



# Sparse Semi-Oblivious Routing

Few Random Paths Suffice

Bernhard Haeupler

ETH Zürich & CMU

Switzerland & USA

bernhard.haeupler@inf.ethz.ch

Antti Roeykskoe

ETH Zürich

Switzerland

antti.roeykskoe@inf.ethz.ch

Goran Zuzic

ETH Zürich

Switzerland

goran.zuzic@inf.ethz.ch

## ABSTRACT

The packet routing problem asks to select routing paths that minimize the maximum edge congestion for a set of packets specified by source-destination vertex pairs. We revisit a semi-oblivious approach to this problem: each source-destination pair is assigned a small set of well-chosen predefined paths before the demand is revealed, while the sending rates along the paths can be optimally adapted to the demand. This approach has been considered in practice in network traffic engineering due to its superior robustness and performance as compared to both oblivious routing and traditional traffic engineering approaches.

We show the existence of sparse semi-oblivious routings: only  $O(\log n)$  paths are selected between each pair of vertices. The routing is  $(\text{poly } \log n)$ -competitive for all demands against the offline-optimal congestion objective, even on worst-case graphs. Even for the well-studied case of hypercubes, no such result was known: our deterministic and oblivious selection of  $O(\log n)$  paths is the first simple construction of a deterministic oblivious structure that near-optimally assigns source-destination pairs to few routes. Prior work shows that a deterministic selection of a single path in a hypercube yields unacceptable performance; our results contrast the current solely-negative landscape of results for semi-oblivious routing. We give the sparsity-competitiveness trade-off for lower sparsities and nearly match it with a lower bound.

Our construction is extremely simple: Sample the few paths from any competitive oblivious routing. Indeed, this natural construction was used in traffic engineering as an unproven heuristic. We give a satisfactory theoretical justification for their empirical effectiveness: the competitiveness of the construction improves exponentially with the number of paths. In other words, semi-oblivious routing benefits from the power of random choices. Finally, when combined with the recent hop-constrained oblivious routing, we also obtain sparse and competitive structures for the completion-time objective.

## CCS CONCEPTS

• **Mathematics of computing** → Paths and connectivity problems; • **Theory of computation** → Distributed algorithms; Graph algorithms analysis; • **Networks** → Network algorithms.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PODC '23, June 19–23, 2023, Orlando, FL, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0121-4/23/06...\$15.00

<https://doi.org/10.1145/3583668.3594585>

## KEYWORDS

packet routing, semi-oblivious routing, oblivious routing

### ACM Reference Format:

Bernhard Haeupler, Antti Roeykskoe, and Goran Zuzic. 2023. Sparse Semi-Oblivious Routing: Few Random Paths Suffice. In *ACM Symposium on Principles of Distributed Computing (PODC '23)*, June 19–23, 2023, Orlando, FL, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3583668.3594585>

## 1 INTRODUCTION

Packet routing through a communication network is a fundamental task that is well-studied in both theoretical and practical contexts. We consider the following version of the task.

*The network is abstracted as an  $n$ -vertex undirected graph  $G = (V, E)$ . Initially, the network receives several packet delivery requests, where the  $i^{\text{th}}$  packet should be transmitted from source  $s_i \in V$  to destination  $t_i \in V$ . The goal is to select a path for each packet in a way that minimizes the maximum edge congestion, i.e., minimizes the maximum number of packets passing over any one edge.*

An offline version of this task is known as the multicommodity flow (MCF) problem, with different packets representing different commodities. However, solving MCF typically assumes knowledge of the entire set of packets upfront, a requirement that is often very restrictive. For this reason, a particularly appealing routing strategy is the so-called *oblivious routing*, where each packet is routed independently (i.e., obliviously) from other packets using a predefined policy while at the same time requiring that the lump of all traffic near-optimally utilizes the network. The line of work on oblivious routings culminated with the celebrated result of Raecke [27], which proves that in every graph one can obliviously route the packets while guaranteeing the maximum congestion (of the most used edge) is  $O(\log n)$ -competitive with the globally-and-offline-optimal maximum congestion.

Motivated by this success, a prominent extension called *semi-oblivious routing* was suggested by Hajiaghayi, Kleinberg, and Leighton [15] for VLSI routing and network traffic engineering in the hope of surpassing the competitiveness of (significantly more stringent) oblivious routing. A semi-obliviously-routed packet is required to obliviously specify a small set of *candidate paths* it can possibly traverse. However, the final choice of over which path the packet is routed is made in a globally-optimal manner *after* the entire set of packets (i.e., the so-called demand matrix) is revealed.

Unfortunately, the theoretical inquiries into the effectiveness of semi-oblivious routing have only yielded negative results: [15] showed that any such routing with polynomially-many candidate paths cannot be  $o(\frac{\log n}{\log \log n})$ -competitive in terms of edge congestion — essentially no better than standard oblivious routing.

In contrast to the theoretical barriers, semi-oblivious solutions to packet routing have found notable success in traffic engineering since installing a new candidate path takes considerable effort that involves updating forwarding tables on geographically-distributed switches; on the other hand, the sending rates over candidate paths can be updated quickly (e.g., using a small snapshot of the global traffic every 15 seconds [19]). Semi-oblivious routing solutions offer superior performance as compared to traditional traffic-engineering approaches and they offer robustness over standard oblivious routing as the set of candidate paths can be chosen more diversely [19, 20].

We provide the first theoretical evidence that semi-oblivious routing is significantly stronger than oblivious routing. As a simple subcase of our results, we show that each graph has a  $O(\log n)$ -sparse semi-oblivious routing that is (poly log  $n$ )-competitive with the offline optimum on all permutation demands. Here,  $\alpha$ -sparse means that  $\alpha$  paths are chosen between each pair of vertices (for a total of  $\alpha n^2$  paths). We note that no standard oblivious routing can be (poly log  $n$ )-sparse, at least without a near-linear competitive ratio. This has compelling consequences.

**Consequence: Deterministic Routing.** One cannot in general *deterministically* assign source-destination pairs to a single path without compromising on either obliviousness (dependencies between pairs) or competitiveness. Indeed, even on the widely-studied case of hypercubes, it is known that the best deterministic oblivious routing has competitiveness  $\tilde{\Theta}(\sqrt{n})$  [18]<sup>1</sup>. We contribute one way to bypass this barrier: *deterministically select a few paths* instead of one. Other methods of deterministic routing of hypercubes were developed, but they mostly involved complicated sorting networks [1, 5] or related ideas [13, 21]. Our deterministic strategy of selecting paths is far simpler, and moreover, our approach works for any graph. It is the first deterministic and oblivious strategy for general graphs that is (poly log  $n$ )-competitive.

**Consequence: Power of Random Choices in Semi-Oblivious Routing.** Consider defining  $\alpha$ -sparse classic oblivious routings in an analogous way, where the support size of paths between each pair of vertices is at most  $\alpha$ . Then, 1-sparse oblivious routing corresponds to the deterministic case and the  $\tilde{\Omega}(\sqrt{n})$  barrier [18] applies. As a simple corollary, any  $\alpha$ -sparse oblivious routing can at best be  $\tilde{\Omega}(\sqrt{n}/\alpha)$ -competitive. On the other hand, our results show that the competitiveness of an  $\alpha$ -sparse semi-oblivious routing is  $\tilde{O}(n^{O(1/\alpha)})$ ; the competitiveness improves exponentially with  $\alpha$ . In other words, each additional path leads to a polynomial improvement in competitiveness; semi-oblivious routing benefits from the “power of a few random choices”, an analog to the fundamental classic result where providing two random choices drastically improves performance [32]. This gives the first compelling theoretical separation between oblivious and semi-oblivious routings.

**Consequence: A Natural Construction and Its Traffic Engineering Applications.** Our  $\alpha$ -sparse semi-oblivious construction is extremely simple: *for each pair of vertices, sample  $\alpha$  random paths from any good oblivious routing distribution*. Indeed, this natural approach was considered for network traffic engineering [19, 20]: they sample from an oblivious routing distribution, due to domain constraints only sampling a small  $\alpha$  number of paths between each

pair of nodes (e.g., they choose  $\alpha = 4$ ), and adapt the sending rates on these paths at real-time (semi-obliviousness). While they empirically find this approach to be very effective, the approach was an unproven heuristic with no a priori reason to work well. The observed “power of a few random choices” offers a compelling theoretical justification for why choosing a small constant sparsity like  $\alpha = 4$  is a practical sweet spot that offers both adequate competitiveness and sparsity. Moreover, our paper not only explains why the approach works well for networks that occur in practice [19], but shows competitiveness for worst-case networks as well.

**Technical Challenges.** While the construction is conceptually extremely simple, its analysis in the context of semi-oblivious routings is deeply involved. It is easy to show using simple randomized rounding [31] that a random  $O(\log n)$ -sparse routing is competitive on a fixed demand with high probability. The main challenge arises because the number of possible demands is exponential in  $n$ , hence the approach fails to be competitive *on all demands*. Indeed, randomized rounding is entirely oblivious, while our results provably require us to exploit semi-obliviousness. A much more intricate analysis is required. In a nutshell, we prove the sampling strategy works using the probabilistic method [2]. We set up a randomized dynamic process: For a fixed demand, pretend to send packets on all candidate paths at once, and delete the edges that get *overcongested* (together with all candidate paths crossing that edge). The goal is to show that, with exponential concentration, many candidate paths remain in the end, hence we can route along them without overcongestions. This enables the union bound over exponentially many demands. The main challenge in formalizing this argument is the dependence between path deletions, and resolving them requires a lot of care.

**Consequence: Optimizing Completion Time.** So far, we focused on minimizing the maximum edge congestion. However, the true objective in traffic engineering [19, 20] was to minimize the *completion time*, i.e., the time until all packets arrive at their destinations (also known as the makespan, or minimizing the delay). The papers [19, 20] only explicitly optimize for congestion, but empirically conclude that this choice implicitly also adequately optimizes the completion time, at least on their benchmark networks.

Unfortunately, this inherently fails on worst-case instances: there exist graphs where optimizing congestion yields non-competitive completion-time guarantees [12]. We show that using the natural sampling strategy with the very recent hop-constrained oblivious routings [12], we can obtain  $O(\log n)$ -sparse semi-oblivious routings that are (poly log  $n$ )-competitive in terms of completion time, even on worst-case instances.

