# Accepted Manuscript


Randomized oblivious integral routing for minimizing power cost

Yangguang Shi, Fa Zhang, Jie Wu, Zhiyong Liu

PII: S0304-3975(15)00628-3
DOI: http://dx.doi.org/10.1016/j.tcs.2015.07.007
Reference: TCS 10323

To appear in: *Theoretical Computer Science*

Received date: 13 November 2014
Revised date: 6 June 2015
Accepted date: 4 July 2015

Please cite this article in press as: Y. Shi et al., Randomized oblivious integral routing for minimizing power cost, *Theoret. Comput. Sci.* (2015), http://dx.doi.org/10.1016/j.tcs.2015.07.007

## Highlights

- Original study on energy saving in oblivious integral routing.
- An $\Omega$(|E|ˆ((a – 1)/(a +1))) lower bound on competitive ratio.
- A random oblivious integral routing algorithm with polylog-tight competitive ratio.
- A general framework to design and analyze oblivious integral routing algorithms.
- Polylog bound on competitive ratio for expanders and hypercubes.

# Randomized Oblivious Integral Routing for Minimizing Power Cost

Yangguang Shi[a,b], Fa Zhang[c], Jie Wu[d], Zhiyong Liu[a,e,*]

[a] *Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road, Haidian District, Beijing, China*
[b] *University of Chinese Academy of Sciences, No.19A Yuquan Road, Beijing, China*
[c] *Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road, Haidian District, Beijing, China*
[d] *Department of Computer and Information Sciences, College of Science and Technology, Temple University, Suite 400, Carnell Hall, 1803 N. Broad Street, Philadelphia, PA, USA*
[e] *State Key Lab for Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road, Haidian District, Beijing, China*

**Abstract**

Given an undirected network $G(V, E)$ and a set of traffic requests $\mathcal{R}$, the minimum power-cost routing problem requires that each $R_k \in \mathcal{R}$ be routed along a single path to minimize $\sum_{e \in E}(l_e)^\alpha$, where $l_e$ is the traffic load on edge $e$ and $\alpha$ is a constant greater than 1. Typically, $\alpha \in (1, 3]$. This problem is important in optimizing the energy consumption of networks.

To address this problem, we propose a randomized oblivious routing algorithm. An oblivious routing algorithm makes decisions independently of the current traffic in the network. This feature enables the efficient implementation of our algorithm in a distributed manner, which is desirable for large-scale high-capacity networks.

An important feature of our work is that our algorithm can satisfy the integral constraint, which requires that each traffic request $R_k$ should follow a single path. We prove that, given this constraint, no randomized oblivious routing algorithm can guarantee a competitive ratio bounded by $o(|E|^{\frac{\alpha-1}{\alpha+1}})$. By contrast, our approach provides a competitive ratio of $O(|E|^{\frac{\alpha-1}{\alpha+1}} \log^{\frac{2\alpha}{\alpha+1}} |V| \cdot \log^{\alpha-1} D)$, where $D$ is the maximum demand of traffic requests. Furthermore, our results also hold for a more general case where the objective is to minimize $\sum_e (l_e)^p$, where $p \geq 1$ is an arbitrary unknown parameter with a given upper bound $\alpha > 1$.

*Corresponding author. TEL.: +86-010-62600634.
*Email addresses:* shiyangguang@ict.ac.cn (Yangguang Shi), zhangfa@ict.ac.cn (Fa Zhang), jiewu@temple.edu (Jie Wu), zyliu@ict.ac.cn (Zhiyong Liu)

Some preliminary results of our work have appeared in a conference version [35] of this paper under the title “Oblivious integral routing for minimizing the quadratic polynomial cost”.

The theoretical results established in proving these bounds can be further generalized to a framework of designing and analyzing oblivious integral routing algorithms, which is significant for research on minimizing $\sum_e (l_e)^\alpha$ in specific scenarios with simplified problem settings. For instance, we prove that this framework can generate an oblivious integral routing algorithm whose competitive ratio can be bounded by $O(\log^\alpha |V| \cdot \log^{\alpha-1} D)$ and $O(\log^{3\alpha} |V| \cdot \log^{\alpha-1} D)$ on expanders and hypercubes, respectively.

*Keywords:* Oblivious Routing, Integral Routing, Randomized Algorithm, Competitive Ratio, Energy Efficiency

## 1. Introduction

In a minimum power-cost routing (MPR) problem, we are given a network $G(V, E)$ and a set of traffic requests $\mathcal{R} = \{R_1, R_2, \cdots, R_k, \cdots\}$. $V$ and $E$ represent the node set and edge set of $G$, respectively. Here we consider a typical case where $G$ is undirected [6], i.e., each edge $e \in E$ is bidirectional. Each traffic request $R_k \in \mathcal{R}$ specifies its source-target pair $\{s_k, t_k\} \in V \times V$ and the demand (i.e., the volume of flow that needs to be routed) $d_k \geq 1$. Routing traffic requests along any edge $e \in E$ will incur a cost that grows superadditively with the load. Formally, let $l_e$ be the flow routed along $e$, the corresponding cost will be a **power** function $f(l_e) = (l_e)^\alpha$, where $\alpha$ is a constant greater than 1 and is typically in the interval $(1, 3]$. The objective is to route every $R_k \in \mathcal{R}$ along a single path to minimize the overall cost $\sum_e f(l_e)$. In the following, we will also use an equivalent form of the overall cost, $\|\vec{l}\|_\alpha^\alpha$, where $\vec{l}$ represents the load vector composed of every $l_e$, and the operator $\|\cdot\|_\alpha^\alpha$ represents the $\alpha$-th power of the $\alpha$-*norm*.

The MPR problem is attracting great attention because of the emergence of energy conservation issues in data networks [6, 8, 21, 28]. Research conducted by the U.S. Department of Energy [1] indicates that over 50 billion kWh of energy is annually consumed by data networks, whereas at least 40% of this can be saved if the electric power consumption[2] of network elements is in proportion to the actual traffic. For this reason, the speed scaling technique has become ubiquitous because it allows network devices to dynamically adjust their electric power consumption according to traffic. The electric power consumption of a network device with the capability of speed scaling can be characterized by the function $P(x) = x^q$ with $q > 1$, where $x$ is the working speed and $q$ is a constant, the value of which depends on the hardware. The value of $q$ is usually assumed to be around 3 [12, 24], while new studies indicate that it can be smaller. For instance, it will respectively take the values 1.11, 1.62, and 1.66 for Intel PXA 270, Pentium M770, and a TCP offload engine [41]. This implies that results

[2] To avoid any ambiguity, throughout this paper we use *electric power consumption* to refer to the electrical power consumed by actual devices, whereas when we talk about the "power-cost", we mean a cost function in the form of $(l_e)^\alpha$.

of the MPR problem will help optimize the electric power consumption of the entire network.

In this paper, we investigate *oblivious routing* strategies [15, 17, 20, 22, 26, 27, 31] for the MPR problem. For an oblivious routing algorithm, each of its routing decisions is made independently of network traffic. This means that the routing paths for each $R_k \in \mathcal{R}$ are determined only using knowledge of the topology of the network $G$, the source-target pair $\{s_k, t_k\}$, and some random bits (if needed), in the absence of any information on the set $\mathcal{R} - R_k$, the value of $d_k$, or the load vector $\vec{l}$. An oblivious routing algorithm can be viewed as precomputing a routing "template" before any traffic request is known. In particular, for a *deterministic* oblivious routing strategy, the corresponding template specifies a unit flow $\mathtt{H}(u, v)$ for each node pair $\{u, v\}$ in $G$ [15, 17, 31]. Then, each $R_k$ will be routed according to the flow $d_k \cdot \mathtt{H}(s_k, t_k)$. By contrast, for a *randomized* oblivious routing strategy, the precomputed template contains a probabilistic distribution over a collection of unit flows $\{\mathtt{H}_1(u, v), \cdots, \mathtt{H}_i(u, v), \cdots\}$ for each $\{u, v\}$ [31]. In such case, each $R_k$ will be routed according to the flow $d_k \cdot \mathtt{H}_i(s_k, t_k)$ with the corresponding probability $p_i(s_k, t_k)$, which implies that traffic requests with the same source-target pair can go through different paths.

An oblivious routing algorithm is attractive because of its simplicity of implementation. Since it allows for the routing strategy to be precomputed and stored in the routing table of every node, the oblivious routing algorithm can be efficiently implemented in a distributed manner [32]. It is especially significant for high-capacity network routers, where traffic requests will dynamically arrive on a transient timescale in the order of nanoseconds [37, 42]. In such a circumstance, path selection based on real-time assessment of the traffic pattern is time-consuming, which implies that a routing algorithm depending on the current traffic may be inefficient. By contrast, oblivious routing algorithms can make timely routing decisions by simply generating random bits and looking up the routing tables, which will be a desirable feature when dealing with the issue of energy efficiency in large-scale high-capacity networks.

To the best of our knowledge, only a few oblivious routing algorithms have been designed to minimize $\|\vec{l}\,\|_\alpha^\alpha$, including [11, 15, 27]. These works, however, only consider the splittable version of MPR, where traffic requests can be partitioned into fractional flows. In this paper, we focus on the **unsplittable** version, which requires that each $R_k \in \mathcal{R}$ should follow a single path. Throughout this paper, we will refer to this requirement as the *integral constraint*. This constraint is important for many practical environments [3], especially for data networks where the frames are not arbitrarily divisible.

When the integral constraint exists, any deterministic oblivious routing algorithm will have to specify a fixed path for each source-target pair [17, 20]. We prove that because of the superadditivity of the cost function, such a routing algorithm cannot provide a competitive ratio of $o(|E|^{\alpha-1})$, which implies a lower bound of $\Omega(|E|)$ on the competitive ratio for the typical case $\alpha \geq 2$. *Competitive ratio* here refers to the largest gap between the cost incurred by the oblivious routing algorithm and the cost associated with the optimal solution [27, 32]. Such a lower bound indicates that randomization is required by oblivious routing strategies to guarantee a satisfactory performance.

### *1.1. Our Results*

In this paper, we propose a *Randomized Oblivious Integral Routing* algorithm, called ROI-Routing, to solve the MPR problem. For each traffic request, we will select a path from a set of precomputed candidates in a randomized manner. This selection procedure will be carried out independently for each traffic request according to a precomputed probability distribution. The number of random bits used for each traffic request $R_k$ is bounded by $O(\log |E|)$. With regard to the performance of ROI-Routing, we prove that:

**Theorem 1.** *ROI-Routing has a competitive ratio of* $O\Big(|E|^{\frac{\alpha-1}{\alpha+1}} \log^{\frac{2\alpha}{\alpha+1}} |V| \cdot \log^{\alpha-1} D\Big)$*, where* $D = \max_k d_k$*.*

Note that the parameter $D$ will only be used in our analysis, whereas our algorithm procedure does not depend on $D$. This competitive ratio is tight up to a polylogarithmic factor $O(\log^{\frac{2\alpha}{\alpha+1}} |V| \cdot \log^{\alpha-1} D)$, since we have the following lower bound:

**Theorem 2.** *No randomized oblivious routing algorithm that satisfies the integral constraint can provide a competitive ratio of* $o(|E|^{\frac{\alpha-1}{\alpha+1}})$ *for the MPR problem.*

An important aspect of our results is that they are not restricted to cases where the precise form of the cost function is known. As mentioned above, the exponent of the power-cost function depends on the hardware, whereas measuring its actual value may be complicated from a practical point of view. For such applications, where the exponent of the cost function is not precisely known, we need to find a solution that is simultaneously satisfactory for every possible cost function. In this paper, the property of being able to yield such solutions will be referred to as function-oblivious [15], and we prove that our algorithm has this property. Formally,

**Theorem 3.** *For the case that the cost function associated with every* $e \in E$ *is* $f_p(l_e) = (l_e)^p$*, where* $p$ *is an arbitrary unknown number in* $[1, \alpha]$ *and* $\alpha$ *is still a given number greater than* 1*, ROI-Routing can still guarantee a competitive ratio of* $O(|E|^{\frac{\alpha-1}{\alpha+1}} \log^{\frac{2\alpha}{\alpha+1}} |V| \cdot \log^{\alpha-1} D)$*.*

**Theorem 4.** *There is no* $o(|E|^{\frac{\alpha-1}{\alpha+1}})$*-competitive randomized oblivious routing algorithm for the case in Theorem 3.*

Note that Theorem 4 is not a trivial application of Theorem 2, since Theorem 4 holds for a more general case where the routing algorithm is allowed to violate the integral constraint.

It is remarkable that the constant hidden in the big $O$ notations of our competitive ratios given by Theorem 1 and Theorem 3 is at most $c_0 2^{\alpha+1} B_\alpha$, where $c_0$ is an absolute constant and $B_\alpha$ is the fractional Bell number with parameter $\alpha$. According to [10], $B_\alpha$ follows Dobiński's formula [13], i.e., $B_\alpha = \sum_{k=1}^{+\infty} \frac{k^\alpha e^{-1}}{k!}$, where $e$ represents Euler's number. The values of $B_\alpha$ for some typical $\alpha$ are given in Table 1.

| $\alpha$ | 1.1 | 1.62 | 1.66 | 2 | 3 |
|---|---|---|---|---|---|
| $B_\alpha$ | 1.0603 | 1.4945 | 1.5386 | 2 | 5 |

Table 1: The values of $B_\alpha$ for some typical $\alpha$. Particularly, the values $1.1, 1.62,$ and $1.66$ are the exponents of the power-cost functions corresponding to Intel PXA 270, Pentium M770, and a TCP offload engine, respectively [41].

Some of our intermediate results obtained in deriving the theorems above can be further extended from the perspective of theory. In particular, the propositions established in proving Theorem 1 can be generalized to a framework to develop and analyze oblivious integral routing algorithms for minimizing $\|\vec{l}\|_\alpha$. This framework is significant for research on MPR in specific scenarios where input instances have good properties that can be used to simplify the problem.

An application of this framework is generating a new oblivious integral routing algorithm $\Psi_I^{\mathcal{E}}$ with a competitive ratio of $O\Big(\Big[\frac{\vartheta(G)\log|V|}{h(G)\log\frac{2\vartheta(G)}{2\vartheta(G)-h(G)}}\Big]^\alpha \log^{\alpha-1} D\Big)$ for MPR, where $\vartheta(G)$ represents the maximum node degree of the nodes in $V$, and $h(G)$ represents the *edge expansion* [23] of $G$. Compared with ROI-Routing, the algorithm $\Psi_I^{\mathcal{E}}$ is more applicable to the scenarios where the networks have well-bounded maximum node degrees and edge expansions. Two classes of networks with extensive applications in both computer science and practical scenarios are specially investigated for purposes of illustration:

- Expander $G_{\mathrm{EX}}$ [23], in which the maximum node degree has a constant upper bound and the edge expansion has a constant lower bound.
- Hypercube $G_{\mathrm{HC}}$ [30], which has a $\Theta(\log|V|)$ maximum node degree and a constant edge expansion.

We prove that the competitive ratio of $\Psi_I^{\mathcal{E}}$ can be respectively bounded by $O(\log^\alpha|V| \cdot \log^{\alpha-1} D)$ and $O(\log^{3\alpha}|V| \cdot \log^{\alpha-1} D)$ on expanders and hypercubes. We then again apply our framework to combine ROI-Routing with $\Psi_I^{\mathcal{E}}$ to generate another oblivious integral routing algorithm $\Psi_I^*$, which has an $O(\log^{\frac{2\alpha}{\alpha+1}}|V| \cdot \log^{\alpha-1} D)$-tight competitive ratio as well as ROI-Routing, while simultaneously preserving the advantages of $\Psi_I^{\mathcal{E}}$ over ROI-Routing on networks with special topologies, including expanders and hypercubes.

### *1.2. Related Works*

The MPR problem was first studied by Andrews et al. [6] to reduce energy consumption in data networks. They proposed a randomized algorithm with an approximation ratio of $2^\alpha\gamma_\alpha(\log_2 D)^\alpha$, where $\gamma_\alpha$ denotes max{1 +

$j_\alpha 2^{\alpha(j_\alpha+1)}e,\ 2 + j_\alpha 2^{\alpha(j_\alpha+1)}\}$ with $j_\alpha = \lceil 2\log_2(\alpha+4)\rceil$. In Table 2, we list the values of $\gamma_\alpha$ for some typical $\alpha$. The best known approximation of this problem was provided by Makarychev and Sviridenko's algorithm [28], the approximation ratio of which is bounded by $(1+\varepsilon)B_\alpha$ for any $\varepsilon > 0$. These algorithms are designed for static scenarios where all traffic requests are known at the beginning of computation and routing decisions are made offline. In particular, their results depend on the global fractional optimal solutions, which are difficult to obtain in dynamic scenarios.

| $\alpha$ | 1.1 | 1.62 | 1.66 | 2 | 3 |
|---|---|---|---|---|---|
| $\gamma_\alpha$ | $1.3194 \times 10^3$ | $1.1463 \times 10^4$ | $5.1336 \times 10^4$ | $2.6722 \times 10^5$ | $3.4204 \times 10^7$ |

Table 2: The values of $\gamma_\alpha$ for some typical $\alpha$.

First investigated by Valiant et al. [38, 39], oblivious routing algorithms have attracted considerable attention due to their efficiency of implementation. As summarized in [15, 32], most of the existing research in the area is devoted to two categories of objectives: congestion minimization (i.e., minimizing $\|\vec{l}\|_\infty$, see [20, 26, 31]) and dilation minimization (i.e., minimizing $\|\vec{l}\|_1$, see [9, 16]). By contrast, only a few researchers [15, 22, 27] have considered the problem of minimizing superlinear power costs using oblivious routing algorithms. Among these, [22] proposed an oblivious routing algorithm for a restricted case where the cost function $f(l_e) = (l_e)^2$ and all traffic requests are directed to the same target. This result does not hold for the general case with arbitrary $\alpha > 1$ or multi-target traffic requests.

Englert and Räcke [15] designed an oblivious routing algorithm to minimize $\|\vec{l}\|_\alpha$. Their result was not constructive for the case $\alpha \neq 2$ until the problem of determining the induced norm of a given matrix was solved by Bhaskara and Vijayaraghavan [11]. When applied to minimize $\|\vec{l}\|_\alpha^\alpha$, their approach can guarantee a competitive ratio of $O(\log^\alpha |V|)$. However, their approach was designed for the splittable case where fractional flow is permitted, and therefore cannot satisfy the integral constraint. Furthermore, it is impossible to achieve such a polylogarithmic competitive ratio when the integral constraint exists because, in such cases, no randomized oblivious routing algorithm can guarantee a competitive ratio of $o(|E|^{\frac{\alpha-1}{\alpha+1}})$. This implies that the integral constraint makes our problem much more difficult for oblivious strategies.

Based on the random walks (also called electric walks [26]), Lawler and Narayanan [27] proposed an oblivious routing algorithm to simultaneously minimize all $L_p$-norms ($p \in [1, \infty)$) of the load vector. Their approach can be viewed as transforming $G$ into an electricity network where each edge has a unit resistance, and routing each traffic request between a node pair $\{u, v\}$ according to a unit electric current that flows into $u$ and out of $v$. Such an approach cannot satisfy the integral constraint either. Furthermore, we prove that any integral routing algorithm that takes the electric current as a probabilistic distribution will yield a high competitive ratio of $\Omega(|E|^{\frac{1}{2}\max\{1,\alpha-2\}})$ for MPR.

