Algorithms for Fault-Tolerant Routing in Circuit Switched Networks

Amitabha Bagchi, Amitabh Chaudhary, and Christian Scheideler
Dept. of Computer Science
Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218, USA
{bagchi,amic,scheideler}@cs.jhu.edu

Petr Kolman
Inst. for Theoretical Computer Science*
Charles University
Malostranské nám. 25
118 00 Prague, Czech Republic
kolman@kam.mff.cuni.cz

*The Institute is supported by the Ministry of Education of the Czech Republic as project LN00A056.

Abstract

In this paper we consider the k *edge-disjoint paths problem* (k-EDP), a generalization of the well-known edge-disjoint paths problem. Given a graph $G = (V, E)$ and a set of terminal pairs (or requests) T, the problem is to find a maximum subset of the pairs in T for which it is possible to select paths such that each pair is connected by k edge-disjoint paths and the paths for different pairs are mutually disjoint. To the best of our knowledge, no nontrivial result is known for this problem for $k > 1$. To measure the performance of our algorithms we will use the recently introduced flow number F of a graph. This parameter is known to fulfill $F = O(\Delta\alpha^{-1} \log n)$, where Δ is the maximum degree and α is the edge expansion of G. We show that a simple, greedy online algorithm achieves a competitive ratio of $O(k^3 \cdot F)$, which naturally extends the best known bound of $O(F)$ for $k = 1$ to higher k. To achieve this competitive ratio, we introduce a new method of converting a system of k disjoint paths into a system of k length-bounded disjoint paths. We also show that any deterministic online algorithm has a competitive ratio of $\Omega(k \cdot F)$.

In addition, we study the k *disjoint flows problem* (k-DFP), which is a generalization of the well-known unsplittable flow problem (UFP). The k-DFP is similar to the k-EDP with the difference that we now consider a graph with edge capacities and the requests can have arbitrary demands d_i. The aim is to find a subset of requests of maximum total demand for which it is possible to select flow paths such that all the capacity constraints are maintained and each selected request with demand d_i is connected by k disjoint paths, each of flow value d_i/k.

The k-EDP and k-DFP problems have important applications in fault-tolerant (virtual) circuit switching, which plays a key role in optical networks.

1 Introduction

This paper was motivated by a talk given by Rakesh Sinha from Ciena Inc. at the DIMACS Workshop on Resource Management and Scheduling in Next Generation Networks, March 26-27, 2001. The speaker pointed out in his talk that standard problems such as the edge-disjoint paths problem and the unsplittable flow problem are insufficient for practical purposes: they do not allow a rapid adaptation to edge faults or heavy load conditions. Instead of having just one path for each request, it would be much more desirable to determine a collection of alternative independent paths for each accepted request that can flexibly be used to ensure rapid adaptability. The paths, however, should be chosen so that not too much bandwidth is wasted under normal conditions. Keeping this in mind, we introduce two new (to the best of our knowledge) optimization problems: the k edge-disjoint paths problem (k-EDP) and the k disjoint flows problem (k-DFP).

In the k-EDP we are given an undirected graph $G = (V, E)$ and a set of terminal pairs (or requests) T. The problem is to find a maximum subset of the pairs in T such that each chosen pair can be connected by k disjoint paths and, moreover, the paths for different pairs are mutually disjoint.

Similarly, in the k-DFP we are given an undirected network $G = (V, E)$ with edge capacities and a set of terminal pairs T with demands d_i, $1 \leq i \leq |T|$. The problem is to find a subset of the pairs of maximum total demand such that each chosen pair can be connected by k disjoint paths, each path is carrying d_i/k units of flow and no capacity constraint is violated.

In order to demonstrate that the k-DFP can be used to achieve fault tolerance together with a high utilization of the network resources and also a rapid adaptability, consider a network G in which new edge faults may occur continuously but the total number of faulty edges at the same time is at most f. In this case, given a request with demand d, the strategy is to reserve $k+f$ disjoint flow paths for it, for some $k \geq 1$, with total demand $(1+f/k)d$. As long as at most f edge faults appear at the same time, it will still be possible to ship a demand of d along the remaining paths. Furthermore, under fault-free conditions, only a fraction f/k of the reserved bandwidth is wasted, which can be made sufficiently small by setting k sufficiently large (which will, of course, be limited by the properties of the network).

