

Oblivious Routing in Directed Graphs with Random Demands

April 3, 2006

1 Introduction

The concept of oblivious routing aims at developing routing algorithms that base their routing decisions only on local knowledge and that therefore can be implemented very efficiently in a distributed environment. Traditionally, for an oblivious routing algorithm the routing path chosen between a source s and a target t may only depend on s and t . Our focus in this lecture is to present oblivious routing algorithms that aim to minimize the congestion, which is defined as the maximum relative load of a network edge, in the context of directed graphs. (The relative load of an edge is the number of routing paths traversing the edge divided by the capacity of the edge.)

Our algorithms were presented by Hajiaghayi, Kim, Leighton and Räcke in [2]. The model that we will present is unusual in two main points. First, the demands between node-pairs are not worst-case as in the standard competitive analysis, but are drawn from a demand distribution that is known in advance. In many practical applications this assumption is justified. Second, the routing algorithm is augmented by the possibility of *demand-dependence*, i.e., the path (or flow) chosen for a node-pair may not only depend on the pair, but also on the demand-value for this pair (note that this is still local knowledge in the sense that the node that sets up a routing path should know about the corresponding demand).

We will show that for any directed graph if the demands for different node-pairs are independent there is a demand-dependent oblivious routing algorithm that is within $O(\log^2 n)$ of the optimum congestion, with high probability. On the other hand, we show that the concept of demand-dependence is necessary for obtaining efficient algorithms. We show that there are directed networks in which traditional oblivious algorithms perform very badly.

1.1 Formal Definition of the Problem

The graph terminology is as follows. The network is represented 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 . We will assume that a capacity function cap is given, assigning a capacity (or bandwidth) to edges in the graph. This models the physical communication potential of the network resources.

We consider oblivious routing when the demands between node-pairs are chosen randomly from a known demand distribution. An oblivious routing scheme consists of a unit flow from s to t for every node-pair (s, t) . The flow for a pair s, t determines how the demand from s to t is routed.

The goal is to minimize the congestion which is defined, as follows. For a given demand-matrix D and a given routing algorithm, we define the *absolute load* of an edge as the amount of flow routed along this edge. The relative load is the absolute load of an edge divided by its capacity. The *edge-congestion* (or just *congestion*) is defined to be the maximum relative load of an edge.

We define $C_{\text{obl}}(D)$ to be the edge-congestion of the routing guided by the flow paths of oblivious routing for the demand matrix D , in which each simple path from s to t for the commodity pair (s, t) gets flow proportional to its share in the routing corresponding to the unit demand. Let $C_{\text{opt}}(D)$ be the optimum edge-congestion for the demand-matrix D , which can be obtained by solving a linear program (we drop D when it is clear from the context). We call the ratio $C_{\text{obl}}(D)/C_{\text{opt}}(D)$ the *competitive ratio* for a demand matrix D .

The goal is to create an oblivious routing scheme (based on knowledge of the demand distribution) such that the competitive ratio is small with high (or at least constant) probability, when choosing the demands from the distribution.

The result in the next section shows that, for general distributions, this goal cannot be achieved. There are graphs for which under general demand distributions any oblivious routing scheme has a large competitive ratio with high probability. Therefore, we usually (unless otherwise stated) refer to demand-distributions in which the demands for different source-target pairs are independent. However, the demand for an individual source-target pair may be chosen according to an arbitrary distribution.

Furthermore, we differentiate between *demand-dependent* and *demand-independent* oblivious routing algorithms. A demand-independent oblivious routing scheme consists of one flow for every source-target pair as described above. A *demand-dependent* routing scheme however defines several $s \rightarrow t$ flows for every pair (s, t) . Which of these flows is used for routing between s and t may depend on the demand between s and t . This concept of demand-dependence makes an oblivious algorithm more powerful while still preserving the fact that only local knowledge is used for making routing decisions (clearly, the demand to be routed between s and t is known to the node that sets up the routing path). We show that in our model of randomized demands the possibility of demand-dependence is extremely important in order to obtain good routing algorithms.