### 1.3. Organization

The remainder of this paper is organized as follows: in Section 2, we introduce and establish a series of probabilistic tools that will be used in our analysis. In Section 3, we establish a sequence of lower bounds on the competitive ratios of oblivious routing algorithms for the MPR problem; in particular, we prove Theorem 2 and Theorem 4. In Section 4, we provide an overview of the decomposition tree [15, 16, 31], a data structure that will be used to identify the candidate paths, and present the details of our algorithm. To analyze the competitive ratio of ROI-Routing, we first fix the candidate paths and study the influence of the randomized selection procedure in Section 5. Section 6 contains an analysis of candidate paths obtained by ROI-Routing, and completes the proof of Theorem 1. Furthermore, we establish Theorem 3 in Section 6, which shows that ROI-Routing is function-oblivious. In Section 7, some of our theoretical results are further generalized to a framework of designing and analyzing oblivious integral routing algorithms for minimizing $\|\vec{l}\|_\alpha^\alpha$. We apply this framework to generate algorithms which can provide a better result on the specific networks with well-bounded maximum node degrees and edge expansions. We summarize our findings and offer concluding thoughts in Section 8.

## 2. Probabilistic Tools

In this section, we state and prove some moment inequalities on the sum of independent random variables. The propositions here will be used in our analysis of the competitive ratios of oblivious routing algorithms.

**Lemma 5** (Jensen's Inequality, [25])**.** *Let $X$ be a random variable and $\varphi$ be a convex function, then we have*

$$\varphi(\mathbb{E}[X]) \;\leq\; \mathbb{E}[\varphi(X)]$$

*where $\mathbb{E}[\cdot]$ represents the expectation of a random variable.*

**Definition 1** (Fractional Bell Number)**.** *For any $p \geq 1$, the fractional Bell number $B_p$ represents the $p$-th moment of a Poisson random variable with expectation 1. It can be obtained using Dobiński's formula [13],*

$$B_p \;=\; \frac{1}{e}\sum_{k=1}^{+\infty}\frac{k^p}{k!}$$

*where $e$ represents Euler's number.*

**Lemma 6** ([8])**.** *Let $\{Y_1, Y_2, \cdots, Y_i, \cdots\}$ be a set of independent random variables with Bernoulli distribution supported on the set $\{0, 1\}$, and $\lambda \doteq \mathbb{E}[\sum_i Y_i]$. For any $p \geq 1$, $\mathbb{E}[(\sum_i Y_i)^p] \leq \mathbb{E}[(\Psi_\lambda)^p]$, where $\Psi_\lambda$ is a Poisson random variable with parameter $\lambda$.*

**Lemma 7** ([8])**.** *For any $p \geq 1$ and $\lambda \geq 0$, $\mathbb{E}[(\Psi_\lambda)^p] \leq \max\{\lambda, \lambda^p\} \cdot \mathbb{E}[(\Psi_1)^p]$.*

Lemma 6 and Lemma 7 directly imply that

$$\mathbb{E}\Big[\Big(\sum_i Y_i\Big)^p\Big] \;\leq\; B_p \cdot \max\Big\{\mathbb{E}\Big[\sum_i Y_i\Big],\ \Big(\mathbb{E}\Big[\sum_i Y_i\Big]\Big)^p\Big\} \tag{1}$$

Note that unlike the result in [10] that is restricted to the discrete case $p \in \mathbb{Z}^+$, Eq. (1) holds for any real $p \geq 1$. In the following, we extend Eq. (1) to a more general case where the Bernoulli random variables are supported on the set $\{0, d\}$ for any $d \in \mathbb{Z}^+$.

**Lemma 8.** *For any $d \in \mathbb{Z}^+$, let $\{Y_1^d, Y_2^d, \cdots, Y_i^d \cdots\}$ be a set of independent random variables with Bernoulli distribution supported on the set $\{0, d\}$. For any $p \geq 1$,*

$$\mathbb{E}\Big[\Big(\sum_i Y_i^d\Big)^p\Big] \;\leq\; B_p \cdot \max\Big\{d^{p-1} \cdot \mathbb{E}\Big[\sum_i Y_i^d\Big],\ \Big(\mathbb{E}\Big[\sum_i Y_i^d\Big]\Big)^p\Big\}$$

*Proof.* For each $Y_i^d$, let $Y_i'$ be a Bernoulli random variable supported on the set $\{0, 1\}$ such that $Y_i^d = d \cdot Y_i'$. Then, we have:

$$\begin{aligned}
\mathbb{E}\Big[\Big(\sum_i Y_i^d\Big)^p\Big] &= \mathbb{E}\Big[\Big(\sum_i Y_i' \cdot d\Big)^p\Big] \\
&= d^p \cdot \mathbb{E}\Big[\Big(\sum_i Y_i'\Big)^p\Big] \\
&\leq B_p \cdot \max\Big\{d^p \mathbb{E}\Big[\sum_i Y_i'\Big],\ \Big(d \cdot \mathbb{E}\Big[\sum_i Y_i'\Big]\Big)^p\Big\} \\
&= B_p \cdot \max\Big\{d^{p-1} \mathbb{E}\Big[\sum_i Y_i^d\Big],\ \Big(\mathbb{E}\Big[\sum_i Y_i^d\Big]\Big)^p\Big\}
\end{aligned}$$

The second equality above follows from the commutative property of the multiplication of random variables and the linearity of the expectation. The inequality above follows from Eq. (1). □

## 3. Lower Bounds on Competitive Ratio

In this section, we investigate the lower bounds on the competitive ratio of any oblivious routing algorithm for the MPR problem, and in particular, prove Theorem 2 and Theorem 4. We begin by proving the lower bound corresponding to deterministic oblivious routing algorithms.

**Theorem 9.** *For the MPR problem, any deterministic oblivious routing algorithm will yield a competitive ratio of $\Omega(|E|^{\alpha-1})$.*

*Proof.* This proof is based on the network $G_1(V_1, E_1)$ shown in Fig. 1. There are $\lfloor |E_1|/2 \rfloor$ edge-disjoint paths of length 2 connecting the node pair $\{u_1, v_1\}$.

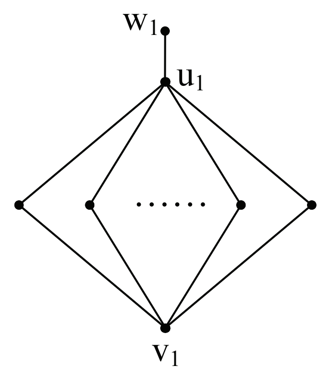



Figure 1: Network $G_1$.

These parallel paths are called the *canonical paths*. We add a node $w_1$ to $G_1$ and connect $u_1$ and $w_1$ iff $|E_1|$ is odd.

Consider a traffic request set $\mathcal{R}_1 = \{R_1, R_2, \cdots, R_{\lfloor |E_1|/2 \rfloor}\}$. For each $R_k \in \mathcal{R}_1$, $\{s_k, t_k\} = \{u_1, v_1\}$ and $d_k = 1$. According to the definition, a deterministic oblivious routing algorithm will route any traffic request between $\{u_1, v_1\}$ by scaling up a same precomputed flow. When the integral constraint exists, such an algorithm will have to route every $R_k \in \mathcal{R}_1$ along a single fixed path. It implies that at least one of the canonical paths will be used by all $R_k \in \mathcal{R}_1$, which will incur a cost of at least $f(\lfloor |E_1|/2 \rfloor) \cdot 2 = 2(\lfloor |E_1|/2 \rfloor)^\alpha$. By contrast, the optimal solution will route each $R_k$ along a distinct canonical path whose cost will be $2\lfloor |E_1|/2 \rfloor$. Thus, the competitive ratio will be at least $\lfloor |E_1|/2 \rfloor^{\alpha-1}$. $\square$

Randomized routing algorithms can guarantee a better competitive ratio than deterministic algorithms. However, it is still impossible for them to yield a polylogarithmic competitive ratio for our problem. To see this, we first consider a typical case where $\alpha = 2$. The lower bound obtained in this typical case then will be extended to a general case with an arbitrary $\alpha > 1$ in the proof of Theorem 2.

**Lemma 10.** *Given the integral constraint, no oblivious routing algorithm can guarantee a competitive ratio bounded by $o(|E|^{1/3})$ for the scenario where the objective is to minimize $\|\vec{l}\|_2^2$, even if it can select paths in a randomized manner.*

*Proof.* Here we consider the network $G_2(V_2, E_2)$ in Fig. 2. It is constructed as follows: nodes $u_2$ and $v_2$ are directly connected by an edge $e_{u_2,v_2}$, called the *short canonical path* between $u_2$ and $v_2$. Moreover, there are $\Delta = \tau^2$ acyclic disjoint paths of length $\tau = \lfloor (|E_2| - 1)^{1/3} \rfloor$ connecting $u_2$ and $v_2$. They are referred to as the *long canonical paths*. For the case that $(|E_2| - 1)^{1/3} \notin \mathbb{Z}^+$, a ring with $|E_2| - \lfloor (|E_2| - 1)^{1/3} \rfloor^3 - 1$ edges will be attached to the node $u_2$

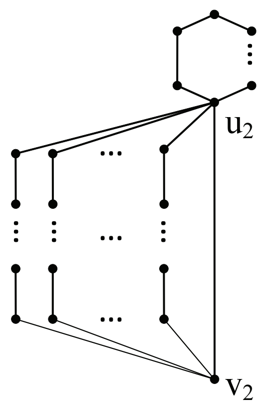



Figure 2: Network $G_2$.

to complement the graph. A randomized oblivious routing algorithm $A$ will integrally route traffic requests between $u_2$ and $v_2$ along the short canonical path with probability $\lambda_A \geq 0$. We now consider two cases:

1. $\lambda_A \geq \frac{\sqrt{5}-1}{2}$. In this case, we construct a set $\mathcal{R}_2$ of $\Delta$ independent traffic requests between $u_2$ and $v_2$. For each request $R_k \in \mathcal{R}_2$, let $d_k = 1$. In such a case, the expected load on the short canonical path will be $\mathbb{E}[l_{e_{u_2,v_2}}] = \lambda_A \cdot \Delta$. Since for any $\alpha > 1$, the power function $f(l_e) = (l_e)^\alpha$ is convex, by Lemma 5 we can bound the corresponding expected cost by:

$$\mathbb{E}\Big[(l_{e_{u_2,v_2}})^2\Big] \;\geq\; \Big(\mathbb{E}[l_{e_{u_2,v_2}}]\Big)^2 \;=\; (\lambda_A \Delta)^2 \;=\; (\lambda_A \tau^2)^2 \tag{2}$$

   However, if we route each request along a distinct long canonical path, the cost will be $\Delta \cdot \tau = \tau^3$. Thus, the competitive ratio will be at least $\frac{(\lambda_A \cdot \tau^2)^2}{\tau^3} = \frac{3-\sqrt{5}}{2}\tau$.

2. $\lambda_A < \frac{\sqrt{5}-1}{2}$. Now, there exists a single traffic request $R_{\text{large}}$ with $d_{\text{large}} = \Delta$ between $u_2$ and $v_2$. $R_{\text{large}}$ will be routed along a long canonical path with a probability of at least $1 - \lambda_A$. Therefore, the expectation of the total cost will be greater than $(1-\lambda_A)\Delta^2\tau = (1-\lambda_A)\tau^5$. By contrast, if we simply route $R_{\text{large}}$ along the short canonical path, the cost will be $\tau^4$. The competitive ratio will be at least $\frac{(1-\lambda_A)\tau^5}{\tau^4} = \frac{3-\sqrt{5}}{2}\tau$.

To sum up, the competitive ratio has a lower bound of $\frac{3-\sqrt{5}}{2}\tau = \frac{3-\sqrt{5}}{2}\lfloor(|E_2| - 1)^{1/3}\rfloor$. $\square$

For the general case where the cost function has an arbitrary exponent $\alpha > 1$, we need to admit $\alpha$ as an argument in the construction of networks to deduce the lower bound in Theorem 2.

*Proof of Theorem 2.* We construct a network $G_3(V_3, E_3)$ in a similar manner to $G_2$. The differences are that the length $\tau$ and the number $\Delta$ of long canonical paths are now set to $\left\lfloor [(|E_3|-1)/2]^{\frac{\alpha-1}{\alpha+1}} \right\rfloor$ and $\left\lceil \frac{|E_3|-1}{2\tau} \right\rceil$, respectively. This theorem can then be proved by plugging the new values of $\Delta$ and $\tau$ into the proof of Lemma 10. □

The lower bounds given above all correspond to the integral constraint. We now proceed to prove Theorem 4 to show that even if the integral constraint is allowed to be violated, the problem is still challenging when we desire the property of *function-oblivious.* For a traffic request set $\mathcal{R}$ and a positive number $p \in [1, \alpha]$, we use $\mathrm{OPT}_I^p(\mathcal{R})$ to denote the cost of the integral solution that is optimal with respect to the cost function $f(x) = x^p$. For a routing algorithm $\Phi$, we use $\vec{l}_\Phi^{\mathcal{R}}$ to represent the load vector incurred by routing $\mathcal{R}$ with $\Phi$. Then, we have:

**Lemma 11.** *No deterministic routing algorithm $\Phi_D$ can guarantee* $\max_{p\in[1,\alpha]}$ $\max_{\mathcal{R}} \left\{ \frac{\|\vec{l}_{\Phi_D}^{\mathcal{R}}\|_p^p}{OPT_I^p(\mathcal{R})} \right\}$ *bounded by* $o\left(|E|^{\frac{\alpha-1}{\alpha+1}}\right)$*, even if it is allowed to violate the integral constraint.*

*Proof.* We construct a network $G_4(V_4, E_4)$ based on the network $G_2$ in Fig. 2. In $G_4$, the length $\tau$ and the number $\Delta$ of the long canonical paths are set as follows:

$$\tau = \left\lfloor \left[ \frac{|E|_4 - 1}{4 \cdot (1/\alpha)^{\frac{1}{\alpha-1}}} \right]^{\frac{\alpha-1}{\alpha+1}} \right\rfloor, \quad \Delta = \left\lceil \frac{|E_4|-1}{2\tau} \right\rceil \tag{3}$$

Note that the value of $\tau$ here is set in a manner different from Theorem 2. Before deducing a lower bound on the competitive ratio with $G_4$, we will first prove that the settings in Eq. (3) are feasible. Since $\alpha > 1$, we have $(1/\alpha)^{1/(\alpha-1)} < 1$. Thus, we have $\tau > 1$ for any $|E_4| \geq 5$. Under the same assumption on $|E_4|$, it can also be inferred that:

$$\frac{|E_4|-1}{2\tau} \geq \left[ \left( \frac{|E_4|-1}{2} \right)^2 \frac{2^{\alpha-1}}{\alpha} \right]^{\frac{1}{\alpha+1}} \geq \left[ \left( \frac{|E_4|-1}{2} \right)^2 \left( 2^{\frac{1}{\ln 2}-1} \ln 2 \right) \right]^{\frac{1}{\alpha+1}} > 1$$

where the third inequality above follows from the fact that the value $\alpha_0 = 1/\ln 2$ can minimize $2^{\alpha-1}/\alpha$. This implies that:

1. $\Delta > 1$ since $\Delta \geq \frac{|E_4|-1}{2\tau}$.
2. $\Delta < 2 \cdot \frac{|E_4|-1}{2\tau} = \frac{|E_4|-1}{\tau}$ since for any positive number $x > 1$, $\lceil x \rceil < x+1 < 2x$. Therefore, $\tau\Delta < |E_4| - 1$.

Hence, the settings in Eq. (3) for $\tau$ and $\Delta$ are consistent.

Consider a traffic request $\mathcal{R}' = \{R_1, R_2, \ldots, R_k, \ldots, R_\Delta\}$ such that for every $R_k \in \mathcal{R}'$, $\{s_k, t_k\} = \{u_2, v_2\}$ and $d_k = 1$. In this case, we have $\mathrm{OPT}_I^1(\mathcal{R}') = \Delta$ (by simply routing all requests along $e_{u_2 v_2}$) and $\mathrm{OPT}_I^2(\mathcal{R}') \leq \Delta \cdot \tau$ (by routing each request along a distinct long canonical path). Let the flow routed by $\Phi_D$ along the short canonical path be $\varepsilon \cdot \Delta$ with $0 \leq \varepsilon \leq 1$. It renders $\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|_1 \geq \varepsilon\Delta + \tau(1-\varepsilon)\Delta$, and $\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|^\alpha_\alpha \geq \varepsilon^\alpha\Delta^\alpha + \tau(1-\varepsilon)^\alpha\Delta$. In such a case, the competitive ratio will be at least:

$$\begin{aligned}
\max\left\{\frac{\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|_1}{\mathrm{OPT}^1_I(\mathcal{R}')}, \frac{\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|^\alpha_\alpha}{\mathrm{OPT}^\alpha_I(\mathcal{R}')}\right\} &\geq \frac{1}{2}\left(\frac{\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|_1}{\mathrm{OPT}^1_I(\mathcal{R}')} + \frac{\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|^\alpha_\alpha}{\mathrm{OPT}^\alpha_I(\mathcal{R}')}\right) \\
&\geq \frac{1}{2}\left[(1-\varepsilon)\tau + \varepsilon^\alpha\Delta^{\alpha-1}/\tau\right] \\
&\geq \frac{1}{2}\left\{(1-\varepsilon)\tau + \varepsilon^\alpha\frac{[(|E_4|-1)/2]^{\alpha-1}}{\tau^\alpha}\right\}
\end{aligned}$$

To derive a lower bound on the competitive ratio, here we consider the case that $\varepsilon$ is set to the value $\varepsilon^*$ that can minimize the above formulation. Such an $\varepsilon^*$ can be found through a second derivative test. Formally, let $h(\varepsilon) = (1-\varepsilon)\tau + \frac{\varepsilon^\alpha[(|E_4|-1)/2]^{\alpha-1}}{\tau^\alpha}$. Taking $\tau$ and $|E_4|$ as constants independent of $\varepsilon$, the derivative and the second derivative of $h$ with respect to $\varepsilon$ will respectively be

$$h'(\varepsilon) = -\tau + \frac{\alpha\varepsilon^{\alpha-1}[(|E_4|-1)/2]^{\alpha-1}}{\tau^\alpha}, \quad h''(\varepsilon) = \frac{\alpha(\alpha-1)\varepsilon^{\alpha-2}[(|E|_4-1)/2]^{\alpha-1}}{\tau^\alpha}$$

By solving the equality $h'(\varepsilon^*) = 0$, it can be obtained that $\varepsilon^* = \left(\frac{1}{\alpha}\right)^{\frac{1}{\alpha-1}} \frac{2\tau^{\frac{\alpha+1}{\alpha-1}}}{|E_4|-1}$. This is the value that we need, since $h''\left(\left(\frac{1}{\alpha}\right)^{\frac{1}{\alpha-1}} \frac{2\tau^{\frac{\alpha+1}{\alpha-1}}}{|E_4|-1}\right) > 0$. Thus, the minimum value of $h$ is:

$$\min_\varepsilon h(\varepsilon) \;=\; h(\varepsilon^*) \;=\; h\left(\left(\frac{1}{\alpha}\right)^{\frac{1}{\alpha-1}} \frac{2\tau^{\frac{\alpha+1}{\alpha-1}}}{|E_4|-1}\right) \;=\; \tau - \left(\frac{1}{\alpha}\right)^{\frac{1}{\alpha-1}} \frac{2(\alpha-1)\tau^{\frac{2\alpha}{\alpha-1}}}{\alpha(|E_4|-1)}$$

Plugging the value of $\tau$ in terms of $|E_4|$ and $\alpha$ into the second item in the equation above, we have:

$$\min_\varepsilon h(\varepsilon) \;\geq\; \tau\left[1 - \left(\frac{1}{\alpha}\right)^{\frac{1}{\alpha-1}} \frac{2(\alpha-1)}{\alpha(|E_4|-1)}\tau^{\frac{\alpha+1}{\alpha-1}}\right] \;\geq\; \tau\left[1 - \frac{\alpha-1}{2\alpha}\right] \;\geq\; \frac{\tau}{2}$$