**Paper Organization.** We first overview important concepts in Section 2.1 followed by an overview of our results for integral semi-oblivious routing in Section 2.2. Related work is presented in Section 3 and the notation used throughout this paper in Section 4. In Section 5, we formally define and construct (fractional) semi-oblivious routings by sampling the candidate paths from an oblivious routing. The section is divided into a technical overview (Section 5.1), a proof structure overview (Section 5.2) and the proof of the main lemma (Section 5.3). This is the conference version of the paper. To fit to the conference page limit, finishing the reduction from the Theorem to the main Lemma, the reductions from the real-valued setting to the integral and completion-time settings and

<sup>1</sup>Tilde (e.g.,  $\tilde{\Theta}$  or  $\tilde{O}$ ) hides poly log  $n$  factors.

the lower bound that shows our construction is near-optimal are deferred to the full version of the paper [36].

## 2 TECHNICAL DISCUSSION

In this section, we give a technical overview of our contribution. However, before stating them we discuss some important concepts (that clarify the technical choices), and then state our results.

### 2.1 Overview of Concepts

A semi-oblivious routing over  $G$  is a simple combinatorial object: a *path system*  $\{P(s, t)\}_{s, t \in V}$ , where each pair of vertices  $(s, t)$  is associated with a collection  $P(s, t)$  representing the candidate paths between  $s, t$ .

**Definition 2.1** (Path System). A path system  $\mathcal{P} = \{P(s, t)\}_{s, t \in V}$  is a collection of sets  $P(s, t)$  of simple paths with endpoints  $s$  and  $t$ , for every vertex pair  $(s, t)$ . We say a path system  $\mathcal{P}$  is  $\alpha$ -sparse if  $|P(s, t)| \leq \alpha$  for all  $(s, t)$ . With slight abuse of notation, we say a path system  $\mathcal{P}$  is  $(\alpha + \text{cut}_G)$ -sparse if  $|P(s, t)| \leq \alpha + \text{cut}_G(s, t)$ , where  $\text{cut}_G(s, t)$  is the minimum cut between  $s$  and  $t$ .

**Competitive Ratio.** To evaluate the quality of our semi-oblivious routing, we perform the following sequence of stages.

- (Stage 1) A graph  $G = (V, E)$  is given as input.
- (Stage 2) We design a path system  $\mathcal{P} = \{P(s, t)\}_{s, t \in V}$ .
- (Stage 3) An arbitrary (possibly adversarially-chosen) demand is revealed (i.e., the multiset of source-destination pairs corresponding to packets).
- (Stage 4) For each packet  $s \rightarrow t$  (with source  $s$  and destination  $t$ ) we choose which of the candidate paths  $P(s, t)$  the packet  $s \rightarrow t$  uses. We are allowed to adaptively use all available global information in a way to minimize the maximum edge congestion (i.e., number of paths going over any single edge).
- (Stage 5) Finally, the maximum edge congestion obtained in this way is compared to the offline optimal one and the ratio is called the *competitive ratio*, the primary quantity we aim to minimize. If  $\mathcal{P}$  has competitive ratio at most  $C$  against all demands, we say  $\mathcal{P}$  is a  $C$ -competitive semi-oblivious routing.

One can consider many types of (semi)oblivious routings. For example, we can construct routings that are either integral or fractional routings, we can compare ourselves to the optimal integral or fractional solution, we can look at demands that are either arbitrary or  $\{0, 1\}$  or permutation (see below), consider  $\alpha$ -sparsity vs.  $(\alpha + \text{cut}_G)$ -sparsity (Definition 2.1), etc. Most of these choices are inconsequential: one can typically inter-reduce results between them with negligible losses. However, some choices are incompatible and keeping track of them introduces technical complexities in the formal statements. Due to this, we explain these choices below and describe which combinations are meaningful.

**Fractional vs. Integral Routing.** When talking about fractional routings, in (Stage 4), for each packet  $s \rightarrow t$  we adaptively assign to each candidate path  $p \in P(s, t)$  a nonnegative weight  $w(p) \geq 0$  such that  $\sum_{p \in P(s, t)} w(p) = 1$  (equivalently, we choose a distribution over  $P(s, t)$ ). These weights define a *fractional unit flow* from  $s$  to  $t$  which we use to route the packet. Similarly, the optimal fractional solution,

to which we compare ourselves in (Stage 5), routes each packet  $s \rightarrow t$  using a (fractional unit) flow (i.e., via a convex combination of paths from  $s$  to  $t$ ). On the other hand, in integral routings, each packet  $s \rightarrow t$  is routed on exactly one path from  $P(s, t)$ . In this case the optimal solution is restricted to be integral (as otherwise no sublinear competitive ratio would be obtainable).

**Types of Demands.** We also define several kinds of demands.

**Definition 2.2.** A demand is a function  $d : V \times V \mapsto \mathbb{R}_{\geq 0}$  from vertex pairs to nonnegative real numbers, such that  $d(v, v) = 0$  for all  $v \in V$ . The demand is *integral* if  $d(s, t)$  is an integer for every  $(s, t)$ . The demand is a  $\{0, 1\}$ -demand if  $d(s, t) \in \{0, 1\}$ . The demand is a *permutation demand* if it is a  $\{0, 1\}$ -demand and  $\sum_s d(s, t) \leq 1$ ,  $\sum_t d(s, t) \leq 1$ . We denote the support of a demand by  $\text{supp}(d) = \{(s, t) : d(s, t) > 0\}$  and define the size  $\text{siz}(d)$  of a demand as  $\text{siz}(d) := \sum_{s \neq t} d(s, t)$ .

Permutation demands will mostly be used to give a technical overview and in the lower bound (since it makes the statement stronger).

**Meaningful Combinations.** In this paper we mostly consider two settings:

- (1) Fractional routings, arbitrary demand, and  $(\alpha + \text{cut}_G)$ -sparse path systems.
- (2) Integral routings,  $\{0, 1\}$ -demands, and  $\alpha$ -sparse path systems.

Other combinations are either weaker or not meaningful (in the sense that no sublinear competitive ratio is achievable). For example, using  $\alpha$ -sparsity in Setting (1) is not meaningful as we need at least  $\text{cut}_G(s, t)$  many candidate paths between  $s$  and  $t$ . To see why, consider two  $n$ -cliques connected via  $n$  edges. The demand is comprised of a single packet going from an arbitrary vertex  $s$  in one clique to an arbitrary  $t$  in the other one: we need at least  $\text{cut}_G(s, t)/\beta = n/\beta$  many paths to be  $\beta$ -competitive since the optimal solution has congestion  $1/n$  and the semi-oblivious congestion is at least  $1/|P(s, t)|$ . Finally, results for Setting (2) are implied by Setting (1).

**Logarithmic- vs. Low-Sparsity Cases.** We also consider cases when  $\alpha$  is small and give the full sparsity-competitiveness trade-off for those cases. However, the case  $\alpha = O\left(\frac{\log n}{\log \log n}\right)$  and poly  $\log n$  competitiveness is of special interest and we refer to it as the *logarithmic-sparsity case*. On the other hand, the general case (when  $\alpha$  can be small) is referred to as the *low-sparsity case*. We note that our results here show an exponential improvement in competitiveness with  $\alpha$ . Moreover, we nearly match the entire trade-off curve with a lower bound.

**Objective: Edge Congestion vs. Completion Time.** The default objective throughout the paper is minimizing the maximum edge congestion, where the congestion of  $e \in E$  is the (potentially fractional) number of packets routed through  $e$ . A compelling alternative is minimizing the completion time: given a set of paths  $P = \{p_1, \dots, p_k\}$  used for routing packets (connecting the source-destination pairs), we aim to minimize “*dilation + congestion*”. Here, congestion is defined as before, and dilation is defined as the maximum number of hops of any path,  $\max_{p \in P} \text{hop}(p)$ . This roughly corresponds to minimizing the time until the last packet arrives at their destination (due to classic reductions [10, 22]), hence the name

completion time. However, due to recent advancements in hop-constrained oblivious routings [12], our results gracefully extend to completion-time-competitive semi-oblivious routings. Namely, we simply sample from a hop-constrained oblivious routing instead of the classic one.

## 2.2 Formal Overview of Results for Integral Routings

In this section, we present our results in the setting of integral routings and demands. For brevity, we omit the analogous results for fractional routings, albeit they directly follow from the results of Section 5. Note that the reduction from the integral setting to the fractional setting and the lower bound proofs are deferred to the full version of the paper [36].

**Logarithmic-Sparsity Case.** We show that in every graph, there exists a set of logarithmically many paths between every vertex pair, through which any  $\{0, 1\}$ -demand can be routed semi-obliviously with polylogarithmic competitiveness. Moreover, we give a matching lower bound that shows that polylogarithmic competitiveness is unachievable with asymptotically less paths, even with the more restricted demand set of permutation demands.

**Theorem 2.3.** *Let  $G$  be a  $n$ -vertex graph with at most a polynomial number of edges. Then, there exists a  $O(\log n / \log \log n)$ -sparse integral semi-oblivious routing on  $G$  that is  $O(\log^3 n / \log \log n)$ -competitive on  $\{0, 1\}$ -demands.*

**Lemma 2.4.** *[Impossibility] There exists an infinite family  $\mathcal{G}$  of simple graphs, such that for any sparsity bound  $g_1 = o\left(\frac{\log n}{\log \log n}\right)$  and competitiveness bound  $g_2 = \text{poly log } n$ , there exists an integer  $n_0$  such that for every  $n$ -vertex graph  $G \in \mathcal{G}$  where  $n \geq n_0$ , there exists no  $g_1(n)$ -sparse integral semi-oblivious routing on  $G$  that is  $g_2(n)$ -competitive on all permutations demands.*

**Low-Sparsity Case.** We now turn our attention to semi-oblivious routings with a sublogarithmic number of paths between every vertex pair. We show that any polynomial competitiveness is achievable with a constant number of sampled paths, and that every additional path yields a polynomial improvement to the competitiveness. The matching lower bound shows that this is tight: for any  $\alpha$ , the sparsity of Theorem 2.5 cannot be improved by more than a constant factor while retaining the same competitiveness.