The following proposition can be easily seen from our model definition.

Proposition 1 *Assume there is an upper bound r such that for each commodity pair the ratio of the maximum demand-value to the minimum demand-value is at most r . Then we can construct an oblivious routing which is always r -competitive.*

We denote the demand from s to t by $D[s, t]$. We refer to the demand-matrix D as a demand-vector D indexed by all commodities. In this case, $\text{demand}(j)$, denotes the demand for commodity j , and $\text{mincut}(j)$, denotes the minimum capacity of an edge set whose removal disconnects the sink of the commodity j from its source (in a “directed” sense).

2 Lower Bounds

Let us start by justifying our model by presenting lower bounds for oblivious routing with randomized demands.

Theorem 2 *There exists a graph for which every demand-independent oblivious routing algorithm has competitive ratio $\Omega(\frac{\sqrt{n}}{\log n})$ with high probability.*

Proof. Consider a graph G consisting of k sources s_1, \dots, s_k , k sinks t_1, \dots, t_k , and two additional nodes u and v . Each source s_i is connected to t_i via a directed edge of capacity 1. Furthermore, there exist infinite capacity edges (s_i, u) and (v, t_i) for each i , and u is connected to v via an edge with capacity \sqrt{k} . Note that each source-target pair s_i, t_i is connected via two paths; one with capacity 1 which is exclusive to the pair s_i, t_i ; and one with capacity \sqrt{k} via edge (u, v) that may be shared by all pairs.

Consider the demand distribution that assigns a demand of \sqrt{k} to commodity i with probability $p := \frac{2(\alpha+1)\ln n}{k}$, and a demand of 1 with probability $1 - p$. A Chernoff Bound shows that with high probability ($\geq 1 - \frac{1}{k^{\alpha+1}}$) the number of high-demand pairs (i.e., pairs with demand \sqrt{k}) is only $\alpha \cdot O(\log n)$. In this case, by routing the high-demand pairs over the high capacity edge, and the remaining commodities directly along the corresponding (s_i, t_i) edge, the optimum congestion is $\alpha \cdot O(\log k)$.

However, an oblivious routing algorithm has to fix the routing paths without knowing the high-demand pairs. We show that every oblivious routing algorithm creates congestion $\Omega(\sqrt{k})$, w.h.p.

Fix an oblivious routing algorithm. Let h denote the number of source-target pairs for which the algorithm routes at least $1/2$ of the demand along the high capacity edge (u, v) . Clearly, if $h > k/2$ the load on edge (u, v) will be at least $\frac{1}{2}h/\sqrt{k} \geq \sqrt{k}/4$.

On the other hand, if $h < k/2$ there exists at least $k/2$ pairs that route at least $1/2$ of their demand via edge (s_i, t_i) . The probability that none of these pairs gets a high demand is only $(1 - \frac{2(\alpha+1)\ln n}{k})^{k/2} \leq \frac{1}{k^{\alpha+1}}$. This means that with high probability one of them will get a high demand which leads to a congestion of $\Omega(\sqrt{k})$.

Combining the result for the optimum and the oblivious algorithm gives that with high probability the competitive ratio is $\Omega(\sqrt{k}/\log k)$. ■

The above theorem shows that demand-dependence is essential for deriving oblivious routing algorithms with low competitive ratio in the randomized demand model. Note that the above proof also rules out an oblivious routing scheme that is good with respect to any demand-distribution.

Further, it is worth mentioning that the concept of demand-dependence does not help for the standard worst case scenario of oblivious routing since the counter-example (see [1]) that proves a lower bound of \sqrt{n} on the competitive ratio in directed graphs works with uniform demands.

Next, we show that in order to obtain a polylogarithmic competitive ratio the assumption that demands for different commodities are independent is necessary.