To sum up, we have

$$\max\left\{\frac{\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|_1}{\mathrm{OPT}^1_I(\mathcal{R}')}, \frac{\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|^\alpha_\alpha}{\mathrm{OPT}^\alpha_I(\mathcal{R}')}\right\} \;\geq\; \frac{\tau}{4} \tag{4}$$

Since $\max_{p\in[1,\alpha]}\max_{\mathcal{R}} \frac{\|\vec{l}^{\mathcal{R}}_{\Phi_D}\|^p_p}{\mathrm{OPT}^p_I(\mathcal{R})} \geq \max_{p\in\{1,\alpha\}} \frac{\|\vec{l}^{\mathcal{R}'}_{\Phi_D}\|^p_p}{\mathrm{OPT}^p_I(\mathcal{R}')}$ and $\tau \geq \frac{1}{2}\left[\frac{|E|_4-1}{4(\frac{1}{\alpha})^{\frac{1}{\alpha-1}}}\right]^{\frac{\alpha-1}{\alpha+1}}$, this theorem follows. $\square$

**Lemma 12.** *For any $\beta \geq 1$, if there exists a randomized routing algorithm $\Phi$ that can guarantee that $\max_{p\in[1,\alpha]} \max_{\mathcal{R}} \left\{ \frac{\|\vec{l}_{\Phi}^{\mathcal{R}}\|_p^p}{OPT_I^p(\mathcal{R})} \right\} \leq \beta$, there must exist a deterministic routing algorithm $\Phi_D$ that can approximate every $OPT_I^p(\mathcal{R})$ ($1 \leq p \leq \alpha$) by $\beta$ when the integral constraint is allowed to be violated.*

*Proof.* Let $\mathcal{L}_{\Phi}^{\mathcal{R}}$ be the set of all load vectors that can be generated by $\Phi$ for the traffic request set $\mathcal{R}$ with non-zero probability. In particular, each load vector $\vec{l}_{\Phi}^{\mathcal{R}}(i) \in \mathcal{L}_{\Phi}^{\mathcal{R}}$ will be generated by $\Phi$ with probability $\Pr_{\Phi}(i)$. The expectation of the cost incurred by $\Phi$ with respect to the cost function $f(x) = x^p$ will be:

$$\sum_{\vec{l}_{\Phi}^{\mathcal{R}}(i)\in\mathcal{L}_{\Phi}^{\mathcal{R}}} \|\vec{l}_{\Phi}^{\mathcal{R}}(i)\|_p^p \cdot \Pr_{\Phi}(i) \quad \geq \quad \Big\| \sum_{\vec{l}_{\Phi}^{\mathcal{R}}(i)\in\mathcal{L}_{\Phi}^{\mathcal{R}}} \vec{l}_{\Phi}^{\mathcal{R}}(i) \cdot \Pr_{\Phi}(i) \Big\|_p^p \tag{5}$$

which follows from the convexity of the power function. It is easy to see that there exists a deterministic *fractional* routing algorithm $\Phi_D$ that can generate the load vector $\sum \vec{l}_{\Phi}^{\mathcal{R}}(i)\Pr_{\Phi}(i)$. Eq. (5) implies that replacing the randomized algorithm $\Phi$ with the corresponding deterministic algorithm $\Phi_D$ will never increase the cost. Thus, this proposition follows. □

It can be directly inferred from Lemma 11 and Lemma 12 that:

**Theorem 13.** *No randomized routing algorithm $\Phi$ can guarantee that $\max_{p\in[1,\alpha]} \max_{\mathcal{R}} \left\{ \frac{\|\vec{l}_{\Phi}^{\mathcal{R}}\|_p^p}{OPT_I^p(\mathcal{R})} \right\}$ is bounded by $o\left(|E|^{\frac{\alpha-1}{\alpha+1}}\right)$, even if it is allowed to violate the integral constraint.*

This theorem directly implies Theorem 4.

## 4. Algorithm Description

Our major contribution in this paper is proposing the ROI-Routing algorithm for MPR. Here, we provide a few important definitions related to ROI-Routing and the algorithm procedure. We start with an overview of the convex combination of decomposition trees [15, 31], a data structure that is used by ROI-Routing.

For a non-empty set $\mathcal{U}$, a *partition* of $\mathcal{U}$ refers to a collection of **non-overlapping** and non-empty subsets $\{\upsilon_1, \upsilon_2, \cdots, \upsilon_z\}$ of $\mathcal{U}$ such that $\bigcup_{i=1}^{z} \upsilon_i = \mathcal{U}$. A decomposition tree $T$ of a network $G(V, E)$ is a rooted tree with the following properties [5, 15, 16, 31]:

I. Each tree node $v^T$ in $T$ corresponds to a non-empty node set $S(v^T) \in V$.

II. The root of $T$ corresponds to $V$.

III. Each leaf node of $T$ corresponds to a *singleton* set of node in $V$.

IV. For any internal node $u^T$ of $T$, the node sets corresponding to the children of $u^T$ form a partition of $S(u^T)$.

It can be inferred from the definition of partition that:

**Proposition 1.** *For each decomposition tree $T$ of $G(V, E)$, each node in $V$ is contained by exactly one singleton set corresponding to a leaf node of $T$.*

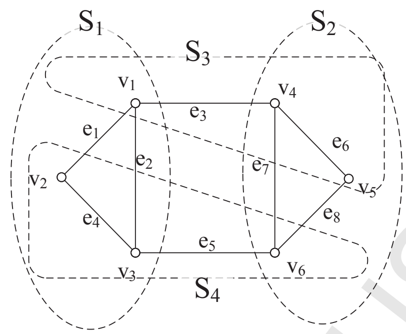
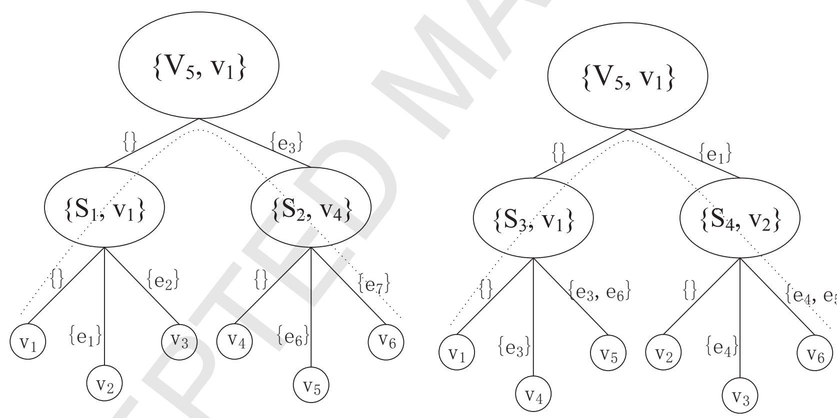



Figure 3: Network $G_5(V_5, E_5)$. $S_1, S_2, S_3$ and $S_4$ are subsets of nodes where $S_1 = \{v_1, v_2, v_3\}, S_2 = \{v_4, v_5, v_6\}, S_3 = \{v_1, v_4, v_5\}$ and $S_4 = \{v_2, v_3, v_6\}$.




(a) Decomposition tree $T_1$.

(b) Decomposition tree $T_2$.

Figure 4: Two decomposition trees with embeddings into $G_5$. The dotted line in each $T_i$ marks the unique acyclic path that connects the leaf nodes respectively corresponding to $v_1$ and $v_6$ in $G_5$.

Each decomposition tree $T$ has an embedding $(\xi, \mathcal{P})$ to the network $G$ [15, 31], where $\xi$ is a function mapping each tree node $v^T \in T$ to a node $\xi(v^T) \in S(v^T)$, and $\mathcal{P}$ is a function mapping each tree edge $e^T = (u^T, v^T)$ to a path $\mathcal{P}(e^T)$ in $G$ between $\xi(u^T)$ and $\xi(v^T)$. For illustration, consider the network $G_5(V_5, E_5)$ shown in Fig. 3. Two decomposition trees, $T_1$ and $T_2$, of the network

$G_5$ are given in Fig. 4. To demonstrate the embeddings, in Fig. 4 we label each leaf node $v^T$ by the corresponding node $\xi(v^T)$, while labeling each internal node $u^T$ by $\{S(u^T), \xi(u^T)\}$. Moreover, each tree edge $e^T$ in Fig. 4 is labeled by the corresponding path $\mathcal{P}(e^T)$. Note that decomposition trees corresponding to the same manner of partitioning $V$ can have different embeddings to $G$, and we will determine the embedding of a specific $T$ in the computation.

In a decomposition tree $T$, let the unique acyclic path connecting the node pair $\{u^T, v^T\}$ in $T$ be $P^T_{u^T,v^T}$. For any pair of nodes $\{u, v\}$ in $G$, we can obtain a path $P_{u,v}(T)$ between them by concatenating the paths $\mathcal{P}(e^T)$ corresponding to each tree edge $e^T \in P^T_{\xi^{-1}(u),\xi^{-1}(v)}$. Here $\xi^{-1}$ is the inverse function of $\xi$. According to Property III of decomposition tree and Proposition 1, the function $\xi$ induces a bijection between $V$ and the leaf nodes of $T$ [15, 31], which implies that $\xi^{-1}$ is well defined and maps each $v \in V$ to a distinct leaf node in $T$. For example, consider a node pair $\{v_1, v_6\}$ in the network $G_5$. On each decomposition tree $T_i$ in Fig. 4, an acyclic path marked by a dotted line connects the leaf nodes respectively corresponding to $v_1$ and $v_6$. Through the embedding function $\mathcal{P}$, these tree paths can be respectively transformed into two different paths $P_{v_1,v_6}(T_1) = \{e_3, e_7\}$ and $P_{v_1,v_6}(T_2) = \{e_1, e_4, e_5\}$ between $v_1$ and $v_6$.

A routing strategy based on a decomposition tree $T$ is assigning the path $P_{s_k,t_k}(T)$ to each traffic request $R_k$. Such a routing strategy can be identified by an $|E| \times \binom{n}{2}$-dimensional matrix $M_T$, whose $j$-th column is the load vector incurred by routing a traffic request $R$ with $d = 1$ between the $j$-th node pair.

**Definition 2** (Convex combination of decomposition trees [15, 31])**.** *Given a network $G(V, E)$, a convex combination $\mathcal{C}^G$ of decomposition trees is a set of decomposition trees $\{T_1, T_2, \cdots, T_i, \cdots\}$, each of which has a non-negative weight $\lambda_i$ such that $\sum_i \lambda_i = 1$. The superscript $G$ in $\mathcal{C}^G$ will be omitted when it is obvious from the context.*

**Definition 3** (Tree-based matrix [15, 31])**.** *A convex combination of decomposition trees, $\mathcal{C}$, can be identified by an $|E| \times \binom{|V|}{2}$-dimensional matrix $M_{\mathcal{C}} = \sum_i \lambda_i M_{T_i}$, which will be referred to as a tree-based matrix.*

To illustrate these definitions, consider the node pair $\{v_1, v_6\}$ of network $G_5$ in Fig. 4 again. Suppose that a convex combination $\mathcal{C}^{G_5}$ consists of the two decomposition trees $T_1$ and $T_2$ shown in Fig. 4 with weights $\lambda_1 = 0.6$ and $\lambda_2 = 0.4$, respectively. For any vector $\vec{x}$, we use $\vec{x}^{\,\mathrm{tr}}$ to represent the transpose of $\vec{x}$. Then, according to the paths $P_{v_1,v_6}(T_1)$ and $P_{v_1,v_6}(T_2)$ given above, the column vectors corresponding to $\{v_1, v_6\}$ in $M_{T_1}$ and $M_{T_2}$ will respectively be $\vec{m}_1^{\mathrm{tr}} = \{0, 0, 1, 0, 0, 0, 1, 0\}$ and $\vec{m}_2^{\mathrm{tr}} = \{1, 0, 0, 1, 1, 0, 0, 0\}$. Thus, in the tree-based matrix $M_{\mathcal{C}^{G_5}}$, the column vector corresponding to $\{v_1, v_6\}$ will be $\vec{m}^{\mathrm{tr}} = 0.6\vec{m}_1^{\mathrm{tr}} + 0.4\vec{m}_2^{\mathrm{tr}} = \{0.4, 0, 0.6, 0.4, 0.4, 0, 0.6, 0\}$.

In addition to the convex combination of decomposition trees and the tree-based matrix, other concepts used in our algorithm are defined as follows:

**Definition 4** (Column selector)**.** *A column selector $\Upsilon$ is a $\binom{|V|}{2} \times |E|$-dimensional matrix of Boolean variables. In particular, $\Upsilon(i, j)$ (i.e., the $j$-th element in the*

*$i$-th row of $\Upsilon$) is 1 if the $j$-th edge $e_j$ is between the $i$-th node pair; otherwise $\Upsilon(i,j) = 0$.*

**Definition 5** (Induced $L_p$ norm)**.** *For a matrix $A$ and any $p \geq 1$, $\|A\|_p$ denotes the induced p-norm of $A$, i.e., $\|A\|_p = \max_{\|x\|_p \neq 0} \frac{\|A \cdot x\|_p}{\|x\|_p}$. Moreover, we use $\|A\|_p^q$ to represent $(\|A\|_p)^q$ for convenience.*

**Lemma 14** ([11, 15])**.** *There exists an absolute constant $c_0 \geq 1$ with the following property. For any $p > 1$, we can compute a convex combination $\mathcal{C}$ of decomposition trees such that $\|M_{\mathcal{C}} \cdot \Upsilon\|_p \leq c_0 \cdot \log_2 |V|$ in polynomial time [3].*

Given the above definitions, we now present the algorithm procedure. Our ROI-Routing algorithm consists of two phases:

1. Precomputation Phase. Given a network $G(V, E)$, we precompute a specific convex combination $\mathcal{C}^*$ of decomposition trees for $G$ such that the corresponding tree-based matrix $M_{\mathcal{C}^*}$ has the property

$$\|M_{\mathcal{C}^*} \cdot \Upsilon\|_\chi \leq c_0 \cdot \log_2 |V| \tag{6}$$

where $\chi$ is defined as follows:

$$\chi = \begin{cases} \alpha & \text{if } (c_0 \cdot \log_2 |V|)^\alpha \geq |E|^{1-\frac{1}{\alpha}}(c_0 \cdot \log_2 |V|); \\ \\ \frac{\alpha+1}{2-(\alpha-1)\frac{\log_2(c_0 \log_2 |V|)}{\log_2 |E|}} & \text{otherwise.} \end{cases} \tag{7}$$

Without loss of generality, in this paper we only consider non-trivial input cases where $|V| \geq 2$ and $|E| \geq 1$. In such cases, we have $\frac{\log_2(c_0 \log_2 |V|)}{\log_2 |E|} > 0$, which implies that $\chi > 1$. According to Lemma 14, we can generate such a convex combination $\mathcal{C}^*$ in polynomial time.

2. Rolling Dice Phase. Whenever a traffic request $R$ is given, we independently select a decomposition tree $T_k^* \in \mathcal{C}^*$ in a randomized manner and route $R$ based on $T_k^*$. The probability $\Pr_k^*$ that a tree $T_k^*$ is selected is equivalent to its weight $\lambda_k^*$. This setting is consistent because the weights are non-negative and $\sum_i \lambda_i^* = 1$.

**Theorem 15.** *The number of random bits used by our algorithm for each traffic request $R_k$ is bounded by $O(\log |E|)$.*

*Proof.* According to [11], we can find $\mathcal{C}^*$ in $O(|E|^{c'})$ steps, where $c'$ is an absolute constant. Each step consists of $O(|E| \log |V|)$ iterations [31], and at most one decomposition tree is obtained in each iteration. This implies that the total number of decomposition trees in $\mathcal{C}^*$ can be bounded by $O(|E|^{c'+1} \log |V|)$. Thus, we need at most $O(\log(|E|^{c'+1} \log |V|)) = O(\log |E|)$ random bits to select a decomposition tree from $\mathcal{C}^*$. □

[3] By simply plugging the constant factors in [11, 15, 16, 31] together, it can be inferred that such a constant $c_0$ can be found in the interval $[1, 68]$.

## 5. Randomized Path Selection

In this part, we will analyze the influence of the Rolling Dice Phase on the competitive ratio independently of the Precomputation Phase. To isolate the Rolling Dice Phase from the Precomputation Phase, we assume that a convex combination $\mathcal{C}$ of decomposition trees is given as input, and the Rolling Dice Phase is carried out according to $\mathcal{C}$. We will prove that for any given $\mathcal{C}$, the Rolling Dice Phase can guarantee that the competitive ratio is bounded by $O\left(\max\{\|M_{\mathcal{C}}\Upsilon\|_1, \|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha\} \log^{\alpha-1} D\right)$.

For ease of reference, we first list the definitions of a few notations used in our analysis:

- $\mathrm{OPT}_F(\mathcal{R})$. The cost of the fractional optimal solution for the traffic request set $\mathcal{R}$.
- $\mathrm{OPT}_I(\mathcal{R})$. The cost of the integral optimal solution for $\mathcal{R}$.
- $\vec{l}^{\mathcal{R}}_{\mathrm{OPT}_F}$. The load vector corresponding to the fractional optimal solution for $\mathcal{R}$. If there is more than one such vector, $\vec{l}^{\mathcal{R}}_{\mathrm{OPT}_F}$ can be any one of them. The notation $\vec{l}^{\mathcal{R}}_{\mathrm{OPT}_I}$ is defined in a similar manner.
- $\vec{l}^{\mathcal{R}}_{\mathrm{OBL}_F}$. The load vector incurred by routing $\mathcal{R}$ according to the given convex combination $\mathcal{C}$ with *fractional* flow in the manner of Englert–Räcke [15], i.e., for each $R_k \in \mathcal{R}$, routing the amount $\lambda_i d_k$ of flow based on each decomposition tree $T_i \in \mathcal{C}$.
- $\vec{l}^{\mathcal{R}}_{\mathrm{OBL}_I}$. The load vector incurred by routing $\mathcal{R}$ through our ROI-Routing algorithm.
- $\vec{l}(e)$. It represents the element of the load vector $\vec{l}$ corresponding to the edge $e$, i.e., $\vec{l}(e) = l_e$. This notation will be used along with the subscripts and superscripts defined above.
- $[A]_i$ and $A(j)$. For a matrix $A$, we use $A_i$ and $A(j)$ to denote its $i$-th row and $j$-th column, respectively. Moreover, we use $A(i,j)$ to represent the $j$-th element in the $i$-th row of $A$.

Let $\mathcal{R}'$ be an arbitrary non-empty subset of $\mathcal{R}$ and $\vec{l}^{\mathcal{R}}_{\mathrm{OBL}_I}(e, \mathcal{R}')$ be the load of the edge $e$ corresponding to the traffic requests in $\mathcal{R}'$ in the case where all requests in $\mathcal{R}$ are routed integrally according to $\mathcal{C}$. The Rolling Dice Phase has the following property:

**Lemma 16.** $\mathbb{E}\left[\vec{l}^{\mathcal{R}}_{OBL_I}(e, \mathcal{R}')\right] = \vec{l}^{\mathcal{R}'}_{OBL_F}(e)$.