**Theorem 2.5.** *Let  $G$  be a  $n$ -vertex graph with at most a polynomial number of edges. Then, for every positive integer  $\alpha = o\left(\frac{\log n}{\log \log n}\right)$ , there exists an  $\alpha$ -sparse integral semi-oblivious routing on  $G$  that is  $n^{O(\alpha^{-1})}$ -competitive on  $\{0, 1\}$ -demands.*

**Lemma 2.6.** *[Impossibility] There exists an infinite family  $\mathcal{G}$  of simple graphs, such that for every constant  $\epsilon > 0$  and  $\alpha = o\left(\frac{\log n}{\log \log n}\right)$ , there exists an integer  $n_0$  such that for every  $n$ -vertex graph  $G \in \mathcal{G}$  where  $n \geq n_0$ , there exists no  $\alpha$ -sparse integral semi-oblivious routing on  $G$  that is  $n^{(\frac{1}{2}-\epsilon)\alpha^{-1}}$ -competitive on all permutation demands.*

**Arbitrary Integral Demands.** Our results generalize to arbitrary integral demands. However, this requires us to consider  $(\alpha + \text{cut}_G)$ -sparsity, as no polylogarithmic competitiveness would be

possible, and to pay an additional logarithmic factor in the competitiveness. We could derive an analogous result for the low-sparsity case, but omit it for brevity.

**Lemma 2.7.** *Let  $G$  be a  $n$ -vertex graph with at most a polynomial number of edges. Then, there exists a  $\left(O\left(\frac{\log n}{\log \log n}\right) + \text{cut}_G\right)$ -sparse  $O\left(\frac{\log^4 n}{\log \log n}\right)$ -competitive integral semi-oblivious routing.*

**Routings for Completion Time.** Recall that “completion time” objective corresponds to minimizing congestion + dilation (where the dilation is the longest path used in the routing). By combining our results with the recent hop-constrained oblivious routings [12], we construct sparse semi-oblivious routings that are competitive in terms of completion time. Note that in this setting, compared to competitiveness in congestion alone, we require quadratic sparsity. One can also derive completion-time results against arbitrary demands, but we omit them for brevity.

**Lemma 2.8.** *Let  $G$  be a  $n$ -vertex graph with at most a polynomial number of edges. Then, there exists a  $O((\log n)^2 / (\log \log n)^2)$ -sparse integral semi-oblivious routing  $\mathcal{P}$  on  $G$  that is  $\text{poly}(\log n)$ -completion time competitive on  $\{0, 1\}$ -demands.*

**Lemma 2.9.** *Let  $G$  be a  $n$ -vertex graph with at most a polynomial number of edges. Then, for every integer  $\alpha = o(\log n / \log \log n)$ , there exists an  $\alpha^2$ -sparse integral semi-oblivious routing  $\mathcal{P}$  on  $G$  that is  $n^{O(\alpha^{-1})}$ -completion time competitive on  $\{0, 1\}$ -demands.*

## 3 RELATED WORK

**Routing on Hypercubes.** Oblivious routing was first studied by Valiant and Brebner [35] for the special case of the hypercube as the underlying graph. This is because parallel computers are often implemented using (a variant of) a hypercube architecture as the topology of choice for connecting its processors, motivating the study of oblivious routings. A very simple strategy known as the “Valiant’s trick” yields a  $(\text{poly log } n)$ -competitive routing (in terms of edge congestion): when routing a packet from  $s_i \rightarrow t_i$ , first greedily route it from  $s_i$  to a random intermediate vertex, and then greedily route it to  $t_i$ . However, this routing is inherently randomized, and [5] and [18] show that any deterministic routing on a hypercube cannot be  $\tilde{o}(\sqrt{n})$ -competitive. Specifically, they show the following slightly-more-general result: in any  $n$ -vertex graph  $G$  with maximum degree  $\Delta$ , for any deterministic oblivious routing  $R$ , there exists a permutation demand  $d$  such that routing it yields congestion at least  $\Omega(\sqrt{n}/\Delta)$ . Oblivious routings on a range of special graphs were also studied, including expanders, Caley graphs, fat trees, meshes, etc. [6, 7, 25, 33, 34].

**Oblivious Routing on General Graphs.** Ræcke [26] first demonstrated that in every graph there exists a  $(\text{poly log } n)$ -competitive oblivious routing. No poly-time construction algorithm was known at the time, hence [4], [16], and [23] gave polynomial time construction algorithms for the *hierarchical decomposition* required by [26]. This line of work culminated in the celebrated result of Ræcke [27] who gave a poly-time construction of the  $O(\log n)$ -competitive oblivious routing scheme by reducing the problem to  $O(\log n)$ -distortion tree embeddings. The  $O(\log n)$  is asymptotically optimal on general graphs, as shown by [3] and [24]. The first

close-to-linear-time construction (at a cost of polylogarithmically-higher competitiveness) was given by [30] by constructing a hierarchical expander decomposition.

**Other Quality Measures.** All aforementioned oblivious routings had the objective of minimizing (maximum) edge congestion and were optimized either for competitiveness or runtime. However, other quality measures are also prominent in the literature. Various  $\ell_p$  norms of edge congestions and generalizations were studied [9, 14, 28]. Notably, very recent work has shown that there exists a (poly log  $n$ )-competitive oblivious routing for congestion+dilation [12], and [11] has shown that these structures can often be constructed in almost-optimal time in the sequential, parallel, and distributed settings. Furthermore, starting with [29], considerable effort has been invested in designing oblivious routing schemes which are *compact*, meaning that the size of the routing tables is small. [8] gave compact oblivious routing schemes in weighted graphs where the hop-lengths (i.e., dilation) of the returned paths are not controlled for, while [11] gave compact oblivious routing schemes which control for both congestion and dilation.

**Semi-Oblivious Routing.** The extension from oblivious to semi-oblivious routing was proposed in [15] to model issues that naturally arise in VLSI design and traffic engineering. However, their results were negative and focus mostly on lower bounds: they excluded the possibility of polynomially-sparse semi-oblivious routing with  $O(1)$ -competitiveness by arguing that every such routing on  $n \times n$  grids cannot be better than  $\Omega\left(\frac{\log n}{\log \log n}\right)$ -competitive. [19] present a practical implementation of a semi-oblivious routing-based algorithm for the traffic engineering problem. They empirically show that  $\alpha$ -sparse semi-oblivious routing offers near-optimal performance and satisfactory robustness even for small constant  $\alpha$  (e.g.,  $\alpha = 4$ ).

## 4 FORMAL NOTATION

**Graphs.** We denote undirected graphs with  $G = (V, E)$ . In place of capacities, we allow  $E$  to contain parallel edges. We only work with undirected and connected graphs with polynomially many edges, and won't state that a graph is undirected and connected every time we declare one. We write  $n := |V|$  and  $m := |E|$ .

**Minimum Cut.**  $\text{cut}_G(s, t) : V \times V \mapsto \mathbb{Z}_{\geq 0}$  denotes the size of the minimum  $(s, t)$ -cut  $G$ . We define  $\text{cut}_G(v, v) = 0$ .

**Routings.** A routing  $R = \{R(s, t)\}_{s, t \in V}$  is a collection of distributions  $R(s, t)$  over simple  $(s, t)$ -paths for every vertex pair  $(s, t)$ . In other words, a routing is a path system that also assigns weights to paths. A routing  $R$  “routes a demand  $d$ ” by assigning a weight  $d(s, t)\mathbb{P}[R(s, t) = p]$  for every path  $p \in \text{supp}(R(s, t))$ . We say a routing  $R$  is *integral* on a demand  $d$  if  $d(s, t)\mathbb{P}[R(s, t) = p]$  is an integer for every  $s, t, p$ . We define the support  $\text{supp}(R(s, t))$  of  $R(s, t)$  to be the set of  $(s, t)$ -paths  $R(s, t)$  assigns nonzero probability to and the support  $\text{supp}(R)$  of  $R$  to be the path system  $\mathcal{P}$  where  $P(s, t) = \text{supp}(R(s, t))$ . We say that a routing  $R$  is supported on a path system  $\mathcal{P}$  if  $\text{supp}(R(s, t)) \subseteq P(s, t)$  for all  $(s, t)$ .

**Congestion.** The *congestion* of an edge  $e$  on a routing  $R$  and demand  $d$  is the sum of weights of paths using that edge:  $\text{cong}(R, d, e) = \sum_{s, t} d(s, t)\mathbb{P}[e \in R(s, t)]$ . The *congestion* of a routing  $R$  on a demand  $d$  is the highest edge congestion  $\text{cong}(R, d) = \max_{e \in E} \text{cong}(R, d, e)$ .

**Dilation.** We denote by  $\text{hop}(p)$  the *hop-length* of  $p$ , or the number of edges in  $p$ . The *dilation*  $\text{dil}(R, d) = \max_{(s, t) \in \text{supp}(d)} \max_{p \in \text{supp}(R(s, t))} \text{hop}(p)$  of a routing  $R$  of  $d$  is the maximum hop-length over paths that  $R$  assigns a positive weight to on  $d$ .

**Optimal Congestion.** For a demand  $d$ , the *optimal congestion*  $\text{opt}_{G, \mathbb{R}}(d) = \min_R \text{cong}(R, d)$  is the minimum congestion over routings  $R$  of  $d$ . For integral  $d$ , the *optimal integral congestion*  $\text{opt}_{G, \mathbb{Z}}(d) = \min_R \text{integral on } d \text{ cong}(R, d)$  is the minimum congestion over routings that are integral on  $d$ .

**Oblivious Routings.** An *oblivious routing* is a routing  $R$ . We say  $R$  is  $C$ -competitive on a set of demands  $D$  for  $C \geq 1$  if, for every demand  $d \in D$ , we have  $\text{cong}(R, d) \leq C \text{opt}_{G, \mathbb{R}}(d)$ . We say  $R$  is  $C$ -competitive for  $C \geq 1$  if it is  $C$ -competitive on the set of all demands.

**Integer Prefix Sets.** We use the notation  $[n] = \{1, \dots, n\}$ .

**Logarithms.** We denote the base-2 logarithm by  $\log$  and the base- $e$  logarithm by  $\ln$ .

## 5 CONSTRUCTING SPARSE FRACTIONAL SEMI-OBLIVIOUS ROUTINGS

In this section, we prove the competitiveness of fractional  $(\alpha + \text{cut}_G)$ -sparse semi-oblivious routings sampled from an oblivious routing.

