Oblivious Routing on Node-Capacitated and Directed Graphs

Mohammad T. Hajiaghayi* Robert D. Kleinberg*†‡ Tom Leighton*† Harald Räcke§

Abstract

Oblivious routing algorithms for general undirected networks were introduced by Räcke [17], and this work has led to many subsequent improvements and applications. Comparatively little is known about oblivious routing in general *directed* networks, or even in undirected networks with node capacities.

We present the first non-trivial upper bounds for both these cases, providing algorithms for k-commodity oblivious routing problems with competitive ratio $O(\sqrt{k}\log(n))$ for undirected node-capacitated graphs and $O(\sqrt{k}\,n^{1/4}\log(n))$ for directed graphs. In the special case that all commodities have a common source or sink, our upper bound becomes $O(\sqrt{n}\log(n))$ in both cases, matching the lower bound up to a factor of $\log(n)$. The lower bound (which first appeared in [6]) is obtained on a graph with very high degree. We show that in fact the degree of a graph is a crucial parameter for node-capacitated oblivious routing in undirected graphs, by providing an $O(\Delta\,\mathrm{polylog}(n))$-competitive oblivious routing scheme for graphs of degree Δ. For the directed case, however, we show that the lower bound of $\Omega(\sqrt{n})$ still holds in low-degree graphs.

Finally, we settle an open question about routing problems in which all commodities share a common source or sink. We show that even in this simplified scenario there are networks in which no oblivious routing algorithm can achieve a competitive ratio better than $\Omega(\log n)$.

1 Introduction

A routing algorithm for large-scale, unstructured networks like the Internet has to meet many partly conflicting criteria as e.g., enabling quick routing decisions, performing well under a variety of different traffic patterns, and working in a distributed fashion in order to keep control overhead low. In particular the latter issue creates serious difficulties, because a lack of coordination due to distributed routing decisions can easily create high-loaded "hot spots" within the network that usually result in a very bad routing performance.

In this paper we consider the online virtual circuit routing problem in a general network. In this problem a sequence of routing requests is received in an online manner. Each request consists of a pair of nodes that wish to communicate, and a routing algorithm has to establish for each request a path through the network that connects the source to the target. We consider two cost-measures, namely the edge-congestion and the node-congestion, which are defined as the maximum load of a network edge and network node, respectively. (The load of an edge (a node) is the number of traversing paths divided by the capacity of the edge (the node).) This cost-measure reflects the goal of minimizing "hot spots", and thereby prevents the emergence of bottlenecks in the network.

Aspnes et al. [2] give an algorithm for edge-congestion that achieves an optimum competitive ratio of $O(\log n)$, where n denotes the number of vertices in the network. Since their algorithm works for directed graphs, a standard reduction from node-capacitated to directed graphs can be used to obtain the same competitive ratio for node congestion, as well. Unfortunately, this result does not properly address the need for distributed routing decisions, as the algorithm is *adaptive*, i.e., its decisions depend on the current load in the network. Therefore Aspnes et al. assume that there is a centralized decision maker that has full instantaneous knowledge of the traffic situation in the network. This can only be implemented with a lot of control overhead.

For the cost measure of edge-congestion in undirected graphs this problem has been recently solved by [17] at the cost of a slightly higher competitive ratio. It is shown that for any undirected network, there is a routing algorithm with polylogarithmic competitive ratio that is oblivious, i.e., the routing decisions are independent from the current load in the network and a *fixed* flow routing is used for any set of demands.

*Department of Mathematics and Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 200 Technology Square, Cambridge, MA 02139, U.S.A., {hajiagha,rdk,ftl}@theory.lcs.mit.edu

†Akamai Technologies, 8 Cambridge Center, Cambridge, MA 02139, U.S.A.

‡Supported by a Fannie and John Hertz Foundation Fellowship.

§School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A., {harry@andrew.cmu.edu}

Type of oblivious routing	Lower bound	Upper bound
undirected edge-capacitated (single-sink)	$\Omega(\log n)$ [this paper]	$O(\log^2 n \log\log n)$ [13] (previously $O(\log^3 n)$ [17])
undirected edge-capacitated (general case)	$\Omega(\log n)$ [7, 15]	$O(\log^2 n \log\log n)$ [13] (previously $O(\log^3 n)$ [17])
undirected node-capacitated (single-sink)	$\Omega(\sqrt{n})$ [this paper]	$O(\min(\sqrt{n}\log n, \Delta\log^2 n \log\log n))$ [this paper]
undirected node-capacitated (general case)	$\Omega(\sqrt{n})$ [this paper]	$O(\min(\sqrt{k}\log^{3/2} n, \Delta\log^2 n \log\log n))$ [this paper]
bounded degree directed (single-sink)	$\Omega(\sqrt{n})$ [this paper]	$O(\sqrt{n}\log n)$ [this paper]
directed (general case)	$\Omega(\sqrt{n})$ [6]	$O(\sqrt{k}n^{1/4}\log n)$ [this paper]
directed with symmetric demands	$\Omega(\sqrt{n})$ [this paper]	$O(\sqrt{k}\log^{5/2} n)$ [this paper]

Table 1: Competitive ratios for different types of oblivious routing. Here Δ denotes the maximum degree of the graph and k denotes the potential number of commodities ($k = O(n^2)$ in the worst case)

1.1 Our contribution. In this paper we analyze the performance of oblivious routing algorithms for the cost-measure edge-congestion in directed graphs and for the cost-measure node congestion in undirected graphs.[1]

As shown by Azar et al. [6], it is not in general possible to get a polylogarithmic competitive ratio in directed graphs due to a lower bound of $\Omega(\sqrt{n})$. We extend this lower bound to undirected node-capacitated graphs, i.e., to the case of node congestion in undirected graphs.