*Proof.* For each traffic request $R_k \in \mathcal{R}$, we construct a $\binom{|V|}{2}$-dimensional vector $\vec{d}_k$. The $i$-th element in $\vec{d}_k$ is set to $\delta(\sigma(k), i) \cdot d_k$, where $\delta$ is the Kronecker delta function and $\sigma(k)$ is the index of the source-target pair of $R_k$. Recalling that the probability with which the tree $T_i \in \mathcal{C}$ is selected is denoted by $\Pr_i$, we have:

$$\begin{aligned}
\mathbb{E}\left[\vec{l}^{\mathcal{R}}_{\mathrm{OBL}_I}(e, \mathcal{R}')\right] &= \sum_{R_k \in \mathcal{R}'} d_k \cdot \sum_i \Pr_i \cdot M_{T_i}(e, \sigma(k)) \\
&= \sum_{R_k \in \mathcal{R}'} d_k \cdot \sum_i \lambda_i \cdot M_{T_i}(e, \sigma(k)) \\
&= \sum_{R_k \in \mathcal{R}'} \left[\sum_i \lambda_i M_{T_i}\right]_e \cdot \vec{d}_k \\
&= \sum_{R_k \in \mathcal{R}'} (M_{\mathcal{C}})_e \cdot \vec{d}_k
\end{aligned}$$

According to [15], $\vec{l}^{\mathcal{R}'}_{\mathrm{OBL}_F} = M_{\mathcal{C}} \cdot \sum_{R_k \in \mathcal{R}'} \vec{d}_k$. Therefore, this lemma follows. □

For two vectors/matrices $A_1$ and $A_2$ with the same dimensions, we say $A_1$ is dominated by $A_2$ iff each element in $A_1$ is no greater than the element with the same index in $A_2$. Such a relation will be denoted by $A_1 \preccurlyeq A_2$ and $A_2 \succcurlyeq A_1$. Then,

**Lemma 17.** *Let $j_1$, $j_2$, $j_3$ be the indices of any three node pairs $\{u_{j_1}, v_{j_1}\}$, $\{u_{j_2}, v_{j_2}\}$, and $\{u_{j_3}, v_{j_3}\}$, respectively. We have $M_{\mathcal{C}}(j_3) \preccurlyeq M_{\mathcal{C}}(j_1) + M_{\mathcal{C}}(j_2)$ if $\{u_{j_1}, v_{j_1}\} \bigcap \{u_{j_2}, v_{j_2}\} \neq \emptyset$ and $\{u_{j_3}, v_{j_3}\} \subseteq \{u_{j_1}, v_{j_1}\} \bigcup \{u_{j_2}, v_{j_2}\}$.*

*Proof.* If $j_3 = j_1$ or $j_3 = j_2$, this proposition trivially holds. Otherwise, since $\{u_{j_3}, v_{j_3}\} \subseteq \{u_{j_1}, v_{j_1}\} \bigcup \{u_{j_2}, v_{j_2}\}$, we assume without loss of generality that $u_{j_3} = u_{j_1}$ and $v_{j_3} = v_{j_2}$. In this case, $v_{j_1} = u_{j_2}$ because $\{u_{j_1}, v_{j_1}\} \bigcap \{u_{j_2}, v_{j_2}\} \neq \emptyset$. For any tree $T \in \mathcal{C}$, it is obvious that

$$P^T_{u^T_{j_1}, v^T_{j_1}} \bigcup P^T_{u^T_{j_2}, v^T_{j_2}} = P^T_{u^T_{j_1}, v^T_{j_1}} \bigcup P^T_{v^T_{j_1}, v^T_{j_2}} \supseteq P^T_{u^T_{j_1}, v^T_{j_2}} = P^T_{u^T_{j_3}, v^T_{j_3}}$$

where $u^T_{j_k} \doteq \xi^{-1}(u_{j_k})$ and $v^T_{j_k} \doteq \xi^{-1}(v_{j_k})$ for each $k \in \{1, 2, 3\}$. In particular, the superset inequality above follows from the fact that $P^T_{u^T_{j_1}, v^T_{j_1}} \bigcup P^T_{v^T_{j_1}, v^T_{j_2}}$ forms a path between $u^T_{j_1}$ and $v^T_{j_2}$, and removing any edge in $P^T_{u^T_{j_1}, v^T_{j_2}}$ will disconnect $u^T_{j_1}$ from $v^T_{j_2}$. Thus, when we map these paths to $G$, the obtained paths $P_{u_{j_3}, v_{j_3}}(T)$ and $P_{u_{j_1}, v_{j_1}}(T) \bigcup P_{u_{j_2}, v_{j_2}}(T)$ have a common sequence of $|P_{u_{j_3}, v_{j_3}}(T)|$ edges. Since $M_T(j)$ is the load vector incurred by routing a unit demand between the $j$-th node pair, $M_T(j_3) \preccurlyeq M_T(j_1) + M_T(j_2)$. Then,

$$M_{\mathcal{C}}(j_3) = \sum_i \lambda_i M_{T_i}(j_3) \preccurlyeq \sum_i \lambda_i (M_{T_i}(j_1) + M_{T_i}(j_2)) = M_{\mathcal{C}}(j_1) + M_{\mathcal{C}}(j_2)$$

Thus, this proposition holds. □

Lemma 17 directly implies the following lemma:

**Lemma 18.** *Let $u, v$ be any two nodes in $G$, and let $P_{u,v}$ be an arbitrary acyclic path connecting $u$ and $v$ in $G$. For any $e \in P_{u,v}$, we denote the index of the pair of its endpoints by $j_e$. Then, $M_{\mathcal{C}}(j^*) \preccurlyeq \sum_{e \in P_{u,v}} M_{\mathcal{C}}(j_e)$, where $j^*$ represents the index of the node pair $\{u, v\}$.*

**Lemma 19.** *For any request set $\mathcal{R}$, let $\vec{l}_I^{\mathcal{R}}$ be the load vector corresponding to an arbitrary integral feasible solution. Then, $\vec{l}_{OBL_F}^{\mathcal{R}} \preccurlyeq M_{\mathcal{C}} \Upsilon \cdot \vec{l}_I^{\mathcal{R}}$.*

*Proof.* Suppose that $\vec{l}_I^{\mathcal{R}}$ is incurred by an integral routing algorithm $\Phi$. We then construct an $|E| \times \binom{|V|}{2}$-dimensional matrix $M_\Phi$ of Boolean variables such that $M_\Phi(i, j) = 1$ iff the traffic request between the $j$-th node pair will be routed by $\Phi$ along the $i$-th edge. Then, $M_\Phi \cdot \sum_{R_k \in \mathcal{R}} \vec{d_k} \preccurlyeq \vec{l}_I^{\mathcal{R}}$. Since [15] indicates that $\vec{l}_{\mathrm{OBL}_F}^{\mathcal{R}} = M_{\mathcal{C}} \cdot \sum_{R_k \in \mathcal{R}} \vec{d_k}$, we can prove Lemma 19 by showing that $M_{\mathcal{C}} \preccurlyeq M_{\mathcal{C}} \Upsilon M_\Phi$.

For the sake of simplicity, let $L = \Upsilon M_\Phi$. According to the definition of $\Upsilon$ and $M_\Phi$, $L(i, j) \geq 1$ if the path specified by $\Phi$ between the $j$-th node pair uses the edge between the $i$-th node pair, and $L(i, j) = 0$ otherwise. Let $i_1^j, i_2^j, \cdots, i_K^j$ be the indices of non-zero elements in $L(j)$, and $(M_{\mathcal{C}} L)(j)$ be the $j$-th column of $M_{\mathcal{C}} L$. Then,

$$(M_{\mathcal{C}} L)(j) \;=\; M_{\mathcal{C}} \cdot L(j) \;=\; \sum_{k=1}^{K} M_{\mathcal{C}}(i_k^j) \cdot L(i_k^j, j) \;\succcurlyeq\; \sum_{k=1}^{K} M_{\mathcal{C}}(i_k^j)$$

According to Lemma 18, $M_{\mathcal{C}}(j) \preccurlyeq \sum_k M_{\mathcal{C}}(i_k^j)$. Thus, $M_{\mathcal{C}}(j) \preccurlyeq (M_{\mathcal{C}} L)(j)$. □

**Lemma 20.** *For any $\vec{l}_I^{\mathcal{R}}$, $\|\vec{l}_{OBL_F}^{\mathcal{R}}\|_p \leq \|M_{\mathcal{C}} \Upsilon\|_p \cdot \|\vec{l}_I^{\mathcal{R}}\|_p$.*

*Proof.* From Lemma 19, we know that $\|\vec{l}_{\mathrm{OBL}_F}^{\mathcal{R}}\|_p \leq \|M_{\mathcal{C}} \Upsilon \cdot \vec{l}_I^{\mathcal{R}}\|_p$. According to Definition 5 of induced norm,

$$\frac{\|\vec{l}_{\mathrm{OBL}_F}^{\mathcal{R}}\|_p}{\|\vec{l}_I^{\mathcal{R}}\|_p} \leq \frac{\|M_{\mathcal{C}} \Upsilon \cdot \vec{l}_I^{\mathcal{R}}\|_p}{\|\vec{l}_I^{\mathcal{R}}\|_p} \leq \max_{\|\vec{l}\| > 0} \frac{\|M_{\mathcal{C}} \Upsilon \cdot \vec{l}\|_p}{\|\vec{l}\|_p} = \|M_{\mathcal{C}} \Upsilon\|_p$$

Thus, this lemma follows. □

Based on the results above, now we can prove our key result from this section:

**Theorem 21.** *The Rolling Dice Procedure can guarantee that the competitive ratio is bounded by $2^{\alpha+1} B_\alpha (\lceil \log_2 D \rceil + 1)^{\alpha - 1} \cdot \max \left\{ \|M_{\mathcal{C}} \Upsilon\|_1, \|M_{\mathcal{C}} \Upsilon\|_\alpha^\alpha \right\}$, where $D = \max_k d_k$.*

*Proof.* Here, we construct an exponentially discrete request set $\widehat{\mathbb{R}}$. Specifically, for each request $R_k \in \mathcal{R}$, there exists a corresponding request $\widehat{R}_k \in \widehat{\mathbb{R}}$ such that $\{\hat{s}_k, \hat{t}_k\} = \{s_k, t_k\}$ and $\hat{d}_k = 2^{\lceil \log_2 d_k \rceil}$, where $\{\hat{s}_k, \hat{t}_k\}$ represents the source-target pair of $\widehat{R}_k$ and $\hat{d}_k$ represents the demand of $\widehat{R}_k$. According to the definition of oblivious routing, the probability of routing $\widehat{R}_k$ along any edge $e$ is equivalent to the probability that $R_k$ goes through $e$, since $R_k$ and $\widehat{R}_k$ have the same source-target pair. Furthermore, as $\hat{d}_k \geq d_k$, we have $\mathbb{E}\left[\|\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\|_\alpha^\alpha\right] \geq \mathbb{E}\left[\|\vec{l}^{\mathcal{R}}_{\mathrm{OBL}_I}\|_\alpha^\alpha\right]$. Thus, the competitive ratio can be bounded by $\mathbb{E}\left[\|\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\|_\alpha^\alpha\right] \Big/ \mathrm{OPT}_I(\mathcal{R})$.

The request set $\widehat{\mathbb{R}}$ can be divided into a sequence of subsets $\widehat{R^1}, \cdots, \widehat{R^j}, \cdots$ such that $\widehat{R^j} = \{\widehat{R}_k \mid \widehat{R}_k \in \widehat{\mathbb{R}} \wedge \hat{d}_k = 2^j\}$, for each $j \in [0, \lceil \log_2 D \rceil]$. Applying Lemma 8, we have:

$$\begin{aligned}\mathbb{E}\Big[\Big(\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\Big(e, \widehat{R^j}\Big)\Big)^\alpha\Big] &\leq B_\alpha \max\Big\{\Big(\mathbb{E}\Big[\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\Big(e, \widehat{R^j}\Big)\Big]\Big)^\alpha, \left(2^j\right)^{\alpha-1} \mathbb{E}\Big[\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\Big(e, \widehat{R^j}\Big)\Big]\Big\} \\ &\leq B_\alpha \max\Big\{\Big(\vec{l}^{\widehat{R^j}}_{\mathrm{OBL}_F}(e)\Big)^\alpha,\ \left(2^j\right)^{\alpha-1} \cdot \vec{l}^{\widehat{R^j}}_{\mathrm{OBL}_F}(e)\Big\}\end{aligned}$$

The second inequality above follows from Lemma 16. For notational convenience, in the following, we will use $\gamma_F^j(e)$ to represent $\vec{l}^{\widehat{R^j}}_{\mathrm{OBL}_F}(e)$. Then, for each $e \in E$:

$$\begin{aligned}\mathbb{E}\left[\left(\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}(e)\right)^\alpha\right] &= \mathbb{E}\Big[\Big(\sum_{j=0}^{\lceil \log_2 D\rceil} \vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\Big(e, \widehat{R^j}\Big)\Big)^\alpha\Big] \\ &\leq \mathbb{E}\Big[(\lceil \log_2 D\rceil + 1)^{\alpha-1} \sum_{j=0}^{\lceil \log_2 D\rceil} \Big(\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\Big(e, \widehat{R^j}\Big)\Big)^\alpha\Big] \\ &\leq (\lceil \log_2 D\rceil + 1)^{\alpha-1} \sum_{j=0}^{\lceil \log_2 D\rceil} \mathbb{E}\left[\Big(\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\Big(e, \widehat{R^j}\Big)\Big)^\alpha\right] \\ &\leq B_\alpha(\lceil \log_2 D\rceil + 1)^{\alpha-1} \sum_{j=0}^{\lceil \log_2 D\rceil} \max\Big\{(\gamma_F^j(e))^\alpha, \left(2^j\right)^{\alpha-1} \gamma_F^j(e)\Big\}\end{aligned}$$

The first inequality above is based on the convexity of the power function [6]. We can now analyze the upper bound on the overall cost:

$$\begin{aligned}\mathbb{E}\Big[\|\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\|_\alpha^\alpha\Big] &\leq \sum_{e\in E} B_\alpha(\lceil \log_2 D\rceil + 1)^{\alpha-1} \sum_j \max\Big\{(\gamma_F^j(e))^\alpha, (2^j)^{\alpha-1}\gamma_F^j(e)\Big\} \\ &= B_\alpha(\lceil \log_2 D\rceil + 1)^{\alpha-1} \sum_j \sum_{e\in E} \max\Big\{(\gamma_F^j(e))^\alpha, (2^j)^{\alpha-1}\gamma_F^j(e)\Big\}\end{aligned}$$

For any two sequences of non-negative numbers $\{a_1, a_2, \cdots, a_N\}$ and $\{b_1, b_2, \cdots, b_N\}$, it is easy to show that $\sum_{i=1}^N \max\{a_i, b_i\} \leq \sum_{i=1}^N (a_i + b_i) \leq 2 \cdot \max\{\sum_{i=1}^N a_i, \sum_{i=1}^N b_i\}$.

Thus,

$$
\begin{aligned}
& \sum_{e\in E} \max\left\{\left(\gamma_F^j(e)\right)^\alpha, (2^j)^{\alpha-1}\gamma_F^j(e)\right\} \\
\le\ & 2\max\left\{\sum_{e\in E}\left(\gamma_F^j(e)\right)^\alpha, (2^j)^{\alpha-1}\sum_{e\in E}\gamma_F^j(e)\right\} \\
=\ & 2\max\left\{\left\|\vec{l}^{\widehat{R^j}}_{\mathrm{OBL}_F}\right\|_\alpha^\alpha,\ (2^j)^{\alpha-1}\left\|\vec{l}^{\widehat{R^j}}_{\mathrm{OBL}_F}\right\|_1\right\} \\
\le\ & 2\max\left\{\left(\left\|M_{\mathcal{C}}\Upsilon\right\|_\alpha\left\|\vec{l}^{\widehat{R^j}}_{\mathrm{OPT}_I}\right\|_\alpha\right)^\alpha, (2^j)^{\alpha-1}\left\|M_{\mathcal{C}}\Upsilon\right\|_1\left\|\vec{l}^{\widehat{R^j}}_{\mathrm{OPT}_I}\right\|_1\right\} \\
\le\ & 2\max\left\{\|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha,\ \|M_{\mathcal{C}}\Upsilon\|_1\right\}\cdot\left\|\vec{l}^{\widehat{R^j}}_{\mathrm{OPT}_I}\right\|_\alpha^\alpha
\end{aligned}
$$

The second inequality above follows from Lemma 20. Due to the integral constraint, $\vec{l}^{\widehat{R^j}}_{\mathrm{OPT}_I}(e)$ must be an integer multiple of $2^j$. In this case, $\left(\vec{l}^{\widehat{R^j}}_{\mathrm{OPT}_I}(e)\right)^\alpha \ge (2^j)^{\alpha-1}\vec{l}^{\widehat{R^j}}_{\mathrm{OPT}_I}(e)$ for each $e\in E$, which implies the fourth inequality above. In summary:

$$
\begin{aligned}
\mathbb{E}\left[\left\|\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OBL}_I}\right\|_\alpha^\alpha\right] &\le B_\alpha(\lceil\log_2 D\rceil+1)^{\alpha-1}2\max\left\{\|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha, \|M_{\mathcal{C}}\Upsilon\|_1\right\}\sum_j\left\|\vec{l}^{\widehat{R^j}}_{\mathrm{OPT}_I}\right\|_\alpha^\alpha \\
&\le 2B_\alpha(\lceil\log_2 D\rceil+1)^{\alpha-1}\max\left\{\|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha, \|M_{\mathcal{C}}\Upsilon\|_1\right\}\left\|\vec{l}^{\widehat{\mathbb{R}}}_{\mathrm{OPT}_I}\right\|_\alpha^\alpha \\
&\le 2^{\alpha+1}B_\alpha(\lceil\log_2 D\rceil+1)^{\alpha-1}\max\left\{\|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha, \|M_{\mathcal{C}}\Upsilon\|_1\right\}\left\|\vec{l}^{\mathcal{R}}_{\mathrm{OPT}_I}\right\|_\alpha^\alpha \\
&= 2^{\alpha+1}B_\alpha(\lceil\log_2 D\rceil+1)^{\alpha-1}\max\left\{\|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha, \|M_{\mathcal{C}}\Upsilon\|_1\right\}\mathrm{OPT}_I(\mathcal{R})
\end{aligned}
$$

The second inequality follows from the superadditivity of the power function [6]. The third inequality holds because $\hat{d}_k \le 2\cdot d_k$ for each $R_k$. Thus, this theorem follows. □

Recall that $\alpha$ is a constant parameter. Thus, we have:

**Corollary 22.** *Given any convex combination $\mathcal{C}$ of decomposition trees, the Rolling Dice Procedure according to $\mathcal{C}$ has a competitive ratio of* $O\Big(\max\{\|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha, \|M_{\mathcal{C}}\Upsilon\|_1\}\cdot\log^{\alpha-1}D\Big)$.

## 6. Minimizing Induced Norms

We have reduced the routing problem to the problem of simultaneously minimizing $\|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha$ and $\|M_{\mathcal{C}}\Upsilon\|_1$. The following theorem gives a lower bound on $\max\{\|M_{\mathcal{C}}\Upsilon\|_1, \|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha\}$.

**Theorem 23.** *There exists a network $G(V,E)$ for which no algorithm can compute a convex combination $\mathcal{C}$ of decomposition trees such that* $\max\{\|M_{\mathcal{C}}\Upsilon\|_1, \|M_{\mathcal{C}}\Upsilon\|_\alpha^\alpha\}$ *is bounded by* $o(|E|^{\frac{\alpha-1}{\alpha+1}})$.