The section starts by formally defining semi-oblivious routings and  $\alpha$ -samples, then states our main Theorem. In Section 5.1, we give a brief overview of the proof of the main Theorem. The proof is divided into the description of the organisation of the proof and the definitions required for it in Section 5.2, a proof of the main Lemma in Section 5.3 and the reduction from the main theorem to the main lemma deferred to the full version of the paper [36].

We start by formally defining semi-oblivious routings. On its face, a semi-oblivious routing is a path system  $\mathcal{P}$  (Definition 2.1) with additional competitiveness properties. To route a demand, the semi-oblivious routing picks the demand-dependent optimal routing supported on a subset of the paths in  $\mathcal{P}$ . For example, if  $\mathcal{P}$  contains every simple  $(s, t)$ -path, the semi-oblivious routing is trivially 1-competitive, thus we are interested in bounding both the competitiveness and the *sparsity* of  $\mathcal{P}$ : the maximum amount of paths between any  $(s, t)$ -pair.

**Definition 5.1** (Semi-Oblivious Routing). *A semi-oblivious routing is a path system  $\mathcal{P}$ . For a demand  $d$ , we define the congestion of (fractionally) routing  $d$  via  $\mathcal{P}$ :*

$$\text{cong}_{\mathbb{R}}(\mathcal{P}, d) = \min_{R \text{ a routing on } \mathcal{P}} \text{cong}(R, d).$$

**Competitiveness.** We say  $\mathcal{P}$  is  $C$ -competitive on a set of demands  $D$  for  $C \geq 1$  if for every demand  $d \in D$  we have  $\text{cong}_{\mathbb{R}}(\mathcal{P}, d) \leq C \text{opt}_{G, \mathbb{R}}(d)$ , and that  $\mathcal{P}$  is  $C$ -competitive if it is  $C$ -competitive on the set of all demands.

**Competitiveness With an Oblivious Routing.** We say  $\mathcal{P}$  is  $C$ -competitive with an oblivious routing  $R$  on a set of demands  $D$  for  $C \geq 1$  if for every demand  $d \in D$  we have  $\text{cong}_{\mathbb{R}}(\mathcal{P}, d) \leq C \text{cong}(R, d)$ . We say  $\mathcal{P}$  is  $C$ -competitive with  $R$  if it is  $C$ -competitive with  $R$  on the set of all demands.

Our methods of constructing semi-oblivious routings are based on sampling paths from an oblivious routing, instead of taking

its whole support. We define samples of an oblivious routing as follows:

**Definition 5.2** (Sample of an Oblivious Routing). *Let  $G$  be a graph,  $R$  be an oblivious routing and  $\alpha \in \mathbb{Z}_{>0}$  be a parameter. An  $\alpha$ -sample of  $R$  is a semi-oblivious routing  $\mathcal{P}$ , where  $P(s, t) \subseteq \text{supp}(R(s, t))$  is a set of  $\alpha$  paths sampled with replacement from the probability distribution  $R(s, t)$ . An  $(\alpha + \text{cut}_G)$ -sample is defined similarly, but with  $(\alpha + \text{cut}_G(s, t))$  sampled paths between  $s$  and  $t$ .*

The following theorem that we prove in the next section states that an  $(\alpha + \text{cut}_G)$ -sample of an oblivious routing is competitive with the oblivious routing with high probability<sup>2</sup>. The second part of the theorem allows us to cut a logarithmic factor from the competitiveness of integral semi-oblivious routings, as the results in that setting are achieved through a reduction where the special condition applies.

We note that it is convenient to measure the competitiveness of a semi-oblivious routing against the result provided by a (standard) oblivious routing, instead of competitiveness against offline solutions. This allows us to apply our results in a black-box manner to the oblivious routings of [12] to achieve completion-time competitive semi-oblivious routings.

**Theorem 5.3** ( $(\alpha + \text{cut}_G)$ -Sample Theorem). *Let  $G = (V, E)$  be a  $n$ -vertex graph with polynomially many edges,  $R$  be an oblivious routing and  $\alpha \in [n]$  be a parameter. Let  $\mathcal{P}$  be an  $(\alpha + \text{cut}_G)$ -sample of  $R$ . Then, with high probability,  $\mathcal{P}$  is*

$$O\left(\log^2(n) \left(\alpha + n^{O(\alpha^{-1})}\right)\right)$$

-competitive with  $R$ , and for every subset  $D$  of  $\{0, 1\}$ -demands such that for  $d \in D$ , we have  $\text{supp}(d) \subseteq \{(s, t) : \text{cut}_G(s, t) = 1\}$ ,  $\mathcal{P}$  is

$$O\left(\log(n) \left(\alpha + n^{O(\alpha^{-1})}\right)\right)$$

-competitive with  $R$  on  $D$ .

## 5.1 Technical Overview

**The Hypercube/Permutation-Demand Subcase.** For simplicity, in this section we overview the proof of a weaker version of Theorem 5.3 for the case of permutation demands on a hypercube. We show that in a hypercube, with high probability, sampling  $\alpha = \Theta(\log n)$  paths between every pair of vertices produces a semi-oblivious routing that can route any permutation demand with congestion  $\text{poly log } n$ . This subcase contains all the main ideas, while avoiding the intricacies involved with general graphs, lower sparsities, and arbitrary demands.

We note that there exists a simple oblivious routing  $R$  for hypercubes: for each packet, we (obliviously) uniformly randomly choose an intermediate node, and then greedily route the packet from the source to the intermediate node, and then to the destination [35]. This strategy ensures that, for any permutation demand, the (expected) congestion of any edge is  $O(1)$ .

Let  $\alpha = \Theta(\log n)$ . Let  $\mathcal{P}$  be a semi-oblivious routing constructed by, for every pair  $s, t \in V$ , taking  $\alpha$  samples of  $(s, t)$ -paths from the oblivious routing  $R$ . We show that, with high probability, for every

<sup>2</sup>We use the phrase “with high probability” to mean that a claim holds with probability at least  $1 - n^{-C}$ , where  $C > 0$  is any sufficiently large constant. In other words, other constants in the statement can be tuned to account for any sufficiently large  $C$ .

permutation demand, we can (adaptively) choose at least one path of the  $\alpha$  sampled paths for each packet, such that the chosen paths yield congestion of at most  $h = \text{poly log } n$ .

**Strategy: Exponentially-Small Failure Per Demand.** Our proof strategy that  $\mathcal{P}$  can (semi-obliviously) route every permutation demand is to show the following: For any fixed permutation demand  $d$ ,  $R$  can route  $d$  with congestion  $h$  except with failure probability that is exponential in the size of the demand (namely  $n^{-\Omega(\text{siz}(d))}$ ). Then, the proof can be finished with a union bound over permutation demands, as the number of permutation demands of size  $s$  is at most exponential in  $s$  (namely  $n^{2s}$ ). Hence, with high probability,  $R$  works for all permutation demands.

We note that there are many standard arguments that show that a fixed demand fails with probability at most  $\exp(-\text{poly log } n)$  (such as randomized rounding [31]). In fact, this requires no semi-obliviousness. However, this is not sufficient as there are exponentially many permutation demands (namely  $n! = \exp(\Theta(n \log n))$ ). A significantly more intricate argument is required.

**Idea: Weak Routing.** It is sufficient to show that the routing  $\mathcal{P}$  can route *at least half* of any permutation demand with congestion  $h$ . Any semi-oblivious routing with that property can be shown to be able to route any permutation demand with congestion  $h \cdot \Theta(\log n) = \text{poly log } n$ , by repeatedly routing half of the demand. Thus, it suffices to show that for any fixed permutation demand  $d$ , with failure probability  $n^{-\Omega(\text{siz}(d))}$ ,  $\mathcal{P}$  can route at least half of  $d$  with congestion  $h$ .

**Idea: Dynamic Process.** Consider all the  $\alpha \cdot \text{siz}(d)$  candidate paths that the sampled  $\mathcal{P}$  can use to route  $d$ . If the congestion of every edge in  $G$  is at most  $h$ , we are done, as selecting arbitrary paths to route the demand cannot achieve higher congestion than selecting every path. Otherwise, we find an overcongested edge  $e$  (with congestion  $t > h$ ) and delete all candidate paths that go through  $e$ . The crucial observation here is that the probability of  $t > h$  paths overcongesting  $e$  is at most  $\exp(-\Omega(t))$ . To explain why, consider a random variable  $X$  that gives the number of sampled paths crossing a fixed edge  $e$ . As the base oblivious routing  $R$  achieves constant congestion, the expectation of the random variable is at most  $\mathbb{E}[X] \leq \Theta(\alpha)$ , so  $t > h \geq 2\mathbb{E}[X]$ . Since the paths are sampled independently, a Chernoff bound ensures  $\Pr[X > t] < \exp(-\Omega(t))$ . We note that the variables are not actually independent, but can be proven to be negatively associated, allowing the use of the bound without changes [17].

We repeat the process for all overcongested edges and delete all congesting paths. After this, if at most half of the candidate paths were deleted, for at least half of the vertex pairs in the demand at least one path remains. Routing the demand between those pairs through those paths routes at least half of the demand with congestion at most  $h$ , as desired.

An uninitiated reader could assume we are done: let the congestions of the overcongested edges be  $t_1, \dots, t_j$ . Assuming independence, the probability of these overcongestions is  $\exp(-\Omega(\sum_i t_i)) \leq n^{-\Omega(\text{siz}(d))}$ , where the last inequality used the fact that weak routing fails, i.e.,  $\sum_i t_i > \alpha \cdot \text{siz}(d)/2 = \Omega(\text{siz}(d) \log n)$ . However, a big issue is that overcongestions are not independent. A more intricate approach needs to be taken.