1.1 Previous results

Since we are not aware of previous results for the k-EDP and the k-DFP for $k > 1$, we will just survey results for the heavily studied case of $k = 1$, that is, the *edge-disjoint paths problem* (EDP) and the more general *unsplittable flow problem* (UFP).

Several results are known about the approximation ratio and competitive ratio achievable for the UFP under the assumption that the maximum demand of a commodity, $d_{\max}$, does not exceed the minimum edge capacity, $c_{\min}$, called here the *no-bottleneck assumption* [1, 12, 4, 7, 10, 13, 14]. If only the number of edges, m, is known, Baveja and Srinivasan [4] present a polynomial time algorithm with approximation ratio $O(\sqrt{m})$. On the lower bound side, it was shown by Guruswami et al. [10] that on directed networks the UFP is NP-hard to approximate within a factor of $m^{1/2-\epsilon}$ for any $\epsilon > 0$. The best result for the EDP and the UFP known so far was given by Kolman and Scheideler [14]. Using a new parameter called the *flow number* F of a network, they show that a simple online algorithm has a competitive ratio of $O(F)$ and prove that $F = O(\Delta\alpha^{-1}\log n)$, where for the EDP Δ is the maximal degree of the network, α is the edge expansion, and n is the number of nodes. For the UFP, Δ has to be defined as the maximal node capacity of the network for the bound above to hold, where the capacity of a node is defined as the sum of the capacities of its edges. Combining the approach of Kolman and Scheideler [14] with the AAP algorithm [1], Chakrabarti et al. [7] recently proved an approximation ratio of $O(\Delta^2\alpha^{-1}\log^2 n)$ for the more general UFP with profits where Δ is just the maximal degree of the network.

We also consider two related problems, the *integral splittable flow problem* (ISF) [10] and the *k-splittable flow problem* (k-SFP). In both cases, the input and the objective (i.e., to maximize the sum of accepted demands) are the same as in the UFP. The difference is that in the ISF all demands are integral and a flow satisfying a demand can be split into several paths, each carrying an integral amount of flow. In the k-SFP[1] a demand may be split into up to k flow paths (of not necessarily integral values). Under the no-bottleneck assumption Guruswami et al. [10] give an $O(\sqrt{md_{\max}}\log^2 m)$ approximation for the ISF. Recent results of Kolman and Scheideler [14] allow to achieve an $O(F)$ randomized competitive ratio and an $O(F)$ deterministic approximation ratio for both of these problems on uniform capacity

[1]The k-splittable flow problem was recently independently introduced by Baier et al. [3].

networks. Although the ISF and the k-SFP on one side and the k-DFP on the other seem very similar at first glance, there is a serious difference between the two. Whereas the ISF and the k-SFP are relaxations of the UFP (they allow the use of more than one path for a single request and the paths are *not* required to be disjoint), the k-DFP is actually a more complex version of the UFP since it requires several *disjoint* paths for a single request.

1.2 New results

The main results or this paper are

- a deterministic online algorithm for the k-EDP with competitive ratio $O(k^3F)$,
- a deterministic offline algorithm for the k-DFP on unit-capacity networks with an approximation ratio of $O(k^3F\log(kF))$,
- a lower bound $\Omega(k \cdot F)$ for the competitive ratio of any deterministic online algorithm for the k-EDP (and thus, obviously, for the k-DFP).

Thus, for constant k, we have matching upper and lower bounds for the k-EDP.

Furthermore, we demonstrate that the disjointness condition of the k paths for every single request seems to be the crucial condition that makes the problems above harder than other problems such as the integral splittable flow problem [10] or the k-splittable flow problem.

Using known techniques, we also show how the online algorithm for the k-EDP can be transformed into an offline algorithm with approximation ratio $O(k^3F)$ for the k-EDP with profits, and we describe how the offline algorithm for the k-DFP can be converted into a randomized online algorithm for the k-DFP with an expected competitive ratio of $O(k^3F\log(kF))$.