*Proof.* Here, we consider the network $G_4$ constructed in Theorem 13. Suppose that there exists a $\mathcal{C}'$ for $G_4$ such that $\max\{\|M_{\mathcal{C}'}\Upsilon\|_1, \|M_{\mathcal{C}'}\Upsilon\|_\alpha^\alpha\} = o(|E|^{\frac{\alpha-1}{\alpha+1}})$. As in Theorem 13, we use $\mathrm{OPT}_I^p(\mathcal{R})$ to represent the cost of integrally routing $\mathcal{R}$ that is optimal with respect to the objective of minimizing $\sum_e (l_e)^p$. According to Lemma 20,

$$\max_{p\in\{1,\alpha\}} \max_{\mathcal{R}} \frac{\|\vec{l}^{\mathcal{R}}_{\mathrm{OBL}_F}\|_p^p}{\mathrm{OPT}_I^p(\mathcal{R})} \;\le\; \max_{p\in\{1,\alpha\}} \|M_{\mathcal{C}'}\Upsilon\|_p^p \;=\; o(|E|^{\frac{\alpha-1}{\alpha+1}})$$

which conflicts with Eq. (4). $\square$

We now prove that the convex combination obtained in the Precomputation Phase, $\mathcal{C}^*$, can minimize $\max\{\|M_\mathcal{C}\Upsilon\|_1, \|M_\mathcal{C}\Upsilon\|_\alpha^\alpha\}$ up to $O(\log^{\frac{2\alpha}{\alpha+1}} |E|)$.

**Lemma 24.** *For any $n \times m$-dimensional matrix $A$, any $p \ge 1$ and $q \ge 1$, $\|A\|_p \le m^{|\frac{1}{p}-\frac{1}{q}|} \cdot \|A\|_q$.*

*Proof.* Let $\vec{x}$ be an arbitrary $m$-dimensional vector. According to the theory of linear algebra, for any $p' > q'$:

$$\|\vec{x}\|_{p'} \;\le\; \|\vec{x}\|_{q'} \;\le\; \|\vec{x}\|_{p'} \cdot m^{\frac{1}{q'}-\frac{1}{p'}} \tag{8}$$

Let $\vec{x}^*$ be an $m$-dimensional vector such that $\frac{\|A\vec{x}^*\|_p}{\|\vec{x}^*\|_p} = \|A\|_p$. We then analyze two cases:

1. $p > q$. According to Eq. (8), $\|A\vec{x}^*\|_p \le \|A\vec{x}^*\|_q$ and $\|\vec{x}^*\|_p \ge m^{\frac{1}{p}-\frac{1}{q}} \cdot \|\vec{x}^*\|_q$. Thus,

$$\|A\|_p = \frac{\|A\vec{x}^*\|_p}{\|\vec{x}^*\|_p} \le \frac{\|A\vec{x}^*\|_q}{m^{\frac{1}{p}-\frac{1}{q}} \cdot \|\vec{x}^*\|_q} \le m^{\frac{1}{q}-\frac{1}{p}} \max_{\|\vec{x}\|_q \neq 0} \frac{\|A\vec{x}\|_q}{\|\vec{x}\|_q} = m^{\frac{1}{q}-\frac{1}{p}} \cdot \|A\|_q$$

2. $p \le q$. In this case, $\|A\vec{x}^*\|_p \le m^{\frac{1}{p}-\frac{1}{q}} \|A\vec{x}^*\|_q$ and $\|\vec{x}^*\|_p \ge \|\vec{x}^*\|_q$. Then,

$$\|A\|_p = \frac{\|A\vec{x}^*\|_p}{\|\vec{x}^*\|_p} \le \frac{m^{\frac{1}{p}-\frac{1}{q}} \cdot \|A\vec{x}^*\|_q}{\|\vec{x}^*\|_q} \le m^{\frac{1}{p}-\frac{1}{q}} \max_{\|\vec{x}\|_q \neq 0} \frac{\|A\vec{x}\|_q}{\|\vec{x}\|_q} = m^{\frac{1}{p}-\frac{1}{q}} \cdot \|A\|_q$$

Hence, $\|A\|_p \le m^{|\frac{1}{p}-\frac{1}{q}|} \cdot \|A\|_q$. $\square$

We can now state our key result regarding the simultaneous minimization of the powers of the induced norms of the tree-based matrix.

**Theorem 25.** $\max\{\|M_{\mathcal{C}^*}\Upsilon\|_1, \|M_{\mathcal{C}^*}\Upsilon\|_\alpha^\alpha\} \le \max\left\{(c_0 \log_2 |V|)^\alpha, (c_0 \log_2 |V|)^{\frac{2\alpha}{\alpha+1}} \cdot |E|^{\frac{\alpha-1}{\alpha+1}}\right\}$.

*Proof.* According to Eq. (7), we consider two cases in the following :

1. $(c_0 \cdot \log_2 |V|)^\alpha \geq |E|^{1-\frac{1}{\alpha}}(c_0 \cdot \log_2 |V|)$. In this case, $\|M_{\mathcal{C}^*}\Upsilon\|_\alpha \leq c_0 \log_2 |V|$. According to Lemma 24,

$$\|M_{\mathcal{C}^*}\Upsilon\|_1 \quad \leq \quad |E|^{1-\frac{1}{\alpha}} c_0 \log_2 |V| \quad \leq \quad (c_0 \log_2 |V|)^\alpha \tag{9}$$

and $\|M_{\mathcal{C}^*}\Upsilon\|_\alpha^\alpha \leq (c_0 \log_2 |V|)^\alpha$. Thus, in this case, $\max\{\|M_{\mathcal{C}^*}\Upsilon\|_1, \|M_{\mathcal{C}^*}\Upsilon\|_\alpha^\alpha\}$ $\leq (c_0 \log_2 |V|)^\alpha$.

2. $(c_0 \cdot \log_2 |V|)^\alpha < |E|^{1-\frac{1}{\alpha}}(c_0 \cdot \log_2 |V|)$. Since $\alpha - 1 > 0$,

$$\log_2(c_0 \log_2 |V|) \quad < \quad \frac{1}{\alpha} \log_2 |E| \tag{10}$$

Eq. (6) indicates that in this case, $\|M_{\mathcal{C}^*}\Upsilon\|_\chi \leq c_0 \log_2 |V|$. According to Eq. (7) and Eq. (10),

$$\chi = \frac{\alpha + 1}{2 - (\alpha - 1)\log_2(c_0 \log_2 |V|)/\log_2 |E|} < \frac{\alpha + 1}{2 - (\alpha - 1)/\alpha} < \alpha \tag{11}$$

According to Lemma 24, we have

$$\begin{aligned}
\|M_{\mathcal{C}^*}\Upsilon\|_\alpha^\alpha &\leq |E|^{\frac{\alpha}{\chi}-1}\|M_{\mathcal{C}^*}\|_\chi^\alpha \\
&\leq |E|^{\frac{\alpha-1}{\alpha+1} - \frac{\alpha(\alpha-1)\log_2(c_0 \log_2 |V|)}{(\alpha+1)\log_2 |E|}}(c_0 \log_2 |V|)^\alpha \\
&= |E|^{\frac{\alpha-1}{\alpha+1}} \cdot \left(|E|^{\frac{\log_2(c_0 \log_2 |V|)}{\log_2 |E|}}\right)^{-\frac{\alpha(\alpha-1)}{\alpha+1}}(c_0 \log_2 |V|)^\alpha \\
&= |E|^{\frac{\alpha-1}{\alpha+1}}(c_0 \log_2 |V|)^{\alpha - \frac{\alpha(\alpha-1)}{\alpha+1}} \\
&= |E|^{\frac{\alpha-1}{\alpha+1}}(c_0 \log_2 |V|)^{\frac{2\alpha}{\alpha+1}}
\end{aligned} \tag{12}$$

As mentioned in Section 4, for any non-trivial input case where $|V| \geq 2$ and $|E| \geq 1$, $\frac{\alpha+1}{2-\frac{\log_2(c_0 \log_2 |V|)}{\log_2 |E|}} \geq 1$. According to Lemma 24,

$$\begin{aligned}
\|M_{\mathcal{C}^*}\Upsilon\|_1 &\leq |E|^{1-\frac{1}{\chi}} \cdot \|M_{\mathcal{C}^*}\Upsilon\|_\chi \\
&= |E|^{\frac{\alpha-1}{\alpha+1} + \frac{(\alpha-1)\log_2(c_0 \log_2 |V|)}{(\alpha+1)\log_2 |E|}}(c_0 \log_2 |V|) \\
&= |E|^{\frac{\alpha-1}{\alpha+1}}(c_0 \log_2 |V|)^{\frac{2\alpha}{\alpha+1}}
\end{aligned} \tag{13}$$

Thus, this theorem follows. □

Since $\alpha$ is a constant parameter, the result of Theorem 25 can be bounded by $O\left(|E|^{\frac{\alpha-1}{\alpha+1}} \log^{\frac{2\alpha}{\alpha+1}} |V|\right)$. According to Theorem 23, this bound is tight up to $O\left(\log^{\frac{2\alpha}{\alpha+1}} |V|\right)$.

By combining Theorem 25 with Corollary 22, Theorem 1 is proved.

### 6.1. Function-oblivious

We now prove that our ROI-Routing algorithm is function-oblivious. Consider the case where each edge $e \in E$ is associated with a cost function $f_p(l_e) = (l_e)^p$, where $p$ is an arbitrary unknown number in $[1, \alpha]$. In such a case, a function-oblivious routing algorithm needs to guarantee a uniform upper bound on the competitive ratios corresponding to every possible $p$.

**Lemma 26** (Riesz-Thorin interpolation theorem [34, 36])**.** *For any $p, q$ which satisfy that $1 \le p < q \le \infty$, let $\theta$ be a number in $[0, 1]$ such that $\frac{1}{p} = \theta + \frac{1-\theta}{q}$. Then, $\|A\|_p \le \|A\|_1^\theta \cdot \|A\|_q^{1-\theta}$.*

**Lemma 27.** *For any $p \in [1, \alpha]$, $\|M_{\mathcal{C}^*}\Upsilon\|_p^p \le \max\left\{(c_0 \log_2 |V|)^\alpha, |E|^{\frac{\alpha-1}{\alpha+1}} \cdot (c_0 \log_2 |V|)^{\frac{2\alpha}{\alpha+1}}\right\}$.*

*Proof.* Let $\beta = \max\left\{c_0 \log_2 |V|, |E|^{\frac{\alpha-1}{\alpha(\alpha+1)}}(c_0 \log_2 |V|)^{\frac{2}{\alpha+1}}\right\}$. Equations (9), (12), and (13) indicate that $\|M_{\mathcal{C}^*}\Upsilon\|_\alpha \le \beta$ while $\|M_{\mathcal{C}^*}\Upsilon\|_1 \le \beta^\alpha$. According to Lemma 26,

$$\|M_{\mathcal{C}^*}\Upsilon\|_p^p \le (\beta^{\alpha\cdot\theta} \cdot \beta^{1-\theta})^p = \left[\beta^{\frac{\alpha(\alpha-p)}{p(\alpha-1)}} \cdot \beta^{1-\frac{\alpha-p}{p(\alpha-1)}}\right]^p = \beta^\alpha$$

Plugging the value of $\beta$ into the equation above, this proof is completed. $\square$

It is easy to see that Theorem 21 holds for the case where the objective is to minimize $\|\vec{l}\|_1$. Combining it with Lemma 27, we obtain:

**Theorem 28.** *For any $p \in [1, \alpha]$, our algorithm has a competitive ratio of $2^{p+1}B_p(\lceil \log_2 D\rceil + 1)^{p-1} \cdot \max\left\{(c_0 \log_2 |V|)^\alpha, |E|^{\frac{\alpha-1}{\alpha+1}}(c_0 \log_2 |V|)^{\frac{2\alpha}{\alpha+1}}\right\}$ with respect to the cost function $\sum_e (l_e)^p$, where $B_p$ is the fractional Bell number with parameter $p$.*

Since $p \le \alpha$, we can infer Theorem 3 directly from Theorem 28. According to Theorem 13, the result of Theorem 28 is tight up to $O\left(\log^{\alpha-1} D \cdot \log^{\frac{2\alpha}{\alpha+1}} |V|\right)$.

## 7. Extension and Application

The theoretical results proposed in the previous sections can be further extended to a framework of generating new oblivious integral routing algorithms and evaluating their performance. In this part, we will show that such a framework is significant for some specific application scenarios of reducing network energy consumption.

*7.1. A Generalized Framework for Oblivious Integral Routing*

Formally, our results on the Rolling Dice Procedure in Section 5 can be generalized as follows. Let $\Psi_F$ be an arbitrary oblivious fractional routing algorithm that operates deterministically, and let $M_{\Psi_F}$ be an $|E| \times \binom{|V|}{2}$-dimensional matrix, the $j$-th column of which is the load vector incurred by using $\Psi_F$ to route a traffic request with unit demand between the $j$-th pair of nodes in the given network $G$. In the following, $M_{\Psi_F}$ will be called the *routing matrix* of $\Psi_F$. We say that $M_{\Psi_F}$ is *path-additive* if for any node pair $\{u, v\}$ and any acyclic path $P_{u,v}$ between $u$ and $v$, $M_{\Psi_F}(j_{u,v}) \preccurlyeq \sum_{e \in P_{u,v}} M_{\Psi_F}(j_e)$, where $j_{u,v}$ represents the index of the node pair $\{u, v\}$, and $j_e$ represents the index of the node pair containing the endpoints of each link $e$ (i.e., the pair of nodes adjacent to $e_k$). For an oblivious integral routing algorithm $\Psi_I$ that operates in a randomized manner, we say that it follows a routing matrix $M_{\Psi_F}$ iff the probability that $\Psi_I$ routes a traffic request between the $j$-th node pair along $e_i$ is equivalent to $M_{\Psi_F}(i, j)$; additionally, $\Psi_I$ is said to be uncoupled iff for any edge $e$ and any two traffic requests $R_{k_1}, R_{k_2}$, the event that $\Psi_I$ routes $R_{k_1}$ along $e$ is stochastically independent of the event that $R_{k_2}$ is routed by $\Psi_I$ along $e$. Then,

**Theorem 29.** *The competitive ratio of an oblivious integral routing algorithm $\Psi_I$ that operates randomly has an $O(\max\{\|M_{\Psi_F}\Upsilon\|_1, \|M_{\Psi_F}\Upsilon\|_\alpha^\alpha\} \log^{\alpha-1} D)$-bound if $\Psi_I$ is uncoupled and follows a path-additive routing matrix $M_{\Psi_F}$.*

*Proof.* To prove this theorem, we now respectively transform Lemma 16 and Lemma 20 to the propositions that hold for $\Psi_I$ and $\Psi_F$:

- For any set $\mathcal{R}$ of traffic requests and any subset $\mathcal{R}' \subseteq \mathcal{R}$, let $\vec{l}^{\mathcal{R}}_{\Psi_I}(e, \mathcal{R}')$ be the part of the load on edge $e$ corresponding to $\mathcal{R}'$ when every traffic request in $\mathcal{R}$ is routed by $\Psi_I$. Recall that we use $\sigma(k)$ to represent the index of the source-target pair of the traffic request $R_k$. Since $\Psi_I$ follows $M_{\Psi_F}$, we have $\mathbb{E}\left[\vec{l}^{\mathcal{R}}_{\Psi_I}(e, \mathcal{R}')\right] = \sum_{R_k \in \mathcal{R}'} d_k \cdot M_{\Psi_F}(e, \sigma(k))$. According to the definition of oblivious routing, $\sum_{R_k \in \mathcal{R}'} d_k \cdot M_{\Psi_F}(e, \sigma(k)) = \vec{l}^{\mathcal{R}'}_{\Psi_F}(e)$, where $\vec{l}^{\mathcal{R}'}_{\Psi_F}$ represents the load vector incurred by fractionally routing $\mathcal{R}'$ with $\Psi_F$. Therefore, similar to Lemma 16, we have

$$\mathbb{E}\left[\vec{l}^{\mathcal{R}}_{\Psi_I}(e, \mathcal{R}')\right] = \vec{l}^{\mathcal{R}'}_{\Psi_F}(e) \tag{14}$$

- Let $\vec{l}^{\mathcal{R}}_{\Psi_F}$ and $\vec{l}^{\mathcal{R}}_{\Phi_I}$ respectively be load vectors incurred by routing $\mathcal{R}$ through $\Psi_F$ and an arbitrary integral routing algorithm $\Phi_I$. Now, we show that, similarly to Lemma 20, we have

$$\|\vec{l}^{\mathcal{R}}_{\Psi_F}\|_p \leq \|M_{\Psi_F}\Upsilon\|_p \cdot \|\vec{l}^{\mathcal{R}}_{\Phi_I}\|_p \tag{15}$$

  The key observation here is that Lemma 20 can be directly inferred from Lemma 19, which only depends on the fact that $\vec{l}^{\mathcal{R}}_{\mathrm{OBL}_F} = M_{\mathcal{C}} \cdot \sum_{R_k \in \mathcal{R}} \vec{d}_k$

and Lemma 18. Correspondingly, it can be derived from the definition of oblivious routing that for any edge $e$,

$$\vec{l}^{\mathcal{R}}_{\Psi_F}(e) \; = \sum_{R_k \in \mathcal{R}} M_{\Psi_F}(e, \sigma(k)) \cdot d_k \; = \sum_{R_k \in \mathcal{R}} [M_{\Psi_F}]_e \cdot \vec{d}_k$$

which implies that $\vec{l}^{\mathcal{R}}_{\Psi_F} = M_{\Psi_F} \cdot \sum_{R_k \in \mathcal{R}} \vec{d}_k$ holds. Furthermore, it is easy to see that Lemma 18 holds for $M_{\Psi_F}$ since we assume that $M_{\Psi_F}$ is path-additive. Thus, we can prove Eq. (15) in a similar manner to the proofs of Lemma 19 and Lemma 20.

Then this theorem can be established in a similar manner to Theorem 21. The only differences are that we need to replace Lemma 16 and Lemma 20 with the above two results, Eq. (14) and Eq. (15), respectively. Note that here we can still use Lemma 8 as the proof of Theorem 21 because $\Psi_I$ is assumed to be uncoupled. □

We remark that Theorem 29 is more general than the results of Section 5 since it is independent of any information on the actual operations of $\Psi_F$ and $\Psi_I$. It provides us a three-step framework for generating new oblivious integral routing algorithms of MPR and for evaluating their performance as follows:

1. Finding a deterministic fractional oblivious routing algorithm $\Psi_F$ with a path-additive routing matrix, and identifying $M_{\Psi_F}$.
2. Turning $M_{\Psi_F}$ into an integral routing algorithm with probabilistic tools. Although our Rolling Dice Procedure can successfully transform $M_{\mathcal{C}}$ into an oblivious integral routing algorithm (i.e., ROI-Routing), it cannot be applied to a general scenario due to its dependence on the existence of the convex combination of decomposition trees. To enhance the usability of our framework, in the following we provide a procedure that can convert any routing matrix $M_{\Psi_F}$ to an oblivious integral routing algorithm, independently of any data structure used by $\Psi_F$.
   The conversion procedure provided here is based on the Raghavan-Thompson (abbrv. R-T) flow decomposition approach [33]. Given a unit flow $\mathtt{H}(u, v)$ between any node pair $\{u, v\}$, the R-T flow decomposition approach can decompose it into at most $|E|$ weighted paths $\Pi = \{\pi_1, \pi_2, \cdots, \pi_i, \cdots\}$ connecting $u$ and $v$ in $O(|E|^2)$ time. Each path $\pi_i$ is associated with a positive weight $\lambda_i$, such that $\sum_{i=1}^{|\Pi|} \lambda_i = 1$ and for any edge $e \in E$, $\sum_{i:e\in\pi_i} \lambda_i$ is equivalent to the part of $H(u, v)$ along $e$.
   We can then generate an oblivious routing algorithm $\Psi_I$ that operates as follows. Let $\Pi_j$ be the set of weighted paths obtained by decomposing the flow identified by $M_{\Psi_F}(j)$ with the R-T flow decomposition approach. For a traffic request $R_k$, $\Psi_I$ will select a path $\pi_k$ from $\Pi_{\sigma(k)}$ independently in a randomized manner, and will route $R_k$ along $\pi_k$. The probability that each $\pi_i \in \Psi_{\sigma(k)}$ is selected will be $\lambda_i$. Obviously, $\Psi_I$ is the routing algorithm we need, since it is uncoupled and follows the routing matrix $M_{\Psi_F}$. Similar to ROI-Routing, the number of random bits used by $\Psi_I$ for each traffic request is also bounded by $O(\log |E|)$ since $|\Pi| \leq |E|$.