Theorem 3 *In the case of general demand-distributions, i.e. when demands for different pairs are not necessarily independent, there exist graphs for which even a demand-dependent oblivious routing scheme has competitive ratio $\Omega(n^{1/3}/\log n)$, with high probability.*

Proof. First, we consider the graph G constructed in the proof of Theorem 2. For each vertex $s_i \in V(G)$, $1 \leq i \leq k$, we place \sqrt{k} vertices $s_i^1, s_i^2, \dots, s_i^{\sqrt{k}}$ which have a directed edge toward s_i and we call them *sources corresponding to s_i* . Note that here $n = O(k\sqrt{k})$. Again all edge-capacities are one. The commodity pairs are all $s_i^h \rightarrow t_i$ for $1 \leq i \leq k$ and $1 \leq h \leq \sqrt{k}$. For each i , with probability $\frac{2\log k}{k}$ all s_i^h , $1 \leq h \leq \sqrt{k}$, get demand one; otherwise, with probability $\frac{6\log k}{\sqrt{k}}$, each s_i^h , $1 \leq h \leq \sqrt{k}$, individually gets a demand one. Thus in total each s_i^h , $1 \leq h \leq \sqrt{k}$ gets demand one with probability $p = \frac{2\log k}{k} + (1 - \frac{2\log k}{k})\frac{6\log k}{\sqrt{k}}$ and gets demand zero with probability $1 - p$. Note that here the demands are only 0 and 1 and thus demand-dependent oblivious routing and demand-independent oblivious routing are equivalent.

First we show that with high probability C_{opt} is in $O(\log k)$. We call an s_i , $1 \leq i \leq k$, *under-saturated* if the number of sources among s_i^h 's, $1 \leq h \leq \sqrt{k}$, which get demand one is at most $8\log k$; we call s_i *saturated* if all s_i^h 's, $1 \leq h \leq \sqrt{k}$, get demand one; otherwise, we call s_i *almost-saturated*. By the Chernoff bound (see a similar case in the proof of Theorem 2), for each $1 \leq i \leq k$, conditioned on the event that s_i is not saturated, with high probability it is under-saturated. Again using the Chernoff bound, we can show that with high probability at most $O(\log k)$ s_i 's are saturated and all other s_i 's are under-saturated. In this case, optimum can route the demand from sources corresponding to a saturated s_i , $1 \leq i \leq k$, via edge (u, v) , and demands from sources corresponding to under-saturated and almost-saturated s_i 's via edge (s_i, t_i) . Thus with high probability the optimum can attend congestion in $O(\log k)$.

We now consider the congestion occurred by an oblivious routing algorithm. Suppose that for a set A of fk of s_i 's, $1 \leq i \leq k$, the oblivious routing algorithm routes at least half of the demands from sources corresponding to s_i via (u, v) . The rest of the proof is almost identical to the proof of Theorem 2. It means the oblivious routing has congestion $\Omega(\sqrt{k})$ with high probability and thus the competitive ratio is in $\Omega(\frac{\sqrt{k}}{\log k}) = \Omega(\frac{n^{1/3}}{\log n})$ with high probability. ■

3 Oblivious Algorithm

In this section, we present a demand-dependent oblivious routing algorithm for general directed graphs with polylogarithmic competitive ratio in our randomized demand model.

Theorem 4 *For every α there is a demand-dependent oblivious routing algorithm that obtains a competitive ratio of $\alpha \cdot O(\log^2 n)$ with probability at least $1 - \frac{1}{n^\alpha}$.*

A crucial step for designing a routing algorithm with low competitive ratio is to derive a lower bound on the optimum congestion for a given demand pattern. For this we partition the demands into classes in the following way. We say that the demand for a commodity j is in class C_k , $k \in \mathbb{Z}$ if

$$2^k \cdot \text{mincut}(j) \leq \text{demand}(j) < 2^{k+1} \cdot \text{mincut}(j) .$$

If the demand for commodity j is in C_k we call j *active* for class C_k . Further, we call a class C_k *active* if at least one commodity is active for this class. The following observation gives a first lower bound on the optimum congestion.