**Idea: Bad Patterns.** To resolve the inter-dependencies, we first fix an arbitrary ordering of edges  $(e_1, e_2, \dots, e_m)$  that is independent of  $\mathcal{P}$  or  $d$ . The dynamic process will iterate over the edges in the fixed order and, if the current edge is overcongested, delete all candidate paths congesting the edge. To analyze the number of paths removed, we define *bad patterns*:  $m$ -tuples of integers  $(t_1, \dots, t_m)$  with  $\sum_i t_i > \alpha \cdot \text{siz}(d)/2$ . A bad pattern describes the path deletions during a failed process. During the processing of a demand, we say that the bad pattern occurs, if for every edge  $e_i$ , we deleted at least  $t_i$  paths while processing  $e_i$ . If more than half of the candidate paths are deleted during the process, at least one bad pattern occurs.

Since we never delete more than zero but at most  $h$  paths during a step, we require that every nonzero  $t_i$  is greater than  $h$ . Thus, there are at most  $\alpha \cdot \text{siz}(d)/h$  nonzero values in the bad pattern, and with simple combinatorial calculations we can bound the number of bad patterns to  $n^{O(\text{siz}(d))}$ . Thus, it suffices to show that any fixed bad pattern occurs with probability at most  $n^{-\Omega(\text{siz}(d))}$ , at which point a union bound finishes the proof.

**Resolving Dependence.** We note that if a bad pattern occurs, in the initial set of sampled paths there must be at least  $t_i$  paths crossing edge  $e_i$  that *do not* cross any overcongested edge  $e_j$  that is “earlier” ( $j < i$ ). Those paths would already have been eliminated while processing edge  $e_j$  before the process reaches  $e_i$ . Thus, after fixing a potential bad pattern  $(t_1, \dots, t_m)$ , we can uniquely assign each path  $p$  to the first edge  $e_i$  (in the ordering) on  $p$  with  $t_i > h$ . Then, if a bad pattern occurs,  $\mathcal{P}$  must have sampled at least  $t_i$  paths assigned to  $e_i$ , the probability of which is  $\exp(-\Omega(t_i))$  as before via a Chernoff bound on negatively associated variables. Crucially, since the assignment is unique and does not depend on the process, these individual probabilities can be multiplied together. More formally, the probability that multiple lower bounds on disjoint subset sums of negatively associated variables simultaneously occur is at most the product of the probabilities of the individual lower bounds [17]. Hence, the probability of a bad pattern is  $\exp(-\Omega(\sum_i t_i))$  and we can conclude the same way as before.

**The General Case.** Recall that in the case of general graphs, it is necessary to sample at least  $\text{cut}_G(s, t)$ -many paths between  $s$  and  $t$  to achieve competitiveness (as explained in Section 2.1). Because the number of sampled paths can now vary, we need to weight the sampled  $(s, t)$ -paths by dividing the  $(s, t)$ -demand equally among them, instead of setting the weight of every path to 1.

**Idea: Special Demands.** However, this causes another issue. Consider the random variables whose sum we bound via Chernoff in the hypercube case. Unlike before, their scales vary wildly, which significantly degrades the concentration of their sum. To resolve this, it is natural to work with *special demands*: demands where  $d(s, t)$  is either 0 or  $\alpha + \text{cut}_G(s, t)$ . Special demands force the random variables comprising the considered sum to be binary. This re-enables Chernoff and fixes the proof, at least for the case of special demands. However, this is enough — we can show that a semi-oblivious routing that is competitive on special demands is competitive on general due to simple bucketing (up to a single logarithmic factor).

Another small issue is that the extra paths grow the set of bad patterns too large. To handle this, we force  $t_i$  to be a multiple of  $h$ ,

and halve the sum requirement. Then, there still exists at least one bad pattern that occurs whenever more than half of the paths get cut, while the total number of bad patterns is sharply cut.

**The Low-Sparsity Case.** To achieve results for  $\alpha \ll \log n$ , we need to use the large deviation version of Chernoff bounds, providing the stronger bound of  $\Pr[X > t] < \exp(-\Omega(t \log(t/\mathbb{E}[X])))$  when  $t \gg \mathbb{E}[X]$ . The extra logarithmic term also allows us to cut the required sparsity to achieve polylogarithmic competitiveness from  $\Omega(\log n)$  to  $O(\log(n)/\log \log n)$ .

## 5.2 Structure of the Proof

In this section, we give a short overview of the formal proof of Theorem 5.3. The most important part of the proof is Lemma 5.6, which is a weaker version of the result: for every fixed demand of a special class of demands we define, with failure probability exponential in the support size of the demand, a sample independent of the demand can route at least half of the total demand with low congestion. Then, a union bound over the special class of demands in the proof of Corollary 5.7 shows that a sample can route at least half of every special demand with low congestion with high probability, and two further reductions generalise from routing half of the demand to routing the full demand (Lemma 5.8) and from special demands to arbitrary demands (Lemma 5.9).

The notion of *weak-competitiveness* formalises the notion of competitively routing half of the total demand. Note that in particular we do not require the semi-oblivious routing to be able to route the subdemand with congestion at most  $C$  times the congestion of the oblivious routing on the subdemand, but instead compare against the congestion on the original demand. In Lemma 5.8, we show that weak competitiveness implies competitiveness in general at the cost of one logarithmic factor.

**Definition 5.4** (Weakly-Competitive Semi-Oblivious Routing). *For  $C \geq 1$ , a semi-oblivious routing  $\mathcal{P}$  is  $C$ -weakly-competitive with an oblivious routing  $R$  on a set of demands  $D$ , if, for every demand  $d \in D$ , there exists a demand  $d'$  and a routing  $R'$  on  $\mathcal{P}$  such that*

- $d'(s, t) \leq d(s, t)$  for all  $s, t$ ,
- $\text{siz}(d') \geq \frac{1}{2} \text{siz}(d)$ ,
- $\text{cong}(R', d') \leq C \text{cong}(R, d)$ .

The special class of demands contains the demands for which the ratio of  $(s, t)$ -demand to the number of  $(s, t)$ -paths  $d(s, t)/(\alpha + \text{cut}_G(s, t))$  is either 0 or 1. The definition of this class is necessary, as the proof of the main Lemma 5.6 requires the ratio to be constant for all nonzero demands to achieve concentration. We prove in Lemma 5.9 that competitiveness on special demands implies competitiveness on general demands at the cost of an (multiplicative) logarithmic factor.

**Definition 5.5** (Special Demands). *Let  $G$  be a  $n$ -vertex graph and  $\alpha \in [n]$  be a fixed parameter. A demand  $d$  is  $\alpha$ -special, if  $d(s, t) \in \{0, \alpha + \text{cut}_G(s, t)\}$  for every  $(s, t)$ . We denote the set of all  $\alpha$ -special demands on  $G$  by  $S_G(\alpha)$ . When  $\alpha$  is clear from the context, we refer to  $\alpha$ -special demands as just “special demands”.*

With these two definitions, we can formally state Lemma 5.6, the core part of our proof. The parameter  $h$  that appears here explicitly can later be set to achieve the “with high probability” guarantee in Theorem 5.3. Section 5.3 is dedicated to the proof of Lemma 5.6.



**Lemma 5.6 (Main Lemma).** *Let  $G = (V, E)$  be a  $n$ -vertex  $m$ -edge graph where  $3 \leq n \leq m$ ,  $\alpha \in [n]$  and  $h \geq 1$  be fixed parameters,  $R$  be an oblivious routing on  $G$  and  $d \in S_G(\alpha)$  be a fixed special demand. Let  $\mathcal{P}$  be an  $(\alpha + \text{cut}_G)$ -sample of  $R$ . Then, with probability at least  $1 - m^{-(h+3)|\text{supp}(d)|}$ ,  $\mathcal{P}$  is  $(\alpha + m^{16(h+7)\alpha^{-1}})$ -weakly-competitive with  $R$  on  $\{d\}$ .*

The proof of Theorem 5.3 from Lemma 5.6 is achieved through three reductions. First, a simple union bound shows that the probability bound of Lemma 5.6 is sufficient for a random sample to be weakly competitive on all special demands, proving Corollary 5.7, a version of the main lemma with the correct sampling order. Then Lemma 5.8 reduces from weak competitiveness on special demands to general competitiveness on special demands, and finally Lemma 5.9 shows competitiveness on general demands for any routing that is competitive on special demands. The three reductions are fairly routine and are deferred to the full version of the paper [36]. Combining Corollary 5.7, Lemma 5.8 and Lemma 5.9, the proof of Theorem 5.3 immediately follows.

**Corollary 5.7.** *Let  $G$  be a  $n$ -vertex  $m$ -edge graph where  $3 \leq n \leq m$ ,  $\alpha \in [n]$  and  $h \geq 1$  be fixed parameters, and  $R$  be an oblivious routing on  $G$ . Let  $\mathcal{P}$  be an  $(\alpha + \text{cut}_G)$ -sample of  $R$ . Then, with probability at least  $1 - m^{-h}$ ,  $\mathcal{P}$  is  $(\alpha + m^{16(h+7)\alpha^{-1}})$ -weakly competitive with  $R$  on  $S_G(\alpha)$ .*

**Lemma 5.8 (weak-to-strong reduction).** *Let  $G$  be a  $n$ -vertex  $m$ -edge graph,  $R$  be an oblivious routing and  $D$  be a set of demands, such that for every demand  $d \in D$ , for every demand  $d'$  such that  $d'(s, t) \in \{0, d(s, t)\}$ , we have  $d' \in D$ . Let  $\mathcal{P}$  be a semi-oblivious routing that is  $C$ -weakly-competitive with  $R$  on  $D$ . Then,  $\mathcal{P}$  is  $O(C \log m)$ -competitive with  $R$  on  $D$ .*

**Lemma 5.9 (special-to-general reduction).** *Let  $G$  be a  $n$ -vertex  $m$ -edge graph,  $\alpha \in [n]$  be a fixed parameter,  $R$  be an oblivious routing and  $\mathcal{P}$  be a semi-oblivious routing on  $G$  that is  $C$ -competitive with  $R$  on  $S_G(\alpha)$ . Then,  $\mathcal{P}$  is  $O(C \log(m))$ -competitive with  $R$  on all demands.*

**Theorem 5.3 ( $(\alpha + \text{cut}_G)$ -Sample Theorem).** *Let  $G = (V, E)$  be a  $n$ -vertex graph with polynomially many edges,  $R$  be an oblivious routing and  $\alpha \in [n]$  be a parameter. Let  $\mathcal{P}$  be an  $(\alpha + \text{cut}_G)$ -sample of  $R$ . Then, with high probability,  $\mathcal{P}$  is*

$$O\left(\log^2(n) \left(\alpha + n^{O(\alpha^{-1})}\right)\right)$$

-competitive with  $R$ , and for every subset  $D$  of  $\{0, 1\}$ -demands such that for  $d \in D$ , we have  $\text{supp}(d) \subseteq \{(s, t) : \text{cut}_G(s, t) = 1\}$ ,  $\mathcal{P}$  is

$$O\left(\log(n) \left(\alpha + n^{O(\alpha^{-1})}\right)\right)$$

-competitive with  $R$  on  $D$ .

