_{su}) \in A^*(s, u)$. However, the weights in $A^*(s, u)$ are incomparable, and thus they are the same weights. Hence, $w(l_{su}) = w(l'_{su}) \in E(s, u)$. ∎

VIII. Evaluation

In this section, we measure the termination time of Algorithm 1 through simulations. We assess the influence of dom-

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.

TABLE VI: Network topologies used in simulations.

Topology	Nodes	Edges	Number of Dominant Paths
AS 1221	50	194	average 4.96, max 27
AS 1755	73	292	average 7.79, max 36
AS 3257	113	558	average 15.06, max 130
AS 6461	129	726	average 21.58, max 292

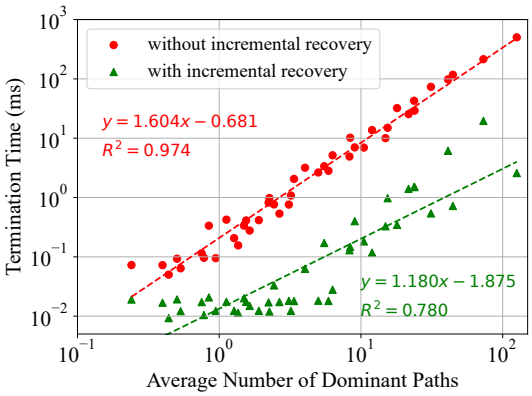


Fig. 7: Termination time of Algorithm 1 under different sizes of dominant path sets.

inant path set sizes, and the effectiveness of the incremental recovery mechanism in case of link failure.

We consider a constrained multi-objective routing problem, a special case of partial order routing. Given a network $G = (V, E)$, link weights, and constraints $C = (T, X, Y, Z)$, i.e., delay $< T$, packet loss rate $< X$, bandwidth $> Y$ and energy consumption (MEOW1) $< Z$, we compute the dominant path set using Algorithm 1. The network topologies used in the simulations are real topologies inferred by Rocketfuel [44] (Table VI). The problem involves four metrics. Rocketfuel topologies provide link delays and bandwidths. Packet loss rates are randomly assigned values between 0 and 0.05, and energy consumption values range between 200 and 1000.

The original size of the dominant path set is shown in Table VI. We introduce constraints of varying scales to manage the size of dominant path set. We first calculate the minimum and maximum values for the four metrics between each node pair, then select 10 sets of data within these extremes as constraints. Delay, packet loss rate, and energy consumption increase, while bandwidth decreases. Paths that do not meet these constraints are discarded during the algorithm's execution, affecting the size of the dominant path set. We also examine the unconstrained case, using the maximum values of delay, packet loss rate, and energy consumption, and the minimum value of bandwidth as constraints. We randomly delete 1% of links in the network and recalculate the dominant path set to evaluate the incremental recovery strategy.

Results are presented in Figures 7 and 8. The termination time of Algorithm 1 grows between linear and quadratic with respect to the number of dominant paths, while the incremental recovery can reduce the termination time by over an order

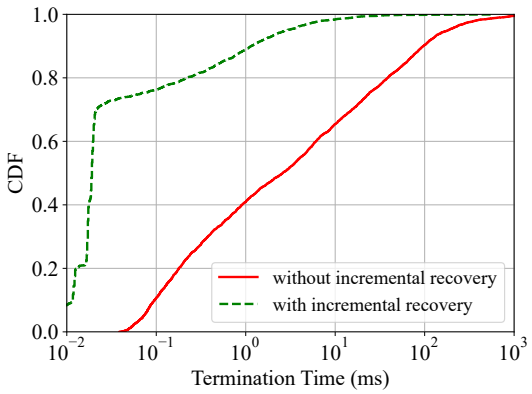


Fig. 8: Cumulative distribution function (CDF) of termination time of Algorithm 1.

of magnitude. Algorithm 1 terminates faster than path vector algorithms. The median termination time without incremental recovery is 2.58 ms, whereas with incremental recovery, it drops to under 0.20 ms, with the 90th percentile around 1 ms. Although the initial computation of all dominant paths may take some time, the algorithm can quickly update the dominant path set after a link failure. For example, with 10 dominant paths, the initial computation requires 8.38 ms, but updates take only 0.20 ms.

