Technical University of Munich
Department of Informatics

Master's Thesis in Informatics

Semi-oblivious Routing Strategies in Directed Graphs

Semioblivious Routing in gerichteten Graphen

Philipp Czerner

15.09.2020

Supervised by Prof. Dr. Harald Räcke

1. Introduction

Routing is the problem of sending packets through a network while minimising the load generated. It is gaining importance with the increasing scale of real-world networks. An area that has received considerable interest in recent years is the topic of *oblivious routing*, which tries to compute possible routes without knowing the actual amount of packets that will be sent. In other words, it identifies routes that perform well under *all* possible traffic conditions and are not subject to change after initial deployment.

This has considerable advantages when compared to *adaptive* routing, which measures current conditions in the network and adapts routes accordingly. In particular, any measurement necessarily involves error and adds complexity to the overall implementation. Additionally, temporary performance degradations can occur while the network updates to optimise for the current situation.

However, oblivious routing comes with drawbacks as well. While it guarantees that its routes will be close to optimal (i.e. to routes optimised for the specific traffic conditions), this is only up to a logarithmic factor, and only in undirected graphs. In directed graphs, not only does such a guarantee not exist, there are also strong (polynomial) lower bounds showing that oblivious routing is not practical in this case.

Semi-oblivious routing tries to present a practical compromise. Instead of fixing the routes completely independent of traffic conditions, it restricts itself to using only a small number of routes, but adapts the rate at which these are used to the current conditions. This is much easier to implement, as the usage rate of a route can be controlled by a simple forwarding decision at the source and no routing tables have to be updated dynamically.

There have been experimental investigations into semi-oblivious routing based on traffic simulations. These have been generally positive, showing that it performs better than oblivious routing. Despite these encouraging practical results, however, the theoretical research has been entirely negative, showing that the worst-case behaviour of semi-oblivious routing matches that of oblivious routing.

In this thesis we investigate this discrepancy, with a focus on the theoretical performance of semi-oblivious routing in directed graphs. As mentioned, it is known that oblivious routing has weak worst-case behaviour in that case, exemplified by simple counterexamples. However, to the best of our knowledge, it is as of yet unknown how semi-oblivious routing performs in directed graphs.[1]

Our main conclusion is negative, showing a lower bound for semi-oblivious routing schemes with a logarithmic number of routes. The main result reads as follows.

Theorem 2. *Let G denote a directed, unweighted, single-sink graph with n nodes. Then any semi-oblivious routing scheme on G using k flows per node has competitive ratio at least $n^{1/20}/23k^{4/5}$.*

In particular, it is not possible for a semi-oblivious routing scheme to be polylogarithmically competitive with only a polylogarithmic number of flows per node.

[1] Note the discussion in Section 3.2.

Existing lower bounds for oblivious routing in directed graphs are based on graphs with multiple different traffic conditions which require the routing scheme to avoid certain areas. As oblivious routing can use only a single route, having two such conditions is sufficient. Semi-oblivious routing, however, is able to choose between multiple routes and can deal with a constant number of situations.

To prove the above result, we therefore construct a graph with a high number of traffic conditions, which cannot be adequately routed with only a logarithmic number of flows. As it turns out, the existence of semi-oblivious routing schemes in this graphs is related to an elementary statement about random arithmetic progressions in coloured finite fields. In particular, we show the following:

Theorem 1. *Let $f_1, ..., f_q \subseteq \mathbb{F}_p$, p prime, s.t. $\sum_c f_c = p$ and $|f_c| \leq \lceil p/q \rceil$ for each c. An arithmetic progression $P \subseteq \mathbb{F}_p$ of length $16q^2$, chosen u.a.r., has $f_c \cap P \neq \emptyset$ for each c with probability at least $\frac{1}{2}$.*

The rest of this thesis is structured as follows: In Section 2 we give an overview of prior work on this subject. Then Section 3 introduces the relevant definitions. A weak lower bound is proved in Section 4, which suffices to rule out polylogarithmically competitive semi-oblivious routing schemes with a constant number of flows per node. Then, Section 5 changes topics to a show an elementary result about random arithmetic progressions in coloured finite fields, which is subsequently used in Section 6 to prove our main result, shown above. Finally, we close out by showing different avenues that were investigated but did not yield interesting results, in particular related to upper bounds, in Section 7, and summarise the overall findings and potential directions for future research in Section 8.

2. Related Work

Oblivious routing in general undirected graphs was first studied by Räcke [16], who showed that it is $\mathcal{O}(\log^3 n)$ competitive. This was later improved to $\mathcal{O}(\log n)$ [11, 17], matching a known lower bound [3, 15]. Additionally, the lower bound of $\Omega(\log n)$ still holds if all commodities share a common source or sink [10].

For directed graphs a lower bound of $\Omega(\sqrt{n})$ is well-known [1], which was improved to $\Omega(n)$ by Ene et al. [7].

Hajiaghayi, Kleinberg, and Leighton [9] show multiple lower bounds for semi-oblivious routing, including a bound of $\Omega(\log n / \log\log n)$ on undirected graphs, almost matching the known lower bound for oblivious routing schemes. They also provide a $\Omega(n^{1/10})$ lower bound in directed graphs, for semi-oblivious routing schemes using at most $\mathcal{O}(n^{1/5})$ paths per node. While this looks superficially similar to our main result, it is based on a different model of semi-oblivious routing schemes, where, as we argue in Section 3.2, semi-oblivious routing schemes are trivially bad when restricted to a sublinear number of paths.

While the theoretical results have been negative, on the practical side semi-oblivious routing seems promising. In particular, multiple empirical simulations comparing semi-oblivious routing to various other approaches show near-optimal performance [13, 14].

Regarding random arithmetic progressions in coloured finite fields we are not aware of any results investigating the exact problem we are interested in. Similar questions are raised in the study of *rainbow* arithmetic progressions, where the elements of the arithmetic progression must have pairwise distinct colours. While Butler et al. [4] show that a large number of colours is required if there are no restrictions on how often each colour appears, there are more favourable results for *equinumerous* colouring of sets of consecutive integers.

Let T_k denote the smallest number $t \in \mathbb{N}$ s.t. for any $n \in \mathbb{N}$ every equinumerous colouring of $\{1, ..., tn\}$ admits a rainbow arithmetic progression of length k. Jungić et al. [12] show that $T_k \in \mathcal{O}(k^3) \cap \Omega(k^2)$, with the upper bound subsequently being improved to $\mathcal{O}(k^2 \log k)$ [8, 6, 2].

This result is in some sense dual to our Lemma 4 (of which Theorem 1 is a generalisation): Instead of maximising the the length of an arithmetic progression s.t. it remains rainbow, we minimise the length of the arithmetic progression s.t. it contains each colour at least once. Interestingly, in both cases the gap between length and number of colours seems to be roughly quadratic (though we provide only an upper bound).

3. Preliminaries

Let $G = (V, E, w)$ denote a directed weighted graph, with V a set of nodes and $E \subseteq V^2$ a set of (directed) edges (also referred to as arcs). Generally speaking, $n := |V|$ is the number of nodes, but for some constructions it is more convenient to use n as some unspecified parameter and end up with $\Theta(n)$ nodes, which does not change the bounds asymptotically. Also, $w : E \to \mathbb{R}_{\geq 0}$ is a set of weights. Applying w to a set of edges refers to taking the sum, so $w(U) := \sum_{e \in U} w(e)$ for $U \subseteq E$.

A *flow* is a function $f : E \to \mathbb{R}_{\geq 0}$. Instead of writing $f((u, v))$ we may instead write $f(u, v)$. The *balance* of f is the difference between outgoing and incoming flow, i.e. $\mathrm{bal}_f(v) := \sum_{(v,u) \in E} f(v, u) - \sum_{(u,v) \in E} f(u, v)$ for each $v \in V$.

We say that a flow f is an *s-t-flow (with value α)* if f transports α units of flow (aka packets) from node s to node t, meaning $\mathrm{bal}_f(s) = \alpha$, $\mathrm{bal}_f(t) = -\alpha$ and $\mathrm{bal}_f(v) = 0$ for $v \in V \setminus \{u, v\}$. Additionally, f is a *unit* s-t-flow if it has value 1. A *path* is an integral unit s-t-flow for some $s, t \in V$, and a node t is *reachable* from s if a unit s-t-flow exists. Same as for weights, applying a flow to a set of edges refers to taking the sum, so $f(A) := \sum_{e \in A} f(e)$ for $A \subseteq E$. We extend addition and scalar multiplication to flows componentwise, so $(f + g)(e) := f(e) + g(e)$ and $(\lambda f)(e) := \lambda f(e)$ for flows f, g, $\lambda \in \mathbb{R}$ and edge $e \in E$.

We associate with G a set of *commodities* $\mathcal{C}$. Each commodity is a pair $(s, t) \in V^2$, with s denoting the *source* and v denoting the *sink* (aka target), s.t. t is reachable from s. Unless otherwise specified, we assume that all valid combinations are possible, i.e. the set of commodities is $\{(s, t) \in V^2 :\ t \text{ is reachable from } s\}$. We usually identify $\mathcal{C}$ with some other set more convenient to write, for example we might write i instead of (s_i, t_i). If all commodities have the same sink, i.e. $\mathcal{C} \subseteq V \times \{t\}$ for some $t \in V$, we say that G is *single-sink*.

A *multi-commodity flow* $f = (f_{(s,t)})_{(s,t)\in\mathcal{C}}$ is a family of flows, s.t. $f_{(s,t)}$ is an s-t-flow. For an edge $e \in E$ we write $f(e) := \sum_{i\in\mathcal{C}} f_i(e)$. If we specify a multi-commodity flow only on a subset of commodities, we assume flows with value zero for the others. We also define a componentwise addition and scalar multiplication operation as $f + g := (f_i + g_i)_{i\in\mathcal{C}}$ and $\lambda f := (\lambda f_i)_{i\in\mathcal{C}}$ for multi-commodity flows f, g and $\lambda \in \mathbb{R}$.

Let $d : \mathcal{C} \to \mathbb{R}_{\geq 0}$ denote *demands*, i.e. it assigns each commodity (s,t) the amount $d(s,t) \geq 0$ that must be sent from s to t. It is often convenient to specify demands only for a subset of commodities $I \subseteq \mathcal{C}$, in which case $d(s,t) := 0$ for $(s,t) \notin I$. A *routing (of d)* then is a multi-commodity flow $R = (R_i)_{i\in\mathcal{C}}$ s.t. R_i has value $d(i)$. The *congestion* of R is the maximum relative load over any edge:

$$\mathrm{cong}(R) := \max_{e\in E} \frac{R(e)}{w(e)}$$

The *optimal congestion (w.r.t. d)* $\mathrm{cong}(d)$ is the minimal congestion of any routing of d.

3.1. Semi-oblivious Routing Schemes

The idea of a semi-oblivious routing scheme is that we fix a small number of flows for each commodity obliviously, i.e. without knowing the demands. Afterwards, we are given the actual demands and distribute the packets for a commodity amongst the chosen flows. We assume that this is done optimally.

Definition 1. A *semi-oblivious routing scheme* S is a mapping of commodities (s,t) to sets of unit s-t-flows $S(s,t)$. We say that S uses k flows per node if $|S(s,t)| \leq k$ for each $(s,t) \in \mathcal{C}$.

An *S-routing (of d)* is a routing R of d with $R(s,t) = \sum_{f\in S(s,t)} \lambda_f f$ for each $(s,t) \in \mathcal{C}$ and some factors $\lambda_f \geq 0$. The *congestion of S (w.r.t. d)* $\mathrm{cong}(S,d)$ is the minimal congestion of any S-routing of d. The *competitive ratio* of S is the maximum ratio between the optimal congestion and the congestion of S for any demands d. Formally,

$$\mathrm{comp}(S) := \max_{\text{demands } d} \frac{\mathrm{cong}(S,d)}{\mathrm{cong}(d)}$$

An *oblivious routing scheme* S is a semi-oblivious routing scheme using one flow per node.

As for an oblivious routing scheme S the value $S(s,t)$ always is a one-element set containing a flow, we identify it with the contained flow when convenient (and thus S with a map from commodities to flows). Note that there is only a single S-routing R if S is oblivious, which has $R = \sum_{i\in\mathcal{C}} d(i)S(i)$.