**PROOF.** For  $m < n$ , since the graph has to be connected, there is exactly one simple path between any two vertices, thus  $\mathcal{P}$  is 1-competitive. Thus, we may assume that  $3 \leq n \leq m$ , as for constant  $n$  and  $m$ , no guarantees are made.

By Corollary 5.7, with high probability, the  $(\alpha + \text{cut}_G)$ -sample  $\mathcal{P}$  is  $(\alpha + m^{O(\alpha^{-1})})$ -weakly competitive with  $R$  on  $S_G(\alpha)$ . From now

on, assume it is. Then, by Lemma 5.8,  $\mathcal{P}$  is  $O\left(\log(m) \left(\alpha + m^{O(\alpha^{-1})}\right)\right)$ -competitive with  $R$  on  $S_G(\alpha)$ . Thus, since  $D \subseteq S_G(\alpha)$ , the sample  $\mathcal{P}$  is  $O\left(\log(m) \left(\alpha + m^{O(\alpha^{-1})}\right)\right)$ -competitive with  $R$  on  $D$ , and by Lemma 5.9,  $\mathcal{P}$  is  $O\left(\log^2(m) \left(\alpha + m^{O(\alpha^{-1})}\right)\right)$ -competitive with  $R$ .

Finally, since  $m$  is polynomially bounded,  $\log m = O(\log n)$  and  $m^{O(\alpha^{-1})} = n^{O(\alpha^{-1})}$ .  $\square$

### 5.3 Proving the Main Lemma

The entirety of this section is dedicated to the proof of Lemma 5.6.

**Lemma 5.6 (Main Lemma).** *Let  $G = (V, E)$  be a  $n$ -vertex  $m$ -edge graph where  $3 \leq n \leq m$ ,  $\alpha \in [n]$  and  $h \geq 1$  be fixed parameters,  $R$  be an oblivious routing on  $G$  and  $d \in S_G(\alpha)$  be a fixed special demand. Let  $\mathcal{P}$  be an  $(\alpha + \text{cut}_G)$ -sample of  $R$ . Then, with probability at least  $1 - m^{-(h+3)|\text{supp}(d)|}$ ,  $\mathcal{P}$  is  $(\alpha + m^{16(h+7)\alpha^{-1}})$ -weakly-competitive with  $R$  on  $\{d\}$ .*

Let  $\text{cnt}_G(s, t) := \alpha + \text{cut}_G(s, t)$  be the number of  $(s, t)$ -paths we sample, and let  $D := \text{siz}(d)$ . Let  $\gamma := \text{cong}(R, d) \left(\alpha + m^{16(h+7)\alpha^{-1}}\right)$  be the maximum allowed congestion of any edge. Index the edges arbitrarily, such that  $E = \{e_1, \dots, e_m\}$ .

We start by defining random variables  $X$  that describe the  $(\alpha + \text{cut}_G)$ -sampling of  $\mathcal{P}$  from  $R$ . Let  $X(s, t)_{i,p}$  be a  $\{0, 1\}$ -random variable, with value 1 if and only if the  $i$ th sampled  $(s, t)$ -path is  $p$ . Then,  $p \in P(s, t)$  if and only if  $X(s, t)_{i,p} = 1$  for some  $i$ . For fixed  $s, t, i$ , exactly one of  $X(s, t)_{i,p}$  will equal 1 and the probability that  $X(s, t)_{i,p}$  equals 1 is  $\mathbb{P}[R(s, t) = p]$ . The variables  $X(s, t)_{i,p}$  do not depend on  $X(s', t')_{i', p'}$  with  $i' \neq i$ . Thus, by Lemma A.2 and Lemma A.3, the variables  $X$  are *negatively associated*.

We certify that the sampled semi-oblivious routing  $\mathcal{P}$  has the desired competitiveness with  $R$  on  $\{d\}$  by giving a process that finds weights  $w_m$  for paths in  $\mathcal{P}$ , such that, routing  $w_m(p)$  of the  $(s, t)$ -demand along the  $(s, t)$ -path  $p$  for all  $(s, t, p)$ , the congestion is at most  $\gamma$  and at least half of the total demand is routed. The process can fail, and our goal is to prove that it fails with probability at most  $m^{-(h+3)|\text{supp}(d)|}$ .

We find  $w_m$  by defining a sequence of path weights  $w_0, \dots, w_m$ . The initial weight  $w_0$  of a  $(s, t)$ -path for  $(s, t) \in \text{supp}(d)$  is simply the number of times it was sampled. As the demand is special, the initial sum of weights of  $(s, t)$ -paths then equals  $d(s, t)$ : using the initial weights all of the demand can be routed, but the resulting congestion might be arbitrarily high. The process iterates over the edges of the graph in the fixed order. While handling edge  $e_k$ , if the congestion applied to the edge by the weights  $w_{k-1}$  is greater than the allowed  $\gamma$ , the weight of all paths crossing the edge is set to 0, and otherwise, all weights remain unchanged. Then, at the end of the process, the congestion of any edge is guaranteed to be at most  $\gamma$ , and it simply remains to bound the probability the final weights  $w_m$  route at least half of the demand.

Formally, the weights  $w_0, \dots, w_m$  are random functions, with  $w_0$  defined as

$$w_0(p) := d(s, t) \frac{\sum_i X(s, t)_{i,p}}{\text{cnt}_G(s, t)} = \sum_i X(s, t)_{i,p} \mathbb{I}[(s, t) \in \text{supp}(d)]$$



(as the demand is special), and for  $k = 1, \dots, m$ , the next weights  $w_k$  being defined based on  $w_{k-1}$  based on the current congestion of edge  $e_k$ :

- Let  $\text{cong}_k := \sum_{p \in \mathcal{P}} w_{k-1}(p) \mathbb{I}[e_k \in p]$  be the congestion of edge  $e_k$  before step  $k$ .
- If  $\text{cong}_k \leq \gamma$ , we let  $w_k := w_{k-1}$ . Otherwise, we let  $w_k(p) := w_{k-1}(p) \mathbb{I}[e_k \notin p]$ .

Note that in both cases,  $w_k(p) \leq w_{k-1}(p)$ .

With the final weights, the subset  $d'$  of the demand we'll route is simply the sum of the final weights,  $d'(s, t) = \sum_{p \in \mathcal{P}(s, t)} w_m(p)$ , and the routing  $R'$  we use distributes load to the sampled paths according to their weights:  $\mathbb{P}[R'(s, t) = p] = \frac{w_m(p)}{d'(s, t)}$  (for  $(s, t) \in \text{supp}(d')$ , the routing can be arbitrary for other pairs). The following lemma, the proof of which is deferred to the full version [36], gives a sufficient condition for the produced  $d'$  and  $R'$  to satisfy the requirements of Lemma 5.6.

**Lemma 5.10.** *Let  $\Delta_k = \sum_{p \in \mathcal{P}} w_{k-1}(p) - w_k(p)$ . Then,*

- $d'(s, t) \leq d(s, t)$  for all  $(s, t)$ ,
- $\text{cong}(R', d') \leq \gamma$ ,
- $\text{siz}(d') = D - \sum_k \Delta_k$ ,

thus, if  $\sum_k \Delta_k \leq \frac{1}{2}D$ ,  $d'$  and  $R'$  satisfy the requirements of Lemma 5.6.

Thus, it only remains to bound the probability that  $\sum_k \Delta_k > \frac{1}{2}D$ . We do this by applying a union bound. As there are far too many possible runs of the process or even sequences of values  $\delta_k$  to union bound over, we instead apply the union bound over a reduced class of simple objects, *bad patterns*.

**Definition 5.11** (Bad Patterns). *A bad pattern  $(b)_k = (b_1, \dots, b_m)$  is a  $m$ -tuple of nonnegative integers, such that  $\frac{1}{4}D \leq \sum_k \gamma b_k \leq D$ .*

**Lemma 5.12.** *If  $\sum_k \Delta_k > \frac{1}{2}D$ , there exists a bad pattern  $b$ , such that  $\Delta_k \geq \gamma b_k$  for all  $k$ .*

Lemma 5.12, the simple proof of which is deferred to the full version of the paper [36], shows that if there is no bad pattern  $(b)_k$  such that  $\Delta_k \geq \gamma b_k$  for all  $k$ , we have  $\sum_k \Delta_k \leq \frac{1}{2}D$ . In Lemma 5.13, the proof of which is deferred into the full version, we bound the number of bad patterns. Then, in Lemma 5.14, we prove that for a fixed bad pattern  $(b)_i$ , the probability that  $\Delta_k \geq \gamma b_k$  for all  $k$  is small. Afterward, a simple union bound combines these three results to upper bound the probability that  $\sum_k \Delta_k > \frac{1}{2}D$ , which by Lemma 5.10 completes the proof of Lemma 5.6.

**Lemma 5.13.** *There are at most  $m^{6D/\alpha}$  bad patterns.*

**Lemma 5.14.** *For a bad pattern  $(b)_k$ ,*

$$\mathbb{P}\left[\bigcap_{k \in [|E|]} \Delta_k \geq \gamma b_k\right] \leq m^{-(h+7)D/\alpha}.$$

**PROOF.** Fix the bad pattern  $(b)_k$  and let  $B = \{k \in [|E|] \mid b_k > 0\}$  be the set of overcongesting edges. We have

$$\mathbb{P}\left[\bigcap_{k \in [|E|]} \Delta_k \geq \gamma b_k\right] \leq \mathbb{P}\left[\bigcap_{k \in B} \Delta_k \geq \gamma b_k\right].$$

Let  $Z_k \subseteq \bigcup_{s, t} \text{supp}(R(s, t))$  be the set of paths that contain edge  $e_k$  but no edge  $e_{k'}$  for  $k' \in B \cap [k-1]$  and  $Z_{k, s, t} = Z_k \cap \text{supp}(R(s, t))$ . Now,  $Z_k$  is the set of paths that could possibly congest edge  $e_k$  if the event occurs, as any path crossing an earlier edge has already been cut, as the event occurring implies those edges are overcongested. Let  $Y_k$  be a random variable giving the number of sampled paths that cross edge  $e_k$  but no earlier bad edge:

$$Y_k = \sum_{(s, t) \in \text{supp}(d)} \sum_{p \in Z_{k, s, t}} \sum_i X(s, t)_{i, p}.$$

We show that  $\bigcap_{k \in B} Y_k \geq \gamma b_k$  is implied by  $\bigcap_{k \in B} \Delta_k \geq \gamma b_k$ , and then upper bound the probability of the former event.

