Simple On-line Algorithms for the Maximum Disjoint Paths Problem*

Petr Kolman
Dept. of Applied Mathematics
Charles University
Malostranské nám. 25
118 00 Prague, Czech Republic
kolman@kam.ms.mff.cuni.cz

Christian Scheideler
Dept. of Computer Science
Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218, USA
scheideler@cs.jhu.edu

December 12, 2001

Abstract

In this paper we study the classical problem of finding disjoint paths in graphs. This problem has been studied by a number of authors both for specific graphs and general classes of graphs. Whereas for specific graphs many (almost) matching upper and lower bounds are known for the competitiveness of on-line algorithms, not much is known about how well on-line algorithms can perform in the general setting. The best results obtained so far use the expansion of a network. We use a different parameter called *routing number* that, as we will show, allows more precise results than the expansion. It enables us to prove tight upper and lower bounds for a class of simple deterministic on-line algorithms, called bounded greedy algorithms. Interestingly, our upper bound on the competitive ratio is even better than the best approximation ratio known for off-line algorithms before our paper. Furthermore, we introduce a refined variant of the routing number and show that this variant allows to construct on-line algorithms with a competitive ratio that can be significantly below the best possible upper bound for deterministic on-line algorithms if only the routing number or expansion of a network is known. We also show that our on-line algorithms can be transformed into efficient algorithms for the related unsplittable flow problem.

*The conference version of the paper appeared in SPAA 2001. Most of the work was done while the authors were members of the Heinz Nixdorf Institute at the Paderborn University. The first author was supported by the DFG-Graduiertenkolleg 124 "Parallele Rechnernetze in der Produktionstechnik", and the second author was supported in part by the DFG-Sonderforschungsbereich 376 "Massive Parallelität: Algorithmen, Entwurfsmethoden, Anwendungen"

1 Introduction

The *disjoint paths problem* (here called DPP) is defined as follows. Given an undirected graph $G = (V, E)$ and a set T of k pairs of nodes (s_i, t_i), $1 \leq i \leq k$, decide whether there exist k edge disjoint paths $P_1, \cdots, P_k$ such that the path P_i connects s_i and t_i. It was shown by Karp [18] that this is an NP–complete problem. The optimization variant of this problem is called the *maximum disjoint paths problem* (MDPP), which is simply to find the maximum subset of T for which there exist edge-disjoint paths. Several approximation algorithms have been proposed for it. A short summary is given in Section 1.1.

A generalization of the MDPP is the *unsplittable flow problem* (UFP) [19]: each edge $e \in E$ is given a capacity of c_e and each request (s_i, t_i) has a demand of d_i. The task is to choose a subset $T' \subseteq T$ such that each request (s_i, t_i) in T' can send d_i flow along a single path, all capacity constraints are kept, and the sum of the demands of the requests in T' is maximized. (More general forms of the UFP also assign a profit to each request, and the aim is to maximize the sum of the profits of the requests in T' [2].) In the *unit-capacity UFP*, $c_e = 1$ for all edges e.

We will mainly concentrate on how to solve the maximum disjoint paths problem in an on-line setting, that is, the requests arrive one after another, and for each of them the algorithm has to decide before knowing the next requests in the input sequence whether to accept it or not. If the request is accepted, a path has to be provided for it that is disjoint to all the paths established previously. This on-line variant of the MDPP is also called *call admission and routing problem* [10, 28]. Our aim is to find algorithms for this problem with a small competitive ratio. The *competitive ratio* of a deterministic on-line algorithm is defined as

$$c = \sup_{\sigma} \frac{OPT(\sigma)}{ON(\sigma)} ,$$

where the suprenum is taken over all possible sequences σ of requests, $ON(\sigma)$ is the number of requests accepted by the on-line algorithm, and $OPT(\sigma)$ is the number of requests accepted by an optimal off-line algorithm [10]. In the case of a randomized on-line algorithm, $ON(\sigma)$ is a random variable. We therefore define the competitive ratio of a randomized algorithm as

$$c = \sup_{\sigma} \frac{OPT(\sigma)}{\mathrm{E}[ON(\sigma)]} .$$

For the rest of the paper we will only compare the performance of our on-line algorithms against oblivious adversaries. Such an adversary does not see the decisions of the algorithm and therefore cannot take them into account for selecting the requests [35, 10]. We note, however, that our upper and lower bounds in Section 2 also hold for adaptive adversaries.

1.1 Previous work

Due to the NP-hardness of the MDPP, a lot of attention has been given to the search for good approximation algorithms for the MDPP problem. However, the problem seems to be hard to approximate. In the off-line setting, the best algoritm for general graphs has approximation ratio $O(\sqrt{m})$ [19]. For directed graphs there is almost a matching lower bound [17]. On the other hand, when more graph parameters than just the number of nodes, n, and the number of edges, m, are used to measure the performance of an algorithm, it is possible to get better results for many classes of graphs. The best published result of this type we are aware of is an $O(\Delta^2 \alpha^{-2} \log^3 n)$ approximation algorithm by Srinivasan [36], where Δ denotes the maximal degree in the graph and α the edge expansion of the graph (we stress that α may be a function of n, e.g. $\Theta(\log^{-1} n)$ for n-node butterfly graphs). The result is based on multicommodity flow algorithms, which is one of the most common approaches for the MDPP and related problems [32, 26, 27, 20, 37]. The other frequently used approach is based on random walks, which was useful especially for expander graphs [31, 11, 12, 14, 13]. Other important results for specific graphs are polylogarithmic and later $O(1)$ approximations for mesh-like graphs [5, 1, 21, 22, 19]. There are also a few results that relate the quality of the approximation ratio to the average path length d_0 in the optimal solution. Srinivasan [36] and later Kolliopoulos and Stein [23] by a different method gave d_0-approximation algorithms.

The $O(\sqrt{m})$ approximation for the MDPP applies also to the UFP ([8, 6], cf. [19]). For a special case of the uniform-capacity UFP (that is, all edges have the same capacity), Baveja and Srinivasan [8] describe an algorithm with approximation ratio $O(\Delta^2 \alpha^{-2} \log^3 n)$.

In the on-line setting, the trivial deterministic lower bound of $\Omega(n)$ for the line shows that in general there is no hope for on-line deterministic algorithms with reasonable competitive ratio. Bartal, Fiat and Leonardi [7] prove this effort to be in vain even for randomized algorithms by giving an $\Omega(n^\epsilon)$ lower bound for randomized on-line algorithms on general networks, where $\epsilon = \frac{2}{3}(1 - \log_4 3)$. As a consequence of these large lower bounds, research has mainly focused on specific topologies. On-line algorithms with an at most polylogarithmic competitive ratio have been found for the line network [15, 16, 4], trees [3, 4, 5], meshes [21], and certain classes of planar graphs [5]. All these algorithms are randomized. The reason for this is that the lower bounds for deterministic algorithms for many of these topologies are much higher. As we already mentioned above, for the line network there is a trivial lower bound of $\Omega(n)$ (e.g. [2]), which can be easily generalized to $\Omega(d)$ for any diameter d tree. Awerbuch et al. [5] mention a deterministic $\Omega(\sqrt{n})$ lower bound for the $\sqrt{n} \times \sqrt{n}$ mesh by Blum, Fiat, Karloff and Rabani. Kleinberg [19] provides an alternative proof. The known deterministic on-line algorithms for the MDPP with at most polylogarithmic competitive ratios are for the hex [5], for graphs with strong expansion properties [20] and for hypercubic networks [24]. Combination of the techniques of Kleinberg and Rubinfeld [20] and the results of Leighton and Rao [26] (see Lemma 1.2 of this paper) yields a competitive ratio of $O(\Delta\ \alpha^{-1} \log n)$ for general graphs for the special case that the sequence of requests forms a complete permutation on V. For arbitrary sequences, the bound on the competitive ratio increases to $O(\Delta^2 \alpha^{-2} \log^2 n)$ (the increase is caused due to the use of random walk techniques).

Most of the afore-mentioned randomized algorithms suffer from the drawback that only the *expected* competitive ratio is good. It may happen that they compute a very poor solution with high probability. Leonardi, Marchetti-Spaccamela, Presciutti and Rosén [29] consider this problem and propose alternative randomized algorithms for trees and meshes with almost optimal competitive ratios that achieve a good solution with high probability.

The problem appears to be much easier when requests allocate only a small fraction of link capacities. If each request requires at most a fraction of $O(1/\log n)$ of the link capacities, then there is an $O(\log n)$–competitive algorithm for general topologies by Awerbuch, Azar and Plotkin [2]. They also give a matching lower bound for this setting.

1.2 Terminology

Before we present our results, we introduce some notation. The *congestion* C of a path collection is defined as the maximum number of paths that share an edge, and the *dilation* D of a path collection is defined as the length of its longest path (measured in the number of edges).

Let S_n denote the set of all permutations from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$. Consider an arbitrary graph G on n nodes. For any permutation $\pi \in S_n$ and any D that is at least the diameter of G, let $C(G, D, \pi)$ be the minimum possible congestion required to route packets in G according to π using paths of length at most D. Then the *D-bounded routing number* $R(G,D)$ of G is defined as

$$R(G,D) = \max_{\pi} \max\{C(G,D,\pi), D\} \ .$$

Furthermore, the *(unbounded) routing number* $R(G)$ of G is defined as $R(G) = \min_D R(G,D)$. The notion of a routing number has been used before (see, for instance, [33]) and is usually defined via the minimum number of *steps*, rather than the minimum possible congestion and dilation, to route a permutation in G. However, since the original definition deviates only by a constant factor from the definition of a routing number above [25], we used the same name. In the case that there is no risk of confusion, we will simply write R instead of $R(G)$ or $R(G,D)$.

Obviously, the following lemma holds.

Lemma 1.1 *For any graph with D-bounded routing number R, there is a path collection for any permutation routing problem with congestion at most R and dilation at most D.*

Next we list the routing number and edge expansion of important classes of networks (we assume that the number of nodes is n).

network	routing number	expansion
line	$\Theta(n)$	$\Theta(1/n)$
$n' \times n'$-mesh	$\Theta(\sqrt{n})$	$\Theta(1/\sqrt{n})$
butterfly	$\Theta(\log n)$	$\Theta(1/\log n)$
hypercube	$\Theta(\log n)$	$\Theta(1)$
expander	$\Theta(\log n)$	$\Theta(1)$

The bounds imply that there should be a close relationship between the routing number and the expansion. In fact, Leighton and Rao [26] proved the following lemma.

Lemma 1.2 *For any graph with expansion α, maximal degree Δ and routing number R it holds that $\Theta(\alpha^{-1}) \leq R \leq \Theta(\Delta\alpha^{-1}\log n)$.*

The next two lemmata show that for constant-degree networks this result is best possible.