3.2. Paths or Flows

Finally, a remark on the above definition of a semi-oblivious routing scheme S. It differs in one important detail from the one by Hajiaghayi, Kleinberg, and Leighton [9]. In particular, they define $S(s,t)$ as a set of *paths* instead of flows. This is equivalent if you

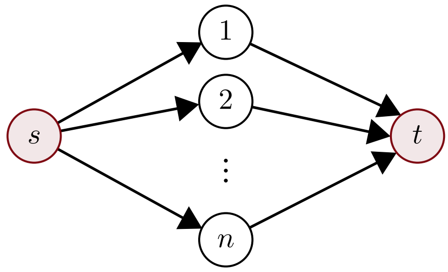

Figure 1: Restricting the number of paths used for routing to a sub-polynomial amount leads to a large congestion.

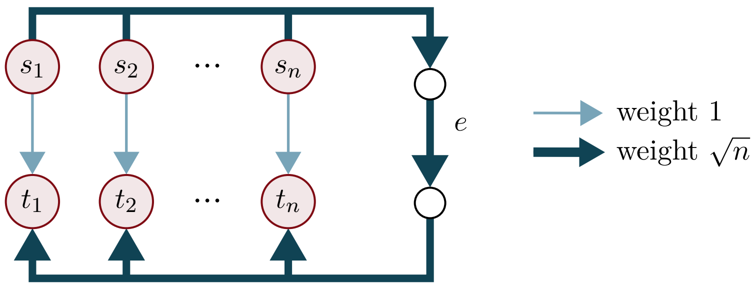

Figure 2: Oblivious routing is $\Omega(\sqrt{n})$-competitive.

allow for a polynomial number of paths or flows per node, but has important consequences for smaller schemes.

As illustrated in Figure 1, restricting the number of paths to, say, $\mathcal{O}(n^{1-\varepsilon})$ for some $\varepsilon > 0$ means that the routing scheme has competitive ratio $\Omega(n^{\varepsilon})$. Therefore the path-based model of semi-oblivious routing is (at least on some instances) *worse* than an ordinary oblivious routing scheme!

We conclude that it makes little sense to study the path-based model with a sublinear number of paths. Instead we will consider the flow-based model, as defined above, which exhibits interesting behaviour even for a small number of flows per node, where it is a natural extension of oblivious routing.

This also implies that the lower bound of Hajiaghayi, Kleinberg, and Leighton regarding directed graphs [9, Theorem 3.5] is of limited interest in this context, as it applies to the path-based model.

4. Simple Lower Bound

Before we begin with proving the main result, we will take a look at a strictly weaker bound, which is much easier to prove. It illustrates some of the problems when constructing lower bounds for semi-oblivious routing.

As mentioned earlier, it is well-known that oblivious routing is $\Omega(n)$-competitive in directed graphs. As our bound for semi-oblivious routing schemes can be viewed as a generalisation of it, we start by presenting a construction similar to the bound utilised by Ene at al. [7, Figure 4].

The graph is illustrated in Figure 2. The basic idea is that there are two scenarios. Either all commodities send one unit of flow, in which case exclusively using the (s_i, t_i) arcs would be (close to) optimal, or only a single pair communicates. In the latter case, we have a single demand to route $\sqrt{n}$ packets from some s_i to t_i, and it is (close to) optimal to use the large edge e.

Lemma 1 (see [1, 7])**.** *Oblivious routing in directed graphs is $\Omega(\sqrt{n})$-competitive.*

Proof. Fix some parameter n and consider a graph $G = (V, E, w)$ which consists of nodes $V := \bigcup_{i=1}^{n}\{s_i, t_i\} \cup \{u, v\}$ and arcs $E := \bigcup_{i=1}^{n}\{(s_i, t_i), (s_i, u), (v, t_i)\} \cup \{(u, v)\}$. All (s_i, t_i) arcs, for $i = 1, ..., n$, have weight 1, the other arcs have weight $\sqrt{n}$. For $i = 1, ..., n$ we have a commodity i with source s_i and sink t_i.

Let S denote any oblivious routing scheme. Additionally we use $E_1 := \{(u, v)\}$ and $E_2 := \{(s_i, t_i) : i = 1, ..., n\}$ to denote two subsets of edges. Removing $E_1 \cup E_2$ from G disconnects all source-target pairs, so each flow $f = S(i)$ for $i \in \mathcal{C}$ has $f(E_1) + f(E_2) = 1$. Hence the average (over f) of either $f(E_1)$ or $f(E_2)$ must be at least 1/2. We consider these cases separately.

If $f(E_1)$ is at least 1/2 on average, then $\sum_{f \in S(\mathcal{C})} \geq n/2$ and setting the demands d to 1 for each commodity would incur a congestion of $\sqrt{n}/2$ on edge e. Meanwhile the optimal routing of d has a congestion of at most 1 by using all (s_i, t_i) arcs, so S is $\Omega(\sqrt{n})$-competitive.

In the other case, we pick an f maximising $f(E_2)$, which therefore has $f(E_2) \geq 1/2$. Now we induce a single demand of $\sqrt{n}$ for commodity i, with i being the commodity corresponding to f, i.e. $f = S(i)$. When routing this demand, S causes a congestion of at least $\sqrt{n}/2$ on the edge (s_i, t_i) while the optimal routing can use arc e exclusively and has only a congestion of at most 1.

So in both cases S is $\Omega(\sqrt{n})$-competitive. □

However, it is obvious that semi-oblivious routing is 1-competitive (i.e. optimal) in this example, using two flows per node. In fact, other lower bounds for oblivious routing in directed graphs, such as the original one by Ene at el. [7, Figure 4] or the earlier bound by Azar et al. [1, Theorem 7.1] exhibit a similar pattern. They have two different kinds of demands, and oblivious routing cannot perform well on both. However, semi-oblivious routing simply uses two flows, one for each situation, and performs optimally.

Hence we generalise the previous example by encoding more than a constant amount of traffic situations in the graph. The graph is illustrated in Figure 3.

While, in contrast to Figure 2, there is only a single sink here, we could also adapt the previous example to the single-sink case. The important difference is that we now separate the edges separating sources and sink into classes. There are $k + 1$ classes, one more than the number of flows used by the semi-oblivious routing scheme. The first class has a single edge with capacity $\sqrt{n}$, for a total capacity of $\sqrt{n}$. For the second class we have α^2 edges with a capacity of $\sqrt{n}/\alpha$, so that the total capacity increases by a factor

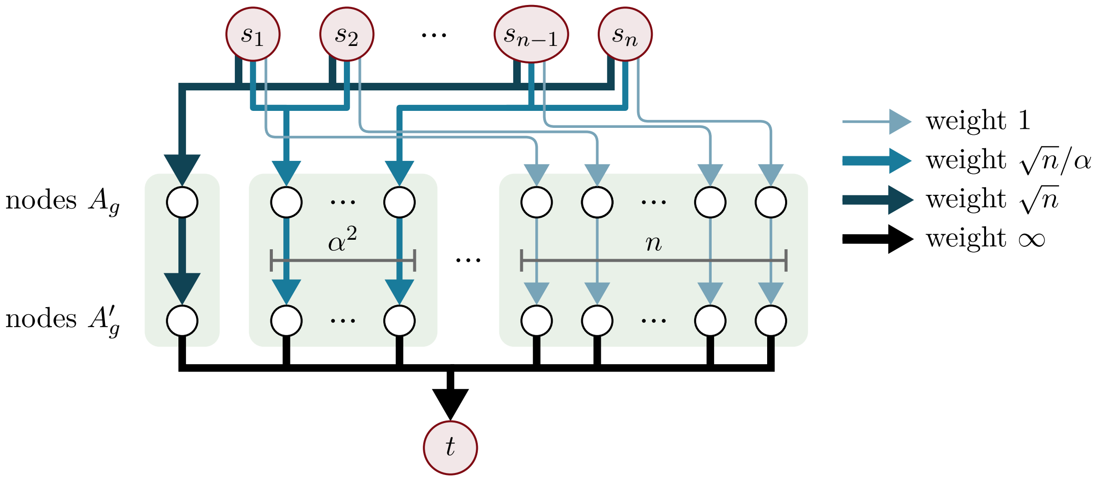

Figure 3: Simple lower bound on semi-oblivious routing. There are $k+1$ classes and $\alpha := \sqrt[2k]{n}$, with k being the number of flows per node used by the routing scheme.

of α. We choose $\alpha := \sqrt[2k]{n}$, meaning that the last class has n nodes of capacity 1, as we increase the number of edges by a factor of α^2 and decrease their capacity by α for each subsequent class.

We then have $k+1$ different traffic situations, one for each class. The corresponding demand enables exactly one commodity for each arc belonging to the class. This makes these arcs ideal for routing the demands: Using an edge belonging to another class either uses an edge with lower weight, or one shared by multiple sources. In either case there is a congestion of at least α.

Lemma 2. *Semi-oblivious routing schemes using at most k flows per commodity have competitive ratio at least $\sqrt[2k]{n}/2(k+1)$.*

Proof. The structure of our counterexample is illustrated in Figure 3. We start with a single sink t and nodes s_i for $i = 1, ..., n$, with n being some parameter (later we will change the bound so that n matches the number of nodes). To connect s_i with t_i, we will partition the set $\{1, ..., n\}$ into groups. For a group $g \subseteq \{1, ..., n\}$ we use a single arc to connect all s_i to t with $i \in g$. More precisely, we add nodes A_g, A'_g and arcs $(A_g, A'_g), (s_i, A_g), (A'_g, t)$ for $i \in g$. The (A'_g, t) arcs have weight ∞, the other weights will be specified in a moment. Commodity i is the source-sink pair (s_i, t).

To get our groups, we use $k+1$ partitions of $M := \{1, ..., n\}$. In the j-th partition, $j = 0, ..., k$, we have r_j groups of equal size (perhaps off by one), and the groups use arcs of weight c_j. So for a group g of the j-th partition the respective (A_g, A'_g) and (s_i, A_g) arcs have weight c_j (though the latter could have weight ∞ just as well). Setting $\alpha := \sqrt[2k]{n}$ we define $r_j := \alpha^{2j}$ and $c_j := \alpha^{k-j}$. We call these groups *class j* groups.

Therefore, to communicate between s_i and t, $i = 1, ..., n$, the scheme S can distribute the flow across $k+1$ paths, one for each class. Hence we can identify each flow with a vector $h \in \mathbb{R}^{k+1}$, where $h(j)$ is the amount of flow using the class j path, with $j = 0, ..., k$.

While S can pick k different flows $h_1, ..., h_k$, we are interested in the maximum flow which S can send over an arc when routing a unit demand. Let $f_i(j) := \max_l h_l(j)$ denote the maximum flow that S can route using the class j path for commodity $i = 1, ..., n$, with $j = 0, ..., k$.

Of course, $h_1, ..., h_k$ are unit flows, so $\sum_j h_l(j) = 1$ for $l = 1, ..., k$, which implies that $\sum_j f_i(j) \leq \sum_{j,l} h_l(j) = k$. In other words, the average of f_i is at most $k/(k+1)$.

Now we want to find a class j where S cannot route much flow using class j paths. Summing up all f_i leads to $\sum_{i,j} f_i(j) \leq nk$. There are $k+1$ groups, so the average value of $\sum_i f_i(j)$ for a class j is at most $nk/(k+1)$. Hence we can fix a j with $\sum_i f_i(j) \leq nk/(k+1)$, i.e. the average of $f_1(j), ..., f_n(j)$ is at most $k/(k+1)$.

Moving on, we need a commodity i for each group in class j s.t. the sum of $f_i(j)$ is low. Again, the average value of $f_i(j)$ in class j is at most $k/(k+1)$, so by picking the smallest value in each group we find a set of commodities $I \subset \{1, ..., n\}$ with $\sum_{i \in I} f_i(j) \leq |I|k/(k+1) = r_j k/(k+1)$. We now set the demand for each commodity in I to c_j and set all other demands to zero.

This means that the optimal routing has congestion 1, by using one class j path for each demand. However, S sends at most $c_j r_j k/(k+1)$ units of flow over class j paths. We will now show that the remaining $c_j r_j/(k+1) = \alpha^{k+j}/(k+1)$ units of flow cause high congestion.