Then we show that for node capacitated graphs the critical parameter is not the number n of graph nodes but the maximum degree Δ of a node in the network by providing an algorithm that obtains competitive ratio $O(\Delta \operatorname{polylog} n)$. This is done via a reduction to the undirected edge-capacitated case. We show that in terms of Δ this result is almost tight up to a polylogarithmic factor. It is worth mentioning that such a result cannot be obtained for directed graphs, since we show that even for directed graphs of degree at most three, the lower bound on the competitive ratio is $\Omega(\sqrt{n})$.

Unfortunately, the algorithm may be far away from optimum if the maximum degree in the graph is $\Delta = \Theta(n)$. Furthermore, it seems to be very difficult to transfer the technique of using a hierarchical decomposition (and in fact constructing such a decomposition) to the case of undirected node-capacitated graphs to get a better result. This seems to be even harder for directed graphs.

Therefore we introduce a new approach to oblivious routing that gives a substantial improvement in the competitive ratio on high-degree node-capacitated networks and on directed graphs. The performance of our algorithm depends on two parameters. The first one is k; the maximum number of commodities that potentially may be routed. This means that the adversary chooses demands only from a restricted predetermined set of at most k source-sink pairs (where k is at most $\binom{n}{2}$). The second parameter is the maximum possible ratio between the throughput of a maximum concurrent flow and the capacity of a sparsest cut. We call this parameter the max-flow min-cut gap and denote it with α. Note that α depends on the underlying graph and on the set of source-sink pairs for which the adversary may create a demand. For example, if we consider general commodities in an undirected (edge-capacitated or node-capacitated) graph $\alpha = O(\log n)$ [11, 12]; if we consider flow problems with symmetric demands in directed graphs (symmetry means that for any two nodes u, v the demand from u to v is equal to the demand from v to u) $\alpha = O(\log^3 n)$ [14, 10], etc. For these cases our algorithm obtains a competitive ratio of $O(\sqrt{\alpha k}\log n) = O(\sqrt{k} \operatorname{polylog} n)$.

Another important implication of our oblivious routing algorithm in directed graphs is that we obtain competitive ratio $O(\sqrt{n}\log n)$ when there is a common source or a common sink that is shared by all commodities.[2] The single-source case arises naturally, for

[1] A simple reduction shows that in directed graphs the cost-measures node congestion and edge congestion are equivalent. Therefore, in the rest of the paper when we consider node capacities we implicitly assume that the underlying graph is undirected.

[2] In the remainder of the paper we will usually refer to this

instance, in communication networks where all clients are receiving their data from a single server such as an HTTP or streaming-media server. Our result in this case is tight up to a logarithmic factor due to the lower bound of $\Omega(\sqrt{n})$ by Azar et al. [6].

As a further result, we disprove the existence of a constant upper bound for single-sink oblivious routing in undirected edge-capacitated graphs by providing an $\Omega(\log n)$ lower bound (the upper bound for this case is $O(\log^2 n \log\log n)$, which is the upper bound for the general multicommodity case [13]). This partially answers a question asked by Azar et al. for oblivious routing with a common source or sink.

The reader is referred to Table 1 to see a complete comparison between the results of this paper and the previous work.

1.2 Related work. The idea of selecting routing paths oblivious to the traffic in the network has been intensively studied for special network topologies, since such algorithms allow for very efficient implementations due to their simple structure. Valiant and Brebner [18] initiate the worst case theoretical analysis for oblivious routing on the hypercube. They design a randomized packet routing algorithm that routes any permutation in $O(\log n)$ steps. This result gives a virtual circuit routing algorithm that obtains a competitive ratio of $O(\log n)$ with respect to edge-congestion.

In [17] it is shown that there is an oblivious routing algorithm with polylogarithmic competitive ratio (w.r.t. edge-congestion) for any undirected graph. However, this result is non-constructive in the sense that only an exponential time algorithm was given for constructing the routing scheme.

This issue was subsequently addressed by Azar et al. [6] who show that the optimum oblivious routing scheme, i.e., the scheme that guarantees the best possible competitive ratio, can be constructed in polynomial time by using a linear program. This result holds for edge-congestion, node-congestion and in arbitrary directed and undirected graphs. Furthermore, they show that there are directed graphs such that every oblivious routing algorithm has a competitive ratio of $\Omega(\sqrt{n})$.

The method by Azar et al. does not give the possibility to derive general bounds on the competitive ratio for certain types of graphs. Another disadvantage of [6] is that it does not give a polynomial time construction of the hierarchy used in [17], which has proven to be useful in many applications (see e.g. [1, 9, 16]). A polynomial time algorithm for this problem was independently given by [8] and [13]. Whereas the first result shows a slightly weaker competitive ratio for the constructed hierarchy than the non-constructive result in the original paper, the second paper by Harrelson, Hildrum and Rao has even improved the competitive ratio to $O(\log^2 n \log\log n)$. This is currently the best known bound for oblivious routing in general undirected graphs.