Lemma 1.3 *For any α, $1/n \leq \alpha \leq 1/\log n$, there exists a constant-degree graph G of size n with expansion $\Theta(\alpha)$ and routing number $\Theta(\alpha^{-1})$.*

Proof. We distinguish between two cases. First, $1/n^{1/2} \leq \alpha \leq 1/\log n$. In this case, consider a d-dimensional butterfly on n' nodes for some n' specified later. We note that $d = \Theta(\log n')$. From the table above it follows that this graph has an expansion of $\Theta(1/d)$ and a flow number of $\Theta(d)$. If we replace now every edge by a path of length ℓ, then the number of nodes of the new graph G increases to $n = \ell \cdot n'$ and the expansion decreases to $\alpha = \Theta(1/(d \cdot \ell))$. Furthermore, the routing number increases to $\Theta(d \cdot \ell)$. Hence, for any desired $\alpha, 1/n^{1/2} \leq \alpha \leq 1/\log n$, the graph G with an expansion α can be obtained by setting $\ell = \lfloor \alpha^{-1}/\log n \rfloor$ and $n' = n/\ell$ in the construction above.

Second, $1/n \leq \alpha \leq 1/n^{1/2}$. In this case, consider the mesh network with $\lfloor \alpha^{-1} \rfloor$ nodes in one dimension and $n/\lfloor \alpha^{-1} \rfloor$ nodes in the other dimension. It is easy to check that this graph has an expansion of $\Theta(\alpha)$ and a routing number of $\Theta(\alpha^{-1})$. □

Lemma 1.4 *For any $1 \leq \alpha = \Omega(\log n/n^{1-\epsilon})$, where ϵ is an arbitrary positive constant, and any $\Delta \geq 0$, there exists a constant-degree graph G of size n with expansion $\Theta(\alpha)$ and routing number $\Theta(\Delta\alpha^{-1}\log n)$.*

Proof. The lemma can be shown by using the same construction as for the butterfly in the proof of Lemma 1.3, but instead of a butterfly we use an expander. □

As implied by the table above, many standard networks have a routing number of $\Theta(\alpha^{-1})$. This will be important below to argue that the routing number usually provides better bounds than the expansion. Now we are ready to state our new results.

1.3 New Results

In this paper, we present a class of simple deterministic algorithms, called *bounded greedy algorithms*, that achieves for any graph G of maximum degree Δ and routing number R a competitive ratio of $O(\Delta \cdot R)$. Using Lemma 1.2, this implies a competitive ratio of $O(\Delta^2 \cdot \alpha^{-1} \log n)$, which is substantially better than the best approximation ratio of $O(\Delta^2 \cdot \alpha^{-2} \log^3 n)$ for *off-line* algorithms that was known before our paper and the competitive ratio of $O(\alpha^{-2} \log^2 n)$ that can be derived from the results of Kleinberg and Rubinfeld [20] (cf. Section 1.1). Since in general the best possible competitive ratio that can be obtained for a deterministic on-line algorithm when applied to a graph with routing number R (resp. expansion α and n nodes) is $\Omega(R)$ (resp. $\Omega(\alpha^{-1} \log n)$), and several standard graphs fulfill that $R = \Theta(\alpha^{-1})$, our $O(\Delta \cdot R)$ bound together with the results in Section 1.2 imply that the routing number is more useful for bounding the competitive ratio than the expansion. Another advantage of the routing number is that, in contrast to the expansion, it is quite easy to construct a constant factor approximation algorithm for the routing number of a graph (see, e.g., [33]).

Furthermore, we present a randomized on-line algorithm that, for any graph G with maximum degree Δ and D-bounded routing number R, achieves a competitive ratio of $O(\Delta \cdot \sqrt{D \cdot R})$ with high probability. Since D can be much smaller than R, this allows to achieve a competitive ratio that can be significantly below the best competitive ratio one can hope to achieve for deterministic on-line algorithms if only R (or α) is known. If we allow an edge to be used by up to two paths, we show that a competitive ratio of $O(\sqrt{\Delta \cdot D \cdot R})$ can even be achieved by a deterministic on-line algorithm. We also provide a lower bound of $\Omega(\sqrt{R} + D)$ on the competitive ratio that holds for all deterministic on-line algorithms.

Consequences of our results are off-line and on-line approximation algorithms for the unit-capacity UFP with same or similar approximation ratios as their counterparts for the MDPP.

2 The Bounded Greedy Algorithm

The *bounded greedy algorithm* (BGA) works as follows [19]. Let L be a suitably chosen parameter. Given a request, reject it if there is no free path of length at most L between its terminal nodes. Otherwise accept it and select any such path for it. First we prove two general lower bounds for the competitive ratio, and afterwards we provide a matching upper bound on the competitive ratio of the bounded greedy algorithm.

Theorem 2.1 *For any R and n for which there is a graph G of size n and routing number R there is a graph G' of size $\Theta(n)$ and routing number $\Theta(R)$ such that the competitive ratio of any deterministic on-line algorithm on G' is at least R.*

Proof. To obtain the graph G', attach a line of R edges to any one node of the graph G. Let $v_0, v_1, \cdots, v_R$ denote the nodes on the line. Consider the following two sequences of requests. The first sequence just consists of (v_0, v_R), and the second consists of (v_0, v_R) followed by (v_i, v_{i+1}) for all $i \in \{0, \ldots, R-1\}$. To have a bounded competitive ratio for the first sequence, any deterministic algorithm must accept (v_0, v_R). Hence, for the second sequence, a deterministic algorithm can only accept (v_0, v_R), whereas an optimal algorithm can accept R requests. $\square$

For the expansion, the following lower bound can be shown.

Theorem 2.2 *For every $1 \leq \alpha^{-1} \leq n/\log n$ there is a constant degree graph of size n with expansion $\Theta(\alpha)$ such that the competitive ratio of any deterministic on-line algorithm on G' is $\Omega(\alpha^{-1}\log n)$.*

Proof. According to [33] it holds that for every $1 \leq \alpha^{-1} \leq n/\log n$ there is a constant degree graph of size n that has an expansion of α and a diameter of $\Omega(\alpha^{-1}\log n)$. Let G be any such graph. Replace every edge in G by a path of length 3. Obviously, the resulting graph G' still has an expansion of $\Theta(\alpha)$ and a diameter of $\Omega(\alpha^{-1}\log n)$. Take now any two nodes v and w that are a distance of $\Omega(\alpha^{-1}\log n)$ apart. Since the on-line algorithm is deterministic, it will choose some fixed path of length $\Omega(\alpha^{-1}\log n)$ to connect these two nodes. Hence, the sequence consisting of (v, w) plus all pairs of nodes that are in the middle of the path pieces of length 3 (formerly representing edges in G) along the path from v to w will result in a competitive ratio of $\Omega(\alpha^{-1}\log n)$. $\square$

Next we prove an upper bound for the bounded greedy algorithm.

Theorem 2.3 *Given a network G of maximum degree Δ and routing number R, the competitive ratio of the BGA with parameter $L = 2 \cdot R$ on G is at most $(\Delta + 4) \cdot R + 1 = O(\Delta^2\alpha^{-1}\log n)$.*

Proof. Let $\mathcal{B}$ denote the set of paths for the requests accepted by the BGA and $\mathcal{O}$ be the set of paths in the optimal solution.

If $\mathcal{O}$ only consists of paths of length at most L, then the competitive ratio of the BGA is clearly at most $L+1$: if a request corresponding to a path $p \in \mathcal{O}$ is rejected by the BGA, then p must intersect in an edge with some other path accepted by the BGA. On the other hand, since the paths in the optimal solution are edge-disjoint, each request accepted and routed by the BGA can cause the rejection of at most L requests that appear in the optimal solution, namely of those that intersect with its route. Hence, in this case, $|\mathcal{O}| \leq (L+1) \cdot |\mathcal{B}|$.

However, there is no such guarantee that the optimal solution consists of short paths only (i.e. of paths of length at most L). There can be many long paths that do not intersect with any path that is used by the BGA. We need to bound the number of these. Fortunately, it is possible to transform an optimal solution that contains long paths into an 'illegal' solution consisting of short paths only that 'do not intersect too much' with the paths accepted by the BGA.

We say that a path $q \in \mathcal{B}$ is a *witness* for a path p if q and p share an edge in G. Obviously, a request that is routed in the optimal solution on a short path and is rejected by the BGA must have a witness in $\mathcal{B}$. Let $\mathcal{O}' \subset \mathcal{O}$ denote the set of paths that are longer than $2 \cdot R$ and that correspond to requests *not* accepted by the BGA and that do *not* have a witness in $\mathcal{B}$. Since all other paths in $\mathcal{O}$ either have a witness or are accepted by the BGA, we know from above that $|\mathcal{O} - \mathcal{O}'| \leq (1 + L) \cdot |\mathcal{B}|$. To be able to bound the size of the set $\mathcal{O}'$, we will transform $\mathcal{O}'$ into a set $\mathcal{S}$ of short paths that have the same pairs of endpoints as the paths in $\mathcal{O}'$ and, moreover, do not intersect too much with the paths of the BGA. To be more specific, we will ensure that (a) each path in $\mathcal{O}'$ has a path in $\mathcal{S}$ of length at most L connecting the same vertices, and (b) each path in $\mathcal{S}$ has a witness in $\mathcal{B}$ but all the paths in $\mathcal{B}$ altogether are witnesses to at most $(\Delta + 1) \cdot R \cdot |\mathcal{B}|$ paths in $\mathcal{S}$. This will complete the proof.