Let $j' \neq j$ denote a class and x the amount of flow that class receives. First we consider the case $j' < j$. Here we have $r_{j'}$ arcs of weight $c_{j'}$, so we get a congestion of at least $x/r_{j'}c_{j'} = x/\alpha^{k+j'}$, which is at least x/α^{k+j-1}.

On the other hand, if $j' > j$ we have more total weight, but each commodity can only use a single arc. So the congestion is $x/r_j c_{j'} = x/\alpha^{k+2j-j'}$, and thus at least x/α^{k+j-1} as well.

To summarise, each unit of flow not send through class j creates a congestion of at least $1/\alpha^{k+j-1}$. There is at least $\alpha^{k+j}/(k+1)$ such flow, so we end up with congestion at least $\alpha/(k+1) = \sqrt[2k]{n}/(k+1)$.

Finally, we have to adjust our bound based on the actual number of nodes used, which is $2n + \sum_j r_j \leq 2n + 2\alpha^{2k} = 4n$. □

In particular, this rules out semi-oblivious routing schemes with a constant number of flows per node and polylogarithmic congestion, but it leaves open the possibility of achieving that congestion with a logarithmic number of flows.

The key issue seems to be that we encode the different traffic situations as separate regions in the graph, each with different weights and numbers of nodes. Those have to differ by (at least) some constant factor to matter, which then limits us to a logarithmic number of situations. For our strong lower bound we thus use another approach.

5. Arithmetic Progressions in Coloured Finite Fields

We will now turn our attention towards a different topic entirely. We consider the finite field $\mathbb{F}_p$ with p elements, p prime. These elements are coloured with q colours, such that each colour appears equally often. Additionally we have a random arithmetic progression

P with a fixed length L. (Later we will choose L as a quadratic function of q.) So $P = \{a, a+b, a+2b, ..., a+(L-1)b\}$, where $a \in \mathbb{F}_p$ and $b \in \mathbb{F}_p \setminus \{0\}$ are chosen uniformly at random (u.a.r.).

The question then is whether the arithmetic progression contains (at least) one element of each colour. In particular, we want to find a small L s.t. this happens with constant probability.

Before talking about colours, we begin by proving a useful property of the arithmetic progression, namely that pairs of elements are distributed u.a.r. in $\mathbb{F}_p$ apart from being distinct. The proof uses some of the same arguments as constructions of universal hash functions of the form $(ax+b \bmod p) \bmod n$, see e.g. Carter and Wegman [5, Proposition 7].

Lemma 3. *Fix some $i, j \in \{0, ..., L-1\}$ with $i \neq j$. Then*

$$\mathbb{P}(a + ib = x \wedge a + jb = y) = \frac{1}{p(p-1)}$$

for all $x, y \in \mathbb{F}_p$ with $x \neq y$. Additionally, $\mathbb{P}(a + ib = x) = 1/p$ for all $x \in \mathbb{F}_p$.

Proof. Recall that the random choice is over a and b, which are chosen u.a.r. with $a, b \in \mathbb{F}_p$ and $b \neq 0$.

To see that the statement holds, note that $a + ib = x \wedge a + jb = y$ is *equivalent* to $b = (x-y)/(i-j) \wedge a = y - bj$. So regardless of the values of x and y there is precisely one choice for a and b that puts the i-th and j-th element into those positions, meaning that the probability is indeed $1/p(p-1)$.

A fortiori, we also get $\mathbb{P}(a + ib = x) = 1/p$ for all i and x, by summing over y. In other words, each individual element of the arithmetic progression is also uniformly distributed. □

5.1. Non-overlapping Colours

To show the general technique, we first start with a simple lemma, which we will later generalise.

Lemma 4. *Let $f : \mathbb{F}_p \to \{1, ..., q\}$ denote a colouring, s.t. $|f^{-1}(c)| \leq \lceil p/q \rceil$ for each c. An arithmetic progression $P \subseteq \mathbb{F}_p$ of length $2q^2$, chosen u.a.r., has $\mathbb{P}(|f(P)| = q) \geq \frac{1}{2}$.*

Proof. Let $L := 2q^2$ denote the length of the arithmetic progression. For $L \geq p$ we have $P = \mathbb{F}_p$ and the statement holds trivially, so assume $L < p$. Recall that we use a and b for the parameters of P, i.e. $P = \{a, a+b, ..., a+(L-1)b\}$, with $a, b \in \mathbb{F}_p$, $b \neq 0$. These are chosen uniformly at random.

For each $0 \leq i < j < L$, let X_{ij} denote a binary random variable, indicating whether the i-th and j-th element of P have the same colour, i.e. $f(a+ib) = f(a+jb)$. We start by arguing that $\mathbb{E}(X_{ij})$ is not too large.

Fix some i, j defined as above and assume that the i-th element lands in position x, i.e. $x = a + ib$. A colour colours at most $\lceil p/q \rceil$ elements, so Lemma 3 implies that the j-th element will be chosen u.a.r. from the at most $\lceil p/q \rceil - 1$ remaining elements of colour $f(x)$. Formally, we get

$$\mathbb{E}(X_{ij}) = \mathbb{P}\big(f(a+ib) = f(a+jb)\big) \le \frac{\lceil p/q \rceil - 1}{p-1} \le \frac{p}{q(p-1)}$$

Now we look at $X := \sum_{ij} X_{ij}$, the total number of pairs which have the same colour. The above bound yields $\mathbb{E}(X) \le \binom{L}{2} p/q(p-1)$. Let $F(c) := |P \cap f^{-1}(c)|$ denote the number of times a colour c appears in P. Then we also get $X = \sum_c \binom{F(c)}{2} = (\sum_c F(c)^2 - L)/2$.

We can lower bound $\sum_c F(c)^2$ by $L^2/|f(P)|$ (see Lemma 16). Of course, $|f(P)| \le q$, so X is at least $L^2/2q - L/2$. On the other hand, if $|f(P)| < q$, meaning that not all colours appear in P, we get $X \ge L^2/2(q-1) - L/2$. All that remains is applying Markov's inequality to $X - (L^2/2q - L/2)$.

$$\mathbb{P}(|f(P)| < q) \le \mathbb{P}\Big(X \ge \frac{L^2}{2(q-1)} - \frac{L}{2}\Big) \le \frac{\mathbb{E}(X) - \big(\frac{L^2}{2q} - \frac{L}{2}\big)}{\frac{L^2}{2(q-1)} - \frac{L}{2} - \big(\frac{L^2}{2q} - \frac{L}{2}\big)}$$

We use the bound on $\mathbb{E}(X)$, then plug in $L = 2q^2$ and $L < P$

$$\le \frac{\frac{L(L-1)p}{2q(p-1)} - \big(\frac{L^2}{2q} - \frac{L}{2}\big)}{\frac{L^2}{2(q-1)} - \frac{L}{2} - \big(\frac{L^2}{2q} - \frac{L}{2}\big)} = \frac{(q-1)(q-1+\frac{L-1}{p-1})}{2q^2} \le \frac{q-1}{2q} \le \frac{1}{2}$$

$\square$

The key idea is that random arithmetic progressions distribute pairs uniformly, which means that we can analyse properties depending on two elements of the arithmetic progression quite well. We use this to reason about the number of colours, by looking at a property (number of pairs sharing a colour) that is minimised when the number of colours is large.

5.2. Overlapping Colours

While the previous proof is nice and elegant, we sadly need a more general version of that lemma. Before we assumed a colouring $f : \mathbb{F}_p \to \{1, ..., q\}$, so each element of $\mathbb{F}_p$ had exactly one colour. However, we are actually interested in allowing the colours to overlap, i.e. we have sets $f_1, ..., f_q \subseteq \mathbb{F}_p$ containing the elements of one colour. As before, each colour should appear with frequency roughly p/q. There is also the technical condition that their sizes still sum up to exactly p, which simplifies the following calculations significantly, but is not actually necessary for the statement to hold.

Theorem 1. *Let $f_1, ..., f_q \subseteq \mathbb{F}_p$, s.t. $\sum_c |f_c| = p$ and $|f_c| \le \lceil p/q \rceil$ for each c. An arithmetic progression $P \subseteq \mathbb{F}_p$ of length $16q^2$, chosen u.a.r., has $f_c \cap P \ne \emptyset$ for each c with probability at least $\frac{1}{2}$.*

The proof will take up the remainder of this section.

Now we have a length of $L := 16q^2$. Again, we use a and b for the parameters of P, i.e. $P = \{a, a+b, ..., a+(L-1)b\}$, with $a, b \in \mathbb{F}_p$, $b \neq 0$. These are chosen uniformly at random. Additionally, we can assume $L < p$ wlog, as else $P = \mathbb{F}_p$ and the statement trivially holds.

If an element of the arithmetic progression P is contained in f_i, with $i \in \{1, ..., q\}$, we say that it has colour i. (So an element can have multiple colours or none.) We use the random variable $l := \sum_i l_i$ to denote the total number of colours in P, where $l_i := |\{c : a+(i-1)b \in f_c\}|$ is the number of colours of the i-th element of the arithmetic progression P, for each $i = 1, ..., L$.

We will later show that l is large, in some sense. Of course, our final goal is proving that P contains q *distinct* colours. For this we define $X_c := |P \cap f_c|$ as the number of times colour c appears in p, for $c = 1, ..., q$; and $X := \sum_c X_c^2$ as the sum of their *squares*. (Note that $\sum_c X_c$ is simply l.)

This is analogous to the previous proof of Lemma 4, where we considered pairs of elements with the same number. Squaring the number of times a colour appears in P has the same effect, but it makes the calculations a bit easier.

Let $N := |\{c : P \cap f_c \neq \emptyset\}|$ denote the number of distinct colours in P. Again, our goal is showing that $N = q$ with probability at least $\frac{1}{2}$.

We start with the observation that X tends to be larger if P contains only few distinct colours.

Lemma 5. $X \geq l^2/N$

Proof. This is a straightforward application of Lemma 16.

$$X = \sum_{c=1}^{q} X_c^2 = \sum_{f_c \cap P \neq \emptyset} X_c^2 \geq \frac{1}{N}\Big(\sum_{f_c \cap P \neq \emptyset} X_c\Big)^2 = \frac{l^2}{N}$$

□

We decompose the statement $N = q$, that our arithmetic progression contains all colours, into two parts. The idea is that we first show that l is likely to be large, and then that X is likely to be small, given that l is large. However, for the second part there are some complications, so we prove a slightly unintuitive statement:

(a) l is large with constant probability, i.e. $\mathbb{P}(l < L/2) \leq 1/4$.

(b) $X - l^2/q \geq L^2/4q(q-1)$ with probability at most $1/4$.

The intuition behind (b) is that we actually want to prove $X \geq l^2/(q-1)$. We would like to use Markov's inequality, but l is not a constant, so we replace it by $L/2$ and show that bound instead. (Due to (a), we know that it is unlikely for l to be smaller than $L/2$.) However, applying Markov's to X itself yields only a weak bound, which is not sufficient for our purposes. Therefore we strengthen the statement by considering the random variable $X - l^2/q$ instead, as we know l^2/q to be a lower bound on X. This leads to the inequality in (b), which we can actually prove.

Now we argue formally why it suffices to prove (a) and (b).

Lemma 6. *If both (a) and (b) hold, then $N = q$.*

Proof.

$$\mathbb{P}(N < q) \overset{(\circ)}{\leq} \mathbb{P}\Big(X \geq \frac{l^2}{q-1}\Big)$$
$$= \mathbb{P}\Big(X \geq \frac{l^2}{q-1} \wedge l \geq \frac{L}{2}\Big) + \mathbb{P}\Big(X \geq \frac{l^2}{q-1} \wedge l < \frac{L}{2}\Big)$$
$$\overset{(*)}{\leq} \mathbb{P}\Big(X \geq \frac{l^2}{q} + \frac{L^2}{4q(q-1)}\Big) + \mathbb{P}\Big(l < \frac{L}{2}\Big) \leq \frac{1}{2}$$

Here, $(\circ)$ is simply Lemma 5. To see that $(*)$ holds, we assume $l \geq L/2$, which leads to the following:

$$X \geq \frac{l^2}{q-1} = \frac{l^2}{q} + \frac{l^2}{q(q-1)} \geq \frac{l^2}{q} + \frac{L^2}{4q(q-1)}$$

□

The general plan for proving (a) is that we bound the expected value and variance of l, and then apply Chebyshev's inequality. The bound for $\mathbb{E}(l)$ is straightforward, for $\mathrm{Var}(l)$ we proceed by carefully estimating how the number of colours behaves for pairs of elements of the arithmetic progression.