3. Find an upper bound $\beta$ on $\max\{\|M_{\Psi_F}\Upsilon\|_1, \|M_{\Psi_F}\Upsilon\|_\alpha^\alpha\}$. Then we can claim that the competitive ratio of $\Psi_I$ is bounded by $O(\beta \log^{\alpha-1} D)$.

According to Theorem 1 and Theorem 2, designing any new oblivious integral routing algorithm cannot help us to significantly improve our results for MPR with the general settings defined in Section 1. However, in some specific application scenarios arising from practice, the input instances have some special properties that can simplify the problem. For these instances, it is possible to achieve a much better upper bound on the competitive ratio through our framework. To show this, in the following we consider a special class of input instances where the network has well-bounded *edge expansion* and node degree, and we design new algorithms using our framework to improve our results for these instances.

*7.2. Expansion and Electric Flow-based Oblivious Fractional Routing*

For a network $G(V,E)$, let $S$ be a non-empty subset of $V$, and $\partial(S)$ be the number of edges with exactly one endpoint in $S$. The *edge expansion* (also called the isoperimetric number in the literature) of $G$ is defined as:

$$h(G) \;=\; \min_{S:|S|\le|V|/2} \frac{\partial(S)}{|S|} \tag{16}$$

The significance of the parameter $h(G)$ is that it can be used to measure the connectivity of the network, i.e., a large edge expansion implies high connectivity [23]. Let $\vartheta(v)$ be the degree of a node $v \in V$, and $\vartheta(G) = \max_{v\in V} \vartheta(v)$, which will be referred to as the maximum node degree of $G$. For any connected network $G$, we have $\frac{2}{|V|} \le h(G) \le \vartheta(G)$.

Note that in the definition of MPR given in Section 1, we make no assumption on any property of network's topology, including the edge expansion, since the networks in the general environment can have an arbitrary topology, especially the national backbone networks [7]. However, in some specific scenarios such as the *data center*, we only need to focus on a regular network topology instead of an arbitrary one. Typically, the topology of the data center network (abbrv. DCN) is designed to have high connectivity (e.g., [2]), which implies a well-bounded edge expansion.

Our approach to achieve a better result on the networks $G$ with well-bounded $h(G)$ will utilize a routing strategy $\Psi_F^{\mathcal{E}}$ based on the electrical flow. Specifically, $\Psi_F^{\mathcal{E}}$ associates a unit resistance to every edge $e \in E$. Let $I_{u,v}(e)$ be the current along the edge $e$ when a unit current flows into $u$ and out of $v$; $\Psi_F^{\mathcal{E}}$ will carry out each traffic request $R_k$ by scaling up $I_{s_k,t_k}(e)$ by a factor of $d_k$ for every $e$. For instance, in Figure. 5 we show a simple network $G_6(V_6, E_6)$ with a corresponding electricity network obtained by associating a resistance of 1 Ohm to each edge in $E_6$. According to Kirchhoff's laws and Ohm's law, if a unit of current flows into $v_1$ and out of $v_2$, the current along the edges $e_1, e_2, e_3$ and $e_4$ will respectively be $0.75A$, $0.25A$, $0.25A$ and $0.25A$, where $A$ represents "ampere". Therefore, for a traffic request $R_k$ with $\{s_k, t_k\} = \{v_1, v_2\}$ and $d_k = 2$, $\Psi_F^{\mathcal{E}}$ will route a flow

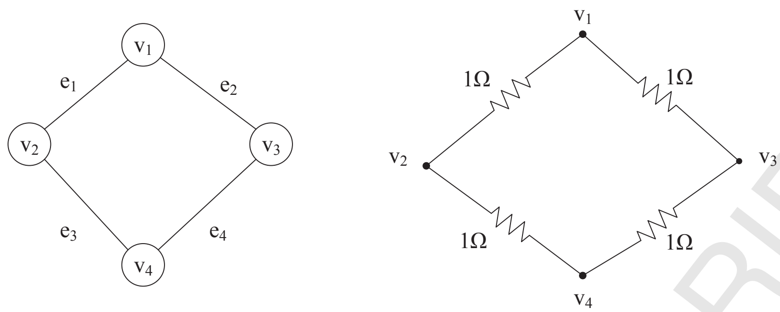



Figure 5: A 4-node 4-edge network $G_6$ with its corresponding electricity network.

of 1.5 along the path $\{e_1\}$ and a flow of 0.5 along the path $\{e_2, e_3, e_4\}$. This indicates that $\Psi_F^{\mathcal{E}}$ cannot satisfy the integral constraint.

The routing matrix $M_{\Psi_F^{\mathcal{E}}}$ of $\Psi_F^{\mathcal{E}}$ can be identified as follows. Similarly to [26], we first designate a direction to every edge such that each $e$ orients from the node with a smaller index to the node with a larger index. Let $\vec{\mathcal{I}}_{u,v}$ be a vector of $|E|$ elements such that for any link $e \in E$, $\vec{\mathcal{I}}_{u,v}(e) = I_{u,v}(e)$ if the current $I_{u,v}(e)$ has the same direction as $e$ and $\vec{\mathcal{I}}_{u,v}(e) = -I_{u,v}(e)$ otherwise. For example, for the network $G_6$ in Fig. 5, the edge $e_3$ will be designated a direction from $v_2$ to $v_4$. Then we have $\vec{\mathcal{I}}_{v_1,v_2}(e_3) = -0.25$, since if a unit current flows into $v_1$ and out of $v_2$, a current of 0.25 Ampere will flow from $v_4$ to $v_2$.

For any vector/matrix $X$, let $X^{\mathrm{tr}}$ be the transpose of $X$. Then, according to Kirchhoff's current law,

$$B^{\mathrm{tr}} \cdot \vec{\mathcal{I}}_{u,v} \;=\; \vec{\delta}_u - \vec{\delta}_v \tag{17}$$

where $\vec{\delta}_x$ is a $|V|$-dimensional vector whose $y$-th element is the Kronecker delta function $\delta(x, y)$, and $B$ is an $|E| \times |V|$-dimensional matrix such that for each $e \in E$, $[B]_e = (\vec{\delta}_u - \vec{\delta}_v)^{\mathrm{tr}}$ if $e$ directs from $u$ to $v$. Let $\vec{\phi}_{u,v}$ be a vector of $|V|$ elements such that $\vec{\phi}_{u,v}(k)$ is the electric potential of the node $k$ when a unit current flows into $u$ and out of $v$. Then, Ohm's law indicates that:

$$B \cdot \vec{\phi}_{u,v} \;=\; \vec{\mathcal{I}}_{u,v} \tag{18}$$

Combining Eq. (17) with Eq. (18), we have $\vec{\mathcal{I}}_{u,v} = B(B^{\mathrm{tr}}B)^{+}(\vec{\delta}_u - \vec{\delta}_v)$ [26], where $(B^{\mathrm{tr}}B)^{+}$ represents the pseudoinverse of $B^{\mathrm{tr}}B$. According to the uniqueness principle [14], such a pseudoinverse is unique. Let $\{u_j, v_j\}$ be the $j$-th node pair. Then for the routing matrix $M_{\Psi_F^{\mathcal{E}}}$, we have

$$M_{\Psi_F^{\mathcal{E}}} \;=\; \mathrm{abs}\Big(\bigotimes_j \vec{\mathcal{I}}_{u_j,v_j}\Big) \;=\; \mathrm{abs}\Big(\bigotimes_j \Big[B(B^{\mathrm{tr}}B)^{+}(\vec{\delta}_{u_j} - \vec{\delta}_{v_j})\Big]\Big) \tag{19}$$

where the symbol $\bigotimes$ represents the direct product, and abs is an operator that takes the absolute value of every element in a matrix. Due to the operator abs, the value of any entry in $M_{\Psi_F^{\mathcal{E}}}$ will not be influenced by exchanging the positions of $\vec{\delta}_{u_j}$ and $\vec{\delta}_{v_j}$ in Eq. (19). Therefore, without loss of generality, we assume that the index of $u_j$ is smaller than that of $v_j$.

The following property of $B^{\text{tr}}B$ can help us associate the performance of $\Psi_F^{\mathcal{E}}$ with the edge expansion of network $G$:

**Lemma 30** ([26])**.** $\|(B^{\text{tr}}B)^{+}\|_1 \ \leq \ \left(4\ln\frac{|V|}{2}\right)\cdot\left[h(G)\cdot\ln\frac{2\vartheta(G)}{2\vartheta(G)-h(G)}\right]^{-1}$

*7.3. New Oblivious Integral Routing Strategy*

Using our framework, in this part we develop a new oblivious integral routing strategy $\Psi_I^{\mathcal{E}}$ based on $M_{\Psi_F^{\mathcal{E}}}$, and prove that compared with ROI-Routing, $\Psi_I^{\mathcal{E}}$ can guarantee a better upper bound on the competitive ratio for the input cases where the network $G$ owns a large edge expansion $h(G)$ and a small maximum node degree $\vartheta(G)$. We first prove that the matrix $M_{\Psi_F^{\mathcal{E}}}$ is path-additive.

**Lemma 31** ([27])**.** *For any three nodes $v_{j_1}, v_{j_2}, v_{j_3} \in V$, $\vec{\mathcal{I}}_{v_{j_1},v_{j_2}} + \vec{\mathcal{I}}_{v_{j_2},v_{j_3}} = \vec{\mathcal{I}}_{v_{j_1},v_{j_3}}$.*

**Lemma 32.** *For any three pairs of nodes $\{u_{j_1}, v_{j_1}\}$, $\{u_{j_2}, v_{j_2}\}$ and $\{u_{j_3}, v_{j_3}\}$ with indices $j_1$, $j_2$ and $j_3$, $M_{\Psi_F^{\mathcal{E}}}(j_3) \preccurlyeq M_{\Psi_F^{\mathcal{E}}}(j_1)+M_{\Psi_F^{\mathcal{E}}}(j_2)$ when $\{u_{j_1}, v_{j_1})\bigcap\{u_{j_2}, v_{j_2}\} \neq \emptyset$ and $\{u_{j_3}, v_{j_3}\} \in \{u_{j_1}, v_{j_1}\}\bigcup\{u_{j_2}, v_{j_2}\}$.*

*Proof.* Since $\{u_{j_1}, v_{j_1}\} \neq \{u_{j_2}, v_{j_2}\}$ and $\{u_{j_1}, v_{j_1})\bigcap\{u_{j_2}, v_{j_2}\} \neq \emptyset$, without loss of generality, we assume that $u_{j_2} = v_{j_1}$, which implies that $\{u_{j_3}, v_{j_3}\} = \{u_{j_1}, v_{j_2}\}$. According to the definition of $M_{\Psi_F^{\mathcal{E}}}$, we have:

$$\begin{aligned}
M_{\Psi_F^{\mathcal{E}}}(j_1) + M_{\Psi_F^{\mathcal{E}}}(j_2) \ &= \ \text{abs}\left(\vec{\mathcal{I}}_{u_{j_1},v_{j_1}}\right) + \text{abs}\left(\vec{\mathcal{I}}_{u_{j_2},v_{j_2}}\right) \\
&= \ \text{abs}\left(\vec{\mathcal{I}}_{u_{j_1},v_{j_1}}\right) + \text{abs}\left(\vec{\mathcal{I}}_{v_{j_1},v_{j_2}}\right) \\
&\succcurlyeq \ \text{abs}\left(\vec{\mathcal{I}}_{u_{j_1},v_{j_1}} + \vec{\mathcal{I}}_{v_{j_1},v_{j_2}}\right) \\
&= \ \text{abs}\left(\vec{\mathcal{I}}_{u_{j_1},v_{j_2}}\right) \\
&= M_{\Psi_F^{\mathcal{E}}}(j_3)
\end{aligned}$$

The third equality above follows from Lemma 31. □

By inductively applying Lemma 32, it can be proved that:

**Lemma 33.** *For any node pair $\{u, v\}$ and any acyclic path $P_{u,v}$ between $u$ and $v$, $M_{\Psi_F}^{\mathcal{E}}(j_{u,v}) \preccurlyeq \sum_{e\in P_{u,v}} M_{\Psi_F}^{\mathcal{E}}(j_e)$.*

Lemma 33 implies that we have found an oblivious fractional routing algorithm with a path-additive routing matrix. Then with the R-T flow decomposition approach, we can convert $\Psi_F^{\mathcal{E}}$ to an oblivious integral routing algorithm $\Psi_I^{\mathcal{E}}$ that is uncoupled and follows $M_{\Psi_F^{\mathcal{E}}}$. According to our framework, we now need to analyze the upper bound on $\max\{\|M_{\Psi_F^{\mathcal{E}}}\Upsilon\|_1, \|M_{\Psi_F^{\mathcal{E}}}\Upsilon\|_\alpha^\alpha\}$.

**Lemma 34** ([29]). *For any $m \times n$-dimensional matrix $A$, $\|A\|_1 = \max_{j=1}^{n} \sum_{i=1}^{m} |A(i,j)|$, and $\|A\|_\infty = \max_{i=1}^{m} \sum_{j=1}^{n} |A(i,j)|$.*

**Lemma 35.** $\|M_{\Psi_F^{\mathcal{E}}} \Upsilon\|_1 = \|B(B^{\mathrm{tr}}B)^+B^{\mathrm{tr}}\|_1$.

*Proof.* Let $\mathcal{B} = \bigotimes_j [B(B^{\mathrm{tr}}B)^+(\vec{\delta}_{u_j} - \vec{\delta}_{v_j})]$. According to Eq. (19), $\|M_{\Psi_F^{\mathcal{E}}} \Upsilon\|_1 = \|\operatorname{abs}(\mathcal{B})\Upsilon\|_1$. Then the $j$-th element in the $i$-row of $\operatorname{abs}(\mathcal{B})\Upsilon$ will be $\sum_k |\mathcal{B}(i,k)| \cdot \Upsilon(k,j)$. Let $\varsigma(j)$ be the index of the node pair containing the endpoints of the $j$-th edge. According to the definition of the column selector $\Upsilon$, there will be only one non-zero element $\Upsilon(\varsigma(j),j) = 1$ in $\Upsilon(j)$. Then we have $\sum_k |\mathcal{B}(i,k)| \cdot \Upsilon(k,j) = |\mathcal{B}(i,\varsigma(j))| = |\sum_k \mathcal{B}(i,k) \cdot \Upsilon(k,j)|$. Therefore,

$$\|M_{\Psi_F^{\mathcal{E}}} \Upsilon\|_1 \;=\; \|\operatorname{abs}(\mathcal{B})\Upsilon\|_1 \;=\; \|\operatorname{abs}(\mathcal{B}\Upsilon)\|_1 \;=\; \|\mathcal{B}\Upsilon\|_1$$

where the last equality follows from Lemma 34. By expanding $\mathcal{B}$ in the above formulation, we have:

$$\|M_{\Psi_F^{\mathcal{E}}} \Upsilon\|_1 = \left\| \bigotimes_j \Big[B(B^{\mathrm{tr}}B)^+(\vec{\delta}_{u_j} - \vec{\delta}_{v_j})\Big] \Upsilon \right\|_1 = \left\| B(B^{\mathrm{tr}}B)^+ \Big[\bigotimes_j (\vec{\delta}_{u_j} - \vec{\delta}_{v_j})\Big] \Upsilon \right\|_1$$

Let $\widetilde{\mathcal{B}} = [\bigotimes_j (\vec{\delta}_{u_j} - \vec{\delta}_{v_j})]\Upsilon$. Then:

$$\widetilde{\mathcal{B}}(i,z) \;=\; \sum_k (\vec{\delta}_{u_k}(i) - \vec{\delta}_{v_k}(i))\Upsilon(k,z) \;=\; \vec{\delta}_{u_{\varsigma(z)}}(i) - \vec{\delta}_{v_{\varsigma(z)}}(i)$$

The last equality holds since for each $z$, the only non-zero $\Upsilon(k,z)$ is $\Upsilon(\varsigma(z),z) = 1$. Recall that without loss of generality, we assume the index of $u_{\varsigma(z)}$ is smaller than the index of $v_{\varsigma(z)}$. Then:

$$\widetilde{\mathcal{B}}(i,z) = \vec{\delta}_{u_{\varsigma(z)}}(i) - \vec{\delta}_{v_{\varsigma(z)}}(i) = \begin{cases} 1 & \text{if the } z\text{-th edge directs from the } i\text{-th node} \\ -1 & \text{if the } z\text{-th edge directs to the } i\text{-th node} \\ 0 & \text{otherwise} \end{cases}$$

Thus, $\widetilde{\mathcal{B}} = B^{\mathrm{tr}}$. This completes the proof. □

It can be proved in a similar way that:

**Lemma 36.** $\|M_{\Psi_F^{\mathcal{E}}} \Upsilon\|_\infty = \|B(B^{\mathrm{tr}}B)^+B^{\mathrm{tr}}\|_\infty$.

**Lemma 37.** $\|B(B^{\mathrm{tr}}B)^+B^{\mathrm{tr}}\|_1 \le \left(8\vartheta(G)\ln\frac{|V|}{2}\right) \cdot \left[h(G) \cdot \ln\frac{2\vartheta(G)}{2\vartheta(G)-h(G)}\right]^{-1}$

*Proof.* We first give the upper bound on $\|B(B^{\mathrm{tr}}B)^+\|_1$. Let $\widehat{\mathcal{B}} = (B^{\mathrm{tr}}B)^+$. Then the sum of the absolute values of all the elements in the $j$-th column of $B\widehat{\mathcal{B}}$ will be:

$$\sum_i |B_i \cdot \widehat{\mathcal{B}}(j)| \;=\; \sum_i \sum_k |B(i,k) \cdot \widehat{\mathcal{B}}(k,j)| \;=\; \sum_k \left| \sum_i B(i,k) \right| \cdot |\widehat{\mathcal{B}}(k,j)|$$

Note that in the column $B(k)$, there are at most $\vartheta(G)$ non-zero elements in $\{-1,1\}$, each of which corresponds to an edge adjacent to the $i$-th node. Therefore, $-\vartheta(G) \le \sum_i B(i,k) \le \vartheta(G)$. According to Lemma 34, we have:

$$
\begin{aligned}
\|B(B^{\mathrm{tr}}B)^{+}\|_1 &= \max_j \sum_i \Big|\sum_k B(i,k)\widehat{\mathcal{B}}(k,j)\Big| \\
&\le \max_j \sum_k \Big(|\widehat{\mathcal{B}}(k,j)| \cdot \sum_i |B(i,k)|\Big) \\
&\le \vartheta(G) \cdot \max_j \sum_k |\widehat{\mathcal{B}}(k,j)| \\
&= \vartheta(G)\|\widehat{\mathcal{B}}\|_1
\end{aligned}
$$

The gap between $\|B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}\|_1$ and $\|B(B^{\mathrm{tr}}B)^{+}\|_1$ can be estimated in a similar manner. In each column of $B^{\mathrm{tr}}$, there will be only two non-zero elements, which are 1 and $-1$. This implies that a column of $B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}$ will be the difference between two columns in $B(B^{\mathrm{tr}}B)^{+}$. Let $\overline{\mathcal{B}} = B(B^{\mathrm{tr}}B)^{+}$, then:

$$
\begin{aligned}
\|B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}\|_1 &\le \max_{j_1,j_2:j_1\neq j_2} \sum_i |\overline{\mathcal{B}}(i,j_1) - \overline{\mathcal{B}}(i,j_2)| \\
&\le \max_{j_1,j_2:j_1\neq j_2} \sum_i (|\overline{\mathcal{B}}(i,j_1)| + |\overline{\mathcal{B}}(i,j_2)|) \\
&\le \max_{j_1} \sum_i |\overline{\mathcal{B}}(i,j_1)| + \max_{j_2} \sum_i |\overline{\mathcal{B}}(i,j_2)| \\
&\le 2\|\overline{\mathcal{B}}\|_1
\end{aligned}
$$

According to Lemma 30, this lemma follows. □

**Lemma 38.** $\|B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}\|_\infty = \|B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}\|_1$.