It remains to describe the transformation of $\mathcal{O}'$ into $\mathcal{S}$. For a path $p \in \mathcal{O}'$ between s and t let $a_{p,1} = s, a_{p,2}, \cdots, a_{p,R}$ denote its first R nodes and $b_{p,1}, \cdots, b_{p,R-1}, b_{p,R} = t$ its last R nodes. Let $\mathcal{L}$ be defined as the (multi-)set $\bigcup_{p \in \mathcal{O}'} \bigcup_{i=1}^{R} \{(a_{p,i}, b_{p,i})\}$. Since the paths in $\mathcal{O}'$ are edge-disjoint, each node of the graph G appears in at most Δ pairs in $\mathcal{L}$. Viewing the pairs as edges in G, Vizing's theorem

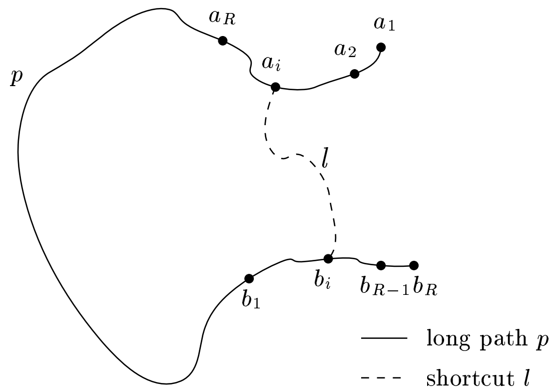

Figure 1: A shortcut l for a long path $p \in \mathcal{O}'$

[9, p. 153] implies that the pairs can be colored with $\Delta + 1$ colors so that no two adjacent pairs have the same color. Combining pairs of two color classes gives a graph that only consists of cycles and paths. Hence, directing the pairs in a suitable way, a combination of two color classes can be seen as a partial permutation routing problem. Thus, altogether the pairs can be split into $\lceil \frac{\Delta+1}{2} \rceil \leq \frac{\Delta+2}{2}$ (partial) permutation routing problems. It follows from Lemma 1.1 that there exists a set of paths connecting the pairs in $\mathcal{L}$ with congestion at most $\frac{\Delta+2}{2} \cdot R$ and dilation at most R. Let $\mathcal{P}$ be such a set of paths. For $l \in \mathcal{P}$ connecting nodes $a_{p,i}$ and $b_{p,i}$ of a long path $p \in \mathcal{O}'$, let p_l denote the path going first from $a_{p,1}$ to $a_{p,i}$ along the path p, then from $a_{p,i}$ to $b_{p,i}$ along l, and finally from $b_{p,i}$ to $b_{p,R}$ along p again (Figure 1).

The length of p_l is at most $2 \cdot R$. Therefore, l will be called a *shortcut* for the path p (recall that p was longer than $2 \cdot R$). The aim is now to choose a subset $\mathcal{P}' \subset \mathcal{P}$ such that each path in $\mathcal{O}'$ has a shortcut in $\mathcal{P}'$ and the paths in $\mathcal{B}$ are witnesses to at most $(\Delta + 2) \cdot R \cdot |\mathcal{B}|$ paths in $\mathcal{P}'$.

Let us perform a random experiment: independently for each long path $p \in \mathcal{O}'$, choose exactly one of its R shortcuts uniformly at random. Let $\mathcal{P}'$ be the set of the chosen shortcuts. For a fixed shortcut $l \in \mathcal{P}$, the probability that l is the chosen one, i.e. $l \in \mathcal{P}'$, is $1/R$. Let

$$X = \{(l, q) \mid l \in \mathcal{P}',\ q \in \mathcal{B},\ q \text{ is a witness for } l\} .$$

Since every path in $\mathcal{P}'$ must have a witness in $\mathcal{B}$, $|\mathcal{P}'| \leq |X|$ for any $\mathcal{P}'$. For every $l \in \mathcal{P}$, let $v_l = |\{q \mid q \in \mathcal{B},\ q \text{ is a witness for } l\}|$ and for every $q \in \mathcal{B}$, let $w_q = |\{l \mid l \in \mathcal{P},\ q \text{ is a witness for } l\}|$. Furthermore, for every $l \in \mathcal{P}$, let the binary random variable X_l be one if and only if l is chosen to be in $\mathcal{P}'$. Since $w_q \leq 2R \cdot (\frac{\Delta+2}{2} \cdot R)$, we obtain

$$\mathrm{E}[|X|] \leq \mathrm{E}\left[\sum_{l \in \mathcal{P}} v_l \cdot X_l\right] = \sum_{l \in \mathcal{P}} \frac{1}{R} \cdot v_l = \frac{1}{R} \sum_{q \in \mathcal{B}} w_q \leq (\Delta + 2) \cdot R \cdot |\mathcal{B}| .$$

It follows that there exists a set $\mathcal{P}'$ with $|X| \leq (\Delta + 2) \cdot R \cdot |\mathcal{B}|$. Since $|\mathcal{O}'| = |\mathcal{P}'|$ and $|\mathcal{P}'| \leq |X|$, this implies that also $|\mathcal{O}'| \leq (\Delta + 2) \cdot R \cdot |\mathcal{B}|$. Recalling that $|\mathcal{O} - \mathcal{O}'| \leq (1 + 2R) \cdot |\mathcal{B}|$, the proof is completed. □

It is worth noting that for the analysis of the BGA we do not need the conflicts between the paths in the transformed optimal and the greedy solution to be distributed evenly in the network. The important thing is the total number of the conflicts.

2.1 Decreasing the maximum path length

Is it possible to decrease the value of the parameter L in the BGA in order to obtain the same or even better bounds on the competitive ratio? For the case that we work with D-bounded routing numbers instead of simply routing numbers, the following theorem holds.

Theorem 2.4 *Given a network G of maximum degree Δ and D-bounded routing number R, the competitive ratio of the BGA with parameter $L = 2 \cdot D$ on G is at most $(\Delta + 4) \cdot R + 1$.*

Proof. The construction is exactly the same as in the proof of Theorem 2.3. Paths longer than $2 \cdot D$ in the optimal solution are suitably transformed into shorter ones and then it is shown that, on average, each path in the greedy solution intersects with at most $(\Delta + 4) \cdot R$ of the transformed paths, and each transformed path has a witness in $\mathcal{B}$. □

The main contribution of the above theorem is that it makes the BGA algorithm more efficient: it is computationally easier to search for paths of length at most $2 \cdot D$ than of length $2 \cdot R$.

Since graphs can be easily constructed where the competitive ratio of BGA$(2 \cdot D)$ is substantially better than that of BGA$(2 \cdot R)$, the question is whether the upper bound given in Theorem 2.4 is tight, or whether a BGA with parameter $L = o(R)$ can achieve a better competitive ratio than $O(R)$. We will show in the next theorem that there are certain limits to this, even when using the BGA in an off-line setting.

Theorem 2.5 *For any network G of size N with D-bounded routing number R where $\Theta(\log N) \leq D \leq \Theta(R)$, there is a network G' of size $\Theta(N)$ with $\Theta(D)$-bounded routing number $\Theta(R)$ so that the competitive ratio of the BGA with parameter L, where $\Theta(D) \leq L \leq \Theta(R)$, is $\Omega(R/L + L)$.*

Proof. To obtain the network G', we attach to a node of G a special network T with $\Theta(R)$ nodes and a diameter of $\Theta(D)$. Before we define this network for all R and D, we start with the situation that $R = \Theta(n)$ and $D = 2 \log n$.

The network T will be a graph consisting of the complete binary tree with n leaves that are connected via additional edges in such a way that they form a linear array. Obviously, no matter how the tree leaves are attached to the binary array, T has a diameter of $2 \log n$ and a $\Theta(\log n)$-bounded routing number of $O(n)$.

Consider now a BGA with parameter $L/2$, where $L = \ell \cdot \log n$ for some $\ell \geq 2$ (w.l.o.g. we will assume that ℓ is a power of two). We connect the leaves of T with the linear array in the following way:

Consider the linear array to be laid out as shown in Figure 2, with $2L$ nodes per 'column' and $n/(2L)$ leaves per 'row'. The node at column i and row j is called $v_{i,j}$. The nodes of every row are connected via a complete binary tree. The roots of these binary trees are connected in such a way that for every $i \in \{0, \ldots, \ell - 1\}$ the roots of the trees in rows $i \cdot \log n, i \cdot \log n + 1, \ldots, (i + 1) \log n - 1$ and rows $(2\ell - (i + 1)) \log n, (2\ell - (i + 1)) \log n + 1, \ldots, (2\ell - i) \log n - 1$ form a tree called T_i. The roots r_i of the trees T_i are connected by another complete binary tree $\hat{T}$ on top to from a single, complete binary tree.

Suppose now that we have a set S of requests $(v_{i,0}, v_{i,L})$ for all $i \in \{0, \ldots, n/(2L) - 1\}$. A path p connecting any of these requests is said to *cover* k leaves of $\hat{T}$ if the number of leaves r_i under (or equal to) the nodes in $\hat{T}$ visited by p is equal to k. Then the following lemma holds.

Lemma 2.6 *Any path p of length at most $L/2$ that connects a request in S must have the property that it covers at least $\ell/2$ leaves of $\hat{T}$.*

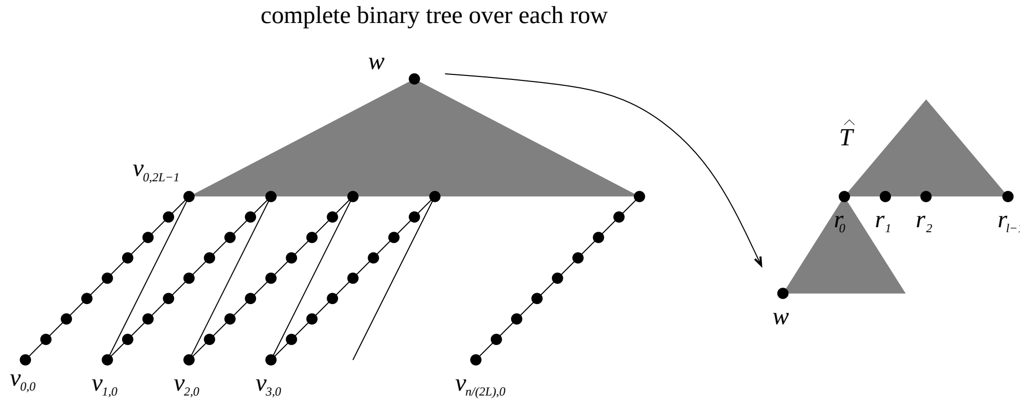

Figure 2: The construction of the network T from a linear array and a binary tree.

Proof. Consider a path p of length at most $L/2$ connecting any request in S. W.l.o.g. let this be the request $(v_{0,0}, v_{0,L})$. Suppose that p covers less than $\ell/2$ leaves of $\hat{T}$. Since each leaf allows to 'bridge' at most $\log n$ nodes along the columns of the linear array, the total vertical distance p can 'bridge' on the linear array from $v_{0,0}$ to $v_{0,L}$ must be less than $L/2$. Since the edges within a subtree T_i cannot be used to have a shortcut for the linear array, p would have to use more than $L/2$ further edges to get from $v_{0,0}$ to $v_{0,L}$. Since p is only allowed to have a length of at most $L/2$, this is a contradiction to our assumption. □

The lemma implies that the BGA with parameter $L/2$ is only able to connect at most 2 of the requests. Because the total number of requests is $n/(2L)$, we arrive at a competitive ratio of at least $R/(4L)$.

If the BGA uses parameter L instead of $L/2$ in our construction above, then all requests can be connected. However, in this case we know that $n/(2L) - 2$ of the corresponding paths must have a length of at least $L/2$. If $n/(2L) \geq 4$, that is $L \leq n/8$, then we extend the requests by adding L new requests requiring a path of length 1 overlapping with each request that got a path of length at least $L/2$. In this case we arrive at a competitive ratio of at least $\Omega(L)$.