Lemma 7. *Statement (a) holds, i.e. $\mathbb{P}(l < L/2) \leq 1/4$.*

Proof. Recall that $l := \sum_i l_i$ with $l_i := |\{c : a + (i-1)b \in f_c\}|$ being the number of colours of the i-th element of P. Lemma 3 implies that each l_i has the same expected value (l_i is a function of the position of the i-th element of P, but the distribution of positions does not depend on i). We have

$$\mathbb{E}(l_1) = \sum_c \mathbb{P}(a \in f_c) = \sum_c \frac{|f_c|}{p} = 1$$

This yields $\mathbb{E}(l) = L$. Now we turn to the variance of l and consider

$$\mathrm{Var}(l) = \mathbb{E}(l^2) - \mathbb{E}(l)^2 = \sum_i \mathbb{E}(l_i^2) + \sum_{i \neq j} \mathbb{E}(l_i l_j) - L^2$$

As we have just argued, the l_i are identically distributed. Additionally, Lemma 3 guarantees that pairs of positions $(a + (i-1)b, a + (j-1)b)$, and therefore pairs (l_i, l_j), are identically distributed, for $i, j \in \{1, ..., L\}, i \neq j$. Hence both $\mathbb{E}(l_i^2)$ and $\mathbb{E}(l_i l_j)$ do not depend on i, j, for $i, j \in \{1, ..., L\}, i \neq j$. So we only need to consider $\mathbb{E}(l_1^2)$ and $\mathbb{E}(l_1 l_2)$.

Let $n_r := \mathbb{P}(l_1 = r) \cdot p$ denote the number of elements of $\mathbb{F}_p$ which have r colours, for $r = 0, ..., q$. Thus, we get $1 = \mathbb{E}(l_1) = \sum_r r n_r / p$ and $\mathbb{E}(l_1^2) = \sum_r r^2 n_r / p$, implying $1 \leq \mathbb{E}(l_1^2) \leq q$. To calculate $\mathbb{E}(l_1 l_2)$ we need to consider that the positions are distinct:

$$\mathbb{E}(l_1 l_2) = \sum_{r \neq s} \frac{rsn_r n_s}{p(p-1)} + \sum_r \frac{r^2 n_r (n_r - 1)}{p(p-1)} = \sum_{r,s} \frac{rsn_r n_s}{p(p-1)} - \sum_r \frac{r^2 n_r}{p(p-1)}$$

To ease notation, we set $\xi := \mathbb{E}(l_1^2)$.

$$= \sum_{r,s} \frac{rsn_r n_s}{p(p-1)} - \frac{\xi}{p-1} = \sum_s \frac{sn_s}{p-1} \sum_r \frac{rn_r}{p} - \frac{\xi}{p-1} = \frac{p-\xi}{p-1}$$

Plugging this into $\mathbb{E}(l^2)$, we get

$$\mathbb{E}(l^2) = L\xi + L(L-1)\frac{p-\xi}{p-1} = L\,\frac{(L-1)(p-\xi)+\xi(p-1)}{p-1}$$
$$= L\,\frac{(p-1)(L-1)+L-1+(p-L)\xi}{p-1}$$
$$= L(L-1) + L\,\frac{L-1+(p-L)\xi}{p-1}$$

Using $1 \leq \xi \leq q$ and $L < p$ we get both an upper and a lower bound for $\mathbb{E}(l^2)$. [2]

$$L^2 \leq \mathbb{E}(l^2) \leq L(L-1) + Lq \leq L(L+q) \tag{1}$$

This yields $\mathrm{Var}(l) \leq L(L+q) - L^2 = Lq$. All that remains for part (a) is applying Chebyshev's inequality. Here we use that $l < L/2$ is implied by $l \leq \mathbb{E}(l)/3$.

$$\mathbb{P}\Big(l < \frac{L}{2}\Big) \leq \mathbb{P}\Big(l \leq \frac{1}{3}\,\mathbb{E}(l)\Big) \leq \mathbb{P}\Big(|l - \mathbb{E}(l)| \geq \frac{2}{3}\,\mathbb{E}(l)\Big) \leq \frac{4Lq}{9\,\mathbb{E}(l)^2} = \frac{1}{36q} \leq \frac{1}{4}$$

□

Now we finish the proof by showing that (b) holds, which basically states that X is not too large. Here we show an upper bound on $\mathbb{E}(X)$, which again comes down to pairs of elements of P being uniformly distributed apart from being distinct. Hence one element of P having a colour c does not make it more likely that another element has the same colour, which in turn decreases X. Finally, we apply Markov's inequality, not to X itself, but to $X - l^2/q$.

Lemma 8. *Statement (b) holds, i.e.* $\mathbb{P}(X - l^2/q \geq L^2/4q(q-1)) \leq 1/4$.

Proof. Recall that we defined X as the sum of X_c^2, where X_c denotes how often colour c appears among elements of P. We start by calculating the expected value of X.

Fix some $c \in \{1, ..., q\}$ and let x_i denote the binary random variable indicating whether $a + (i-1)b \in f_c$ for $i = 1, ..., L$, i.e. the i-th element of the arithmetic progression has colour c. Of course, $X_c = \sum_i x_i$ and $X_c^2 = \sum_{i,j} x_i x_j$. Again, we use that pairs in the arithmetic progression are distributed u.a.r. apart from being distinct (see Lemma 3).

$$\mathbb{E}(X_c^2) = \sum_i \mathbb{E}(x_i^2) + \sum_{i \neq j} \mathbb{E}(x_i x_j) = \mathbb{E}(X_c) + L(L-1)\mathbb{P}(x_1 \wedge x_2)$$

[2] Of course, $\mathbb{E}(l^2) \geq L^2 = \mathbb{E}(l)^2$ is trivial.

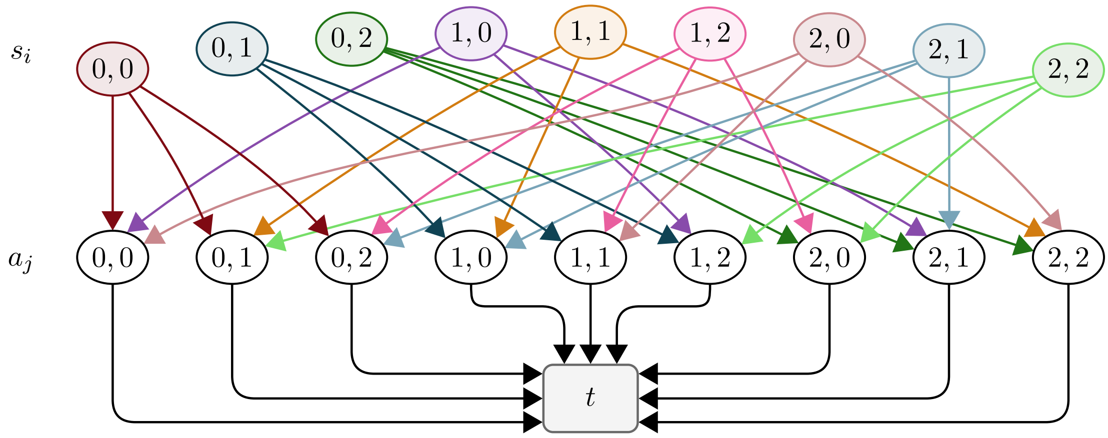

Figure 4: Construction used to prove the strong lower bound for semi-oblivious routing, with $p = 3$. All arcs have weight 1.

Using

$$\mathbb{P}(x_1 \wedge x_2) = \frac{|f_c|(|f_c| - 1)}{p(p-1)} \leq \frac{|f_c|}{q(p-1)}$$

we get

$$\mathbb{E}(X) \leq \sum_c \mathbb{E}(X_c) + L(L-1) \sum_c \frac{|f_c|}{q(p-1)} = L + \frac{Lp(L-1)}{q(p-1)} \tag{2}$$

Recall that $X \geq l^2/N$ and $N \leq q$, so we can apply Markov's inequality to $X - l^2/q$, and show that it is likely at most $L^2/4q(q-1)$.

$$\begin{aligned}
&\mathbb{P}\Big(X - \frac{l^2}{q} \geq \frac{L^2}{4q(q-1)}\Big) \leq \mathbb{E}\Big(X - \frac{l^2}{q}\Big) \cdot \frac{4q(q-1)}{L^2} \\
&\overset{(1,2)}{\leq} \Big(L + \frac{Lp(L-1)}{q(p-1)} - \frac{L^2}{q}\Big) \cdot \frac{4q(q-1)}{L^2} \\
&= \Big(q + \frac{p(L-1) - L(p-1)}{p-1}\Big) \cdot \frac{4(q-1)}{L} \\
&= \Big(q - \frac{p-L}{p-1}\Big) \cdot \frac{4(q-1)}{L} \leq \frac{4(q-1)^2}{L} \leq \frac{1}{4}
\end{aligned}$$

□

6. Strong Lower Bound

The problem with our previous bound for semi-oblivious routing was that we had a dedicated section of the graph for each type of traffic conditions, which limits the number of scenarios we can create. Instead, our graph in this section will be highly symmetric, but the choice of demands allows us to create many different bottlenecks, too many for a semi-oblivious routing scheme with a small number of flows per node to cover. An instantiation is shown in Figure 4, though the precise structure is perhaps a bit difficult to determine from just the drawing.

To construct the graph G, we fix a prime p and set $n := p^2$. We have source nodes s_i for $i = 1, ..., n$ and a single sink t. Commodity i is the pair (s_i, t). Between sources and sinks we add a layer of intermediate nodes a_j, $j = 1, ..., n$, with arcs (a_j, t). All arcs have weight 1. We will refer to these intermediate nodes as *links*. Finally, there are some arcs between the s_i and the a_j. Their precise structure is the intricate part.

Let $\mathbb{F}_p$ denote the finite field with p elements. For both a commodity $i =: (i_0, i_1)$ and a link $a_j =: (j_0, j_1)$ we identify $i, j \in \{1, ..., n\}$ with elements in $\mathbb{F}_p^2$, using some fixed bijection. We will refer to a_j as *type* j_1 link and use $\text{Type}(j_1) := \{(j_0, j_1) : j_0 \in \mathbb{F}_p\}$ to refer to the set of all type j_1 links.

Each node a_j gives rise to a linear function $f : \mathbb{F}_p^2 \to \mathbb{F}_p^2$ as $f(x) = j_0 + j_1 x$. We connect s_i to a_j if i lies on the graph of f, i.e. $f(i_0) = j_0 + j_1 i_0 = i_1$.

We start by mentioning a few properties of G. There are $2n + 1$ nodes, $n + n^{1.5}$ edges, and each commodity i is connected to $\sqrt{n}$ links and vice versa. Different commodities are "spread out", in that they use mostly disjoint sets of links:

Lemma 9. *Two distinct commodities i, i' share at most one link.*

Proof. Let $f : \mathbb{F}_p^2 \to \mathbb{F}_p^2$ with $f(x) = a + bx$ denote any linear polynomial in $\mathbb{F}_p$. For the link corresponding to f to be connected to both i and i', it must fulfil $f(i_0) = i_1 \wedge f(i'_0) = i'_1$. This implies $a = i_1 - bi_0 = i'_1 - bi'_0$ and thus $b(i_0 - i'_0) = i'_1 - i_1$. If $i_0 = i'_0$ then $i_1 \neq i'_1$ (as i, i' are distinct) and the equation has no solution. Else, we get a unique value for b and thus for a, meaning that there is exactly one solution. □

Earlier on, we gave each link a type. Later it will become convenient to group links, so that we can pick a subset of groups to put load on. The following lemma makes the type a convenient choice for grouping.