*Proof.* According to Lemma 34, for any matrix $A$, $\|A\|_1 = \|A\|_\infty$ if $A$ is symmetric. In the following, we will prove that the matrix $B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}$ is symmetric by showing that it is equivalent to its transpose:

$$
\Big[B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}\Big]^{\mathrm{tr}} = (B^{\mathrm{tr}})^{\mathrm{tr}}\Big[(B^{\mathrm{tr}}B)^{+}\Big]^{\mathrm{tr}}B^{\mathrm{tr}} = B\Big[(B^{\mathrm{tr}}B)^{\mathrm{tr}}\Big]^{+}B^{\mathrm{tr}} = B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}
$$

where the second equality follows from the property of pseudoinversion. □

**Theorem 39.** *We have* $\|M_{\Psi_F^{\varepsilon}}\Upsilon\|_p^p \le \Big[\Big(8\vartheta(G)\ln\frac{|V|}{2}\Big)\Big/\Big[h(G)\ln\frac{2\vartheta(G)}{2\vartheta(G)-h(G)}\Big]\Big]^p$ *for any* $p \ge 1$.

*Proof.* When $p=1$, this theorem trivially follows from Lemma 35 and Lemma 37. Now we consider the case where $p>1$. Lemma 26 indicates that:

$$
\begin{aligned}
\|M_{\Psi_F^{\varepsilon}}\Upsilon\|_p^p &\le \Big(\Big\|M_{\Psi_F^{\varepsilon}}\Upsilon\Big\|_1^{\frac{1}{p}} \cdot \Big\|M_{\Psi_F^{\varepsilon}}\Upsilon\Big\|_\infty^{\frac{p-1}{p}}\Big)^p \\
&= \Big\|B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}\Big\|_1 \cdot \Big\|B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}\Big\|_\infty^{p-1} \\
&= \Big\|B(B^{\mathrm{tr}}B)^{+}B^{\mathrm{tr}}\Big\|_1^p
\end{aligned}
$$

where the first equality follows from Lemma 35 and Lemma 36, and the last equality follows from Lemma 38. Then this theorem follows from Lemma 37. □

According to Theorem 29, the competitive ratio of $\Psi_I^{\mathcal{E}}$ for MPR can be bounded by $O\Big(\Big[\Big(\vartheta(G)\log|V|\Big)\cdot\Big(h(G)\log\frac{2\vartheta(G)}{2\vartheta(G)-h(G)}\Big)^{-1}\Big]^{\alpha}\cdot\log^{\alpha-1}D\Big)$. To see how $\Psi_I^{\mathcal{E}}$ improves the result of ROI-Routing on networks with well-bounded edge expansions and node degrees, here we first consider a class of networks called *expanders*, which has a large variety of applications in computer science [23]. A network $G_{\mathrm{EX}}$ is said to be an expander if its maximum node degree $\vartheta(G_{\mathrm{EX}})$ has a constant upper bound and its edge expansion $h(G_{\mathrm{EX}})$ has a constant lower bound. According to Theorem 29 and Theorem 39, we have:

**Corollary 40.** *The algorithm $\Psi_I^{\mathcal{E}}$ can guarantee that the competitive ratio is bounded by $O(\log^{\alpha}|V|\cdot\log^{\alpha-1}D)$ on expanders $G_{EX}$.*

Another class of networks considered here for illustration are the *hypercubes* $G_{\mathrm{HC}}$. A hypercube $G_{\mathrm{HC}}$ contains $2^n$ nodes, each of which has a label of $n$-bit binary digits. Any two nodes $u, v$ in $G_{\mathrm{HC}}$ are connected iff their labels differ in exactly one digit. This implies that $\vartheta(G_{\mathrm{HC}}) = \log_2|V|$. Moreover, it can be inferred from *Cheeger's inequality* [4] that $h(G_{\mathrm{HC}}) = 1$ [40]. Then we have:

**Corollary 41.** *The competitive ratio of the algorithm $\Psi_I^{\mathcal{E}}$ can be bounded by $O(\log^{3\alpha}|V|\cdot\log^{\alpha-1}D)$ on hypercubes $G_{HC}$.*

*Proof.* From Theorem 29 and Theorem 39, we can infer that the competitive ratio of $\Psi_I^{\mathcal{E}}$ for MPR on hypercubes can be bounded by $O\Big(\Big[\Big(\vartheta(G_{\mathrm{HC}})\ln|V|\Big)\Big(h(G_{\mathrm{HC}})\log\frac{2\vartheta(G_{\mathrm{HC}})}{2\vartheta(G_{\mathrm{HC}})-h(G_{\mathrm{HC}})}\Big)^{-1}\Big]^{\alpha}\log^{\alpha-1}D\Big) = O\Big(\Big[\log^2|V|\Big(\log\frac{2\log|V|}{2\log|V|-1}\Big)^{-1}\Big]^{\alpha}\log^{\alpha-1}D\Big)$. Then, we need to reduce $\Big(\log\frac{2\log|V|}{2\log|V|-1}\Big)^{-1}$ to a simplified form. Since:

$$\begin{aligned}
2^{2\log_2|V|\cdot\log_2\frac{2\log_2|V|}{2\log_2|V|-1}} &= \Big(\frac{2\log_2|V|}{2\log_2|V|-1}\Big)^{2\log_2|V|} \\
&= \Big[\Big(1-\frac{1}{2\log_2|V|}\Big)^{2\log_2|V|}\Big]^{-1} \\
&\geq \exp(1)
\end{aligned}$$

we have $2\log_2|V|\cdot\log_2\frac{2\log_2|V|}{2\log_2|V|-1} \geq \log_2(\exp(1))$, which means

$$\Big(\log_2\frac{2\log_2|V|}{2\log_2|V|-1}\Big)^{-1} \leq \frac{2}{\log_2(\exp(1))}\log_2|V|$$

Therefore, the competitive ratio can be bounded by $O\Big(\log^{3\alpha}|V|\cdot\log^{\alpha-1}D\Big)$. □

To sum up, on both expanders and hypercubes, the upper bound on the competitive ratio of the algorithm $\Psi_I^{\mathcal{E}}$ is better than the $O\Big(|E|^{\frac{\alpha-1}{\alpha+1}}\log^{\frac{2\alpha}{\alpha+1}}|V|\cdot\log^{\alpha-1}D\Big)$-bound guaranteed by ROI-Routing.

*7.4. Combination*

We have shown that the algorithm $\Psi_I^{\mathcal{E}}$ can guarantee a polylogarithmic competitive ratio on the networks with special topologies. However, such a good result does not hold for every possible network. Formally, we have:

**Theorem 42.** *Any oblivious integral routing algorithm $\Phi_I'$ following $M_{\Psi_F^{\mathcal{E}}}$ cannot guarantee an $o\left(|E|^{\frac{1}{2}\max\{1,\alpha-1\}}\right)$-bound on the competitive ratio for every network.*

**Remark.** *Note that this theorem only requires that $\Phi_I'$ follows $M_{\Psi_F^{\mathcal{E}}}$, but makes no assumption on whether $\Phi_I'$ is uncoupled or not.*

*Proof.* We construct a network $G_6(V_6, E_6)$ in a similar manner to $G_2$ in Fig. 2. The only difference is that in $G_6$, $\Delta = \tau = \lfloor(|E_6| - 1)^{1/2}\rfloor$. Let the node pair in $G_6$ which corresponds to $\{u_2, v_2\}$ in $G_2$ be $\{u_6, v_6\}$. By Ohm's law and Kirchhoff's integral theorem, when a unit current flows into $u_6$ and out of $v_6$:

- There is no current in the $(|E_6| - \Delta\tau)$-node ring attached to $u_6$, if such a ring exists.
- The amount of current flowing across the short canonical path is 1/2.

Consider the case where there is only one traffic request $R_1$ between $(u_6, v_6)$ with $d_1 = 1$. The optimal cost of routing $R_1$ will then be 1. However, $\Phi_I'$ will route $R_1$ along one of the long canonical paths with probability 1/2, which will incur an expected cost of $\tau/2$. Another case here is that there are $\Delta$ traffic requests $R_1, \cdots, R_\Delta$ between $(u_6, v_6)$ with $d_k = 1$. Routing them with $\Phi_I'$ will burden the short canonical path with an expected load of $\Delta/2$. According to Lemma 5, the expectation of the cost incurred by $\Phi_I'$ will be at least $(\Delta/2)^\alpha$. By contrast, the strategy of routing each traffic request along a distinct long canonical path accrues a cost of $\Delta\tau$. Thus, the competitive ratio of $\Phi_I'$ will be at least $\max\left\{\frac{\tau}{2}, \frac{(\Delta/2)^\alpha}{\Delta\tau}\right\}$. Plugging the values of $\Delta$ and $\tau$ in terms of $|E_6|$ into this equation completes this proof. □

The difference between Theorem 2 and Theorem 42 indicates that, when we take every possible network topology into consideration, there exists a gap of $\Omega\left(|E|^{\frac{1}{6}\max\{1,\alpha-1\}}\right)$ between the competitive ratios of the best possible oblivious integral routing algorithm and the algorithm $\Psi_I^{\mathcal{E}}$. By contrast, Theorem 1 indicates that the algorithm ROI-Routing can narrow this gap to $O(\log^{\frac{2\alpha}{\alpha+1}}|V| \cdot \log^{\alpha-1} D)$. An interesting problem is that considering how to guarantee a competitive ratio that is tight up to a polylogarithmic factor as well as ROI-Routing, while simultaneously preserving the advantages of $\Psi_I^{\mathcal{E}}$ on special networks, such as expanders and hypercubes.

Our approach to this issue is combining ROI-Routing with $\Psi_I^{\mathcal{E}}$. Corresponding to the first step of our framework, we first generate the matrices $M_{\mathcal{C}^*}$ and $M_{\Psi_F^{\mathcal{E}}}$ respectively with the Precomputation Phase defined in Section 4 and

Eq. (19), and then choose the one among $\{M_{\mathcal{C}^*}, M_{\Psi_F^\varepsilon}\}$ to minimize $F(\mathcal{M})$, where $F(\mathcal{M}) = \max\{\|\mathcal{M}\Upsilon\|_1, \|\mathcal{M}\Upsilon\|_\alpha^\alpha\}$ is a function defined over the set $\{M_{\mathcal{C}^*}, M_{\Psi_F^\varepsilon}\}$. The minimization of $F(\mathcal{M})$ requires the calculation of the exact values of the induced norms of $M_{\mathcal{C}^*}\Upsilon$ and $M_{\Psi_F^\varepsilon}\Upsilon$. In particular, $\|M_{\mathcal{C}^*}\Upsilon\|_1$ and $\|M_{\Psi_F^\varepsilon}\Upsilon\|_1$ can be identified through Lemma 34. We now show that we can approximate $\|M_{\mathcal{C}^*}\Upsilon\|_\alpha^\alpha$ and $\|M_{\Psi_F^\varepsilon}\Upsilon\|_\alpha^\alpha$ by a factor of $1-\varepsilon$ for any $0 < \varepsilon < 1$ in polynomial time. It has been proved in [11] that:

**Lemma 43** (Bhaskara-Vijayaraghavan's iteration algorithm [11])**.** *For any $\varepsilon \in (0,1)$ and any $n \times n$-dimensional matrix $A$ that only contains non-negative elements, there exists an iteration algorithm that can obtain an $n$-dimensional vector $\vec{x}$ satisfying $\frac{\|Ax\|_p}{\|x\|_p} \geq (1-\varepsilon)\|A\|_p$ in the time polynomial in $n$ and $\frac{1}{\varepsilon}$.*

Let $\vec{x}_{\mathcal{C}^*}$ and $\vec{x}_{\Psi_F^\varepsilon}$ be two $|E|$-dimensional vectors obtained by the Bhaskara-Vijayaraghavan's iteration algorithm such that $\frac{\|M_{\mathcal{C}^*}\Upsilon\vec{x}_{\mathcal{C}^*}\|_\alpha}{\|\vec{x}_{\mathcal{C}^*}\|_\alpha} \geq (1-\varepsilon)\|M_{\mathcal{C}^*}\Upsilon\|_\alpha$ and $\frac{\|M_{\Psi_F^\varepsilon}\Upsilon\vec{x}_{\Psi_F^\varepsilon}\|_\alpha}{\|\vec{x}_{\Psi_F^\varepsilon}\|_\alpha} \geq (1-\varepsilon)\|M_{\Psi_F^\varepsilon}\Upsilon\|_\alpha$ for some properly chosen constant $\varepsilon > 0$, and $\mathcal{M}^*$ be an $|E| \times \binom{|V|}{2}$-dimensional matrix defined as follows:

$$\mathcal{M}^* = \begin{cases} M_{\mathcal{C}^*} & \text{if } \max\{\|M_{\mathcal{C}^*}\Upsilon\|_1, \frac{\|M_{\mathcal{C}^*}\Upsilon\vec{x}_{\mathcal{C}^*}\|_\alpha^\alpha}{\|\vec{x}_{\mathcal{C}^*}\|_\alpha^\alpha}\} \leq \max\{\|M_{\Psi_F^\varepsilon}\Upsilon\|_1, \frac{\|M_{\Psi_F^\varepsilon}\Upsilon\vec{x}_{\Psi_F^\varepsilon}\|_\alpha^\alpha}{\|\vec{x}_{\Psi_F^\varepsilon}\|_\alpha^\alpha}\} \\ M_{\Psi_F^\varepsilon} & \text{otherwise} \end{cases} \tag{20}$$

Then we have:

**Lemma 44.** *$\mathcal{M}^*$ can minimize $F(\mathcal{M})$ up to a constant factor of $\left(\frac{1}{1-\varepsilon}\right)^\alpha$.*

*Proof.* Without loss of generality, here we assume that $F(M_{\mathcal{C}^*}) \leq F(M_{\Psi_F^\varepsilon})$. Then this lemma trivially holds when $\mathcal{M}^* = M_{\mathcal{C}^*}$. For the case where $\mathcal{M}^* = M_{\Psi_F^\varepsilon}$, we have:

$$\begin{aligned} F(\mathcal{M}^*) &\leq \left(\frac{1}{1-\varepsilon}\right)^\alpha \max\left\{\|M_{\Psi_F^\varepsilon}\Upsilon\|_1, \frac{\|M_{\Psi_F^\varepsilon}\Upsilon\vec{x}_{\Psi_F^\varepsilon}\|_\alpha^\alpha}{\|\vec{x}_{\Psi_F^\varepsilon}\|_\alpha^\alpha}\right\} \\ &\leq \left(\frac{1}{1-\varepsilon}\right)^\alpha \max\left\{\|M_{\mathcal{C}^*}\Upsilon\|_1, \frac{\|M_{\mathcal{C}^*}\Upsilon\vec{x}_{\mathcal{C}^*}\|_\alpha^\alpha}{\|\vec{x}_{\mathcal{C}^*}\|_\alpha^\alpha}\right\} \\ &\leq \left(\frac{1}{1-\varepsilon}\right)^\alpha \max\{\|M_{\mathcal{C}^*}\Upsilon\|_1, \|M_{\mathcal{C}^*}\Upsilon\|_\alpha^\alpha\} \end{aligned}$$

The first inequality follows from $\|\mathcal{M}^*\Upsilon\|_\alpha = \|M_{\Psi_F^\varepsilon}\Upsilon\|_\alpha \leq \frac{1}{1-\varepsilon}\frac{\|M_{\Psi_F^\varepsilon}\Upsilon\vec{x}_{\Psi_F^\varepsilon}\|_\alpha}{\|\vec{x}_{\Psi_F^\varepsilon}\|_\alpha}$. The second inequality follows from Eq. (20). The last one follows from Definition 5 of induced $L_p$-norm. Therefore, this lemma is established. $\square$

Then, we can apply the procedure given in the second step of our framework to generate an oblivious integral routing algorithm $\Psi_I^*$ that is uncoupled and follows $\mathcal{M}^*$. Combining Theorem 25, Theorem 29, Theorem 39 and Lemma 44 together, we have:

**Theorem 45.** *The competitive ratio of the algorithm $\Psi_I^*$ can be bounded by* $O\Big(\min\Big\{|E|^{\frac{\alpha-1}{\alpha+1}}\log^{\frac{2\alpha}{\alpha+1}}|V|, \Big[\Big(\vartheta(G)\log|V|\Big)\Big(h(G)\log\frac{2\vartheta(G)}{2\vartheta(G)-h(G)}\Big)^{-1}\Big]^{\alpha}\Big\}\log^{\alpha-1}D\Big)$.

Obviously, such a competitive ratio is tight up to a factor of $O(\log^{\frac{2\alpha}{\alpha+1}}|V|\cdot\log^{\alpha-1}D)$, and also has upper bounds $O(\log^{\alpha}|V|\cdot\log^{\alpha-1}D)$ and $O(\log^{3\alpha}|V|\cdot\log^{\alpha-1}D)$ on expanders and hypercubes, respectively. Furthermore, according to Lemma 27 and Theorem 39, this competitive ratio holds for any cost function $\|\vec{l}\|_p^p$ with $1\le p\le\alpha$, which means that the algorithm $\Psi_I^*$ also has the property of function-oblivious. In Appendix A, we will use the pseudocode to provide more details on the implementation of $\Psi_I^*$.

## 8. Conclusion

In this paper, we investigate the minimum power-cost routing (MPR) problem. It involves an undirected network $G(V,E)$ where each edge $e$ is associated with a superlinear cost function $f(l_e)=(l_e)^{\alpha}$ and a set of traffic requests $\mathcal{R}$, and requires the minimization of the cost of routing $\mathcal{R}$ in $G$. For this problem, we proposed an oblivious routing algorithm — ROI-Routing. The property of being oblivious to the network traffic enables ROI-Routing to be efficiently implemented in a distributed manner, which is significant for large-scale high-capacity networks.

Our research is different from related work on oblivious routing algorithms because ROI-Routing is designed for the unsplittable version of the MPR problem, where the integral constraint needs to be satisfied. Compared with the splittable version, the unsplittable version is closer to a real network configuration, but is more difficult to solve. Specifically, we proved that given the integral constraint, no randomized oblivious routing algorithm can yield a competitive ratio of $o(|E|^{\frac{\alpha-1}{\alpha+1}})$, whereas ROI-Routing can guarantee a competitive ratio of $O\Big(|E|^{\frac{\alpha-1}{\alpha+1}}\log^{\frac{2\alpha}{\alpha+1}}|V|\cdot\log^{\alpha-1}D\Big)$, which is tight up to a polylogarithmic factor $O\Big(\log^{\alpha-1}D\cdot\log^{\frac{2\alpha}{\alpha+1}}|V|\Big)$.