To see this, we assume the contrary: that there exists a sampling  $x(s, t)_{i, p}$ , such that with nonzero probability,  $X(s, t)_{i, p} = x(s, t)_{i, p}$  holds for all  $s, t, i, p$ , and when  $X(s, t)_{i, p} = x(s, t)_{i, p}$ ,  $\Delta_k \geq \gamma b_k$  holds for all  $k$ , but there is some  $k'$  such that  $Y_{k'} < \gamma b_{k'}$ . But then,

$$\begin{aligned} \text{cong}_{k'} &= \sum_{p \in \mathcal{P} \mid e_{k'} \in p} w_{k'-1}(p) \\ &\leq \sum_{p \in \mathcal{P} \mid e_{k'} \in p} w_0(p) \prod_{k \in B \cap [k'-1]} \mathbb{I}[e_k \notin p] \\ &= \sum_{p \in Z_{k'}} w_0(p), \end{aligned}$$

as for  $k \in B$ ,  $w_k(p) = w_{k-1}(p) \mathbb{I}[e_k \notin p]$ , as  $\Delta_k > \gamma b_k > 0$  implies the edge  $e_k$  is overcongested, thus all paths crossing it were cut during step  $k$ , and  $w_k(p) \leq w_{k-1}(p)$  holds for all  $k$ , thus also for  $k \notin B$ . Finally,

$$\begin{aligned} \sum_{p \in Z_{k'}} w_0(p) &= \sum_{(s, t)} \sum_{p \in Z_{k', s, t}} w_0(p) \\ &= \sum_{(s, t)} \sum_{p \in Z_{k', s, t}} \frac{d(s, t)}{\text{cnt}_G(s, t)} \sum_i x(s, t)_{i, p} \\ &= \sum_{(s, t) \in \text{supp}(d)} \sum_{p \in Z_{k', s, t}} \sum_i x(s, t)_{i, p} \\ &= Y_{k'} \end{aligned}$$

where we use that the demand is special. Now, we have a contradiction:  $\text{cong}_{k'} \leq Y_{k'}$ , but also  $Y_{k'} < \gamma b_{k'} \leq \Delta_k = \text{cong}_{k'}$ . Thus, it suffices to bound the probability of the event  $\bigcap_{k \in B} Y_k \geq \gamma b_k$ .

Next, notice that by definition, for  $k \in B$ , the sets  $Z_k$  are disjoint. Thus, since the random variables  $X(s, t)_{i, p}$  are negatively associated, by Lemma A.4, we have  $\mathbb{P}[\bigcap_{k \in B} Y_k \geq \gamma b_k] \leq \prod_{k \in B} \mathbb{P}[Y_k \geq \gamma b_k]$ .

We now bound the probability  $\mathbb{P}[Y_k \geq \gamma b_k]$ . We have

$$\begin{aligned} \mathbb{E}[Y_k] &= \mathbb{E}\left[\sum_{(s,t) \in \text{supp}(d)} \sum_{p \in Z_{k,s,t}} \sum_i X(s,t)_{i,p}\right] \\ &= \sum_{(s,t)} \sum_{p \in Z_{k,s,t}} \sum_i \mathbb{I}[(s,t) \in \text{supp}(d)] \mathbb{E}[X(s,t)_{i,p}] \\ &= \sum_{(s,t)} \sum_{p \in Z_{k,s,t}} \sum_i \frac{d(s,t)}{\text{cnt}_G(s,t)} \mathbb{P}[R(s,t) = p] \\ &= \sum_{(s,t)} \sum_{p \in Z_{k,s,t}} d(s,t) \mathbb{P}[R(s,t) = p] \\ &\leq \sum_{(s,t)} \sum_{\substack{p \in \text{supp}(R(s,t)) \\ e_k \in p}} d(s,t) \mathbb{P}[R(s,t) = p] \\ &\leq \text{cong}(R, d). \end{aligned}$$

Let  $\delta = \frac{\gamma}{\mathbb{E}[Y_k]}$ . By the above,  $\delta \geq \alpha + m^{16(h+7)\alpha^{-1}} \geq 2$ . Thus, as  $Y_k$  is a sum of disjoint negatively associated  $\{0, 1\}$ -random variables, by Chernoff for negatively associated random variables (Lemma A.5),

$$\mathbb{P}[Y_k \geq \gamma b_k] = \mathbb{P}[Y_k \geq \delta b_k \mathbb{E}[Y_k]] \leq \exp\left(\frac{-\mathbb{E}[Y_k] \delta b_k \ln(\delta b_k)}{4}\right).$$

Since  $\mathbb{E}[Y_k] \delta = \gamma$  and  $\delta \geq m^{16(h+7)\alpha^{-1}}$ , we have  $\ln(\delta b_k) \geq \ln(\delta) \geq 16(h+7)\alpha^{-1} \ln m$ , thus,

$$\mathbb{P}[Y_k \geq \gamma b_k] \leq \exp\left(-\gamma b_k \frac{16(h+7)}{4\alpha} \ln m\right) = m^{-\gamma b_k \frac{4(h+7)}{\alpha}}.$$

Now, with an upper bound on  $\mathbb{P}[Y_k \geq \gamma b_k]$ , we can upper bound the product of these terms. Since  $b$  is a bad pattern, we have  $\sum_{k \in [|E|]} \gamma b_i = \sum_{k \in B} \gamma b_i \geq \frac{1}{4}D$ . Thus, as desired,

$$\begin{aligned} \mathbb{P}\left[\bigcap_{k \in [|E|]} \Delta_k \geq \gamma b_k\right] &\leq \prod_{k \in B} \mathbb{P}[Y_k \geq \gamma b_k] \\ &\leq m^{-(\sum_{k \in B} \gamma b_k) \frac{4(h+7)}{\alpha}} \\ &\leq m^{-\frac{(h+7)D}{\alpha}}. \quad \square \end{aligned}$$

With the proof of Lemma 5.14 complete, we can now finish the proof of Lemma 5.6. Let  $\mathcal{B}$  be the set of bad patterns, and let  $E_b$  be the event that, for bad pattern  $b \in \mathcal{B}$ , we have  $\bigcap_{k \in [|E|]} \Delta_k \geq \gamma b_k$ .

By Lemma 5.12, if  $\sum_k \Delta_k > \frac{1}{2}D$ , there exists a bad pattern  $b$  such that  $E_b$  happens. Thus, by a union bound,

$$\mathbb{P}\left[\sum_k \Delta_k \leq \frac{1}{2}D\right] \geq 1 - \mathbb{P}\left[\bigcup_{b \in \mathcal{B}} E_b\right] \geq 1 - \sum_{b \in \mathcal{B}} \mathbb{P}[E_b].$$

Now, by Lemma 5.13 we have  $|\mathcal{B}| \leq m^{4D/\alpha}$ , and by Lemma 5.14 we have  $\mathbb{P}[E_b] \leq m^{-(h+7)D/\alpha}$ . Thus,

$$\sum_{b \in \mathcal{B}} \mathbb{P}[E_b] \leq \sum_{b \in \mathcal{B}} m^{-(h+7)D/\alpha} \leq m^{4D/\alpha} m^{-(h+7)D/\alpha} = m^{-(h+3)D/\alpha}.$$

Thus, by Lemma 5.10,  $\mathcal{P}$  has the desired property with failure probability  $m^{-(h+3)D/\alpha}$ . Finally,

$$D = \sum_{s,t} d(s,t) = \sum_{(s,t) \in \text{supp}(d)} \text{cnt}_G(s,t) \geq \alpha |\text{supp}(d)|,$$

thus  $m^{-(h+3)D/\alpha} \leq m^{-(h+3)|\text{supp}(d)|}$ , as desired. This completes the proof of Lemma 5.6.

## A NEGATIVE ASSOCIATION

In this appendix we state the definitions and Lemmas regarding negatively associated random variables used in the proof of Lemma 5.6. For an overview of negative association, see [17].

**Definition A.1** (Negatively Associated Random Variables). *Random variables  $X = (X_1, \dots, X_n)$  are negatively associated if for every two functions  $f(X)$  and  $g(X)$  that depend on disjoint sets of indices  $I$  and  $J$  of  $[n]$ , that are either both monotone increasing or monotone decreasing, we have*

$$\mathbb{E}[f(X)g(X)] \leq \mathbb{E}[f(X)] \mathbb{E}[g(X)].$$

The following two lemmas let us easily show the negative-associatedness of random variables.

**Lemma A.2.** *If  $X = (X_1, \dots, X_n)$  are zero-one random variables, exactly one of which is 1, they are negatively associated.*

**Lemma A.3.** *If  $X = (X_1, \dots, X_n)$  are negatively associated,  $Y = (Y_1, \dots, Y_m)$  are negatively associated, and  $X$  and  $Y$  are independent, then  $Z = (X_1, \dots, X_n, Y_1, \dots, Y_m)$  are negatively associated.*

Applying the definition of negative association to indicator functions on the sum of the relevant indices, we get

**Lemma A.4.** *Let  $X = (X_1, \dots, X_n)$  be negatively associated,  $I_1, \dots, I_m$  be disjoint subsets of  $[n]$ ,  $X_I = \sum_{i \in I} X_i$  and  $\gamma_1, \dots, \gamma_m$  be real numbers. Then,*

$$\mathbb{P}\left[\bigcap_j X_{I_j} \geq \gamma_j\right] \leq \prod_j \mathbb{P}[X_{I_j} \geq \gamma_j].$$

Chernoff bounds hold for negatively associated variables just like they hold for independent random variables. We use the following variant.