Lemma 10. *Fix some $x \in \mathbb{F}_p^2$. Each commodity i is connected to exactly one type x link.*

Proof. The equation $i_1 = j_0 + j_1 i_0 = j_0 + a i_0$ has a unique solution for j_0. □

We will now consider the maximum amount of flow that is possible to send within G, while keeping the congestion constant. Each node s_i has degree $\sqrt{n}$, so clearly a single commodity can route at most $\mathcal{O}(\sqrt{n})$ flow. Additionally, there are n links a_j for a total capacity of n, which means that routing significantly more than $\sqrt{n}$ commodities with a demand of $\sqrt{n}$ is not possible within constant congestion. Interestingly, we can match this bound, i.e. it is possible to route $\sqrt{n}$ flow for any "small" number of commodities with constant congestion.

Lemma 11. *Let $I \subset \{1, ..., n\}$ denote a subset of at most $\sqrt{n}$ commodities. Then demands d with $d(i) := \sqrt{n}$ for $i \in I$ have an optimal routing with congestion at most 2.*

Proof. Assume that we have some optimal routing f which minimises the maximum congestion. Additionally, amongst those routings we pick one with the smallest number of arcs (a_j, t) which have maximum congestion. Let $J \subseteq \{1, ..., n\}$ denote the set of links a_j of those arcs, i.e. $J := \{a_j : f(a_j, t) = \max_{j'} f(a_{j'}, t)\}$. Additionally, let $I' \subseteq I$ denote the commodities which are connected to a link $a_j \in J$, meaning that $I' := \{i \in I : (s_i, a_j) \in E, j \in J\}$.

For an $i \in I'$ we know that *all* connected links must be in J. If that were not the case, then we would have a commodity with both an adjacent maximally congested arc (a_j, t) and one with a strictly smaller congestion, say $(a_{j'}, t)$. By shifting a small amount of flow from the former to the latter, we reduce the number of arcs with maximum congestion (or, if a_j was the only one, we reduce the maximum congestion instead). But this contradicts our choice of f.

Therefore the arcs of links in J form a cut of I'. In particular, we have that each unit of flow sent by commodities in I' passes over such an arc exactly once, and that no other commodity uses those arcs. So the congestion of f is simply $|I'|\sqrt{n}/|J|$.

Now we conclude by arguing that a cut of I' has to be large, as I' is connected to many links. Recall that each commodity is connected to $\sqrt{n}$ links. If we count each connected link for each commodity, we may, of course, count some links multiple times. In particular, a link a_j being counted $r > 1$ times means that there are $r(r-1)/2 \geq r-1$ pairs of commodities in I' which share a_j.

Due to Lemma 9 we know that two commodities share at most one links, so by subtracting the total number of pairs in I' we ensure that we underestimate the number of connected links. Therefore $|J| \geq |I'|\sqrt{n} - |I'|(|I'|-1)/2 \geq |I'|\sqrt{n}/2$ (recall that $|I'| \leq |I| \leq \sqrt{n}$), and the congestion of f is at most 2. $\square$

Now consider some semi-oblivious routing scheme S using k flows per commodity $i \in \{1, ..., n\}$ and with competitive ratio at most C. Our goal will be to construct a small subset of commodities $I \subset \{1, ..., n\}$ s.t. S cannot route a demand $\sqrt{n}$ for each commodity in I with low congestion. As Lemma 11 states, the optimal routing of any such subset has constant congestion, leading to a bound on C.

For a node s_i we have the k unit s_i-t-flows $S(i)$, which we denote by $f_1, ..., f_k$. To simplify our analysis, we want to round these flows. Given a flow f_c, for $c = 1, ..., k$, we say that a link a_j has colour c if $f_c(a_j, t) \geq 1/2\sqrt{n}$. So some links may not have any colours, while some may have multiple.

If there are at most $\sqrt{n}/8Ck$ links of colour c, we call the flow f_c *sparse*, else we call it *dense*. Later we want to choose a subset of links so that one is used by each flow $f_c \in S(i)$, i.e. that we pick one link of each colour. However, sparse flows use only a few links and are difficult to cover. Instead, we will now argue that S can transport only a small amount using sparse flows (while remaining within a good congestion), so that our subsequent analysis only needs to concern itself with dense flows.

Lemma 12. *Using only sparse flows, S can route at most $\sqrt{n}/2$ flow from s_i to t_i with congestion $2C$.*

Proof. Pick some arbitrary sparse flow $f_c \in S(i)$. Node s_i is connected to $\sqrt{n}$ links a_j, so there are at most $\sqrt{n}$ links a_j without colour c, each of which has $f(a_j, t) < 1/2\sqrt{n}$. Hence they carry at most $1/2$ units of flow from f_c, and conversely the links with colour c carry at least $1/2$ units of flow. (Recall that f_c is a unit flow.)

As f_c is sparse, there are only $r := \sqrt{n}/8Ck$ links with colour c, so there is some link a_j with $f_c(a_j) \geq 1/2r$. Then, however, f_c can transport at most $2C/f_c(a_j) \leq 4Cr = \sqrt{n}/2k$ flow within a congestion of $2C$. There are at most k such flows, yielding the desired bound. $\square$

Let $J \subset \mathbb{F}_p^2$ denote some subset of links. If, for any dense flow f_c, there is $j \in J$ with link a_j having colour c, we say that J *covers* i. We use $\text{cover}(J) := \{i \in \mathbb{F}_p^2 : J \text{ covers } i\}$ to refer to the set of links covered by J.

We are interested in finding a set J which fulfils the following properties:

(a) There is some subset of types $T \subseteq \mathbb{F}_p$ s.t. J contains precisely the links with type in J, i.e. $J = \bigcup_{j_1 \in T} \text{Type}(j_1) = \{(j_0, j_1) \in \mathbb{F}_p^2 : j_1 \in T\}$.

(b) The set T is an arithmetic progression of length $L := 16(8Ck)^2$, i.e. there exist $a, b \in \mathbb{F}_p, b \neq 0$ with $T = \{a, a+b, ..., a+(L-1)b\}$.

(c) At least a $\frac{1}{2}$ fraction of commodities are covered by J, i.e. $|\text{cover}(J)| \geq n/2$.

First, note that J uses $L\sqrt{n}$ links, which is much less than the $n/2$ commodities it covers. We want to "scale this down", i.e. use roughly L links to cover $\sqrt{n}/2$ commodities. From this we can deduce the demands for our counterexample. Loosely speaking, if we consider sending $\sqrt{n}$ flow for each of the covered commodities, S has to put one unit of flow onto the (modified) J for each demand it routes, while the optimal routing can distribute them with congestion 2 and avoid that bottleneck (see Lemma 11).

The specific structure of J (all links of certain types, which form an arithmetic progression) enables us to do the "scaling down". We will now show the existence of J, before moving on to the proof of our lower bound.

Lemma 13. *A set J fulfilling conditions (a)-(c) exists.*

Proof. We modify our colouring based on dense flows s.t. it meets the prerequisites of Theorem 1. Fix some commodity i. Recall that for each dense flow f_c of i at least a $1/8Ck$ fraction of links have colour c. We can remove colours from links so that each f_c covers *exactly* a $1/8Ck$ fraction, and add otherwise meaningless dense flows, with the purpose to have $8Ck$ dense flows in total. These then yield a colouring of links connected to commodity i, which we map to a colouring of $\mathbb{F}_p$ by mapping each link to its type.

Now we can apply Theorem 1, so choosing an arithmetic progression of types $T \subseteq \mathbb{F}_p$ with length $L = 16(8Ck)^2$ u.a.r. leads to the links of those types covering all dense flows f_c with probability at least $1/2$. Therefore we can fix a T s.t. this happens for a $1/2$ fraction of commodities, which then yields the set J fulfilling (a)-(c). $\square$

Before we start with the proof of Theorem 2, we want explain the intuition behind it, in particular the reasoning for choosing an arithmetic progression.

Starting with the types T we have a set of links J with size $|T|\sqrt{n} = L\sqrt{n}$ covering $n/2$ commodities. As mentioned, we want to scale this down, so that we have roughly L

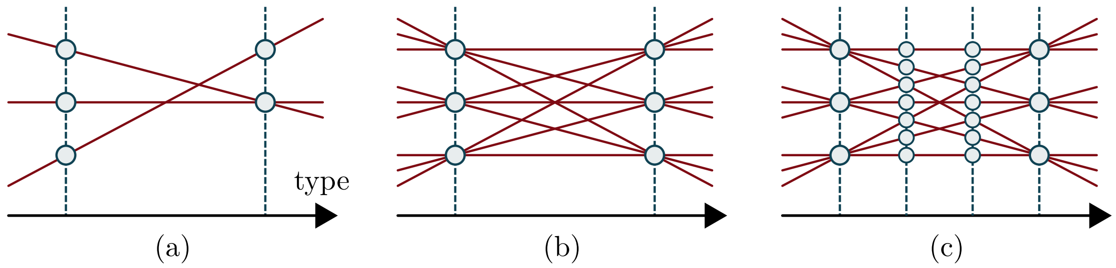

Figure 5: Covering nodes using links of some fixed types T. Points correspond to links, with type as x-coordinate. Dashed lines are the links with types T. (a) To cover a set of commodities (solid lines) we need the links corresponding to the intersections. (b) For $|T| = 2$ we can cover m^2 commodities using m links of each type. (c) As long as the types T are equidistant, we need only $|T|m$ links of each type to cover m^2 commodities.

links covering roughly $\sqrt{n}$ commodities. Those commodities can be routed concurrently with small congestion (Lemma 11), but the semi-oblivious routing scheme puts too much flow into the small set of links, creating a large congestion there.

The question then is how many commodities we can cover using links of types T. The way we connected commodities and links was based on linear polynomials in $\mathbb{F}_p$, where each commodity corresponds to a point in $\mathbb{F}_p^2$, each link to a line, and a commodity is connected to all links with lines crossing its point. So for a link a_j the set of commodities connected to it forms a line in $\mathbb{F}_p^2$, it is precisely the set $\{(x, j_0 + j_1 x) : x \in \mathbb{F}_p\}$. (The type of the link is the slope of the line.)

From the condition for a commodity i and a link a_j being connected, namely $j_0 + j_1 i_0 = i_1$, it is obvious that you can turn this statement around: the set of links connected to a commodity i also forms a line, which is $\{(i_1 - i_0 x, x) : x \in \mathbb{F}_p\}$.

Viewing points in the two-dimensional plane $\mathbb{F}_p^2$ as links, we group them by their types. Conveniently, the links of one type then also form a line. We already know that using all links with types in T we cover (half of) all commodities. So, if for any one commodity i we consider its line in $\mathbb{F}_p^2$ and take the links corresponding to the intersections of the line of i and the type-lines of types in T, then these links will cover i.

This, as well as the following arguments, are illustrated in Figure 5. Of course, this means covering a single commodity with $|T| = L$ links, which is not a good exchange ratio. But if $T = \{j_1, j_1'\}$, i.e. we only had two types, we could take some m consecutive links of each of the two types, and then each pair of links covers one commodity, for m^2 commodities in total. This is a rate of 1 link for $m/2$ commodities, which is much better.

Sadly, there are more than two types. However, the types are an arithmetic progression, which turns out to be enough. Let us now consider the setting in $\mathbb{R}^2$ instead of $\mathbb{F}_p^2$. Now we have equidistant types T, of which we pick the smallest and largest, calling them j_1 and j_1', respectively. Again, we take m consecutive links of j_1 and j_1' each, and all of the m^2 commodities connected to those links. Additionally, we take all links of the other types connected to those commodities. This already ensures that the commodities are covered, and we only need to upper bound the number of links we have used.

Here, the situation is as in Figure 5c (note that the x-coordinate corresponds to the type, which is the second element of the pair representing the link in $\mathbb{F}_p^2$). The coordinates of the additional links we use are given by convex combinations of the type j_1 and type j_1' links. So if these links have y-coordinates, say, y and y', then the y-coordinate of an additional link is given by $y^* := \lambda y + (1-\lambda)y'$. Crucially, $\lambda = q/(|T|-1)$ with $q \in \{0, ..., |T-1|\}$, so λ is an integer fraction with small numerator and denominator. As both y and y' are from a range of m consecutive integers, y^* can only take one of $|T|m$ values.