IX. Conclusion and Discussion

In this paper, we have presented a comprehensive study of non-commutative routing. The non-commutative algebras introduced in Section IV are also applicable to various similar scenarios. Table V primarily examines three common routing algorithms, and our analysis can be easily extended to other algorithms as well. We first identify the necessary algebraic properties for an algorithm to converge and route on the optimal path. Then the solvability of routing problems with various algebraic properties using this algorithm can be obtained from Table IV.

During the execution of Algorithm 1, the weights elected by nodes can only be eliminated when more preferred weights arrive. If a link failure occurs, a new computation instance needs to be initiated to calculate for the nodes affected by the failure. Although computational load can be lessened through incremental recovery strategy, in the future, more mechanism designs can be explored to address link failures.

Acknowledgment

This work is supported in part by the National Key R&D Program of China under Grant 2024YFB2906800, and the National Natural Science Foundation of China under Grant No.62221003, No.62472267, and No.62394322. Xingang Shi is the corresponding author.

References

[1] S. Sharma and S. K. Jena, "Cluster based multipath routing protocol for wireless sensor networks," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 2, pp. 14–20, 2015.

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.

[2] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green internet of things," *IEEE Systems Journal*, vol. 11, no. 2, pp. 983–994, 2015.

[3] P. N. Bideh, J. Sönnerup, and M. Hell, "Energy consumption for securing lightweight iot protocols," in *Proceedings of the 10th International Conference on the Internet of Things*, 2020, pp. 1–8.

[4] J. L. Sobrinho, "Network routing with path vector protocols: Theory and applications," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 49–60.

[5] Y. Yang and J. Wang, "Design guidelines for routing metrics in multihop wireless networks," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 1615–1623.

[6] T. Lengauer and D. Theune, "Efficient algorithms for path problems with general cost criteria," in *Automata, Languages and Programming: 18th International Colloquium Madrid, Spain, July 8–12, 1991 Proceedings 18*. Springer, 1991, pp. 314–326.

[7] J. L. Sobrinho and M. A. Ferreira, "Routing on multiple optimality criteria," in *Proceedings of the 2020 ACM SIGCOMM Conference*, 2020, pp. 211–225.

[8] M. I. Henig, "The shortest path problem with two objective functions," *European Journal of Operational Research*, vol. 25, no. 2, pp. 281–291, 1986.

[9] K. Rana and R. G. Vickson, "A model and solution algorithm for optimal routing of a time-chartered containership," *Transportation Science*, vol. 22, no. 2, pp. 83–95, 1988.

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

[11] 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.

[12] P. Hansen, "Bicriterion path problems," in *Multiple criteria decision making theory and application*. Springer, 1980, pp. 109–127.

[13] A. Warburton, "Approximation of pareto optima in multiple-objective, shortest-path problems," *Operations research*, vol. 35, no. 1, pp. 70–79, 1987.

[14] C. H. Papadimitriou and M. Yannakakis, "On the approximability of trade-offs and optimal access of web sources," in *Proceedings 41st annual symposium on foundations of computer science*. IEEE, 2000, pp. 86–92.

[15] G. Tsaggouris and C. Zaroliagis, "Improved fptas for multiobjective shortest paths with applications," 2005.

[16] S. Vassilvitskii and M. Yannakakis, "Efficiently computing succinct trade-off curves," *Theoretical Computer Science*, vol. 348, no. 2-3, pp. 334–356, 2005.

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

[18] M. Reitblatt, M. Canini, A. Guha, and N. Foster, "Fattire: Declarative fault tolerance for software-defined networks," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013, pp. 109–114.