Observation 5 *For a demand-vector D let k_{max} denote the number of the highest active class. Then $C_{\text{opt}}(D) \geq 2^{k_{\text{max}}}$.*

However, this observation only gives a very crude way for lower bounding the optimum congestion. For our application we need an additional bound that is based on a more sophisticated classification scheme.

Let C_{opt}^k and C_{obl}^k denote the congestion of the optimum and the oblivious routing algorithm, respectively, for routing commodities in class \mathcal{C}_k . Similarly, we define $C_{\text{opt}}^{\leq k}$ and $C_{\text{obl}}^{\leq k}$ as the optimum and oblivious congestion, respectively, for routing commodities in classes with number at most k . Define ℓ as

$$\ell := \max \left\{ k \mid \mathbf{E}[C_{\text{opt}}^{\leq k} \mid \text{classes } \mathcal{C}_s, s > \ell \text{ are inactive}] > 8 \ln n \cdot 2^{k+1} \right\} \quad (1)$$

if the maximum exists and $\ell := -\infty$ otherwise. We merge all demands in classes \mathcal{C}_k , $k \leq \ell$ into one class \mathcal{B} (i.e., a commodity j is in \mathcal{B} if $\text{demand}(j) \leq 2^{\ell+1} \cdot \text{mincut}(j)$). If the maximum in Equation 1 does not exist, \mathcal{B} is the empty set. We call \mathcal{B} the *base class* and classes \mathcal{C}_k , $k > \ell$ are called *higher order classes*.

The following simple but crucial observation directly follows from this classification scheme.

Observation 6 *For every higher order class \mathcal{C}_k , $\mathbf{E}[C_{\text{opt}}^k \mid \mathcal{C}_k \text{ is highest active class}] \leq O(\log n \cdot 2^k)$.*

Based on the class-definitions the demand-dependent oblivious routing algorithm constructs a *demand-independent* routing scheme for each higher order class and a demand-dependent routing scheme for the base class \mathcal{B} . A commodity j is routed by first determining the class j belongs to, and then routing according to the $s_j \rightarrow t_j$ flow in the routing scheme of this class. In Section 3.1, we show that for each higher order class \mathcal{C}_k there is an oblivious routing scheme such that for any α

$$\Pr \left[C_{\text{obl}}^k \geq \alpha \cdot O(\log n \cdot \mathbf{E}[C_{\text{opt}}^k \mid \mathcal{C}_k \text{ is highest active class}]) \mid \mathcal{C}_k \text{ is highest active class} \right] \leq \frac{1}{n^\alpha} . \quad (2)$$

Furthermore, it can be shown that

$$C_{\text{opt}} \geq \frac{1}{2} \mathbf{E}[C_{\text{opt}}^{\mathcal{B}} \mid \mathcal{B} \text{ is highest active class}] \quad (3)$$

holds with probability at least $1 - \frac{1}{n^\delta}$ where δ is the constant used in the definition of the base class \mathcal{B} . Finally we present a demand-dependent routing scheme for the class \mathcal{B} for which

$$C_{\text{obl}}^{\mathcal{B}} \leq \alpha \cdot O(\log^2 n \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{B}} \mid \mathcal{B} \text{ is highest active class}]) \quad (4)$$

holds with probability at least $1 - \frac{1}{n^\alpha}$. The following proof combines these results to yield Theorem 4.

Proof of Theorem 4. We choose δ in the definition of \mathcal{B} and the parameter in Equation 4 such that equations 3 and 4 hold together with probability at least $1 - \frac{1}{2n^\alpha}$. In the following we assume that both equations hold.