Theorem 2. *$C \geq n^{1/20}/23k^{4/5}$.*

Proof. We start by finding a small subset of commodities $I \subset \mathbb{F}_p$ with at least a $\frac{1}{2}$ fraction covered by J. (Additionally, we want I to have a specific structure.) For $x \in \mathbb{F}_p, a \in T$ let $J_a^x := \{(x,a), (x+1,a), ..., (x+m-1,a)\}$ denote a set of $m := \sqrt[4]{n}$ consecutive type a links starting at x. By choosing an appropriate offset x we ensure that a modified J, where the type a links are restricted to the ones in J_a^x, still covers $n/2\sqrt[4]{n} = n^{3/4}/2$ links. (Note that J_a^x contains a $1/\sqrt[4]{n}$ fraction of type a links.)

More precisely, if we choose the offset x u.a.r. and let $X := \sum_{i \in \mathrm{cover}(J)} X_i$ denote the number of covered commodities, with X_i indicating whether commodity $i \in \mathrm{cover}(J)$ is covered by $J' := J_a^x \cup \bigcup_{j_1 \in T \setminus \{a\}} \mathrm{Type}(j_1)$.

A commodity $i \in \mathbb{F}_p$ is connected to exactly one link of type a, therefore an i covered by J remains covered if its connected type a link is in J_a^x. This happens with probability $|J_a^x|/|\mathrm{Type}(a)| = 1/\sqrt[4]{n}$, meaning that $\mathbb{E}(X_i) \geq 1/\sqrt[4]{n}$. We then have $\mathbb{E}(X) \geq |\mathrm{cover}(J)|/\sqrt[4]{n}$ which, using property (c), is at least $n^{3/4}/2$.

Hence we can fix some a s.t. at least $n^{3/4}/2$ commodities are covered. Applying the same argument again for type $a' := a + (L-1)b$, we get an offset x' and a set $J'' := J_a^x \cup J_{a'}^{x'} \cup \bigcup_{j_1 \in T \setminus \{a,a'\}} \mathrm{Type}(j_1)$ covering at least $\sqrt{n}/2$ commodities.

While we have reduced the number of links we need for types a and a', we need to do the same to the other types. For these, however, we will preserve the commodities $I := \mathrm{cover}(J'')$ which are covered, since the set I has about the right size (between $\sqrt{n}/2$ and $\sqrt{n}$). Instead we simply choose all links connected to I of that type. Formally, for a type $j_1 \in T \setminus \{a, a'\}$ we set

$$J_{j_1}^* := \{(j_0, j_1) : i \in I, (s_i, a_j) \in E\}$$

and

$$J^* := J_a^x \cup J_{a'}^{x'} \cup \bigcup_{j_1 \in T \setminus \{a,a'\}} J_{j_1}^*$$

Now we need to show that the $J_{j_1}^*$ are “small”, which is where we will use property (b), that T is an arithmetic progression.

Fix some type $j_1 \in T \setminus \{a, a'\}$. For a link (j_0, j_1) to be in $J_{j_1}^*$, it has to be connected to some commodity $i \in I$, which itself is connected to a link $(y, a) \in J_a^x$ and a link

$(y', a') \in J^{x'}_{a'}$. This yields a system of equations

$$\begin{aligned}
& j_0 + j_1 i_0 = i_1 \wedge y + a i_0 = i_1 \wedge y' + a' i_0 = i_1 \\
\Rightarrow\ & j_0 + j_1 i_0 = y + a i_0 \wedge (a - a') i_0 = y' - y \\
\Rightarrow\ & j_0 = y + (a - j_1)\frac{y' - y}{a - a'} = y + (y' - y)\frac{j_1 - a}{a' - a}
\end{aligned}$$

We are trying to determine how many values j_0 could take. The calculations above happen in $\mathbb{F}_p$; but it will be convenient to consider sets of values in $\mathbb{N}$, as those are easier to count. Let $\pi : \mathbb{F}_p \to \{0, ..., p-1\}$ denote the mapping from an element of $\mathbb{F}_p$ to their smallest representative in $\mathbb{N}$. We can count values of j_0 in the following manner:

$$\Big|\Big\{ y + (y' - y)\frac{j_1 - a}{a' - a} : (y, a) \in J^x_a, (y', a') \in J^{x'}_{a'} \Big\}\Big|$$

Scaling by $(a' - a)/b$ (recall that b is the distance between subsequent elements in the arithmetic progression T) and adding the constant $(m + x - x')(j_1 - a)/b - x(a' - a)/b$ does not change the cardinality, as we are still in $\mathbb{F}_p$. (Also recall that $m = |J^x_a| = \sqrt[4]{n}$).

$$\begin{aligned}
&= \Big|\Big\{ (y - x)\frac{a' - a}{b} + (y' - x' - (y - x) + m)\frac{j_1 - a}{b} : y, y' \Big\}\Big| \\
&\le \Big|\Big\{ \pi(y - x)\,\pi\Big(\frac{a' - a}{b}\Big) + \pi(y' - x' - (y - x) + m)\,\pi\Big(\frac{j_1 - a}{b}\Big) : y, y' \Big\}\Big| =: |M|
\end{aligned}$$

We have $(a' - a)/b = L - 1$, with L the length of the arithmetic progression. Similarly, $(j_1 - a)/b \in \{1, ..., L-2\}$ holds and we have $y' - x' - (y - x) + m \in \{1, ..., 2m - 1\}$ due to the definition of J^x_a implying $y - x, y' - x' \in \{0, ..., m - 1\}$. So the largest number in M is $(m-1)(L-1) + (2m-1)(L-2) < 3mL$, which is an upper bound on the size of $J^*_{j_1}$.

To summarize, $|J^*| \le |T| 3mL = 3\sqrt[4]{n}L^2$.

Finally, we can set up demands for our lower bound. For each commodity in I we set the demand to $\sqrt{n}$, so $d(i) := \sqrt{n}$ for $i \in I$. Due to Lemma 11, the optimal routing of d has congestion 2. As S has competitive ratio C, we therefore know that it has congestion at most $2C$. Lemma 12 then shows that sparse flows carry at most $\sqrt{n}/2$ flow for each commodity $i \in I$, so the remaining $\sqrt{n}/2$ units of flow must use dense flows.

However, i is covered by J^*, therefore for each each dense flow f_c of i we have a link with color c, i.e. one with at least $1/2\sqrt{n}$ flow, in J^*. Thus the $\sqrt{n}/2$ units of flow routed using dense flows at i induce a load of at least $1/4$ in J^*. We have $|I| \ge \sqrt{n}/2$, so the total load on J^* is at least $\sqrt{n}/2$. This load then uses the arcs (a_j, t) for $j \in J^*$.

Again, the congestion is at most $2C$. Together with the load on J^* this yields

$$2C \ge \frac{\sqrt{n}}{2|J^*|} \ge \frac{\sqrt[4]{n}}{6L^2} = \frac{\sqrt[4]{n}}{3 \cdot 2^{21} C^4 k^4} \;\Rightarrow\; C \ge \frac{n^{1/20}}{23\, k^{4/5}}$$

which is the bound we wanted to show. □

7. Additional Work

In this section we discuss various approaches to proving upper and lower bounds for semi-oblivious routing in directed graphs. Our lower bound from Theorem 2 already settles the case of $\mathcal{O}(n^{1/16-\varepsilon})$ flows per node for any $\varepsilon > 0$, by showing that then a polylogarithmic congestion cannot be achieved. From the other direction, it is clear that a semi-oblivious routing scheme containing all paths is optimal, and there are at most $n!$ paths in a directed graph, 2^n if it is acyclic. The remaining case are semi-oblivious routing schemes with a polynomial, or at least subexponential, number of flows per node.

7.1. Counting Arguments

For any specific demand d it is easy to add the flows from its optimal routing to our semi-oblivious routing scheme S, to ensure that S performs optimally on that demand. At first glance it might seem futile to add flows for each demand to S, as there are uncountably many of them, but as we now show it suffices to perform well on binary demands, of which there are at most an exponential number.

Lemma 14. *Let S be an semi-oblivious routing scheme which is C-competitive on binary demands. Then a semi-oblivious routing scheme with competitive ratio $\mathcal{O}(C \log n)$ exists.*

Proof. Let m denote the number of edges in G. We first change S by rounding its flows f to put at least $1/2n^2$ units of flow onto any arc e with $f(e) > 0$. This increases congestion by a factor of at most 2, as f can transport at most $m/2n^2 \leq 2$ packets using these edges. So our modified S is still $2C$-competitive.

Now we fix some arbitrary demands d. We can scale d s.t. its minimum is 1, then round each demand to be a power of two, which changes congestion by at most a factor of 2. Now we decompose d into classes, i.e. we construct demands d_j with $d_j(i) = 2^j$ for $i \in \{i : d(i) = 2^j\}$. For demands d_j let R_j denote an S-routing of d_j with minimal congestion which uses at most $2n^2$ flows (see Lemma 17). (So R has congestion at most $2C \operatorname{cong}(d_j) \leq 2C \operatorname{cong}(d)$.)

For each j we now round R_j s.t. each commodity i puts at least $d_j(i)/4n^4$ packets on each flow it uses. Again, as there are at most $2n^2$ flows in total, R_j can transport at most $d_j(i)/2$ using flows with fewer than $d_j(i)/4n^4$ packets, so this increases congestion by at most a factor of 2. Note that $d_j(i) = 2^j$ for each i, so R_j routes at least $2^j/4n^4$ packets with each flow it uses.

We will now fix some edge $e \in E$ and analyse the congestion on e. Let j denote the highest class j s.t. S sends flow over e when routing d_j, i.e. $j = \max\{j' : R_{j'}(e) > 0\}$. As R_j puts at least $2^j/4n^4$ packets into each flow f it uses, and any flow f of S with $f(e) > 0$ has $f(e) \geq 1/2n^2$, we have that $R_j(e) \geq 2^j/4n^6$. Conversely, let $j^* := j - 8\log_2 n - 3$ denote a class. Then any class $j' = 1, ..., j^*$ can put at most $2^{j'}n^2$ flow on e, and $2^{j'}n^2 = 2^{j'-j^*} \cdot 2^j/8n^6$. Hence classes below j' increase the load on e by at most a factor of two.

For classes $j' = j^* + 1, ..., j$ we use that $R_{j'}$ has congestion at most $4C \operatorname{cong}(d)$. So if we simply add all routings to get $R := \sum_j R_j$ we conclude that R has congestion at most $\mathcal{O}(C \operatorname{cong}(d) \log n)$, and S therefore is $\mathcal{O}(C \log n)$ competitive. ☐

For now we consider graphs where each pair of nodes is a commodity. We can try looking at the different binary demands as a black box, so we construct a semi-oblivious routing scheme by adding flows for specific demands, and then try to approximate the demands we did not add. If we group the (binary) demands by cardinality, we note that there is only one demand of with n^2 commodities (the all-ones demand) and only n^2 demands with one commodity (the unit demands).

Using $n^2 + 1$ flows per node we could cover those extremes. The largest number of demands are the ones with cardinality $n^2/2$, of which there are more than $(n^2/2)^{n^2/2}$. For these it seems possible to approximate them by using the flow for the all-ones demand. Simply reusing any such flow unchanged does not work (it may have a large congestion for some commodities, causing it to route the other commodities sub-optimally), so it is unclear how to do so.

Even more problematic are the demands of size n. There are still exponentially many of them, and they seem difficult to approximate by both the all-ones demand and the unit demands, as the number of commodities differs by a factor of n from each of them.

Incidentally, similar arguments lead to the construction of our strong lower bound of Section 6. For the single-sink case there are only n commodities, so the construction revolves around demands of cardinality $\sqrt{n}$. That construction, however, is inherently polynomial: we choose an arithmetic progression of types, of which there are only n in total, and we choose offsets for the two types, again with n possibilities in total. So we are only able to construct up to n^2 different traffic conditions.

7.2. Cuts

In this section we will assume that the semi-oblivious routing scheme uses paths instead of flows, as this simplifies the presentation.

Characterising routing schemes with cuts seems difficult for semi-oblivious routing. However, there is a similar notion: the set of maximally congested edges.