Other distributed routing and admission control algorithms are also proposed by Awerbuch and Azar [3] and Awerbuch and Leighton [4, 5] which route flows with a rate that is within a $(1+\epsilon)$ factor of the optimal, but these are not real-time algorithms and take at least a polylogarithmic number of rounds to converge.

2 Formal definition of the problem

We represent the network as a graph $G = (V, E)$ (directed or undirected), where V denotes the set of vertices (or nodes) and E denotes the set of edges. We denote the number of vertices by n. The degree of a vertex v is denoted by $d(v)$, and the maximum degree of a node in G by Δ. We will assume that a capacity function cap is given, assigning a capacity (or bandwidth) to either nodes or edges in the graph. This models the physical communication potential of the network resources.

For this network, we are further given a set K of commodities (let $k = |K|$). Each commodity $(s,t) \in K$ specifies a source node $s \in V$ and a target node $t \in V$ that potentially want to communicate. An oblivious routing scheme for K specifies a unit flow between source and target for each commodity in K. The unit flow for a commodity $(s,t) \in K$ defines a "routing rule" that describes how demand between source s and target t is routed through the network. In the rest of the paper, when we do not specify K explicitly, we implicitly assume that K is the set of all $\binom{n}{2}$ node pairs.

For a given set of demands and a given routing algorithm, we define the *absolute load* of an edge (a node) as the amount of data routed along this edge (the node). (Note that a message that starts at node $v \in V$ will contribute to the load of v.) The relative load is the absolute load of an edge or a node divided by the respective capacity. The *node-congestion* is the maximum relative load of any node in the network, and the *edge-congestion* is defined as the maximum relative load of an edge.

The goal is to design an oblivious routing algorithm that always achieves an edge or node congestion that is close to the best possible. To formalize this we introduce the notion of a demand vector D that specifies the demand for every commodity in K. We denote the optimum edge congestion and optimum node congestion

scenario as the *single-sink case*. However, all results for a single sink hold for a single source, as well.

that can be obtained for demands D with $\mathrm{OPT_E(D)}$ and $\mathrm{OPT_V(D)}$, respectively. When there is no ambiguity we use $\mathrm{OPT_E(D)}$ and $\mathrm{OPT_V(D)}$ also to denote the acutal routing that achieves the optimum congestion.

Let for some oblivious routing strategy OBL, $\mathrm{congestion}_E(\mathrm{OBL}, D)$ and $\mathrm{congestion}_V(\mathrm{OBL}, D)$ denote the edge congestion and node congestion, respectively, achieved for demand vector D. The competitive ratio of the oblivious routing scheme is defined as

$$c_E(\mathrm{OBL}) := \max_D \left\{ \frac{\mathrm{congestion}_E(\mathrm{OBL}, D)}{\mathrm{OPT}_E(D)} \right\}$$

and

$$c_V(\mathrm{OBL}) := \max_D \left\{ \frac{\mathrm{congestion}_V(\mathrm{OBL}, D)}{\mathrm{OPT}_V(D)} \right\} ,$$

for edge congestion and node congestion, respectively.

3 Oblivious routing on directed graphs

We consider oblivious routing algorithms for a directed graph $G = (V, E)$ with a set K of k commodities given in advance. The capacity of an edge $e \in E$ will be denoted by $\mathsf{cap}(e)$. For a commodity $i \in K$, the source and the target for commodity i will be denoted by s_i, t_i respectively. The demand for commodity i will be denoted by d_i. We will assume that the commodities and their demand pattern belong to a class for which the max-flow min-cut gap is at most α, where α is a parameter known to the algorithm and depends on the specifics of the type of flow problem being considered. Thus, for instance, $\alpha = 1$ if we are considering single-sink or single-source flow problems, $\alpha = O(\log n)$ if we are considering general multicommodity flows in undirected graphs, $\alpha = O(\log^3 n)$ if we are considering multicommodity flow problems with symmetric demands in directed graphs, and $\alpha = O(\sqrt{n})$ if we are considering general multicommodity flows in directed graphs.

THEOREM 3.1. *There is an oblivious routing algorithm* OBL *that achieves competitive ratio* $c_E(\mathrm{OBL}) = O(\sqrt{\alpha k} \log n)$ *on directed graphs.*

Proof. The high level idea for the proof is as follows. We partition the set of commodities into subsets $K_1, K_2, \ldots$ and bound the load created on an edge by any of these subsets individually. (Each subset may use a different routing strategy). We show that if the demands of a particular routing problem can be routed (by an optimal algorithm) with congestion C, i.e., $C = \mathrm{OPT_E(D)}$, an edge will get load at most $\sqrt{\alpha k} \cdot C$ from such a subset. Moreover, each edge gets non-negligible load from only $O(\log n)$ of the subsets. This means that the oblivious routing algorithm is $O(\sqrt{\alpha k} \log n)$-competitive.