**Lemma A.5** (Chernoff Bound for Negative Association). *Let  $X = (X_1, \dots, X_n)$  be negatively associated 0/1-random variables,  $I \subseteq [n]$  be a subset of indices,  $X_I = \sum_{i \in I} X_i$  and  $\mu = \mathbb{E}[X_I]$ . Then, for all  $\delta \geq 2$ ,*

$$\mathbb{P}[X_I \geq \delta \mu] \leq \exp\left(-\frac{1}{4} \delta \mu \ln(\delta)\right).$$

## ACKNOWLEDGMENTS

The author ordering was randomized using <https://www.aeaweb.org/journals/policies/random-author-order/generator>. We kindly ask that citations of this work use `\textcircled{r}` as the author separator instead of commas: Zuzic `\textcircled{r}` Haeupler `\textcircled{r}` Roeykskoe.

This work has been Supported in part by NSF grants CCF-1814603, CCF-1910588, NSF CAREER award CCF-1750808, a Sloan Research Fellowship, and funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 949272).

## REFERENCES

- [1] Miklós Ajtai, János Komlós, and Endre Szemerédi. 1983. An  $O(n \log n)$  sorting network. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*. 1–9.

- [2] Noga Alon and Joel H Spencer. 2016. *The probabilistic method*. John Wiley & Sons.
- [3] Yair Bartal and Stefano Leonardi. 1997. On-Line Routing in All-Optical Networks. In *Automata, Languages and Programming, 24th International Colloquium, ICALP '97, Bologna, Italy, 7-11 July 1997, Proceedings (Lecture Notes in Computer Science, Vol. 1256)*, Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela (Eds.). Springer, 516–526. [https://doi.org/10.1007/3-540-63165-8\\_207](https://doi.org/10.1007/3-540-63165-8_207)
- [4] Marcin Bienkowski, Mirosław Korzeniowski, and Harald Räcke. 2003. A practical algorithm for constructing oblivious routing schemes. In *SPAA 2003: Proceedings of the Fifteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, June 7-9, 2003, San Diego, California, USA (part of FCRC 2003)*, Arnold L. Rosenberg and Friedhelm Meyer auf der Heide (Eds.). ACM, 24–33. <https://doi.org/10.1145/777412.777418>
- [5] Allan Borodin and John E. Hopcroft. 1985. Routing, Merging, and Sorting on Parallel Models of Computation. *J. Comput. Syst. Sci.* 30, 1 (1985), 130–145. [https://doi.org/10.1016/0022-0000\(85\)90008-X](https://doi.org/10.1016/0022-0000(85)90008-X)
- [6] Costas Busch and Malik Magdon-Ismail. 2010. Optimal oblivious routing in hole-free networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Springer, 421–437.
- [7] Costas Busch, Malik Magdon-Ismail, and Jing Xi. 2008. Optimal oblivious path selection on the mesh. *IEEE Trans. Comput.* 57, 5 (2008), 660–671.
- [8] Philipp Czerner and Harald Räcke. 2020. Compact Oblivious Routing in Weighted Graphs. In *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference) (LIPIcs, Vol. 173)*, Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 36:1–36:23. <https://doi.org/10.4230/LIPIcs.ESA.2020.36>
- [9] Matthias Englert and Harald Räcke. 2009. Oblivious routing for the Lp-norm. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 32–40.
- [10] Mohsen Ghaffari and Bernhard Haeupler. 2016. Distributed algorithms for planar networks ii: Low-congestion shortcuts, mst, and min-cut. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms (SODA)*. 202–219.
- [11] Mohsen Ghaffari, Bernhard Haeupler, and Harald Räcke. 2021. Hop-Constrained Expander Decompositions, Oblivious Routing, and Universally-Optimal Distributed Algorithms. *arXiv preprint* (2021).
- [12] Mohsen Ghaffari, Bernhard Haeupler, and Goran Zuzic. 2021. Hop-constrained oblivious routing. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, Samir Khuller and Virginia Vassilevska Williams (Eds.). ACM, 1208–1220. <https://doi.org/10.1145/3406325.3451098>
- [13] Milos D Grammatikakis, D Frank Hsu, Miro Kraetzl, and Jop F Sibeyn. 1998. Packet routing in fixed-connection networks: A survey. *J. Parallel and Distrib. Comput.* 54, 2 (1998), 77–132.
- [14] Anupam Gupta, Mohammad T Hajiaghayi, and Harald Räcke. 2006. Oblivious network design. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. 970–979.
- [15] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tom Leighton. 2007. Semi-oblivious routing: lower bounds. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, Nikhil Bansal, Kirk Pruhs, and Clifford Stein (Eds.). SIAM, 929–938. <http://dl.acm.org/citation.cfm?id=1283383.1283483>
- [16] Chris Harrelson, Kirsten Hildrum, and Satish Rao. 2003. A polynomial-time tree decomposition to minimize congestion. In *SPAA 2003: Proceedings of the Fifteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, June 7-9, 2003, San Diego, California, USA (part of FCRC 2003)*, Arnold L. Rosenberg and Friedhelm Meyer auf der Heide (Eds.). ACM, 34–43. <https://doi.org/10.1145/777412.777419>
- [17] Kumar Joag-Dev and Frank Proschan. 1983. Negative Association of Random Variables with Applications. *The Annals of Statistics* 11, 1 (1983), 286 – 295. <https://doi.org/10.1214/aos/1176346079>
- [18] Christos Kaklamanis, Danny Krizanc, and Thanasis Tsantilas. 1991. Tight Bounds for Oblivious Routing in the Hypercube. *Math. Syst. Theory* 24, 4 (1991), 223–232. <https://doi.org/10.1007/BF02090400>
- [19] Praveen Kumar, Yang Yuan, Chris Yu, Nate Foster, Robert Kleinberg, Petr Lapukhov, Chiunlin Lim, and Robert Soulé. 2018. Semi-Oblivious Traffic Engineering: The Road Not Taken. In *15th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2018, Renton, WA, USA, April 9-11, 2018*, Sujata Banerjee and Srinivasan Seshan (Eds.). USENIX Association, 157–170. <https://www.usenix.org/conference/nsdi18/presentation/kumar>
- [20] Praveen Kumar, Yang Yuan, Chris Yu, Nate Foster, Robert Kleinberg, Petr Lapukhov, Chiun Lin Lim, and Robert Soulé. 2018. Semi-oblivious traffic engineering with smore. In *Proceedings of the Applied Networking Research Workshop, ser. ANRW, Vol. 18*. 21.
- [21] Bradley C Kuszmaul. 1990. Fast, deterministic routing, on hypercubes, using small buffers. *IEEE Trans. Comput.* 39, 11 (1990), 1390–1393.
- [22] Frank Thomson Leighton, Bruce M Maggs, and Satish B Rao. 1994. Packet routing and job-shop scheduling in (congestion+ dilation) steps. *Combinatorica* 14, 2 (1994), 167–186.
- [23] Aleksander Madry. 2010. Fast Approximation Algorithms for Cut-Based Problems in Undirected Graphs. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*. IEEE Computer Society, 245–254. <https://doi.org/10.1109/FOCS.2010.30>
- [24] Bruce M. Maggs, Friedhelm Meyer auf der Heide, Berthold Vöcking, and Matthias Westermann. 1997. Exploiting Locality for Data Management in Systems of Limited Bandwidth. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*. IEEE Computer Society, 284–293. <https://doi.org/10.1109/SFCS.1997.646117>
- [25] Michael O Rabin. 1989. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM (JACM)* 36, 2 (1989), 335–348.
- [26] Harald Räcke. 2002. Minimizing Congestion in General Networks. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*. IEEE Computer Society, 43–52. <https://doi.org/10.1109/SFCS.2002.1181881>
- [27] Harald Räcke. 2008. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, Cynthia Dwork (Ed.). ACM, 255–264. <https://doi.org/10.1145/1374376.1374415>
- [28] Harald Räcke. 2009. Survey on Oblivious Routing Strategies. In *Mathematical Theory and Computational Practice, 5th Conference on Computability in Europe, CIE 2009, Heidelberg, Germany, July 19-24, 2009. Proceedings (Lecture Notes in Computer Science, Vol. 5635)*, Klaus Ambos-Spies, Benedikt Löwe, and Wolfgang Merkle (Eds.). Springer, 419–429. [https://doi.org/10.1007/978-3-642-03073-4\\_43](https://doi.org/10.1007/978-3-642-03073-4_43)
- [29] Harald Räcke and Stefan Schmid. 2019. Compact Oblivious Routing. In *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany (LIPIcs, Vol. 144)*, Michael A. Bender, Ola Svensson, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 75:1–75:14. <https://doi.org/10.4230/LIPIcs.ESA.2019.75>
- [30] Harald Räcke, Chintan Shah, and Hanjo Täubig. 2014. Computing Cut-Based Hierarchical Decompositions in Almost Linear Time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, Chandra Chekuri (Ed.). SIAM, 227–238. <https://doi.org/10.1137/1.9781611973402.17>
- [31] Prabhakar Raghavan and Clark D Tompson. 1987. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* 7, 4 (1987), 365–374.
- [32] Andrea W Richa, M Mitzenmacher, and R Sitaraman. 2001. The power of two random choices: A survey of techniques and results. *Combinatorial Optimization* 9 (2001), 255–304.
- [33] Christian Scheideler. 1998. *Universal routing strategies for interconnection networks*. Vol. 1390. Springer Science & Business Media.
- [34] Eli Upfal. 1984. Efficient schemes for parallel communication. *Journal of the ACM (JACM)* 31, 3 (1984), 507–517.
- [35] Leslie G. Valiant and Gordon J. Brebner. 1981. Universal Schemes for Parallel Communication. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing, May 11-13, 1981, Milwaukee, Wisconsin, USA*. ACM, 263–277. <https://doi.org/10.1145/800076.802479>
- [36] Goran Zuzic, Bernhard Haeupler, and Antti Royskoe. 2023. Sparse Semi-Oblivious Routing: Few Random Paths Suffice. *CoRR* abs/2301.06647 (2023). <https://doi.org/10.48550/arXiv.2301.06647> arXiv:2301.06647 Full version of this paper.