Lemma 15. *Let S denote a semi-oblivious routing scheme using paths with competitive ratio C. Then there are demands d and a subset of edges $U \subseteq E$ s.t. $\operatorname{cong}(S,d)/\operatorname{cong}(d) = C$ and for each commodity i with $d(i) > 0$ all paths in $S(i)$ have an edge in U.*

Proof. Let d denote demands s.t. $\operatorname{cong}(S,d)/\operatorname{cong}(d) = C$ and R an S-routing of d with minimal congestion and, amongst those S-routings, one where the set U of edges with maximal congestion, i.e. $U := \{e \in E : R(e)/w(e) = \operatorname{cong}(S,d)\}$, is smallest.

Consider some commodity i. Assume for contradiction that i has two paths $p, p' \in S(i)$ s.t. p has an edge in U and p' does not. Thus each edge of p' would have a congestion strictly smaller than $\operatorname{cong}(S,d)$ and moving a tiny amount of flow from p to p' would decrease the number of maximally congested edges, which contradicts our choice of R.

Therefore we know that either all paths of i have an edge in U, or none do. Let $i \in I$ iff the former holds, i.e. $I := \{i \mid d(i) > 0 \land \forall p \in S(i) : p \cup U \neq \emptyset\}$.

Let $d' : I \to \mathbb{R}_{\geq 0}$ denote modified demands, with $d'(i) := d(i)$ for $i \in I$. We now claim that the congestion on d remains unchanged, i.e. $\operatorname{cong}(S,d')/\operatorname{cong}(d') = C$. First we remark that d was chosen s.t. $\operatorname{cong}(S,d)/\operatorname{cong}(d)$ is the competitive ratio of S and

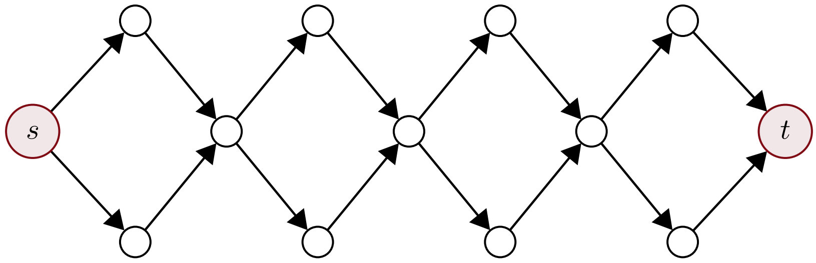

Figure 6: The graph G_0, which has exponentially many paths.

thus maximal. As $\operatorname{cong}(d') \leq \operatorname{cong}(d)$ we already know $\operatorname{cong}(S, d') \leq \operatorname{cong}(S, d)$. If we show $\operatorname{cong}(S, d') \geq \operatorname{cong}(S, d)$ as well, then $\operatorname{cong}(d') \geq \operatorname{cong}(d)$ and thus our claim would follow.

We will show this by contradiction: assume $\operatorname{cong}(S, d') < \operatorname{cong}(S, d)$. Let R' denote an S-routing of d' with congestion $\operatorname{cong}(S, d')$. We decompose R into R_0 and R_1, where $R_0 := (R_i)_{i \in I}$ contains only the flows for commodities in I and $R_1 := (R_i)_{i \notin I}$ contains the others. Now we can set $R^* := R_1 + (1 - \lambda)R_0 + \lambda R'$ for some $0 \leq \lambda \leq 1$. For $\lambda = 0$ we simply have $R^* = R$, for $\lambda = 1$ we have replaced the routing for commodities in I with R'.

For an edge $e \in U$ we know that $R_1(e) = 0$, as commodities in I' do not use edges in U, and $R'(e) < R_0(e)$, as e is maximally congested in R and R' has strictly lower congestion than R. So we get $R^*(e)/w(e) < R(e)/w(e) = C$ for $\lambda > 0$. Additionally, for an edge $e \notin U$ we have $R^*(e)/w(e) = R(e)/w(e) < C$ for $\lambda = 0$, as e is not one of the maximally congested edges. As R^* is continuous in λ, we can fix a $\lambda > 0$ s.t. we still have $R^*(e)/w(e) < C$ for all $e \notin U$. For the other edges we had this already, hence $\operatorname{cong}(R^*) < \operatorname{cong}(R)$, which contradicts our choice of R.

This proves the statement, with d' as demands and U as subset of edges. $\square$

It is tempting to use the above lemma to bound the competitive ratio C of a semi-oblivious routing scheme S. The lower bound is clear, given such a set U and demands d we know that $C \geq \sum_i d(i)/w(U) \operatorname{cong}(d)$. (We know that the optimal routing of demands $d/\operatorname{cong}(d)$ has congestion 1, and S routes each of these packets over U.)

For the upper bound one could imagine using Lemma 15 to show the other direction. However, while it yields demands d and a set U with $\operatorname{cong}(S, d) = C$, this congestion might be caused by paths using edges in U multiple times. Hence the actual competitive ratio may be higher than $\sum_i d(i)/w(U) \operatorname{cong}(d)$.

7.3. Graphs with Exponentially Many Paths

In a graph with polynomially many paths, a semi-oblivious routing scheme can incorporate all possible paths and route optimally. Hence we need a graph with an exponential number of paths, the prototypical example of which is shown in Figure 6, which we will denote G_0.

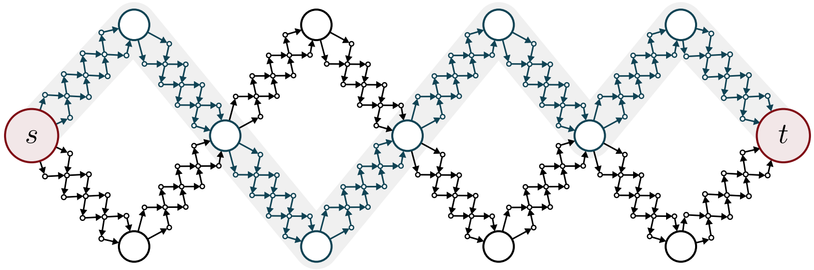

Figure 7: Recursive construction for an $\Omega(\log n/\log\log n)$ lower bound. Each edge is replaced by a copy of G_0. A random s-t-path P is highlighted, which the optimal routing will avoid, but the semi-oblivious routing scheme must use at least partially.

Let S denote some polynomial set of s-t-paths in G_0. Then a random s-t-path shares roughly half of its edges with all paths in S, with high probability. For this it suffices if G_0 has length $\Omega(\log n)$, as we choose the constant of G after knowing the size of S. So we can force a semi-oblivious routing scheme to put half of its flow into the "wrong" spot.

Hajiaghayi, Kleinberg, and Leighton construct a lower bound using this kind of approach [9, Theorem 3.3]. After picking a random s-t-path P in G_0, they know that the a semi-oblivious routing scheme S puts roughly half of its flow onto the path while the optimal routing can avoid it entirely. So they induce a demand from s to t of 1. Then we will later "punish" the semi-oblivious routing scheme by adding a demand (u, v) for each edge $(u, v) \in P$. The optimal routing is not affected, as it does not use any edges from P and can route these demands without additional congestion.

The key is that we now proceed recursively, i.e. we replace each edge in G with a copy of G_0 (see Figure 7). As this replaces an edge of weight 1 by a structure with capacity 2 we also double all existing demands. Now we add the next set of demands, setting $d(u, v) := 1$ for each $(u, v) \in P$. (For this, P is not a path but a set of pairs of nodes.) After this, we refine P s.t. it is a valid path for our new graph; here we simple replace each pair $(u, v) \in P$ by a random path from u to v. This (roughly) halves the flow that the semi-oblivious routing scheme sends over P, but we doubled the demand earlier, so in total the flow does not change. Again, the optimal routing can send the the new demands without using edges on (the refined) P, but S routes roughly half of the flow of the new demands over P.

This repeats for $\Omega(\log n/\log\log n)$ iterations. The graph increases in size in each iteration; after this number of iterations we end up with n nodes. In the end we get a competitive ratio of $\Omega(\log n/\log\log n)$.

However, it appears difficult to apply a similar procedure and get a bound that is stronger than logarithmic, since the recursion naturally increases the size of the graph by a constant factor.

7.4. Upper Bounds

In this section we will present a construction for semi-oblivious routing schemes that *does not work*. However, it highlights some interesting problems.

Definition 2. A directed graph $G = (V, E, w)$ is a *level-graph* if there is a partition $V_1 \dot\cup V_2 \dot\cup ... \dot\cup V_l = V$ of nodes into *levels*, arcs only go from a level to the subsequent level, i.e. $E \subseteq (V_1 \times V_2) \cup (V_2 \times V_3) \cup ... \cup (V_{l-1} \times V_l)$, and the set of commodities $\mathcal{C}$ has $\mathcal{C} \subseteq V_1 \times V_l$, i.e. flow is only sent from the first to the last level.

A *graph isomorphism* $\pi : V \to V$ is a bijection s.t. $(u, v) \in E$ iff $(\pi(u), \pi(v)) \in E$ for all $u, v \in V$. We call a level graph G *node-symmetric*, if for all nodes $u, v \in V_i$ there is a graph isomorphism π with $\pi(u) = v$.

When constructing lower bounds, there is a tradeoff based on the number of commodities using a single edge. If there are only few of them, then it is not possible for the arc to have a high congestion. On the other hand, having lots of commodities traversing an edge means that they “mix”, i.e. the packet going over that edge can use the same set of paths, regardless of which commodity it originally belonged to. If two commodities have similar sets of paths, then there are fewer different demands the routing scheme has to handle, as it does not matter which of the two commodities is active.

To investigate this, we first apply a few restrictions. First, we require that all commodities have a common sink. This allows flow of different sources to “mix” freely; during routing we only need to consider a single-commodity flow.

As it appears difficult to deal with “traps”, i.e. edges that can be overloaded when certain commodities are enabled, s.t. these they should be avoided when routing. Therefore we also require G to be a node-symmetric level-graph, so that it does not matter which node you route to (in some sense).

(As it turns out, even a node-symmetric graph can have very asymmetric behaviour when talking about groups of nodes, so this restriction does not seem very fruitful.)

We will now talk about the construction of a potential semi-oblivious routing scheme S in a node-symmetric, single-sink level-graph G which, as mentioned above, does not actually work.

Let t denote the unique sink and v an arbitrary source node (i.e. a level 1 node). To construct the semi-oblivious routing scheme, we first do a random walk on G starting at v, yielding a sequence of distributions $\mu_1, ..., \mu_k$, with $\mu_1(v) = 1$ and $\mu_1(u) = 0$ for $u \neq v$.

Note that the support of μ_i are precisely the nodes reachable after i steps. We then choose a subsequence of levels $r_0, ..., r_l$ with length $l \in \mathcal{O}(\log n)$, s.t. the size of the support at most doubles between each r_j, relative to the size of the level. Formally, we have $r_0 = 0$ and $|\mu_i|/|V_i| \leq 2|\mu_{r_j}|/|V_{r_j}|$ for $r_j < i < r_{j+1}$. We use $\text{trace}(p) := (p \cap V_{r_1}, ..., p \cap V_{r_l})$ to denote the nodes visited by an v-t-path p on the levels given by r_i.

Let X denote the random variable corresponding to the path taken by the random walk starting at v. From X we can construct a flow in G, referred to as $\text{flow}(X)$, by setting $\text{flow}(X)(e) := \mathbb{P}(e \in X)$.

Finally, our semi-oblivious routing scheme is given by

$$S(v, t) := \{\text{flow}(X \mid \text{trace}(X) = T) : T \in V_{r_1} \times ... \times V_{r_l}\}$$

The idea is that S allows us to adjust the distribution of packets at each level r_i. We could use this to ensure that these distributions remain close to uniform.

Proof idea. Fix some demands $d : V_0 \times \{t\} \to \mathbb{R}_{\geq 0}$. Due to Lemma 14 it suffices to consider only binary demands when trying to achieve a polylogarithmic congestions.

In the following, R denotes a routing of the (binary) demands d and $\mu_i : V_i \to \mathbb{R}_{\geq 0}$ the distribution of packets at level i w.r.t. R.

At each level, we want to choose a routing that ensures two things, both up to a constant factor:

(a) The packet distribution is uniform amongst reachable nodes.

(b) Packets follow the probabilities of the random walk.