We first define a specific multicommodity flow problem in G, with commodity set K, called the *canonical flow problem.* In the canonical flow problem, the demand $\mathsf{dem}(i)$ for commodity $i \in K$ is equal to the capacity of a minimum cut in G separating s_i from t_i. For a set of commodities $T \subseteq K$ we use the notation $\mathsf{dem}(T)$ to denote $\sum_{i \in T} \mathsf{dem}(i)$. For an edge set S, we use the notation $\mathsf{cap}(S)$ to denote the sum $\sum_{e \in S} \mathsf{cap}(e)$. The set of commodities $i \in K$ such that S contains a cut separating s_i from t_i will be denoted by $T(S)$.

LEMMA 3.1. *G contains a (possibly empty) edge set S such that*

$$\frac{\mathsf{dem}(T(S))}{\mathsf{cap}(S)} > \sqrt{k/\alpha}, \tag{3.1}$$

and such that the flow problem "has no bottlenecks outside S," i.e. the flow problem with graph $G \setminus S = (V, E \setminus S)$, commodities $K \setminus T(S)$, and demands $\mathsf{dem}(i)$ has a solution with congestion $\leq \sqrt{\alpha k}$. Such a set S may be computed in polynomial time, given an oracle which takes a flow problem as input and produces a cut whose sparsity is within a factor α of the maximum concurrent flow.

Proof. To construct such a set S, we initialize S to be the empty set. As long as the flow problem has a bottleneck outside S, the α-approximate max-flow min-cut theorem ensures that there is an edge cut S' in $G \setminus S$ and a set of commodities $T' \subseteq K \setminus T(S)$ such that every commodity in T' is separated by S' in $G \setminus S$, and $\mathsf{dem}(T')/\mathsf{cap}(S') > \sqrt{\alpha k}/\alpha = \sqrt{k/\alpha}$. Now

$$\frac{\mathsf{dem}(T(S \cup S'))}{\mathsf{cap}(S \cup S')} \geq \frac{\mathsf{dem}(T(S) \cup T')}{\mathsf{cap}(S \cup S')}$$

$$= \frac{\mathsf{dem}(T(S)) + \mathsf{dem}(T')}{\mathsf{cap}(S) + \mathsf{cap}(S')} > \sqrt{k/\alpha},$$

hence the edge cut $S \cup S'$ satisfies (3.1). So we may replace S with $S \cup S'$ and repeat the process. □

To define the commodity set K_1, let S_1 be the edge set in G whose existence is guaranteed by the lemma. Define the *deletion threshold* for S_1 to be

$$\theta(S_1) = \frac{\sqrt{k/\alpha} \cdot \mathsf{cap}(S_1)}{2|T(S_1)|}.$$

Let $K_1 = \{i \in K : \mathsf{dem}(i) \geq \theta(S_1)\}$. Delete all the commodities in K_1 from K, and recursively apply the same procedure on the set of remaining commodities, to define cuts $S_2, S_3, \ldots$ and commodity sets $K_2, K_3, \ldots$.

Having specified how the commodity sets $K_1, K_2, \ldots$ are defined, we will now define the oblivious routing scheme for K_j. Let $A_j = K_j \setminus T(S_j)$, $B_j = K_j \cap T(S_j)$. By construction of S_j, the flow problem with commodities A_j and demands $\mathsf{dem}(i)$ has a solution with congestion $\le \sqrt{\alpha k}$. We may modify this flow solution, without increasing the congestion of any edge by a factor of more than 2, so that it does not use any edge of capacity $< \theta(S_j)/2n^2$. (Each commodity $i \in A_j$ has $\mathsf{dem}(i) > \theta(S_j)$, hence less than half of its flow is routed along flow paths containing an edge of capacity $< \theta(S_j)/2n^2$. Reroute all such flow, by at most doubling the flow on each path which avoids such low-capacity edges.) This defines the routing scheme for commodities in A_j. For a commodity $i \in B_j$, the routing is determined by computing a congestion-minimizing single commodity flow for i, then modifying it as above so that the congestion of each edge increases by a factor of at most 2, and no edges of capacity $< \theta(S_j)/2n^2$ are utilized.

We claim that this oblivious routing scheme for the commodity set K_j is $O(\sqrt{\alpha k})$-competitive for flow problems limited to that commodity set. If $\{d_i : i \in K_j\}$ is a demand pattern admitting a flow solution with congestion C, then $d_i \le C\mathsf{dem}(i)$ for each i, so the commodities in A_j place congestion at most $2C\sqrt{\alpha k}$ on any edge. Turning our attention now to the commodities in B_j, let $W = \sum_{i \in B_j} d_i$. Each commodity $i \in B_j$ places a congestion of at most $2d_i/\mathsf{dem}(i)$ on any edge. (This is because i is routed according to a 2-approximation to the congestion-minimizing single-commodity flow, and $\mathsf{dem}(i)$ was defined to be the maximum amount of flow that can be routed with congestion 1.) Recalling that $\mathsf{dem}(i) \ge \theta(S_j)$ for each $i \in B_j$, we find that the commodities in B_j produce a combined congestion of at most $W/\theta(S_j)$ on any edge. But $W \le C\mathsf{cap}(S_j)$, since each commodity in B_j is separated by the cut S_j and there exists a flow solution with congestion C. This gives, for each edge e,

$$\begin{aligned}
\text{congestion}(e) &\le \frac{W}{\theta(S_j)} \\
&\le \frac{C\mathsf{cap}(S_j)}{\theta(S_j)} \\
&= \frac{2C\mathsf{cap}(S_j)|T(S_j)|}{\sqrt{k/\alpha}\cdot \mathsf{cap}(S_j)} \\
&= C \cdot 2\sqrt{\alpha} \cdot \frac{|T(S_j)|}{\sqrt{k}} \\
&\le 2C\sqrt{\alpha k}.
\end{aligned}$$