[19] R. Soulé, S. Basu, P. J. Marandi, F. Pedone, R. Kleinberg, E. G. Sirer, and N. Foster, "Merlin: A language for provisioning network resources," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 2014, pp. 213–226.

[20] R. Beckett, R. Mahajan, T. Millstein, J. Padhye, and D. Walker, "Don't mind the gap: Bridging network-wide objectives and device-level configurations," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 328–341.

[21] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," in *Proceedings of the 2013 ACM SIGCOMM Conference*, 2013, pp. 15–26.

[22] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: experience with a globally-deployed software defined wan," in *Proceedings of the 2013 ACM SIGCOMM Conference*, 2013, pp. 3–14.

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

[24] T. G. Griffin and J. L. Sobrinho, "Metarouting," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, 2005, pp. 1–12.

[25] A. J. Gurney and T. G. Griffin, "Lexicographic products in metarouting," in *2007 IEEE International Conference on Network Protocols*. IEEE, 2007, pp. 113–122.

[26] T. G. Griffin and A. J. Gurney, "Increasing bisemigroups and algebraic routing," in *International Conference on Relational Methods in Computer Science*. Springer, 2008, pp. 123–137.

[27] M. L. Daggitt, A. J. Gurney, and T. G. Griffin, "Asynchronous convergence of policy-rich distributed bellman-ford routing protocols," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 103–116.

[28] J. L. Sobrinho and M. A. Ferreira, "From non-optimal routing protocols to routing on multiple optimality criteria," *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 294–307, 2023.

[29] H. Singh and S. Singh, "Energy consumption of tcp reno, newreno, and sack in multi-hop wireless networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, pp. 206–216, 2002.

[30] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002, pp. 146–156.

[31] ——, "Adaptive transmission power for optimal energy reliable multi-hop wireless communication," 2003.

[32] Q. Dong, S. Banerjee, M. Adler, and A. Misra, "Minimum energy reliable paths using unreliable wireless links," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 449–459.

[33] X.-Y. Li, H. Chen, Y. Shu, X. Chu, and Y.-W. Wu, "Energy efficient routing with unreliable links in wireless networks," in *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2006, pp. 160–169.

[34] Y.-D. Bromberg and V. Issarny, "Indiss: Interoperable discovery system for networked services," in *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, 2005, pp. 164–183.

[35] Y.-D. Bromberg, P. Grace, and L. Réveillere, "Starlink: runtime interoperability between heterogeneous middleware protocols," in *2011 31st International Conference on Distributed Computing Systems*. IEEE, 2011, pp. 446–455.

[36] C. Lerche, N. Laum, F. Golatowski, D. Timmermann, and C. Niedermeier, "Connecting the web with the web of things: lessons learned from implementing a coap-http proxy," in *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*. IEEE, 2012, pp. 1–8.

[37] H. Derhamy, J. Eliasson, and J. Delsing, "Iot interoperability—on-demand and low latency transparent multiprotocol translator," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1754–1763, 2017.

[38] G. Bouloukakis, N. Georgantas, P. Ntumba, and V. Issarny, "Automated synthesis of mediators for middleware-layer protocol interoperability in the iot," *Future Generation Computer Systems*, vol. 101, pp. 1271–1294, 2019.

[39] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service location protocol, version 2," 1999. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2608

[40] U. Forum, "Upnp device architecture 1.1," Tech. Rep., 2008. [Online]. Available: https://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf

[41] J. L. Sobrinho and R. P. Santos, "Correctness of eigrp generalized for arbitrary routing metrics and policies," in *2023 IEEE 31st International Conference on Network Protocols (ICNP)*. IEEE, 2023, pp. 1–11.

[42] E. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[43] J. McQuillan, I. Richer, and E. Rosen, "The new routing algorithm for the arpanet," *IEEE transactions on communications*, vol. 28, no. 5, pp. 711–719, 1980.

[44] 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.

Authorized licensed use limited to: Tsinghua University. Downloaded on October 15,2026 at 12:29:01 UTC from IEEE Xplore. Restrictions apply.