Replacing now each (tree and linear array) edge in T of size R/δ by a path of length δ with $\delta = D/\log n$ also yields all other combinations of R and D, which concludes the proof of the theorem. □

The proof of Theorem 2.5 can also be used to prove the following result.

Theorem 2.7 *For any network G of size n with D-bounded routing number R there is a network G' of size $\Theta(n)$ with $\Theta(D)$-bounded routing number $\Theta(R)$ so that the competitive ratio of any deterministic on-line algorithm applied to G' is at least $\Omega(\sqrt{R} + D)$.*

Proof. The proof is based on the construction of T for $L = \Theta(\sqrt{R})$. ($\sqrt{R}$ minimizes the bound $R/L + L$ in Theorem 2.5). As in the proof of Theorem 2.5, we consider the set of requests $(v_{i,0}, w_{i,L})$ for all $i \in \{0, \ldots, n/(2L) - 1\}$. Any deterministic on-line algorithm either has to reject many of these requests or must connect many of them with a path of length at least $L/2$. In the latter case we extend the requests by many short requests that have to be rejected. This causes for both cases the competitive ratio to be at least $\Omega(\sqrt{R})$. If $D \geq \sqrt{R}$, then it follows from Theorem 2.1 that the competitive ratio can be improved to $\Omega(D)$. □

Iterative BGA. As mentioned already in the introduction, there are algorithms [23, 36] with approximation ratio d_0, where d_0 denotes the average path length in the optimal solution, for the given instance of the problem. Consider the following off-line modification of the BGA, called *Iterative* BGA: run the BGA $\log n$ times, starting with parameter $L = 1$, and doubling L in each subsequent run. Finally, as your solution, choose the best one.

Theorem 2.8 *For an instance of the MDPP, let d_0 denote the average path length in the optimal solution. Then the approximation ratio of the Iterative BGA is $8 \cdot d_0$.*

Proof. Clearly, since one of the choices for L must be in the interval $[2 \cdot d_0, 4 \cdot d_0 - 1]$, and since at least half of the paths in the optimal solution are shorter than $2 \cdot d_0$, using the witnessing argument in Theorem 2.3 results in an approximation ratio of at most $2(1 + (4d_0 - 1)) = 8 \cdot d_0$. □

2.2 Unsplittable flow problem

The BGA can also be efficiently used for solving the unit-capacity unsplittable flow problem in the off-line setting. Consider the following procedure. First, sort all the requests according to their demands, starting with the heaviest. Then run the BGA with $L = 2 \cdot R$ on the requests in this order.

Theorem 2.9 *Consider any unit-capacity UFP on a graph G, and let R denote the routing number of G and Δ be the maximal degree in G. Then the approximation ratio of the BGA when run on requests ordered according to their demands is $O(\Delta\ R) = O(\Delta^2 \alpha^{-1} \log n)$.*

Proof. For simplicity we will assume that each edge in G has integral capacity C and that also all requests are integral. This will influence our bounds only by a constant.

As usual, let $\mathcal{B}$ denote the set of paths for the requests accepted by the BGA and $\mathcal{O}$ be the set of paths in the optimal solution. The notion of the witness has to be modified. For this purpose the following notion will be useful. For a path $p \in \mathcal{B}$ or $p \in \mathcal{O}$ let $d(p)$ denote the demand of the corresponding request. For a set $\mathcal{Q}$ of paths let $||\mathcal{Q}|| = \sum_{p \in \mathcal{Q}} d(p)$, that is, $||\mathcal{Q}||$ denotes the sum of demands of its paths (for simplicity we will sometimes talk about a demand of a path, meaning the demand of the corresponding request). For an edge $e \in E$ and a path $p \in \mathcal{O}$, let $D(e,p)$ denote the sum of demands of all paths from $\mathcal{B}$ passing through e whose demand is at least as large as the demand of p, that is, $D(e,p) = ||\{q \mid q \in \mathcal{B},\ e \in q,\ d(q) \geq d(p)\ \}||$. A path $q \in \mathcal{B}$ is a *witness* for a path p if $d(q) \geq d(p)$ and q and p intersect in an edge e such that $D(e,p) + d(p) > C$. We say that q serves as a witness on the edge e and p has a witness on the edge e. Let $D(e) = ||\{q \mid q \in \mathcal{B},\ e \in q\ \}||$. We start with a simple observation.

Lemma 2.10 *For any path p and edge e: if p has a witness on e then $D(e,p) \geq C/2$.*

Proof. Let q be the witness of p on e. Assume, by contradiction, that $D(e,p) < C/2$. Then it easily follows that $d(q) < C/2$. Since $d(p) \leq d(q)$ and $D(e,p) + d(p) > C$ by the witness definition, we have a contradiction. □

Let $\mathcal{O}' \subset \mathcal{O}$ be the set of paths that are longer than $2 \cdot R$ and that correspond to requests *not* accepted by the BGA and that do *not* have a witness in $\mathcal{B}$. The next two bounds on $||\mathcal{O} - \mathcal{O}'||$ and $||\mathcal{O}'||$ complete the proof. □

Lemma 2.11

$$||\mathcal{O} - \mathcal{O}'|| \leq (1 + 4 \cdot R) \cdot ||\mathcal{B}|| \quad .$$

Proof. Consider the following partitioning of $\mathcal{O} - \mathcal{O}'$ into two parts. Let $\mathcal{O}_1 \subseteq \mathcal{O} - \mathcal{O}'$ consist of the paths corresponding to requests accepted by the BGA and let $\mathcal{O}_2 = (\mathcal{O} - \mathcal{O}') - \mathcal{O}_1$. First note that each $p \in \mathcal{O}_2$ must have a witness in $\mathcal{B}$. Let $E' \subseteq E$ denote the set of edges on which some path from $\mathcal{O}_2$ has a witness. Then $||\mathcal{O}_2|| \leq \sum_{e \in E'} C \leq 2 \cdot \sum_{e \in E'} D(e) \leq 4 \cdot R \cdot ||\mathcal{B}||$, with the help of Lemma 2.10 and the fact that all paths in $\mathcal{B}$ are of length at most $2 \cdot R$. Obviously $||\mathcal{O}_1|| \leq ||\mathcal{B}||$ which concludes the proof. □

Lemma 2.12

$$||\mathcal{O}'|| \leq 4 \cdot \Delta \cdot R \cdot ||\mathcal{B}|| \quad .$$

Proof. For this purpose we are going to modify the set of flows $\mathcal{O}'$ into a set of short flows only. For a path $p \in \mathcal{O}'$ between s and t let $a_{p,1} = s, a_{p,2}, \cdots, a_{p,R}$ denote its first R nodes and $b_{p,1}, \cdots, b_{p,R-1}, b_{p,R} = t$ its last R nodes. Let $\mathcal{L}$ be the multiset $\bigcup_{p \in \mathcal{O}'} \bigcup_{i=1}^{R} \bigcup_{j=1}^{d(p)} \{(a_{p,i}, b_{p,i})\}$ (recall our assumption that all $d(p)$'s are integral). Since the paths in $\mathcal{O}'$ satisfy the capacity constraints, each node of the graph G appears in at most $\Delta \cdot C$ pairs in $\mathcal{L}$. Thus, the pairs can be split into $\frac{\Delta \cdot C + 2}{2}$ (partial) permutation routing problems. It follows from Lemma 1.1 that there exists a set of paths connecting the pairs in $\mathcal{L}$ with congestion at most $R \cdot \frac{\Delta \cdot C + 2}{2}$ and dilation at most R.

Consider now the following random experiment: for each long path $p \in \mathcal{O}'$, choose uniformly and independently at random exactly one its $R \cdot d(p)$ shortcuts. Let $\mathcal{P}'$ be the set of the chosen shortcuts. For every $\ell \in \mathcal{P}$, let the binary random variable X_ℓ be one if and only if ℓ is chosen to be in $\mathcal{P}'$. Assume now that each of the shortcuts is used to carry the original demand of the corresponding long flow. Since the BGA processed the requests according to their demands, starting from the heaviest, each of the shortcuts in $\mathcal{P}'$ will have a witness. Let $E' \subseteq E$ denote the set of edges on which some path from $\mathcal{P}$ has a witness. Since $||\mathcal{O}'|| = ||\mathcal{P}'||$, it suffices to give an upper bound on $||\mathcal{P}'|| = E\left[||\mathcal{P}'||\right]$.

$$\begin{aligned} \mathrm{E}\left[||\mathcal{P}'||\right] &\overset{(1)}{\leq} \mathrm{E}\left[\sum_{e \in E'} \sum_{\ell \in \mathcal{P}: e \in \ell} d(\ell) \cdot X_\ell\right] \\ &\overset{(2)}{=} \sum_{e \in E'} \sum_{\ell \in \mathcal{P}: e \in \ell} \frac{d(\ell)}{R \cdot d(\ell)} \overset{(3)}{\leq} \sum_{e \in E'} \frac{1}{R} \cdot R \cdot \frac{\Delta \cdot C + 2}{2} \\ &\overset{(4)}{\leq} 2 \cdot \Delta \sum_{e \in E'} D(e) \overset{(5)}{\leq} 4 \cdot \Delta \cdot R \cdot ||\mathcal{B}|| \end{aligned}$$

The following facts were used in the reasoning:

(1) each path in $\mathcal{P}'$ must pass at least one edge in E'

(2) $E[X_\ell] = \frac{1}{R \cdot d(\ell)}$

(3) the congestion of paths for $\mathcal{L}$ is at most $R \cdot \frac{\Delta \cdot C + 2}{2}$

(4) Lemma 2.10 and the assumption that $\Delta \geq 2$

(5) all paths in $\mathcal{B}$ are of length at most $2 \cdot R$

□

The problem with the UFP in the on-line setting is that an acceptance of a single request with very small demand may cause a rejection of a request with very large demand which results in a competitive ratio that cannot be bounded in terms of the network G. That is why it is interesting to focus on problem instances for the on-line case that satisfy an additional constraint: an instance of the UFP is ϵ*–bounded* for $\epsilon > 0$, if the maximum demand is at most $1 - \epsilon$ [19]. Then we arrive at the following result for the on-line unit-capacity UFP:

Theorem 2.13 *Consider any unit-capacity UFP on a graph G, and let R denote the routing number of G and Δ the maximal degree in G. If the sequence of requests is ϵ–bounded then the competitive ratio of the BGA is $O(\epsilon^{-1}\Delta\ R) = O(\epsilon^{-1}\Delta^2\alpha^{-1}\log n)$.*

Proof. Consider the following modification of $D(e,p)$: let $D(e,p)$ be the sum of demands of paths in $\mathcal{B}$ passing through the edge e that were accepted by the BGA prior to appearance of the request corresponding to p in the input sequence. Then the bound of Lemma 2.10 changes to $||D(e,p)|| \geq \epsilon \cdot C$. □

3 Relaxing the Disjointness Constraint

In this section, slightly different conditions on the established paths are considered. Instead of insisting on edge–disjointness, a congestion of two is allowed, both for the on-line algorithm as well as for the optimal off-line solution. This will substantially simplify the proof of the performance of the on-line algorithm. In the next section it will be shown how this relaxation can be avoided at the cost of introducing randomization in the decisions.