In addition to being oblivious to traffic, ROI-Routing has the property of being function-oblivious, which is essential for scenarios in which the precise value of the degree of the cost function is unavailable. We proved that for any $p\in[1,\alpha]$, ROI-Routing can guarantee a uniform upper bound of $O\Big(|E|^{\frac{\alpha-1}{\alpha+1}}\log^{\frac{2\alpha}{\alpha+1}}|V|\cdot\log^{\alpha-1}D\Big)$ on the competitive ratio for the case where the cost function is the $p$-th power of the load. This result was also proved to be tight up to a polylogarithmic factor $O\Big(\log^{\alpha-1}D\cdot\log^{\frac{2\alpha}{\alpha+1}}|V|\Big)$.

The theoretical results obtained in the analysis of ROI-Routing can be generalized to a framework that can help researchers design and analyze oblivious integral routing algorithms for specific input instances. To illustrate the significance of this framework, we apply it to generate routing algorithms $\Psi_I^{\mathcal{E}}$ and $\Psi_I^*$, which can guarantee a better competitive ratio than ROI-Routing on the networks with well-bounded maximum node degrees and edge expansions.

In particular, $\Psi_I^{\mathcal{E}}$ has a competitive ratio of $O\Big(\Big[\frac{\vartheta(G)\log|V|}{h(G)\log\frac{2\vartheta(G)}{2\vartheta(G)-h(G)}}\Big]^{\alpha}\log^{\alpha-1}D\Big)$ for MPR, which can be respectively bounded by $O(\log^{\alpha}|V|\cdot\log^{\alpha-1}D)$ and $O(\log^{3\alpha}|V|\cdot\log^{\alpha-1}D)$ on expanders and hypercubes. Another algorithm $\Psi_I^*$, which combines ROI-Routing with $\Psi_I^{\mathcal{E}}$, has a competitive ratio that is tight up to $O\Big(\log^{\alpha-1}D\cdot\log^{\frac{2\alpha}{\alpha+1}}|V|\Big)$ like ROI-Routing, while simultaneously having the same upper bounds as $\Psi_I^{\mathcal{E}}$ on the expanders and hypercubes.

An interesting problem is determining the competitive ratio that can be achieved by $\Psi_I^*$ on the emerging network topologies designed for data centers, including BCube [18], DCell [19], etc. This problem is challenging since it is not easy to bound the edge expansions of these network topologies. This will be the subject of our future work.

**Acknowledgements**

This research was supported in part by the China National Natural Science Foundation (NSFC) and Hong Kong RGC Joint Project grant No. 61161160566, NSFC International Coordination Project grant No. 61020106002, and NSFC Project for Innovation Groups grant No. 61221062.

## Appendix A. Description of Algorithm in Pseudocode

In Algorithm 1, we give the pseudocode of the algorithm $\Psi_I^*$. Particularly, three functions will be defined and implemented in this part:

- Get_Routing_Matrix: Corresponding to the first step of our framework, this function will generate the routing matrix $\mathcal{M}^*$ defined in Eq. (20) by respectively computing $M_{\mathcal{C}^*}$ and $M_{\Psi_F^*}$, and comparing their induced $L_p$-norms.
- Get_Candidate_Paths: Corresponding to the second step of our framework, this function takes a network $G$, a routing matrix $M$ and a node index $i$ as parameters, and converts $M$ to a series of integral paths that connect the $i$-th node $v_i$ to each $v_j$ with $j \neq i$. The input parameter $M$ has a default value $M_{\Psi_F^*}$. For each $v_j \neq v_i$, this function yields a path set $\Pi_j$ and a weight vector $\vec{\lambda}_j$, where each element $\vec{\lambda}_j(i)$ is the weight associated with the corresponding path $\Pi(i)$. The output is a hash table which stores every key-value pair $(j, [\Pi_j, \vec{\lambda}_j])$.
- Find_Path: Based on the hash table given by Get_Candidate_Paths, this function will select a path for a given traffic request $R_k$ in a randomized manner.

In Algorithm 1, we assume that the following functions are provided by external libraries:

- Calculate_Covex_Combination: It refers to the algorithm proposed in [11, 15] which takes a network $G$ and a real number $p > 1$ as input parameters and can output a convex combination $M_{\mathcal{C}^*}$ of decomposition trees and the corresponding matrix $M_{\mathcal{C}^*}$ such that $\|M_{\mathcal{C}}\Upsilon\|_p \leq c_0 \log_2 |V|$.
- Bhaskara_Vijayaraghavan_Iteration: It refers to Bhaskara-Vijayaraghavan's iteration algorithm proposed in [11], which takes a non-negative square matrix $A$, a real number $p > 1$ and a real number $\varepsilon > 0$ as the input parameters, and returns a $(1-\varepsilon)$-approximation of $\|A\|_p$.
- Raghavan_Thompson_Decomposition: It refers to the R-T flow decomposition algorithm proposed in [33]. Given a network $G$, an $|E|$-dimensional load vector, a source node $s$ and a target node $t$, this function will decompose it into a series $\Pi$ of weighted paths between $s$ and $t$.

Additionally, the following functions are assumed to be provided by the system:

- zeros: This function takes two integers $m, n$ as input parameters and outputs a $m \times n$-dimensional matrix which only contains zeros.
- Matrix_Transpose: It calculates the transpose of a given matrix.
- Matrix_Multiplication: It calculates the multiplication of two given matrices.
- pinv: This function will return a pseudoinverse of a given matrix.
- binomial: Calling binomial($n$, $k$) will get the binomial coefficient $\binom{n}{k}$.
- abs: This function will return the absolute value of each element in the input parameter.
- max: Returning the larger of two input parameters.
- new hashtable: This function will yield a new hashtable which is empty.
- index: Given a node $v$ in the network, this function will return the index of $v$ as an integer in $[1, |V|]$.
- random: This function returns a random number uniformly distributed in the interval $[0, 1]$.

**Algorithm 1** A full description of the algorithm $\Psi_I^*$, **Part 1**

**function** GET_ROUTING_MATRIX(Network $G$, Real $\alpha$, Real $\varepsilon$)
/* *First compute* $M_{\mathcal{C}^*}$ */
**if** $(c_0 \cdot \log_2 |V|)^\alpha \geq |E|^{1-\frac{1}{\alpha}}(c_0 \cdot \log_2 |V|)$ **then**
$\chi = \alpha$;
**else**
$\chi = (\alpha+1)\left[2-(\alpha-1)\frac{\log_2(c_0 \log_2 |V|)}{\log_2 |E|}\right]^{-1}$;
**end if**
$[\mathcal{C}^*, M_{\mathcal{C}^*}] =$ CALCULATE_COVEX_COMBINATION($G$, $\chi$);
/* *Proceed to compute* $M_{\Psi_F^\varepsilon}$ */
$B =$ ZEROS($|E|$, $|V|$)
**for** $k = 1 \to |E|$ **do**
**for** $i = 1 \to |V|$ **do**
**for** $j = i+1 \to |V|$ **do**
**if** $e_k$ connects $v_i$ and $v_j$ **then**
$B(k,i) = 1$; $B(k,j) = -1$;
**end if**
**end for**
**end for**
**end for**
$B^{\mathrm{tr}} =$ MATRIX_TRANSPOSE($B$);
tmp = MATRIX_MULTIPLICATION($B^{\mathrm{tr}}$, $B$);
$\widehat{\mathcal{B}} =$ PINV(tmp);
$\overline{\mathcal{B}} =$ MATRIX_MULTIPLICATION($B$, $\widehat{\mathcal{B}}$);
$N =$ BINOMIAL($|V|$, 2); $M_{\Psi_F^\varepsilon} =$ ZEROS($|E|$, $N$);
$k = 1$;
**for** $i = 1 \to |V|$ **do**
**for** $j = i+1 \to |V|$ **do**
$\vec{\delta}_i =$ ZERO($|V|$, 1); $\vec{\delta}_i(i) = 1$;
$\vec{\delta}_j =$ ZERO($|V|$, 1); $\vec{\delta}_j(j) = 1$;
$M_{\Psi_F^\varepsilon}(k) =$ MATRIX_MULTIPLICATION($\overline{\mathcal{B}}$, $\vec{\delta}_i - \vec{\delta}_j$);
$M_{\Psi_F^\varepsilon}(k) =$ ABS($M_{\Psi_F^\varepsilon}(k)$);
$k = k+1$;
**end for**
**end for**
/* *Start to to minimize* $F(\mathcal{M})$ */
$y_1 = y_2 = 0$;
$A_1 =$ MATRIX_MULTIPLICATION($M_{\mathcal{C}^*}$, $\Upsilon$);
$A_2 =$ MATRIX_MULTIPLICATION($M_{\Psi_F^\varepsilon}$, $\Upsilon$);
/* *Compute* $\|M_{\mathcal{C}^*}\Upsilon\|_1$ *and* $\|M_{\Psi_F^\varepsilon}\Upsilon\|_1$ */
**for** $j = 1 \to |E|$ **do**
$\mathrm{sum}_1 = \mathrm{sum}_2 = 0$;

**Algorithm 2** A full description of the algorithm $\Psi_I^*$, **Part 2**

        **for** $i = 1 \to |E|$ **do**
            $\text{sum}_1 = \text{sum}_1 + \textsc{abs}(A_1(i,j))$;
            $\text{sum}_2 = \text{sum}_2 + \textsc{abs}(A_2(i,j))$;
        **end for**
        $y_1 = \textsc{max}(y_1, \text{sum}_1)$;
        $y_2 = \textsc{max}(y_2, \text{sum}_2)$;
    **end for**
    tmp = $\textsc{Bhaskara\_Vijayaraghavan\_Iteration}(A_1, \alpha, \varepsilon)$;
    $y_1 = \textsc{max}(\text{tmp}, y_1)$;
    tmp = $\textsc{Bhaskara\_Vijayaraghavan\_Iteration}(A_2, \alpha, \varepsilon)$;
    $y_2 = \textsc{max}(\text{tmp}, y_2)$;
    **if** $y_1 \le y_2$ **then** $\mathcal{M}^* = M_{\mathcal{C}^*}$;
    **else** $\mathcal{M}^* = M_{\Psi_F^\varepsilon}$;
    **end if**
    **return** $\mathcal{M}^*$;
**end function**

**function** $\textsc{Get\_Candidate\_Paths}$(Network $G$, Integer $i$, Matrix $M = \mathcal{M}^*$)
    ans = $\textsc{new hashtable}(\ )$;
    **for** $j = 1 \to |V|$ **do**
        **if** $i \neq j$ **then**
            **if** $i < j$ **then**
                $k = (i-1) * |V| - i * (i-1)/2 + (j-i)$;
            **else**
                $k = (j-1) * |V| - j * (j-1)/2 + (i-j)$
            **end if**
            $[\Pi, \vec{\lambda}] = \textsc{Raghavan\_Thompson\_Decomposition}(G, M(k), \text{i}, \text{j})$;
            ans.$\textsc{add}(j, [\Pi, \vec{\lambda}])$;
        **end if**
    **end for**
    **return** ans;
    /* *The weight paths in* ans *will be stored in the routing table of* $v_i$ */
**end function**

**function** $\textsc{Find\_Path}$(Hashtables HT, Request $R_k$)
    /* HT *is a series of hashtables, where* $\text{HT}(i,j)$*stores the weighted paths between* $v_i$ *and* $v_j$. */
    $i = \textsc{index}(s_k)$; $j = \textsc{index}(t_k)$;
    $[\Pi, \vec{\lambda}] = HT(i,j)$;
    $r = \textsc{random}(\ )$;
    **for** $k = 1 \to$ **sizeof**$(\Pi)$ **do**
        **if** $\vec{\lambda}(k) \ge r$ **then**

**Algorithm 3** A full description of the algorithm $\Psi_I^*$, **Part 3**

```
            return Π(k);
        else
            r = r − λ⃗(k)
        end if
    end for
end function
```

**References**


[1] Vision and roadmap: Routing telecom and data centers toward efficient energy use, `http://www1.eere.energy.gov/industry/datacenters/pdfs/vision_and_roadmap.pdf` (May 2009).

[2] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, H. Liu, Energy proportional datacenter networks, in: ACM SIGARCH Computer Architecture News, vol. 38, ACM, 2010, pp. 338–347.

[3] R. Ahuja, T. Magnanti, J. Orlin, Network flows: theory, algorithms, and applications, chap. Multicommodity Flows, Prentice Hall, 1993, pp. 649–694.

[4] N. Alon, Eigenvalues and expanders, Combinatorica 6 (2) (1986) 83–96.

[5] M. Andrews, Approximation algorithms for the edge-disjoint paths problem via raecke decompositions, in: Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS '10, IEEE Computer Society, 2010, pp. 277–286.

[6] M. Andrews, A. F. Anta, L. Zhang, W. Zhao, Routing for power minimization in the speed scaling model, IEEE/ACM Transactions on Networking 20 (1) (2012) 285 –294.

[7] M. Andrews, L. Zhang, Bounds on fiber minimization in optical networks with fixed fiber capacity, in: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 1, 2005, pp. 409–419 vol. 1.

[8] E. Bampis, A. Kononov, D. Letsios, G. Lucarelli, M. Sviridenko, Energy efficient scheduling and routing via randomized rounding, in: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2013, December 12-14, 2013, Guwahati, India, pp. 449–460.

[9] Y. Bartal, Probabilistic approximation of metric spaces and its algorithmic applications, in: Proceedings of the 37th Annual Symposium on Foundations of Computer Science, FOCS '96, IEEE Computer Society, 1996.


ACCEPTED MANUSCRIPT

[10] D. Berend, T. Tassa, Improved bounds on bell numbers and on moments of sums of random variables, Probability and Mathematical Statistics 30 (2010) 185–205.

[11] A. Bhaskara, A. Vijayaraghavan, Approximating matrix p-norms, in: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '11, SIAM, 2011, pp. 497–511.

[12] D. Brooks, P. Bose, S. Schuster, H. Jacobson, P. Kudva, A. Buyuktosunoglu, J.-D. Wellman, V. Zyuban, M. Gupta, P. Cook, Power-aware microarchitecture: design and modeling challenges for next-generation microprocessors, IEEE Micro 20 (6) (2000) 26–44.

[13] G. Dobiński, Summirung der reihe $\sum n^m/n!$, für m= 1, 2, 3, 4, 5, Arch. für Mat. und Physik 61 (1877) 333–336.

[14] P. G. Doyle, J. L. Snell, Random walks and electric networks, Carus mathematical monographs 22.

[15] M. Englert, H. Räcke, Oblivious routing for the lp-norm, in: Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 32–40.

[16] J. Fakcharoenphol, S. Rao, K. Talwar, A tight bound on approximating arbitrary metrics by tree metrics, in: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, STOC '03, ACM, 2003, pp. 448–455.

[17] N. Goyal, N. Olver, F. Shepherd, Dynamic vs. oblivious routing in network design, in: Algorithms - ESA 2009, vol. 5757 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 277–288.

[18] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, S. Lu, Bcube: a high performance, server-centric network architecture for modular data centers, ACM SIGCOMM Computer Communication Review 39 (4) (2009) 63–74.

[19] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, S. Lu, Dcell: a scalable and fault-tolerant network structure for data centers, ACM SIGCOMM Computer Communication Review 38 (4) (2008) 75–86.

[20] A. Gupta, M. T. Hajiaghayi, H. Räcke, Oblivious network design, in: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, SODA '06, ACM, New York, NY, USA, 2006, pp. 970–979.

[21] A. Gupta, R. Krishnaswamy, K. Pruhs, Online primal-dual for non-linear optimization with applications to speed scaling, in: Approximation and Online Algorithms, vol. 7846 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 173–186.

[22] P. Harsha, T. P. Hayes, H. Narayanan, H. Räcke, J. Radhakrishnan, Minimizing average latency in oblivious routing, in: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms, SODA '08, Society for Industrial and Applied Mathematics, 2008, pp. 200–207.

[23] S. Hoory, N. Linial, A. Wigderson, Expander graphs and their applications, Bulletin of the American Mathematical Society 43 (4) (2006) 439–561.

[24] Intel, Enhanced intel speedstep technology for the intel pentium m processor, in: Intel White Paper 301170-001, 2004.

[25] J. Jensen, Sur les fonctions convexes et les inégalités entre les valeurs moyennes, Acta Mathematica 30 (1) (1906) 175–193.

[26] J. Kelner, P. Maymounkov, Electric routing and concurrent flow cutting, Theoretical Computer Science 412 (32) (2011) 4123–4135.

[27] G. Lawler, H. Narayanan, Mixing times and lp bounds for oblivious routing, in: Proceedings of the 5th Workshop on Analytic Algorithmics and Combinatorics (ANALCO), 2009, pp. 66–74.

[28] K. Makarychev, M. Sviridenko, Solving optimization problems with diseconomies of scale via decoupling, in: 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS '14), 2014.

[29] C. D. Meyer, Matrix analysis and applied linear algebra, chap. 5.2 Matrix norms, Society for Industrial and Applied Mathematics, 2000, pp. 279–285.

[30] R. Motwani, P. Raghavan, Randomized algorithms, chap. 4.2 Routing in a parallel computer, Chapman & Hall/CRC, 2010, pp. 74–79.

[31] H. Räcke, Optimal hierarchical decompositions for congestion minimization in networks, in: Proceedings of the 40th annual ACM symposium on Theory of computing, STOC '08, ACM, 2008, pp. 255–264.

[32] H. Räcke, Survey on oblivious routing strategies, in: Mathematical Theory and Computational Practice, vol. 5635 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 419–429.

[33] P. Raghavan, C. D. Tompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs, Combinatorica 7 (4) (1987) 365–374.

[34] M. Riesz, Sur les maxima des formes bilinéaires et sur les fonctionnelles linéaires, Acta Mathematica 49 (3-4) (1927) 465–497.

[35] Y. Shi, F. Zhang, Z. Liu, Oblivious integral routing for minimizing the quadratic polynomial cost, in: Frontiers in Algorithmics, vol. 8497 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 216–228.

[36] G. O. Thorin, Convexity theorems, generalizing those of m. riesz and hadamard: With some applications, Ph.D. thesis, Almqvist & Wiksells-boktr. (1948).

[37] R. S. Tucker, The role of optics and electronics in high-capacity routers, Journal of Lightwave Technology 24 (12) (2006) 4655–4673.

[38] L. G. Valiant, A scheme for fast parallel communication, SIAM Journal on Computing 11 (2) (1982) 350–361.

[39] L. G. Valiant, G. J. Brebner, Universal schemes for parallel communication, in: Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing, STOC '81, ACM, 1981, pp. 263–277.

[40] U. Vazirani, S. Rao, Sparse cuts and cheeger's inequality, `http://www.cs.berkeley.edu/~satishr/cs270/sp11/rough-notes/Sparse-cuts-spectra.pdf` (2011).

[41] A. Wierman, L. Andrew, A. Tang, Power-aware speed scaling in processor sharing systems, in: INFOCOM 2009, IEEE, 2009, pp. 2007 –2015.

[42] G. Zervas, M. De Leenheer, L. Sadeghioon, D. Klonidis, Y. Qin, R. Nejabati, D. Simeonidou, C. Develder, B. Dhoedt, P. Demeester, Multi-granular optical cross-connect: Design, analysis, and demonstration, IEEE/OSA Journal of Optical Communications and Networking 1 (1) (2009) 69–84.