Now, we distinguish two cases. First suppose that no higher order classes are active. Then equations 3 and 4 already guarantee a competitive ratio of at most $\alpha \cdot O(\log^2 n)$.

Now, suppose that there is an active higher order class. Let k_{max} denote the number of the highest active class (all probabilities are conditioned on the event that $\mathcal{C}_{k_{\text{max}}}$ is the highest active class). Observation 5 gives that the optimum congestion is at least $2^{k_{\text{max}}}$. It remains to bound the congestion of the oblivious routing algorithm. For classes $\mathcal{C}, \mathcal{C}'$ and any threshold t

$$\begin{aligned} & \Pr [C_{\text{opt}}^{\mathcal{C}} \leq t \mid \mathcal{C}' \text{ is highest active class}] \\ & \geq \Pr [C_{\text{opt}}^{\mathcal{C}} \leq t \mid \mathcal{C} \text{ is highest active class}] . \end{aligned}$$

Therefore we can set \mathcal{C}' to $\mathcal{C}_{k_{\text{max}}}$ and use Equation 2 (Lemma 7) to derive a high-probability upper bound on $C_{\text{obl}}^{\mathcal{C}}$ that also holds for our case where probabilities are conditioned on the event that $\mathcal{C}_{k_{\text{max}}}$ is the highest active class (instead on the event that \mathcal{C} is the highest class). Then we want to combine these bounds to get a bound for C_{obl} . However, since Equation 2 only holds with high probability we can apply this bound only to a polynomial number of classes in order to guarantee that with high probability the bound holds for each such class. In fact, it is sufficient if we apply Equation 2 only for the classes $\mathcal{C}_{k_{\text{max}}-2 \log n}, \dots, \mathcal{C}_{k_{\text{max}}}$. For all other classes \mathcal{C} we apply

```

COMPUTEROUTINGScheme ( $\mathcal{D}, \mathcal{C}$ )
  for  $i = 1$  to  $n^3$  do
    draw a demand-vector  $D_i$ 
    according to distribution  $\mathcal{D}_{\mathcal{C}}$ 
    compute  $D'_i$  from  $D_i$  by setting all
    demands not active for class  $\mathcal{C}$  to zero.
    compute an optimal
    multicommodity flow opt-flow( $D_i$ )
  end
  sum-flow :=  $\sum_i$  opt-flow( $D_i$ )
  normalize sum-flow to get a multicommodity flow in
  which the demand from each source to its sink is 1.

```

Figure 1: The algorithm for computing the oblivious routing scheme for a higher order class \mathcal{C} .

$C_{\text{obl}}^k \leq n^2 \cdot 2^{k+1}$. Let $k' := k_{\max} - 2 \log n$. We get with high probability

$$\begin{aligned}
C_{\text{obl}} &\leq C_{\text{obl}}^{\mathcal{B}} + \sum_{k=\ell+1}^{k_{\max}} C_{\text{obl}}^k \\
&\leq C_{\text{obl}}^{\mathcal{B}} + \sum_{k=\ell+1}^{k'-1} C_{\text{obl}}^k + \sum_{k=k'}^{k_{\max}} C_{\text{obl}}^k \\
&\leq O(\log^2 n \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{B}} \mid \mathcal{B} \text{ highest act. class}]) \\
&\quad + \sum_{k=k'}^{k_{\max}} O(\log n \cdot \mathbf{E}[C_{\text{opt}}^k \mid \mathcal{C}_k \text{ highest act. class}]) \\
&\quad + \sum_{k=\ell+1}^{k'-1} O(n^2 \cdot \mathbf{E}[C_{\text{opt}}^k \mid \mathcal{C}_k \text{ highest act. class}]) \\
&\leq O(\log^2 n \cdot C_{\text{opt}}) + \sum_{k=k'}^{k_{\max}} O(\log^2 n \cdot 2^k) \\
&\quad + \sum_{k=\ell+1}^{k'-1} O(n^2 \cdot 2^k) \\
&\leq O(\log^2 n \cdot C_{\text{opt}} + \log^2 n \cdot 2^{k_{\max}}) \\
&\leq O(\log^2 n \cdot C_{\text{opt}}) .
\end{aligned}$$

■

3.1 Routing algorithms for demand-classes

In this section, we present the following lemma that is crucial for the performance analysis of our demand-dependent oblivious routing algorithm as described in the previous section.