Suppose we have a graph of D-bounded routing number R. The algorithm is again rather simple. We may think of the graph as a graph with two copies of each edge, a blue and a red one. Given a request (s,t), accept it whenever there exists a free path with at most $2\epsilon R$ blue edges and at most $2 \cdot D$ red edges for some fixed $\epsilon \in [D/R, 1]$. We stress that this is a deterministic algorithm.

Theorem 3.1 *Suppose we have a network G of maximum degree Δ and D-bounded routing number R. For any $\epsilon \in [D/R, 1]$, the competitive ratio of the BGA with parameters $(2\epsilon R, 2D)$ is at most $4 \cdot (\epsilon R + (\Delta + 3) \cdot D/\epsilon)$.*

Proof. We call a path q a *witness* for a path p if p and q share an edge in G, no matter whether their colors match or not. Let $\mathcal{B}$ denote the set of paths accepted by the BGA and $\mathcal{O}$ be the set of paths in the optimal solution. Let $\mathcal{O}' \subset \mathcal{O}$ denote the subset of paths that are longer than $2 \cdot (\epsilon R + D)$, that correspond to requests *not* accepted by the BGA and that do not have a witness in $\mathcal{B}$. Then $|\mathcal{O} - \mathcal{O}'| \leq 4 \cdot (\epsilon R + D) \cdot |\mathcal{B}|$.

As in the proof of Theorem 2.3, we are going to transform the paths in $\mathcal{O}'$ into paths fulfilling the restrictions of the BGA that, at the same time, do not intersect much with paths of the BGA. For a path $p \in \mathcal{O}'$ between s and t let $a_{p,1} = s, a_{p,2}, \cdots, a_{p,\epsilon R}$ denote its first ϵR nodes and $b_{p,1}, \cdots, b_{p,\epsilon R-1}, b_{p,\epsilon R} = t$ its last ϵR nodes. Let $\mathcal{L}$ be the set $\bigcup_{p\in\mathcal{O}'} \bigcup_{i=1}^{\epsilon R}\{(a_{p,i}, b_{p,i})\}$. Since every edge is used by at most two paths in $\mathcal{O}'$, each node of the graph G appears in at most $2 \cdot \Delta$ requests in $\mathcal{L}$. Similar to the proof of Theorem 2.3 there exists a set of paths connecting the requests in $\mathcal{L}$ with congestion at most $(\Delta+2) \cdot R$ and dilation at most D. Now, each path $p \in \mathcal{O}'$ chooses uniformly and independently at random exactly one of its possible shortcuts, say (a_p, b_p). We route all the shortcuts on the red edges and everything else (i.e., the initial and final parts of the paths in $\mathcal{O}'$) on the blue edges. This may cause up to two paths in $\mathcal{O}'$ to use the same blue edge. By exactly the same argument as in the proof of Theorem 2.3, it is possible to show that the expected congestion of shortcuts on red edges is $(\Delta + 2)/\epsilon$ only. Let $\mathcal{S}$ denote the set of all selected shortcuts. Obviously, every path in $\mathcal{S}$ must have a witness in $\mathcal{B}$. In particular, there must be a path in $\mathcal{B}$ using the corresponding edge also as a red edge. Since each path of the BGA consists of at most $2 \cdot D$ red edges, each path in $\mathcal{B}$ is a witness to at most $2 \cdot D \cdot (\Delta + 2)/\epsilon$ paths in $\mathcal{S}$. Thus, putting together the bounds on $|\mathcal{O} - \mathcal{O}'|$ and on $|\mathcal{O}'| = |\mathcal{S}|$, the competitive ratio of the algorithm is as desired. □

Obviously, the (asymptotically) best possible competitive ratio is reached when $\epsilon R = \Delta D/\epsilon$. If ϵ is required to be more than 1 for this, we simply use the BGA to obtain a competitive ratio of $O(R)$. Otherwise, we obtain the following result.

Corollary 3.2 *Suppose we have a network G of maximum degree Δ and D-bounded routing number R. Then the competitive ratio of the BGA with parameters $(2\sqrt{R/(\Delta \cdot D)}, 2D)$ is $O(\sqrt{\Delta \cdot D \cdot R})$.*

It is worth noting that the sizes of optimal solutions for congestion one and congestion two problems may differ dramatically. Think about the brick wall and let $a_1, \cdots, a_m$ denote the border nodes on the upper side, going from left to right, and $b_1, \cdots, b_m$ denote the border nodes on the lower side, going from right to left. If $\bigcup_{i=1}^{m}(a_i, b_i)$ is the set of requests, then the size of an optimal solution for congestion one is only 1, whereas for congestion two it is m.

4 The Shrewd Algorithm

In this section, we will present a randomized on-line algorithm that achieves a competitive ratio that is similar to the deterministic algorithm in Section 3. It consists of a preprocessing phase and a path selection phase.

4.1 Preprocessing

Suppose that we have a graph of maximum degree Δ and D-bounded routing number R. Before the algorithm selects any path, it first computes a path system $\mathcal{T}$ (consisting of a path for every source-destination pair) with dilation D and congestion $n \cdot R + O(\sqrt{n \cdot R \cdot \log n})$. This can be done in polynomial time [33]. Since the $O(\sqrt{n \cdot R \cdot \log n})$ term is significantly smaller than $n \cdot R$, we will assume in the following for simplification reasons that the congestion is at most $n \cdot R$. Afterwards, the algorithm randomly selects a *well-connected subset* $W \subseteq V$ of nodes: each node decides independently at random with probability $\frac{1}{\epsilon R}$ to belong to W, for a suitably chosen ϵ. Let n' denote the size of W. Obviously, $E[n'] = \frac{n}{\epsilon R}$. Furthermore, it follows from the Chernoff bounds that $n' = \Theta(\frac{n}{\epsilon R})$ w.h.p. if $\epsilon \leq 1/\log n$. Let $\mathcal{Q} \subseteq \mathcal{T}$ be a collection of paths that contains all paths in $\mathcal{T}$ for all pairs of nodes in W. The set $\mathcal{Q}$ will serve as a path system for W. Define the (*absolute*) congestion of an edge with regard to $\mathcal{Q}$ as the number of paths traversing it, and the *relative* congestion of an edge as its absolute congestion divided by n'. These parameters have the following property.

Lemma 4.1 *For any fixed edge, its absolute congestion concerning $\mathcal{Q}$ is at most $\mathrm{E}[n']/\epsilon$ in the expected case and, for $\epsilon \leq 1/\log n$, also $O(\mathrm{E}[n']/\epsilon)$ with high probability. Furthermore, its expected relative congestion is at most $1/\epsilon$.*

The proof of the lemma can be found in the appendix. In the following, we call all edges with congestion of n' or more due to paths in $\mathcal{Q}$ *heavy* edges and all other edges *light*.

4.2 Path selection

Given a request, the *Shrewd algorithm* accepts it whenever there is a free path between its terminal nodes consisting of at most $2 \cdot \epsilon R + 4 \cdot D$ edges of which at most $4 \cdot D$ are heavy. Such paths are called *legal paths*. What is the idea behind this strategy? The heavy edges are (usually) edges that represent bottlenecks in the path system. When selecting paths for the requests, the algorithm avoids using too many bottleneck edges per path, because a single path passing through many bottlenecks could cause the rejection of many subsequent requests.

Theorem 4.2 *Suppose we have a network G of maximum degree Δ and D-bounded routing number R. For any $\epsilon \in [D/R, 1]$, the competitive ratio of the shrewd algorithm with parameters $(2\epsilon R, 4 \cdot D)$ is $O(\Delta(\epsilon R + D/\epsilon))$ in the expected case and also $O(\Delta(\epsilon R + D/\epsilon))$ w.h.p. if $\epsilon \leq 1/\log n$.*

Proof. Let $\mathcal{O}$ denote the set of paths in the optimal solution, $\mathcal{B}$ the set of paths accepted by the shrewd algorithm and let $\mathcal{O}' \subset \mathcal{O}$ consist of all illegal paths in $\mathcal{O}$ whose corresponding requests were rejected by the shrewd algorithm and that, moreover, have no witness in $\mathcal{B}$ (i.e., they do not intersect in an edge with any path in $\mathcal{B}$). The proof idea is the same again: we transform $\mathcal{O}'$ into an 'illegal' solution of almost the same size consisting of legal paths only that do not intersect much with paths in $\mathcal{B}$. By 'illegal' solution we mean that the modified paths are not mutually edge-disjoint. The transformation heavily depends on the path system $\mathcal{Q}$ for the well-connected subset W. It is done in two main steps. First, we show how to connect for most of the paths $p \in \mathcal{O}'$ its end nodes s and t to two nodes $a_s, a_t \in W$. Then, in the second step, for each pair (a_s, a_t) a path between a_s and a_t is constructed. This is done with the help of $\mathcal{Q}$. The main difficulty in the proof is to ensure that the resulting modified paths will be legal and that they will not intersect much with paths from $\mathcal{B}$.

Step 1: With each terminal node s of a path $p \in \mathcal{O}'$ a subpath p_s of the path p is associated. The subpath p_s is the shorter one of the following two:

- a subpath of length ϵR starting in s,
- a minimal subpath containing $D/2$ heavy edges starting in s.

In the first case, s and p_s are called *insecure*, in the other case *secure*. Since each path in $\mathcal{O}'$ between, say, s and t, contains either more than $4D$ heavy edges or more than $2\epsilon R + 4D$ edges altogether, the two constructed subpaths p_s and p_t are (node) disjoint and are well defined.

First we show how to connect secure nodes to nodes from W. Each secure subpath p_s chooses uniformly and independently at random one of its $D/2$ heavy edges, say f_s (choice 1), and then each of the chosen heavy edges chooses, again uniformly and independently at random, one of the paths from the path system $\mathcal{Q}$ that are passing through it (choice 2). Let q_s denote the path chosen by f_s. The desired node $a_s \in W$ for s is the one of the two terminal nodes of q_s that is closer to f_s w.r.t. q_s (Figure 3). Clearly, the combination of p_s and q_s between s and a_s contains at most D heavy edges and at most $\epsilon R + D$ edges in total.