Our algorithms for the k-EDP and k-DFP are based on a simple concept, which is a natural extension of the *bounded greedy algorithm* (BGA) that has already been studied in several papers [12, 13, 14]: for every request for which there are still k disjoint flow paths of total length at most L available without violating the capacity constraints, select any such system of k paths for it. The core of the paper is in the analysis of this simple algorithm. The problem is to show that this strategy works even if the optimal offline algorithm connects many requests via k disjoint paths of total length more than L. In order to solve this problem we use a new technique, based on Menger's theorem and the Lovász Local Lemma, that converts large systems of k disjoint paths into small systems of k disjoint paths. Previously, shortening strategies were only known for $k = 1$ [13, 14].

1.3 Basic notation and techniques

Many of the previous techniques for the EDP and related problems do not allow us to prove strong upper bounds on approximation or competitive ratios due to the use of inappropriate parameters. If m is the only parameter used, an upper bound of $O(\sqrt{m})$ is essentially the best possible for the case of directed networks [10]. Much better ratios can be shown if the expansion or the routing number [16] of a network are used. These measures give very good bounds for low-degree networks with uniform edge capacities, but are usually very poor when applied to networks of high degree or highly nonuniform degree or edge capacities. To get more precise bounds for the approximation and competitive ratios of algorithms, Kolman and Scheideler [14] introduced a new network measure, the *flow number* F. Not only does the flow number lead to more precise results, it also has the major advantage that, in contrast to the expansion or the routing number, it can be computed exactly in polynomial time. Hence we will use the flow number in this paper as well.

Before we can introduce the flow number, we need some notation. In a *concurrent multicommodity flow problem* there are k commodities, each with two terminal nodes s_i and t_i and a demand d_i. A *feasible* solution is a set of flow paths for the commodities that obey capacity constraints but need not meet the specified demands. An important difference between this problem and the unsplittable flow problem is that the commodity between s_i and t_i can be routed along multiple paths. The *(relative) flow value of a feasible solution* is the maximum f such that at least $f \cdot d_i$ units of commodity i are simultaneously routed for each i. The *max-flow* for a concurrent multicommodity flow problem is defined as the maximum flow value over all feasible solutions. For a path p in a solution, the *flow value of p* is the amount of flow routed along it. A special class of concurrent multicommodity flow problems is the *product multicommodity flow problem* (PMFP). In a PMFP, a nonnegative weight $\pi(u)$ is associated with each node $u \in V$. There is a commodity for every pair of nodes and the demand for the pair (u, v) is equal to $\pi(u) \cdot \pi(v)$.

Suppose we have a network $G = (V, E)$ with arbitrary non-negative edge capacities. For every node v, let the *capacity* of v be defined as $c(v) = \sum_{w:\{v,w\}\in E} c(v,w)$ and the capacity of G be defined as $\Gamma = \sum_v c(v)$. Given a concurrent multicommodity flow problem with feasible solution $\mathcal{S}$, let the *dilation* $D(\mathcal{S})$ of $\mathcal{S}$ be defined as the length of the longest flow path in $\mathcal{S}$ and the *congestion* $C(\mathcal{S})$ of $\mathcal{S}$ be defined as the inverse of its flow value (i.e., the congestion tells us how many times the edge capacities would have to be increased in order to fully satisfy all the original demands, along the paths of $\mathcal{S}$). Let I_0 be the PMFP in which $\pi(v) = c(v)/\sqrt{\Gamma}$ for every node v, that is, each pair of nodes (v,w) has a commodity with demand $c(v) \cdot c(w)/\Gamma$. The *flow number* $F(G)$ of a network G is the minimum of $\max\{C(\mathcal{S}), D(\mathcal{S})\}$ over all feasible solutions $\mathcal{S}$ of I_0. When there is no risk of confusion, we will simply write F instead of $F(G)$. Note that the flow number of a network is invariant to scaling of capacities.