Lemma 7 *For a high order class \mathcal{C} there is an oblivious routing scheme such that*

$$\Pr[C_{\text{obl}}^{\mathcal{C}} \geq \alpha \cdot O(\log n \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{C}} \mid \mathcal{C} \text{ is highest active class}]) \mid \mathcal{C} \text{ is highest active class}] \leq \frac{1}{n^\alpha} .$$

Although the proof of the lemma is omitted, the algorithm in Figure 1 is used in such proof. Some important aspects of this algorithm are presented in the following lemma.

Lemma 8 *With high probability sum-flow fulfills the following properties.*

1. For any high-expectation commodity j , $\text{sumdem}(j) = \Omega(\mathbf{E}[\text{demand}(j) \mid \mathcal{C} \text{ is highest active class}] \cdot n^3)$.
2. The load on any edge is at most $O(n^3 \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{C}} \mid \mathcal{C} \text{ is highest active class}])$.
3. The load on any edge for routing a specific demand j is only $O(p_j \cdot n^3 \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{C}} \mid \mathcal{C} \text{ is highest active class}])$, where p_j denotes the probability that commodity j is active for class \mathcal{C} .

Claim 9 *If sum-flow fulfills the properties of Lemma 8 the oblivious routing scheme achieves congestion $O(\log n \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{C}} \mid \mathcal{C} \text{ is highest active class}])$ with high probability.*

The following lemma shows the existence of a good demand-dependent routing scheme for the base class \mathcal{B} .

Lemma 10 *There is a demand-dependent oblivious routing scheme for \mathcal{B} that obtains congestion $C_{\text{obl}}^{\mathcal{B}} = O(\log^2 n \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{B}} \mid \mathcal{B} \text{ highest act. class}])$ with high probability.*

Proof. The routing scheme works as follows. All demands for a commodity j that are smaller than $2^\ell/n^2$ are routed according to a maximum $s_j \rightarrow t_j$ flow. Since, there are at most n^2 commodities, these demands can cause a congestion of at most $2^\ell \leq O(\log^2 n \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{B}} \mid \mathcal{B} \text{ highest act. class}])$.

We partition the remaining demands into $2 \log n$ classes \mathcal{B}_i , $1 \leq i \leq 2 \log n$ such that commodity j is in class \mathcal{B}_i if $2^{\ell-i} \leq \text{demand}(j) \leq 2^{\ell-i+1}$. For each such class we can compute the oblivious routing scheme as we did for classes \mathcal{C}_k in the proof of Lemma 7 because demands for a commodity in a class only vary by a constant factor.

Then $C_{\text{opt}}^{\mathcal{B}_i}$ is less than $O(\log n \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{B}} \mid \mathcal{B} \text{ highest active class}])$ for each class with high probability. Therefore with high probability $C_{\text{opt}}^{\mathcal{B}} \leq \sum_i C_{\text{opt}}^{\mathcal{B}_i} \leq O(\log^2 n \cdot \mathbf{E}[C_{\text{opt}}^{\mathcal{B}} \mid \mathcal{B} \text{ highest active class}])$. ■

References

- [1] Yossi Azar, Edith Cohen, Amos Fiat, Haim Kaplan, and Harald Räcke. Optimal oblivious routing in polynomial time. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 383–388. ACM Press, 2003.
- [2] Mohammad Hajiaghayi, Jeong Han Kim, Tom Leighton, and Harald Räcke. Oblivious routing in directed graphs with random demands. In *STOC*, 2005.