We have now seen, for each edge e, that if the optimal flow solution has congestion C, then in the oblivious flow solution each commodity set K_j will place congestion at most $4C\sqrt{\alpha k}$ on edge e. It remains to bound the number of commodity sets which place a non-negligible load on e. Observe that $\theta(S_j) < \frac{\mathsf{dem}(T(S_j))}{2|T(S_j)|}$, i.e. the deletion threshold for S_j is less than half the average canonical demand of commodities crossing S_j. But all such commodities have demand $< \theta(S_{j-1})$ (since they were not deleted in a prior step), so $\theta(S_j) < \frac{1}{2}\theta(S_{j-1})$. This means that the deletion threshold drops by a factor of at least 2 in each stage of the construction. For any edge e, the commodities in K_j do not use e unless $\theta(S_j) \le 2n^2\mathsf{cap}(e)$. Let j_0 denote the least such j, and let $j_1 = j_0 + 8\log_2(n)$. If $j > j_1$ then $\theta(S_j) < \mathsf{cap}(e)/n^2 \le \mathsf{cap}(e)/k$, so all commodities in $\cup_{j>j_1} K_j$ could route their entire demand across edge e and still produce congestion $< C$. That means the congestion of e is $O(C + (j_1 - j_0)(\sqrt{\alpha k})C) = O(C \cdot \sqrt{\alpha k}\log n)$ as desired. □

4 Oblivious routing on undirected node capacitated graphs

In this section, we present an upper bound $O(\Delta \operatorname{polylog} n)$ for the performance ratio of oblivious routing on undirected node-capacitated graphs. We note that this upper bound is almost tight within a polylogarithmic factor, since in the construction mentioned in the proof of Theorem 5.2 (see Section 5), the lower bound is $\Omega(\sqrt{n}) = \Omega(\Delta)$.

THEOREM 4.1. *There is an oblivious routing algorithm* OBL *that achieves competitive ratio $c_V(\mathrm{OBL}) = O(\Delta \log^2 n \log\log n)$ on undirected node-capacitated graphs.*

Proof. The proof follows from a reduction from the node-capacitated case to the edge-capacitated case and then use the result of Räcke [17] (and its improvement by Harrelson, Hildrum, and Rao [13]) for the edge-capacitated case.

From the given undirected graph $G = (V, E)$, we construct another graph $G' = (V', E')$ as follows. For each $v \in V$, we place $d(v)$ vertices $v_1, v_2, \ldots, v_{d(v)}$ in V' (we say vertices $v_1, v_2, \ldots, v_{d(v)}$ in V' are *clones* of vertex v in V). Edges are defined as follows. We order the neighbors of a vertex v from 1 to $d(v)$ arbitrarily, and place an edge from v_i to u_j if and only if v is the ith neighbor of u and u is the jth neighbor of v in the aforementioned orderings. The capacities of all such edges are ∞. In addition, we put a clique on vertices $v_1, v_2, \ldots, v_{d(v)} \in V'$ corresponding to vertex $v \in V$. The capacity of each edge in this clique is $c(v)/d(v)$.

Now, we construct the oblivious routing on G' using the algorithm of Harrelson et al. [13] with competitive ratio $O(\log^2 n \log\log n)$ and obtain the final oblivious routing in G, by contracting all vertices $v_1, v_2, \ldots, v_{d(v)}$ into one vertex v (and thus a path which goes through clones of a vertex $v \in V$ in G' goes through v in G.) Next, we show that the competitive ratio of the resulting oblivious routing is the desired bound.

Consider a demand vector (matrix) D and the corresponding $\mathrm{OPT}_V(D)$ for the node congestion. We show that we can route the corresponding demand matrix D' in G' with edge congestion at most $2\mathrm{OPT}_V(D)$ (here by corresponding demand matrix we mean for any demand d from u to v in G, there is a demand d from an arbitrary copy u_i, $1 \le i \le d(u)$, to an arbitrary copy v_j, $1 \le j \le d(v)$). Consider a path $P = (v = v^1, v^2, \ldots, v^k = u)$ in G which carries a non-zero ϵ-fraction of the demand from u to v. We consider the corresponding path P' in G between the clones of vertices $v^1, v^2, \ldots, v^k$. The only issue is routing along the cliques. Consider the sub-path v^{i-1}, v^i, v^{i+1} of P. Assume that $\{v_l^{i-1}, v_j^i\}, \{v_k^i, v_q^{i+1}\} \in E'$ (note that $v^{i-1} \neq v^{i+1}$ and thus $k \neq j$). We pass the whole ϵ-fraction through edges $\{v_l^{i-1}, v_j^i\}$ and $\{v_k^i, v_q^{i+1}\}$. However to send the ϵ-fraction of the demand from v_j^i to v_k^i, we send $\frac{\epsilon}{d-1}$-fraction along edge $\{v_j^i, v_k^i\}$ and $\frac{\epsilon}{d-1}$-fraction via each path v_j^i, v_l^i, v_k^i where $l \neq j$ and $l \neq k$. Now, it is easy to see that from any ϵ-fraction of demands which goes through a vertex $v \in V$, each edge in the corresponding clique gets at most a $\frac{\epsilon}{d-1}$-fraction. Since the capacity of each edge is $c(v)/d(v)$, it means the total congestion of edges in the clique is at most $\frac{d(v)\beta}{d(v)-1} \le 2\beta$, where β is the congestion of v in $\mathrm{OPT}_V(D)$. Since the edges between clones of different nodes $u, v \in V$ have infinite capacity, the total congestion of the solution in G' corresponding to $\mathrm{OPT}_V(D)$ in G is at most $2 \cdot \mathrm{OPT}_V(D)$.