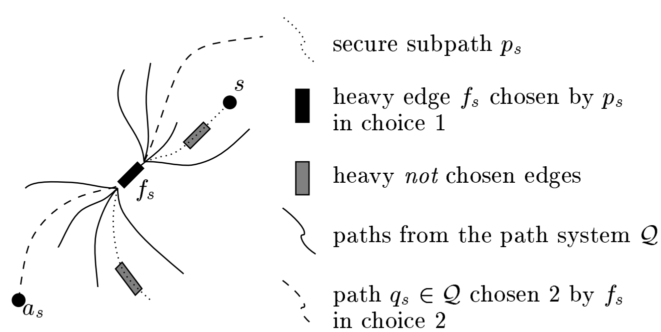

Figure 3: Connection between a secure node s and a node $a_s \in W$

In the case of the insecure nodes, only a part of them will be provided with a node from W. Since the nodes in W were chosen independently at random and since the insecure subpaths are quite long (ϵR edges), many of them will contain a node from W. If the subpath p_s contains at least one node from W, then the closest of them to the terminal node s is chosen as the desired a_s. If there is no such node on it, then no node $a_s \in W$ is provided for s and the corresponding request will therefore not be able to participate in step 2.

If $|\mathcal{O}'| = O(\epsilon R + \Delta \cdot D/\epsilon)$, we do not care how many of the paths in $\mathcal{O}'$ cannot be connected to two nodes in W. Since $|\mathcal{B}| \geq 1$, the competitive ratio claimed in Theorem 4.2 would immediately follow. Otherwise, let $\mathcal{O}'' \subseteq \mathcal{O}'$ denote the subset of paths for which nodes a_s and a_t in W can be provided. The following lemma states that $\mathcal{O}''$ contains many of the paths in $\mathcal{O}'$. It's proof can be found in the appendix.

Lemma 4.3 *For any $\mathcal{O}'$ with $|\mathcal{O}'| = \omega(\epsilon R + \Delta \cdot D/\epsilon)$, $|\mathcal{O}''| \geq |\mathcal{O}'|/4$ with high probability.*

Step 2: Let $\mathcal{L} = \bigcup_{p \in \mathcal{O}''} \{(a_{p,s}, a_{p,t})\}$. It remains to provide connections for all the pairs (a_s, a_t) in $\mathcal{L}$. For this we use the path system $\mathcal{Q}$ and Valiant's trick with random intermediate destinations: each pair $(a_s, a_t) \in \mathcal{L}$ chooses uniformly and independently at random an intermediate destination $c_{st} \in W$ (choice 3) and uses the paths from the path system $\mathcal{Q}$ to connect a_s with c_{st} and c_{st} with a_t (Figure 4). For a path $p \in \mathcal{O}''$ from s to t, let l_p denote the path between a_s and a_t via c_{st} as described above.

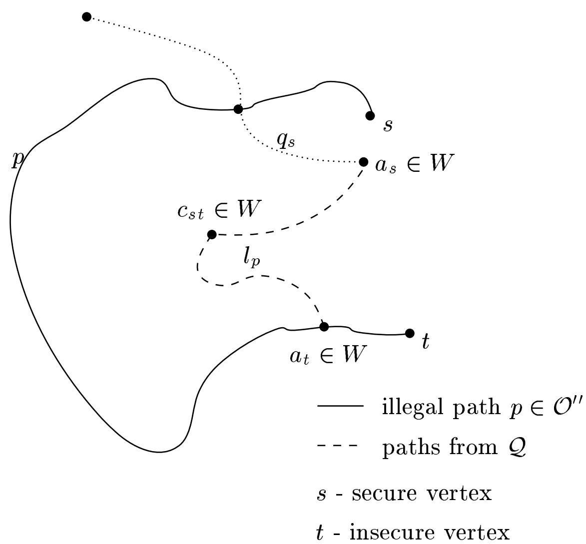

Figure 4: A modification of an illegal path p.

From the description of the modification it follows that all the modified paths are legal.

Bounding the congestion: Let k_l be the total number of light edges and k_h the total number of heavy edges used by the paths in $\mathcal{B}$. Let $\mathcal{P}$ denote the set of all paths for the pairs in $\mathcal{L}$, that is $\mathcal{P} = \bigcup_{p \in \mathcal{O}''} l_p$, and let $\mathcal{U} = \bigcup_{s \text{ secure}} q_s$.

Consider any node $u \in W$. We are going to bound the number of pairs in $\mathcal{L}$ in which u appears. First, we bound the number of secure nodes that chose u as a_s or a_t. For each heavy edge f, let y_f be the number of paths in $\mathcal{Q}$ that are passing through f and terminate in u. Let $H_u = \{f \mid y_f > 0\}$. Note that $\sum_{f \in H_u} y_f \leq n' \cdot D$, because u is terminal node of n' paths in $\mathcal{Q}$ and all paths in $\mathcal{Q}$ have length at most D. For an edge $f \in H_u$, the probability that f was chosen in choice 1 is at most $\frac{2}{D}$. For an edge $f \in H_u$ that *was chosen* in choice 1, the probability that f chose in choice 2 a path terminating in u is at most $\frac{y_f}{n'}$. Thus, the expected number of secure nodes that chose u as a_s or a_t is bounded by $\sum_{f \in H_u} \frac{2}{D} \cdot \frac{y_f}{n'} \leq 2$. Concerning insecure nodes, at most Δ of them can choose u as a_s or a_t since the paths in $\mathcal{O}'$ are edge-disjoint. In total, the expected number (with respect to the random choices 1, 2 and 3) of pairs from $\mathcal{L}$ terminating in u is at most $\Delta + 2$.

Consider now any light edge e in the graph. We want to bound the congestion of paths in $\mathcal{U}$ and $\mathcal{P}$ on e. A path $q \in \mathcal{U}$ adds to the congestion of e if there is a heavy edge f on q such that f was chosen in choice 1 and, moreover, the edge f chose the path q in choice 2 and q passes through e (Figure 5). A path $l \in \mathcal{P}$ between a_s and a_t adds to the congestion of e if the pair (a_s, a_t) chose such a node c_{st} in choice 3 that either the path between a_s and c_{st} or between c_{st} and a_t in the path system $\mathcal{Q}$ is passing through e.

For each heavy edge f, let x_f be the number of paths in $\mathcal{Q}$ that are passing both through e and f. Let $H_e = \{f \mid x_f > 0\}$. Note that $\sum_{f \in H_e} x_f \leq n' \cdot D$, because e is a light edge and all paths in $\mathcal{Q}$ have length at most D. For an edge $f \in H_e$, the probability that f was chosen in choice 1 is at most $\frac{2}{D}$. For an edge $f \in H_e$ that *was chosen* in choice 1, the probability that f chose in choice 2 a path going through e is at most $\frac{x_f}{n'}$. The expected congestion of paths from $\mathcal{U}$ on e is thus bounded by $\sum_{f \in H_e} \frac{2}{D} \cdot \frac{x_f}{n'} \leq 2$.

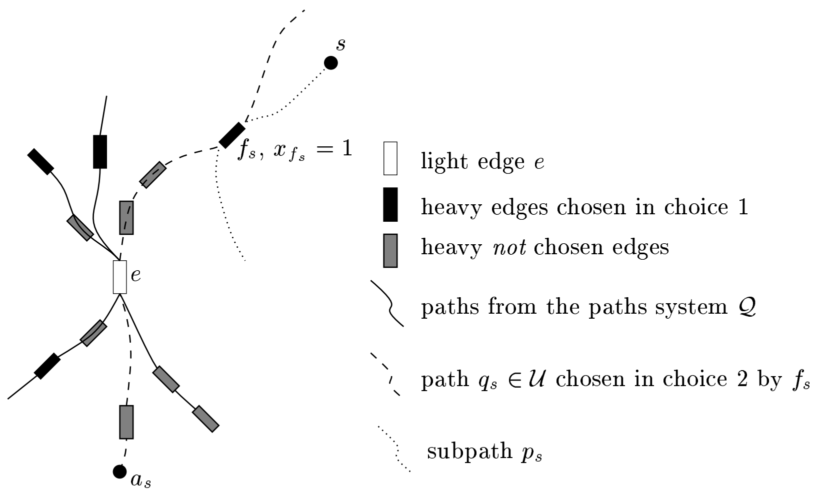

Figure 5: Congestion of paths in $\mathcal{U}$ on a light edge e.

For each node $u \in W$, let x_u be the number of paths in $\mathcal{Q}$ that terminate in u and go through e. Note that $\sum_{u \in W} x_u \leq n'$. We noticed above that for a fixed $u \in W$, the expected number of pairs from $\mathcal{L}$ in which u appears is at most $\Delta + 2$. For each of these pairs (a_s, a_t), the probability that it chose such an intermediate node c_{st} in choice 3 that either the path between a_s and c_{st} or between c_{st} and a_t from $\mathcal{Q}$ is passing through e (depending on whether $a_s = u$ or $a_t = u$) is at most $\frac{x_u}{n'}$. The expected congestion of paths from $\mathcal{P}$ on e is thus bounded by $\sum_{u \in W} (\Delta + 2) \cdot \frac{x_u}{n'} \leq \Delta + 2$. In total, the expected congestion on a light edge is $\Delta + 4$.

Recall the congestion bound in Lemma 4.1. Consider now any heavy edge e in the graph. Let the random variable C_e denote the number of paths in $\mathcal{Q}$ that traverse e. Using the same arguments as above, the expected congestion of paths from $\mathcal{U}$ on e is at most $2C_e/n'$, and the expected congestion of paths from $\mathcal{P}$ on e is at most $(\Delta + 2)C_e/n'$. Thus, the expected number of conflicts between the modified paths and the paths in $\mathcal{B}$ is at most $(\Delta + 4) \cdot k_l + (\Delta + 4) \sum_{\text{heavy } e \in \mathcal{B}} C_e/n'$. We conclude that there exist choices 1, 2 and 3 with that many conflicts at most. According to Lemma 4.1 we know that $\mathrm{E}[C_e/n'] \leq 1/\epsilon$ (recall that C_e/n' is the relative congestion of e) and that for $\epsilon \leq 1/\log n$ both $n' = \Theta(\mathrm{E}[n'])$ and $C_e = O(\mathrm{E}[n']/\epsilon)$ w.h.p. Hence, both in the expected case and the high probability case with $\epsilon \leq 1/\log n$, the number of conflicts between the modified paths and the paths in $\mathcal{B}$ is at most $(\Delta+4) \cdot k_l + (\Delta+4) \cdot O(k_h/\epsilon)$. This is also the maximal number of paths in $\mathcal{S}$ that have a witness in $\mathcal{B}$.