To be precise, (a) refers to the distribution μ_i on level i fulfilling $\mu_i(v) \leq \alpha/|M_i|$ for each $v \in M_i$, with M_i denoting the set of reachable nodes on level i and α some constant factor. (Of course, μ_i is zero for non-reachable nodes.)

Assume both (a) and (b) hold. Then, we claim, S is $\mathcal{O}(1)$-competitive: Any routing of d has to distribute its flow at each level. Distributing it uniformly on as many nodes as possible and then executing a random walk minimises the congestion at that level, as G is node-symmetric. Conditions (a) and (b) ensure that, apart from a constant factor, we do just that for all levels.

We will now try to construct a routing R fulfilling both conditions. Initially, R is chosen s.t. the packets follow a random walk. Then we iterate over the levels $r_1, ..., r_l$, to progressively update R s.t. conditions (a) and (b) hold.

Assuming that at level r_j packets are distributed almost uniformly, meaning $\mu_{r_j}(v) \leq 2/|M_{r_j}|$ for $v \in M_{r_j}$, we will now show that on levels $r_j, ..., r_{j+1}-1$ we have $\mu_i(v) \leq 4/|M_i|$. (Note that $\mu_i(V_i) = 1$ due to our normalisation of d.)

Let $x_i := \max_v \mu_i(v)$ denote the maximum number of packets in any level i node. If we do a random walk starting with x_i packets in *every* level i node, then at level $i' > i$ each node will have exactly $x_i \cdot |V_i|/|V_{i'}|$ packets, due to G being node-symmetric. Consequently, $x_{i'} \leq x_i \cdot |V_i|/|V_{i'}|$ is an upper bound on the distribution $\mu_{i'}$. Setting $i := r_j$ and $r_j < i' < r_{j+1}$ we get $x_{r_j} \leq 2/|M_{r_j}|$ from our assumption and thus

$$x_{i'} \leq \frac{2|V_{r_j}|}{|M_{r_j}||V_{i'}|} \leq \frac{4}{|M_{i'}|}$$

The latter part follows from our choice of the r_j: the amount of reachable nodes (relative to level size) by a *single* random walk can at most double between levels r_j and i'. Doing multiple random walks simply means that we take the union of the reachability sets, which then fulfils $|M_{i'}|/|V_{i'}| \leq 2 \cdot |M_{r_j}|/|V_{r_j}|$.

Our assumption holds for level $r_0 = 0$, as our demands d are binary and normalised. However, and this is where the proof fails, it is not clear how to adjust the distribution for subsequent r_j. So, now we are going to talk about the problems arising in this step.

Problems. Firstly, $\mu_{i'+1}$ is not necessarily close to uniform for any constant. A simple example would be a step of the random walk starting at a single node, which has one outgoing arc with large weight and many with small weight. This can occur even in unweighted graphs, when a group of q nodes $v_1, ..., v_q$ is fully connected with q nodes on the next level (so they have q^2 edges between them) and each of the v_i has an additional $\sqrt{q}$ outgoing edges to other nodes.

One can require that G has bounded-degree, but, while this would ensure that $\mu_{i'+1}$ is close to uniform, that is not actually sufficient.

The problem arises where we have almost all packets concentrated in a few nodes, while a large number of nodes are almost empty. To make the distribution uniform again, we would have to increase the number of packets arriving at the latter by more than a constant factor, and it is not clear that this is possible with good congestion.

Worse, even if we adjust the probabilities of the random walk by only a constant factor, it is not clear that these factors do not compound.

On a higher level, it seems that requiring node-symmetry is not very meaningful, as groups can exhibit the same kinds of problematic behaviour. For the uniform distributions we relied on the notion of reachability, which does not hold up well in weighted graphs. (A node can be barely reachable or well connected, but we cannot differentiate.) Even in unweighted graphs reachability appears to fall apart in this context, as connections between groups of nodes are still "weighted", due to having more or fewer edges between them.

Additionally, random walks are strictly local. If a node has two outgoing edges, one with a large weight and one with an extremely large weight, the former will be used less, even if the relevant bottlenecks occur later. While the idea of this whole construction was to use the local optimisation provided by random walks and then fix the global behaviour by adjusting the probabilities at the levels r_j, it does not work out that well.

8. Conclusion

Semi-oblivious routing is a promising extension of oblivious routing, performing well in empirical evaluations. However, as of yet there are no clear theoretical results which indicate an improvement over oblivious routing. We investigate semi-oblivious routing in the context of directed graphs, using a flow-based model, and rule out polylogarithmically competitive semi-oblivious routing schemes with a polylogarithmic number of flows.

The existence of this bound is related to an elementary question about random arithmetic progressions in coloured finite fields. In particular, we show that an equinumerous colouring admits a short arithmetic progression (length quadratic in the number of colours) covering all colours.

There are a number of interesting questions left open for further research. Most importantly, it is unclear whether a polynomial number of flows per node are sufficient to achieve a polylogarithmic competitivity. We believe that this is the case, in particular for single-sink graphs, as only edges used by many commodities are problematic, but it seems that having many of these edges makes the graph well-connected in some sense.

More generally, it is unclear whether semi-oblivious routing provides a clear benefit in *any* situation. Regarding congestion, it is known that this is not the case in undirected graphs and grids. As we have shown, in directed graphs roughly $\Omega(n^{1/16})$ flows per node are needed at the very least, which already limits its feasibility there as well. So advantages in this regard would have to be in other classes of graphs, e.g. balanced graphs[3].

One could also investigate whether semi-oblivious routing improves different kinds of cost measures, such as fault tolerance or compactness, i.e. the space needed to store routing information in.

Finally, regarding our result on arithmetic progressions, we only proved a quadratic upper bound between the number of colours and the length of the sequence, and leave open whether this is sharp. We believe so, due to a symmetry between the gap in this problem and the maximum length of a rainbow arithmetic progression.

References

[1] Yossi Azar et al. "Optimal oblivious routing in polynomial time". In: *Journal of Computer and System Sciences* 69.3 (2004), pp. 383–394. DOI: 10.1145/780542.780599. URL: https://ttic.uchicago.edu/~harry/pdf/optimal_oblivious_journal.pdf.

[2] József Balogh, William Linz, and Letícia Mattos. *Long rainbow arithmetic progressions.* 2019. arXiv: 1905.03811 [math.CO].

[3] Yair Bartal and Stefano Leonardi. "On-line routing in all-optical networks". In: *Proc. International Colloquium on Automata, Languages, and Programming (ICALP).* Springer. 1997, pp. 516–526.

[4] Steve Butler et al. "Rainbow arithmetic progressions". In: *Journal of Combinatorics* 7.4 (2016), pp. 595–626. ISSN: 2150-959X. DOI: 10.4310/joc.2016.v7.n4.a3. URL: https://arxiv.org/pdf/1404.7232.

[5] J.Lawrence Carter and Mark N. Wegman. "Universal classes of hash functions". In: *Journal of Computer and System Sciences* 18.2 (1979), pp. 143–154. ISSN: 0022-0000. DOI: 10.1016/0022-0000(79)90044-8. URL: http://www.sciencedirect.com/science/article/pii/0022000079900448.

[6] David Conlon, Jacob Fox, and Benny Sudakov. *Independent arithmetic progressions.* 2019. arXiv: 1901.05084 [math.CO].

[7] Alina Ene et al. "Routing under Balance". In: New York, NY, USA: Association for Computing Machinery, 2016, pp. 598–611. ISBN: 9781450341325. DOI: 10.1145/2897518.2897654. URL: https://arxiv.org/pdf/1603.09009.

[8] Jesse Geneson. *A note on long rainbow arithmetic progressions.* 2018. arXiv: 1811.07989 [math.CO].

[3] A directed graph G is α-balanced, if for every cut C the weight of edges entering C differs by at most a factor of α from the weight of edges leaving C.

[9] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tom Leighton. “Semi-Oblivious Routing: Lower Bounds”. In: SODA ’07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 929–938. ISBN: 9780898716245. URL: https://www.cs.cornell.edu/~rdk/papers/semiobliv.pdf.

[10] Mohammad Taghi Hajiaghayi et al. “Oblivious routing on node-capacitated and directed graphs”. In: *ACM Transactions on Algorithms* 3.4 (2007), p. 51. DOI: 10.1145/1290672.1290688. URL: https://www.cs.cornell.edu/~rdk/papers/OblivSODA05.pdf.

[11] Chris Harrelson, Kirsten Hildrum, and Satish Rao. “A polynomial-time tree decomposition to minimize congestion”. In: *Proc. 15th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*. 2003, pp. 34–43. DOI: 10.1145/777412.777419. URL: https://dl.acm.org/doi/pdf/10.1145/777412.777419.

[12] Veselin Jungić et al. “Rainbow Arithmetic Progressions and Anti-Ramsey Results”. In: *Combinatorics, Probability and Computing* 12.6 (Nov. 2003), pp. 599–620. ISSN: 0963-5483. DOI: 10.1017/S096354830300587X. URL: http://math.mit.edu/~fox/paper-antiRamsey.pdf.

[13] Praveen Kumar et al. *Kulfi: Robust Traffic Engineering Using Semi-Oblivious Routing*. 2016. arXiv: 1603.01203 [cs.NI].

[14] Praveen Kumar et al. “Semi-Oblivious Traffic Engineering: The Road Not Taken”. In: *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. Renton, WA: USENIX Association, Apr. 2018, pp. 157–170. ISBN: 978-1-939133-01-4. URL: https://www.usenix.org/conference/nsdi18/presentation/kumar.

[15] Bruce M Maggs et al. “Exploiting locality for data management in systems of limited bandwidth”. In: *Proc. 38th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 1997, pp. 284–293.

[16] Harald Räcke. “Minimizing congestion in general networks”. In: *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2002, pp. 43–52. DOI: 10.1109/SFCS.2002.1181881. URL: https://ttic.uchicago.edu/~harry/pdf/min_congestion.pdf.

[17] Harald Räcke. “Optimal hierarchical decompositions for congestion minimization in networks”. In: *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*. ACM. 2008, pp. 255–264. DOI: 10.1145/1374376.1374415. URL: http://www.cs.cornell.edu/~abrahao/tdg/papers/p255.pdf.

A. Sum of Squares Inequality

The following inequality is well-known. Intuitively, it states that a sum of squares is minimised when all summands are equal (assuming that their sum is held constant).

Lemma 16. *Let* $x_1, ..., x_n \in \mathbb{R}$. *Then*

$$\sum_i x_i^2 \geq \frac{1}{n}\Big(\sum_i x_i\Big)^2$$

Proof. We apply the Cauchy-Bunyakovsky-Schwarz inequality, yielding

$$\frac{1}{n}\Big(\sum_{i=1}^n 1 \cdot x_i\Big)^2 \leq \frac{1}{n}\Big(\sum_{i=1}^n 1^2\Big) \cdot \Big(\sum_{i=1}^n x_i^2\Big) = \sum_{i=1}^n x_i^2$$

□

B. Sparse Routings

While a semi-oblivious routing scheme S may consist of many flows per node, for any specific demand it only needs to use a small amount of them. Recall that w denotes the weights in G.

Lemma 17. *Let d denote demands. Then there exists an S-routing R with congestion* $\mathrm{cong}(S, d)$ *s.t.* $R(i) = \sum_{f \in S(i)} \lambda_{i,f} f$ *for each commodity i, and there are at most* $2n^2$ *strictly positive* $\lambda_{f,i}$ *for commodity i and* $f \in S(i)$.

Proof. Let $\alpha := \mathrm{cong}(S, d)$ and let $\mathcal{C}$ denote the set of commodities. The space of all S-routings with congestion α is given by

$$\begin{aligned}
&\sum_{i \in \mathcal{C}} \sum_{f \in S(i)} \lambda_{f,i} f(e) \leq \alpha w(e) && \forall e \in E \\
&\sum_{f \in S(i)} \lambda_{f,i} \geq d(i) && \forall i \in \mathcal{C} \\
&\lambda_{f,i} \geq 0 && \forall i \in \mathcal{C}\ \forall f \in S(i)
\end{aligned}$$

As is known from linear programming, if a solution to this system exists, there is a one with at most $|E| + |\mathcal{C}| \leq 2n^2$ nonzero entries. □