Now if we use the oblivious routing of Harrelson et al. [13] in G' the resulting edge congestion is at most $2O(\log^2 n \log\log n)$ of the congestion of the solution constructed from $\mathrm{OPT}_V(D)$ and thus at most $O(\log^2 n \log\log n)\mathrm{OPT}_V(D)$. Now, when we return from the oblivious routing in G' to the oblivious routing in G, we have a blow-up factor of at most $(\Delta - 1)/2$. This is because when we contract all clones of a vertex in order to obtain the final oblivious routing in G, the the congestion of each vertex v becomes at most $\frac{d(v)(d(v)-1)}{2}\frac{1}{c(v)}\frac{c(v)\beta}{d(v)} = \frac{d(v)-1}{2}\beta$ where β is the maximum congestion of an edge in the corresponding clique of v in G' (here $\frac{d(v)(d(v)-1)}{2}$ is the number of edges in the clique and $\frac{c(v)\beta}{d(v)}$ is the absolute load of a single edge). It means the overall congestion of a node using our oblivious routing in G is at most $O(\Delta \log^2 n \log\log n)\,\mathrm{OPT}_V(D)$ as desired. □

5 Lower bounds

In this section, we first disprove the existence of a constant upper bound for single-sink oblivious routing in undirected edge-capacitated graphs by providing an $\Omega(\log n)$ lower bound.

THEOREM 5.1. *There is an undirected graph G such that for any oblivious routing algorithm* OBL, *$c_E(\mathrm{OBL}) = \Omega(\log n)$, even if all commodity pairs share a common sink.*

Proof. The proof is an extension of the proof by Bartal and Leonardi [7] and Maggs et al. [15] for general multicommodity online routing. Consider an $N \times N$ 2-dimensional mesh M, with $N = \sqrt{n-1}$. Without loss of generality, we assume that $n = (2^{k-1}+1)^2+1$ for some integer $k \ge 1$. Let $M[x, y]$ denote the vertex in row x and column y in the mesh (thus for M, $0 \le x, y \le 2^{k-1}$). We add a super-sink s which is connected to all vertices $M[2^{k-1}, y]$ for $0 \le y \le 2^{k-1}$. We consider the scenario in which each node of M is a possible source that may communicate with s. All edge-capacities are one.

The request sequence is defined recursively in $k-1$ stages as follows. At the ith stage of the recursion where $i = 1, 2, \ldots, k-1$, we define the requests for a $2^{k-i} \times 2^{k-i}$ sub-mesh M_i determined from the previous step. We set $M_1 = M$ in the beginning. In the ith stage each vertex $M_i[x, 2^{k-i-1}]$ where $2^{k-i-2} \le x < 3 \times 2^{k-i-2}$ issues a request of demand one. The number of edges of the mesh M_i is $2 \times 2^{k-i} \times (2^{k-i}+1)$. At the ith stage, any amount of the demand congests a path containing at least 2^{k-i-2} edges of sub-mesh M_i even if the path leaves the sub-mesh. The number of requests at this stage is 2^{k-i-1}. Thus the total load by these requests on sub-mesh M_i is at least $2^{k-i-2} \times 2^{k-i-1}$. Hence the average load on edges of M_i is at least $\frac{2^{k-i-2} \times 2^{k-i-1}}{2 \times 2^{k-i} \times (2^{k-i}+1)} \ge 1/32$.

We partition M_i into four submeshes of size $2^{k-i-1} \times 2^{k-i-1}$ such that each submesh contains a corner of M_i. An edge is contained in at most two of the sub-meshes. Therefore, one of the submeshes has average load ad least $((1/32)/2)/4$. This submesh is chosen as submesh M_{i+1} in the following stage. We can easily prove by induction that the average load of the edges in the sub-mesh M_i is at least $i/256$. This means there exists an edge in M_{k-1} with load $\Omega(\log n)$.

On the other hand, if we know the sequence of requests in advance, in each step we can use those edges in M_i which are not in M_{i+1} to route the demands of M_i to the super-sink. In this way, all edges of M_{i+1} are free of load and we can use them for the requests in the subsequent stages (Figure 1a shows an example for $k = 5$).