Summary: For a path $p \in \mathcal{O}''$ between s and t, let p' denote its modification as described in the two steps. That is, p' goes from s to a_s first, then from a_s to a_t via c_{st}, and finally from a_t to t. Let $\mathcal{S} = \bigcup_{p \in \mathcal{O}''} p'$ denote the set of the modifications. With high probability (with respect to the *initial* random choice of W), $|\mathcal{S}| \geq |\mathcal{O}'|/4$, that is, most of the paths in $\mathcal{O}'$ have a modification in $\mathcal{S}$ (Lemma 4.3). All paths in $\mathcal{S}$ are legal and because requests corresponding to them were rejected by the shrewd algorithm, each of them must have a witness in $\mathcal{B}$. Thus, in the expected case, $|\mathcal{S}| \leq (\Delta+4) \cdot k_l + (\Delta+4) \cdot k_h/\epsilon \leq (\Delta+4) \cdot (2\epsilon \cdot R + 4 \cdot D/\epsilon) \cdot |\mathcal{B}| = O(\Delta \cdot (\epsilon R + D/\epsilon)) \cdot |\mathcal{B}|$. Also, w.h.p. $|\mathcal{S}| = O(\Delta \cdot (\epsilon R + D/\epsilon)) \cdot |\mathcal{B}|$ for $\epsilon \leq 1/\log n$. Hence, the shrewd algorithm is $O(\Delta \cdot (\epsilon R + D/\epsilon))$–competitive in the expected case and also $O(\Delta \cdot (\epsilon R + D/\epsilon))$–competitive w.h.p. if $\epsilon \leq 1/\log n$. □

Choosing the best possible ϵ, we arrive at the following result.

Corollary 4.4 *Suppose we have a network G of maximum degree Δ and D-bounded routing number R. The competitive ratio of the shrewd algorithm with parameters $(2\sqrt{D \cdot R}, 4 \cdot D)$ is $O(\Delta\sqrt{D \cdot R})$ in the expected case and also $O(\Delta\sqrt{D \cdot R})$ w.h.p. if $R \geq D \log^2 n$.*

In the same way as the BGA, the shrewd algorithm can be used to solve the unit-capacity UFP problem. Use several runs of the shrewd algorithm to transform the expected competitive ratio into an approximation ratio that holds w.h.p. for any R and D.

Corollary 4.5 *Consider any unit-capacity UFP on a graph G of maximum degree Δ and D-bounded routing number R. Then the approximation ratio of the shrewd algorithm with parameters $(2\sqrt{D \cdot R}, 4 \cdot D)$, when run on requests ordered according to their demands, is $O(\Delta\sqrt{D \cdot R})$, w.h.p.*

Corollary 4.6 *Consider any ϵ–bounded unit-capacity UFP on a graph G of maximum degree Δ and D-bounded routing number R. Then the competitive ratio of the shrewd algorithm with parameters $(2\sqrt{D \cdot R}, 4 \cdot D)$ is $O(\epsilon^{-1}\Delta\sqrt{D \cdot R})$, w.h.p.*

5 Conclusions

In this paper we presented a simple deterministic on-line algorithm for general networks with an optimal competitive ratio when using the routing number of a network. Furthermore, we introduced a new parameter called the D-bounded routing number and showed that with the help of this parameter on-line algorithms can be constructed with a competitive ratio that can be significantly below the best possible upper bound of a deterministic on-line protocol if only the routing number of a graph is known. Our upper and lower bounds for the case of using bounded routing numbers are not tight. It is therefore an interesting open question what the best possible competitive ratio is that can be reached by deterministic or randomized on-line algorithms in this setting. Furthermore, it would be interesting to know whether simple algorithms can reach such an optimal ratio.

References

[1] Y. Aumann and Y. Rabani. Improved bounds for all optical routing. In *Proc. of the 6th ACM-SIAM Symposium on Discrete Algorithms*, pages 567–576, 1995.

[2] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive on-line routing. In *Proc. of the 34th IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[3] B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosén. Competitive non-preemptive call control. Manuscript, 1993.

[4] B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosén. Competitive non-preemptive call control. In *Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 312–320, 1994.

[5] B. Awerbuch, R. Gawlick, T. Leighton, and Y. Rabani. On-line admission control and circuit routing for high performance computing and communication. In *Proc. of the 35th IEEE Symposium on Foundations of Computer Science*, pages 412–423, 1994.

[6] Y. Azar and O. Regev. Strongly polynomial algorithms for the unsplittable flow problem. In *Proc. of the 8th Conference on Integer Programming and Combinatorial Optimization*, 2001.

[7] Y. Bartal, A. Fiat, and S. Leonardi. Lower bounds for on-line graph problems with application to on-line circuit and optical routing. In *Proc. of the 28th ACM Symposium on the Theory of Computing*, pages 531–540, 1996.

[8] A. Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research*, 25, 2000.

[9] B. Bollobas. *Modern Graph Theory.* Springer, New York, 1998.

[10] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[11] A. Broder, A. Frieze, and E. Upfal. Existence and construction of edge-disjoint paths on expander graphs. *SIAM Journal on Computing*, 23:976–989, 1994.

[12] A. Broder, A. Frieze, and E. Upfal. Static and dynamic path selection on expander graphs: A random walk approach (extended abstract). In *Proc. of the 29th ACM Symposium on Theory of Computing*, pages 531–539, 1997.

[13] A. Broder, A. Frieze, and E. Upfal. Static and dynamic path selection on expander graphs: A random walk approach. *RSA: Random Structures & Algorithms*, 14, 1999.

[14] A. Frieze. Disjoint paths in expander graphs via random walks: A short survey. In *RANDOM: International Workshop on Randomization and Approximation Techniques in Computer Science*, 1998.

[15] J. Garay, I. Gopal, S. Kutten, Y. Mansour, and M. Yung. Efficient on-line call control algorithms. In *Proceedings of the 2nd Israeli Symposium on Theory of Computing and Systems*, pages 285–293, 1993.

[16] J. Garay, I. Gopal, S. Kutten, Y. Mansour, and M. Yung. Efficient on-line call control algorithms. *Journal of Algorithms*, 23:180–194, 1997.

[17] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. In *Proc. of the 31st ACM Symposium on Theory of Computing*, pages 19–28, 1999.

[18] R. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[19] J. Kleinberg. *Approximation Algorithms for Disjoint Paths Problems.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1996.

[20] J. Kleinberg and R. Rubinfeld. Short paths in expander graphs. In *Proc. of the 37th IEEE Symposium on Foundations of Computer Science*, pages 86–95, 1996.

[21] J. Kleinberg and É. Tardos. Disjoint paths in densely embedded graphs. In *Proc. of the 36th IEEE Symposium on Foundations of Computer Science*, pages 52–61, 1995.

[22] J. Kleinberg and É. Tardos. Approximations for the disjoint paths problem in high-diameter planar networks. *Journal of Computer and System Sciences*, 57:61–73, 1998.

[23] S. Kolliopoulos and C. Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *Proc. of the 6th Conference on Integer Programming and Combinatorial Optimization*, pages 153–162, 1998.

[24] P. Kolman. Short disjoint paths on hypercubic graphs. Technical Report 2000-481, Charles University, KAM-DIMATIA Series, 2000. Submitted to Journal of Interconnection Networks.

[25] F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet routing and job-shop scheduling in $O(\text{congestion} + \text{dilation})$ steps. *Combinatorica*, 14(2):167–186, 1994.

[26] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proc. of the 29th IEEE Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[27] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787–832, 1999.

[28] S. Leonardi. On-line network routing. In A. Fiat and G. Woeginger, editors, *Online Alogithms*, volume 1442 of *Lecture Notes in Computer Science*, pages 242–267. Springer, 1998.

[29] S. Leonardi, A. Marchetti-Spaccamela, A. Presciutti, and A. Rosén. On-line randomized call control revisited. In *Proc. of the 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 323–332, 1998.

[30] A. Panconesi and A. Srinivasan. Fast randomized algorithms for distributed edge coloring. In *Proc. of the ACM Symposium on Principles of Distributed Computing*, pages 251–262, 1992.

[31] D. Peleg and E. Upfal. Constructing disjoint paths on expander graphs. *Combinatorica*, 9, 1989.

[32] P. Raghavan and C. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[33] C. Scheideler. *Universal Routing Strategies for Interconnection Networks.* Lecture Notes in Computer Science 1390, Springer Verlag, 1998.

[34] J. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM Journal on Discrete Mathematics*, 8(2):223–250, 1995.

[35] D. Sleator and R. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

[36] A. Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In *Proc. of the 38th IEEE Symposium on Foundations of Computer Science*, pages 416–425, 1997.

[37] A. Srinivasan. A survey of the role of multicommodity flow and randomization in network and routing. In S. R. P. Pardalos and J. Rolim, editors, *Randomization Methods in Algorithm Design*, volume 43 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science.*, pages 271–302. American Mathematical Society, 1999.

A Proof of Lemma 4.1

We first prove bounds for the absolute congestion. Because the congestion of $\mathcal{T}$ is at most $n \cdot R$ and every node is chosen independently at random with probability $\frac{1}{\epsilon R}$ to belong to W, the expected congestion at any fixed edge is at most

$$n \cdot R \cdot \left(\frac{1}{\epsilon R}\right)^2 = \frac{n}{\epsilon R} \cdot \frac{1}{\epsilon} = \frac{E[n']}{\epsilon} .$$

The main problem for finding a bound for the congestion that holds with high probability is that the probabilities for the paths traversing an edge may not be independent.

In the following, we assume that $\epsilon \leq 1/\log n$. Suppose for a moment that the paths had independent probabilities to be taken. Let $p = (\frac{1}{\epsilon R})^2$ represent this probability, and for any sequence $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$ let

$$P_j(z) = \sum_{i_1 < i_2 < \ldots < i_j} z_{i_1} z_{i_2} \cdots z_{i_j} .$$

Consider some fixed edge e. Let $\mathcal{T}_e$ denote the set of all paths in $\mathcal{T}$ that are traversing e, and assume these paths to be numbered from 1 to $m = |\mathcal{T}_e|$. Furthermore, for every path i let the binary random variable X_i be one if and only if path i belongs to $\mathcal{Q}$, and let $X = \sum_i X_i$. Then we would have that

$$\mathrm{E}[P_j(X_1, \ldots, X_m)] = P_j(p, \ldots, p)$$

for any $j \in \{1, \ldots, m\}$. We will not be able to show this for our situation, but what we can show is that up to some large enough $k \in \{1, \ldots, m\}$ there is a p' close to p with the property that

$$\mathrm{E}[P_j(X_1, \ldots, X_m)] \leq P_j(p', \ldots, p')$$

for all $j \leq k$. As we will see later, this allows us to use Chernoff bounds to estimate the probability that X is far away from its expected value.