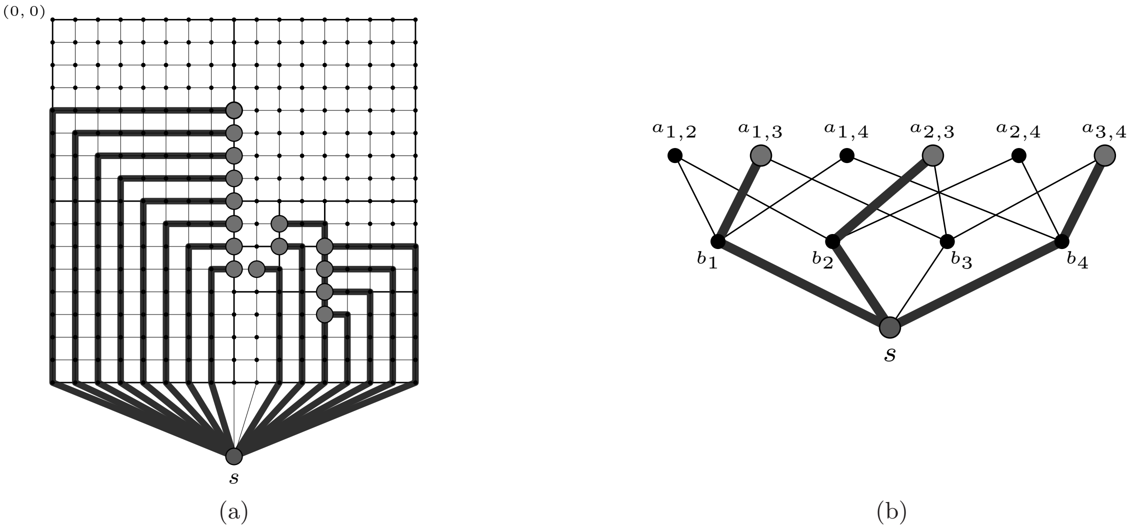

Figure 1: (a) The mesh M with a common sink used in Theorem 5.1. The gray nodes represent a possible set of demand-nodes according to the recursive construction. The routing paths give an optimal routing for these demand nodes. (b) The network for proving the lower bound for node-capacitated oblivious routing.

Thus, we can choose the requests in such a way that the oblivious routing algorithm has congestion $\Omega(\log n)$, while an optimum solution routes all requests with congestion one. This finishes the proof. $\Box$

Note that the above lower bound does not only hold for oblivious routing schemes but for any online algorithm. In addition, Theorem 5.1 shows that the result of Harrelson et al. [13] who give a competitive ratio of $O(\log n \log\log n)$ for oblivious routing on planar graphs is tight up to a factor of $O(\log\log n)$ even for single-sink oblivious routing (since the graph considered above is planar).

The following theorem gives a lower bound of $\Omega(\log n)$ for the single-sink oblivous routing problem in general undirected node-capacitated graphs. This shows that our upper bound of $O(\sqrt{n}\log n)$ given in Section 3 is tight up to a logarithmic factor.

Theorem 5.2. *There is an undirected graph G such that for any oblivious routing algorithm* OBL, $c_V(\text{OBL}) = \Omega(\sqrt{n})$*, where the set of commodity pairs shares a common sink.*

Proof. We essentially use the same example as introduced by Azar et al. [6]. It consists of a graph G with three levels. The first level contains $\binom{k}{2}$ nodes denoted by a_{ij} for $1 \le i < j \le k$. The second level contains k nodes denoted by b_i for $1 \le i \le k$ and the third level contains a super-sink s. Each node a_{ij} is connected via two undirected edges to nodes b_i and b_j. Further, each node b_i is connected to the sink. All nodes have capacity one, execept the sink s which has infinite capacity.

Any oblivious routing scheme defines a unit flow from each node a_{ij} on the first level to the sink. Hence, there is a node b_x such that at least a flow of $\binom{k}{2}/k$ from its direct neighbors on the first level (i.e., nodes a_{ix} for $i < x$ and nodes a_{xj} for $j > x$). Suppose that all these neighbors of b_x send a demand of one and all other demands are zero. Then the oblivous algorithm has congestion at least $\binom{k}{2}/k$ at node b_x. However, the optimum algorithm can route this demand with congestion 1 by using the paths $a_{ix} - b_i - s$ for demands from nodes a_{ix} and the paths $a_{xj} - b_j - s$ for demands from a_{xj}-nodes. The proof follows immediately, since $k = \Omega(\sqrt{n})$. Figure 1b illustrates the case for $k = 4$ and $x = 3$. $\Box$

The following theorem shows that the result of Theorem 4.1, which provides an $O(\Delta \operatorname{polylog} n)$-competitive oblivious routing algorithm for node-capacitated undirected graphs, cannot be extended to directed graphs.

Theorem 5.3. *There is a directed graph G of degree at most three such that for any oblivious routing algorithm* OBL, $c_E(\text{OBL}) = \Omega(\sqrt{n})$*, where the set of commodity pairs share a common sink.*

Proof. Consider the directed version of the graph used in the proof of Theorem 5.2, i.e., each edge is directed towards the higher level node. Now, we replace each node b_i and the sink s by a directed tree, in which edges are directed towards the root. A graph edge that is directed to a b_i or to the sink s is attached to a unique leaf node of the corresponding tree, and the outgoing edge of node b_i is attached to the root. Since all flow that is sent to such a b_i-tree has to traverse the outgoing edge attached to the root, the analysis of Theorem 5.2 works for the new graph. Since the degree of each node in the new graph is at most three, the theorem follows. □

Finally, we note that the construction considered in the proof of Theorem 5.2 also shows that the performance of oblivious routing with symmetric demands in edge-capacitated directed graphs is at least $\Omega(\sqrt{n})$. To see this, consider the directed version of the graph G in which all edges are directed towards the higher level node, and add edges with infinite capacity from s to all nodes on the first level. In this graph symmetry of demands does not change the problem and therefore the $\Omega(\sqrt{n})$ lower bound holds.