Proposition A.1 *For any $k \in \{1, \ldots, m\}$ it holds with $p' = \min[1, \frac{k^2}{m}] + \min[1, \frac{2k \cdot n}{m}] \cdot \frac{1}{\epsilon R} + \left(\frac{1}{\epsilon R}\right)^2$ that*

$$\mathrm{E}[P_j(X_1, \ldots, X_m)] \leq P_j(p', \ldots, p')$$

for any $j \leq k$.

Proof. Let path i be represented by its source-destination pair (s_i, t_i). Take any subset $U \subseteq \mathcal{T}_e = \{(s_1, t_1), \ldots, (s_m, t_m)\}$ of size at most $k-1$ and any $(s_i, t_i) \in \{(s_1, t_1), \ldots, (s_m, t_m)\} \setminus U$. We distinguish between three cases.

1. If both s_i and t_i already appear in other pairs in U, then we only know that

$$\Pr\left[X_{(s_i,t_i)} = 1 \mid \prod_{(s_l,t_l) \in U} X_{(s_l,t_l)} = 1\right] = 1 .$$

2. If exactly one of s_i and t_i appears in U, then

$$\Pr\left[X_{(s_i,t_i)} = 1 \mid \prod_{(s_l,t_l) \in U} X_{(s_l,t_l)} = 1\right] = \frac{1}{\epsilon R} .$$

3. If none of s_i and t_i appear in U, then

$$\Pr\left[X_{(s_i,t_i)} = 1 \mid \prod_{(s_l,t_l)\in U} X_{(s_l,t_l)} = 1\right] = \left(\frac{1}{\epsilon R}\right)^2 .$$

Suppose now that (s_i, t_i) is chosen uniformly at random out of $\{(s_1, t_1), \ldots, (s_m, t_m)\} \setminus U$. Then the probability for case (1) is at most

$$\min\left[1, \frac{(k-1)^2 - (k-1)}{m-(k-1)}\right] \le \min\left[1, \frac{k^2}{m}\right] ,$$

since there can be at most $\binom{2(k-1)}{2} \le (k-1)^2$ pairs of nodes that use nodes in U. Furthermore, the probability for case (2) is at most

$$\min\left[1, \frac{2(k-1)\cdot(n-1)-(k-1)}{m-(k-1)}\right] \le \min\left[1, \frac{2k\cdot n}{m}\right] ,$$

because in the worst case every one of the at most $2(k-1)$ different nodes in U has all of the $n-1$ paths from $\mathcal{T}$, that can have it as endpoint, running through e. Combining these probabilities with the probabilities in the cases above, we get that for (s_i, t_i) chosen uniformly at random out of $\{(s_1, t_1), \ldots, (s_m, t_m)\} \setminus U$,

$$\Pr\left[X_{(s_i,t_i)} = 1 \mid \prod_{(s_l,t_l)\in U} X_{(s_l,t_l)} = 1\right] \le \min\left[1, \frac{k^2}{m}\right] + \min\left[1, \frac{2k\cdot n}{m}\right]\cdot\frac{1}{\epsilon R} + \left(\frac{1}{\epsilon R}\right)^2 .$$

Hence, for any $j \le k$,

$$\mathrm{E}[P_j(X_1, \ldots, X_m)] \le P_j(p', \ldots, p') ,$$

where $p' = \min[1, \frac{k^2}{m}] + \min[1, \frac{2k\cdot n}{m}]\cdot\frac{1}{\epsilon R} + \left(\frac{1}{\epsilon R}\right)^2$. □

Using the techniques in [34] (in particular, see inequality (1) on page 227), it follows from Proposition A.1 that for $m = R\cdot n$, $\mu = p'\cdot m$, and any $\delta > 0$,

$$\Pr[X \ge (1+\delta)\mu] \le \mathrm{e}^{-(\delta^2\mu/3+\delta\mu/3+k/2)} .$$

According to the definition of p',

$$\mu = \Theta\left(k^2 + \frac{2k\cdot n}{\epsilon R} + \frac{n\cdot R}{(\epsilon R)^2}\right) = \Theta(k^2 + \mathrm{E}[n']\cdot k + \mathrm{E}[n']/\epsilon) .$$

Choosing $k = \Theta(\log n)$, we obtain that $\mu = \Theta(E[n']/\epsilon)$ for any $\epsilon \le 1/\log n$. Using this in the probability bound above, we obtain a polynomially small probability that $X \ge (1+\delta)\mu$ for some constant $\delta > 0$. Since the probability bound also holds for all $m \le R\cdot n$ (just introduce dummy paths to get back to $m = R\cdot n$), the proof for the bounds of the absolute congestion is completed.

Next we consider the relative congestion. Consider some fixed edge e. Let C_e denote the absolute congestion caused by paths in $\mathcal{Q}$ traversing e. Furthermore, let $\bar{C}_e = C_e/n'$ represent its relative congestion. It holds that

$$\begin{aligned}\mathrm{E}[\bar{C}_e] &= \sum_{c,m}\frac{c}{m}\cdot\Pr[C_e = c \wedge n' = m] \\ &= \sum_m \frac{1}{m}\cdot\Pr[n' = m]\cdot\sum_c c\cdot\Pr[C_e = c \mid n' = m] .\end{aligned}$$

Furthermore,

$$\Pr[n' = m] = \binom{n}{m} \left(\frac{1}{\epsilon R}\right)^m \left(1 - \frac{1}{\epsilon R}\right)^{n-m}$$

and if $n' = m$, then the probability for a fixed path to belong to $\mathcal{Q}$ is equal to

$$\frac{\binom{n-2}{m-2}}{\binom{n}{m}} = \frac{m(m-1)}{n(n-1)} \leq \left(\frac{m}{n}\right)^2 .$$

Hence,

$$\sum_c c \cdot \Pr[C_e = c \mid n' = m] = \mathrm{E}[C_e \mid n' = m] = n \cdot R \cdot \left(\frac{m}{n}\right)^2 = \frac{R \cdot m^2}{n} .$$

Therefore,

$$\begin{aligned}
\mathrm{E}[\bar{C}_e] &\leq \sum_{m=1}^{n} \frac{1}{m} \cdot \binom{n}{m} \left(\frac{1}{\epsilon R}\right)^m \left(1 - \frac{1}{\epsilon R}\right)^{n-m} \cdot \frac{R \cdot m^2}{n} \\
&= \frac{R}{n} \sum_{m=1}^{n} m \cdot \binom{n-1}{m-1} \frac{n}{m} \cdot \left(\frac{1}{\epsilon R}\right)^m \left(1 - \frac{1}{\epsilon R}\right)^{n-m} \\
&= \frac{R}{n} \cdot n \cdot \frac{1}{\epsilon R} \sum_{m=1}^{n} \binom{n-1}{m-1} \left(\frac{1}{\epsilon R}\right)^{m-1} \left(1 - \frac{1}{\epsilon R}\right)^{n-m} \\
&= \frac{1}{\epsilon} .
\end{aligned}$$

This completes the proof of the lemma.

B Proof of Lemma 4.3

For every path $q \in \mathcal{O}'$, let the binary random variable X_q be one if and only if two nodes in W can be provided for q. Furthermore, let the binary random variable Y_q be one if and only if among the first and among the last ϵR nodes in p (called in the following *source part* and *destination part*) there is at least one node in W. Obviously, if $Y_q = 1$ then also $X_q = 1$. Thus,

$$\sum_{q \in \mathcal{O}'} Y_q \leq \sum_{q \in \mathcal{O}'} X_q .$$

Hence, we obtain for $X = \sum_q X_q$ and $Y = \sum_q Y_q$ that

$$\Pr[Y \leq c] \geq \Pr[X \leq c]$$

for all $c \geq 0$. Thus, for any p with $\Pr[Y \leq c] \leq p$ it also holds that $\Pr[X \leq c] \leq p$. We will therefore continue in the following to bound $\Pr[Y \leq c]$.

Since the nodes decide independently of each other to belong to W, the probability that the source part resp. destination part of the path q contains a node in W is equal to

$$1 - \left(1 - \frac{1}{\epsilon R}\right)^{\epsilon R} \geq 1 - \frac{1}{\mathrm{e}} .$$

Hence, $\Pr[Y_p = 1] \geq (1 - 1/\mathrm{e})^2$ for all $q \in \mathcal{O}'$, which implies that $\mathrm{E}[Y] \geq (1 - 1/\mathrm{e})^2 |\mathcal{O}'| \geq \frac{2}{5}|\mathcal{O}'|$. Unfortunately the Y_p are not independent. This is due to the fact that parts of paths may overlap.

However, knowing that a certain set of paths has no node in W can only increase the probability that also some other path has no node in W, since some of its nodes may be contained in these paths and every node not contained in any one of these paths still has an independent probability of belonging to W. Hence, for any $q \in \mathcal{O}'$ and any subset of paths $U \subseteq \mathcal{O}' \setminus \{q\}$ we have

$$\Pr\left[(1 - Y_q) = 1 \mid \prod_{p \in U}(1 - Y_p) = 1\right] \leq \Pr[(1 - Y_q) = 1]$$

Thus, the random variables $Z_q = (1 - Y_q)$ are self-weakening with parameter $\lambda = 1$ (see [34, 30] for the definition). This implies [34, 30] that we can use the usual Chernoff bounds to obtain that for $\mu = (1 - (1 - 1/\mathrm{e})^2)|\mathcal{O}'|$ and for any $0 < \delta \leq 1$ we have

$$\Pr[\sum_q Z_q \geq (1 + \delta)\mu] \leq \mathrm{e}^{-\delta^2\mu/2} .$$

Hence,

$$\Pr[\sum_q (1 - Y_q) \geq (1 + \delta)\mu] \leq \mathrm{e}^{-\delta^2\mu/2} ,$$

and therefore

$$\Pr[Y \leq |\mathcal{O}'| - (1 + \delta)\mu] \leq \mathrm{e}^{-\delta^2\mu/2} .$$

Since

$$|\mathcal{O}'| - (1 + \delta)\mu = ((1 - 1/\mathrm{e})^2 - \delta(1 - (1 - 1/e)^2))|\mathcal{O}'| \geq (2/5 - \delta \cdot 3/5)|\mathcal{O}'|$$

and $\Pr[Y \leq c] \geq \Pr[X \leq c]$ for all c, we get

$$\Pr[X \leq (2/5 - \delta \cdot 3/5)|\mathcal{O}'|] \leq \mathrm{e}^{-\delta^2\mu/2} .$$

As any network of maximum degree Δ must have a diameter of at least $\log_{\Delta-1} n$ and we have $\epsilon \leq 1$, it follows that $\Delta \cdot D/\epsilon \geq \log n$. Thus, $|\mathcal{O}'| = \omega(\log n)$ and therefore also $\mu = \omega(\log n)$. Hence, the probability that $X \leq |\mathcal{O}'|/4$ is polynomially small in n, which proves the lemma.