6 Discussion and open problems

In this paper, we presented non-trivial upper and lower bounds for oblivious routing in node-capacitated and directed graphs. In particular, we obtained almost tight upper and lower bounds (up to an $O(\log n)$ factor) for single-sink oblivious routing.

The main open problem is whether we can obtain a competitive ratio of $O(\sqrt{n}\,\mathrm{polylog}\, n)$ for oblivious routing in general node-capacitated graphs when the number k of potential commodities is asymptotically larger than n (in this case, we have an upper bound of $O(\sqrt{k}\,\mathrm{polylog}\, n)$). The problem for directed graphs seems more challenging, since the best known max-flow min-cut gap is $O(\sqrt{n})$ in this case (In all currently known algorithms for oblivious routing this max-flow min-cut gap plays an important role).

The results of Harrelson et al. [13] for the competitive ratio of oblivious routing in edge-capacitated undirected graphs are parameterized in terms of the max-flow min-cut gap α of the considered graph-class. They obtain an $O(\alpha \log n \log\log n)$-competitive oblivious routing algorithm. On planar graphs, e.g., α is constant. Unfortunately, our reduction from the node-capacitated case to the edge-capacitated case does not preserve planarity. Hence, it is an interesting open problem whether for planar graphs an $O(\Delta\,\mathrm{polylog}\, n)$-competitive oblivious routing algorithm can be obtained for the node-capacitated version of the problem.

Last but not least, there is a slight gap of $O(\log n)$ between the lower bounds and the upper bounds for the single-sink case in both directed and undirected graphs. Closing these gaps is another interesting open problem.

References

[1] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. S. Naor, *A general approach to online network optimization problems*, in Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2004, pp. 577–586.

[2] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, *On-line routing of virtual circuits with applications to load balancing and machine scheduling*, J. ACM, 44 (1997), pp. 486–504.

[3] B. Awerbuch and Y. Azar, *Local optimization of global objectives: Competitive distributed deadlock resolution and resource allocation*, in Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1994, pp. 240–249.

[4] B. Awerbuch and T. Leighton, *A simple local-control approximation algorithm for multicommodity flow*, in Proceedings of the 34th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1993, pp. 459–468.

[5] ———, *Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks*, in Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, ACM Press, 1994, pp. 487–496.

[6] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke, *Optimal oblivious routing in polynomial time*, in Proceedings of the thirty-fifth ACM symposium on Theory of computing, ACM Press, 2003, pp. 383–388.

[7] Y. Bartal and S. Leonardi, *On-line routing in all-optical networks*, Theoret. Comput. Sci., 221 (1999), pp. 19–39. ICALP '97 (Bologna).

[8] M. Bienkowski, M. Korzeniowski, and H. Räcke, *A practical algorithm for constructing oblivious routing schemes*, in Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures, ACM Press, 2003, pp. 24–33.

[9] C. Chekuri, S. Khanna, and F. B. Shepherd, *The all-or-nothing multicommodity flow problem*, in Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, ACM Press, 2004, pp. 156–165.

[10] G. Even, J. Naor, B. Schieber, and M. Sudan, *Approximating minimum feedback sets and multicuts in directed graphs*, Algorithmica, 20 (1998), pp. 151–174.

[11] N. Garg, V. V. Vazirani, and M. Yannakakis, *Approximate max-flow min-(multi)cut theorems and their applications*, SIAM J. Comput., 25 (1996), pp. 235–251.

[12] ———, *Multiway cuts in node weighted graphs*, J. Algorithms, 50 (2004), pp. 49–61.

[13] C. Harrelson, K. Hildrum, and S. Rao, *A polynomial-time tree decomposition to minimize congestion*, in Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures, 2003, pp. 34–43.

[14] P. N. Klein, S. A. Plotkin, S. Rao, and É. Tar-

DOS, *Approximation algorithms for Steiner and directed multicuts*, J. Algorithms, 22 (1997), pp. 241–269.

[15] B. M. Maggs, F. M. auf der Heide, B. Vocking, and M. Westermann, *Exploiting locality for data management in systems of limited bandwidth*, in Proceedings of the 38th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1997, pp. 284–293.

[16] B. M. Maggs, G. L. Miller, O. Parekh, R. Ravi, and S. L. M. Woo, *Solving symmetric diagonally-dominant systems by preconditioning.* manuscript, 2003.

[17] H. Räcke, *Minimizing congestion in general networks*, in Proceedings of the 43rd Symposium on Foundations of Computer Science, IEEE Computer Society, 2002, pp. 43–52.

[18] L. G. Valiant and G. J. Brebner, *Universal schemes for parallel communication*, in Proceedings of the 13th ACM symposium on Theory of computing, 1981, pp. 263–277.