The smaller the flow number, the better are the communication properties of the network. For example, $F(\text{line}) = \Theta(n)$, $F(\text{mesh}) = \Theta(\sqrt{n})$, $F(\text{hypercube}) = \Theta(\log n)$, $F(\text{butterfly}) = \Theta(\log n)$, and, $F(\text{expander}) = \Theta(\log n)$. In the analysis of the presented algorithms, a useful tool will be the *Shortening lemma* [14].

Lemma 1.1 (Shortening Lemma) *For any network with flow number F it holds: for any $\epsilon \in (0,1]$ and any feasible solution $\mathcal{S}$ to an instance of the concurrent multicommodity flow problem with a flow value of f, there exists a feasible solution with flow value $f/(1+\epsilon)$ that uses paths of length at most $2 \cdot F(1+1/\epsilon)$. Moreover, the flow through any edge e not used by $\mathcal{S}$ is at most $\epsilon \cdot c(e)/(1+\epsilon)$.*

Another useful class of concurrent multicommodity flow problems is the *balanced multicommodity flow problem* (or short BMFP). A BMFP is a multicommodity flow problem in which the sum of the demands of the commodities originating and the commodities terminating in a node v is at most $c(v)$ for every $v \in V$. We will make use of the following property of the problem [14]:

Lemma 1.2 *For any network G with flow number F and any instance I of a BMFP for G, there is a feasible solution for I with congestion and dilation at most $2F$.*

Apart from the flow number we will also need Chernoff bounds [11], the (symmetric form of the) Lovász Local Lemma [9] and Menger's theorem [6, p. 75].

Lemma 1.3 (Chernoff Bound) *Consider any set of n independent binary random variables $X_1, \ldots, X_n$. Let $X = \sum_{i=1}^{n} X_i$ and μ be chosen so that $\mu \geq \mathrm{E}[X]$. Then it holds for all $\delta \geq 0$ that*

$$\Pr[X \geq (1+\delta)\mu] \leq \mathrm{e}^{-\min[\delta^2,\, \delta]\cdot\mu/3} .$$

Lemma 1.4 (Lovász Local Lemma) *Let $A_1, \ldots, A_n$ be "bad" events in an arbitrary probability space. Suppose that each event is mutually independent of all other events but at most b, and that $\Pr[A_i] \leq p$ for all i. If $\mathrm{e}p(b+1) \leq 1$ then, with probability greater than 0, no bad event occurs.*

Lemma 1.5 (Menger's theorem) *Let s and t be distinct vertices of G. Then the minimal number of edges separating s from t is equal to the maximal number of edge-disjoint s-t paths.*

1.4 Organization of the paper

In Section 2 we present our upper and lower bounds for the k-EDP and some related problems, and in Section 3 we present our upper bounds for the k-DFP. The paper ends with a conclusion and open problems.

2 Algorithms for the k-EDP

Consider the following extension of the bounded greedy algorithm: Let L be a suitably chosen parameter. Given a request, if it is possible to find k edge-disjoint paths, $p_1, p_2, \ldots p_k$ between the terminal nodes of the request that are mutually disjoint with the previously selected paths and that fulfill $\sum_{i=1}^{k} |p_i| \leq L$, where $|p|$ is the length (i.e., the number of edges) of a path p, then accept the request and select any such collection of paths for it. Otherwise, reject the request. Let us call this algorithm k-BGA.

Note that the problem of finding k edge-disjoint paths $p_1, p_2, \ldots p_k$ of total length at most L can be reduced to the classical min-cost (integral) flow problem, which can be solved by standard methods in polynomial time [8, Chapter 4]. It is worth mentioning that if there were a bound of L/k on the length of every path, the problem would not be tractable any more (cf. [5]).

2.1 The upper bound

Theorem 2.1 *Given a network G of flow number F, the competitive ratio of the k-BGA with parameter $L = 20k^3F$ is $O(k^3F)$.*

Proof. In the following, we call the k edge-disjoint paths that were selected for a request a *k-system*. A k-system is *small* if it has at most L edges.

Let $\mathcal{B}$ be the solution obtained by the k-BGA and $\mathcal{O}$ be the optimal solution. For notational simplicity we allow a certain ambiguity. Sometimes $\mathcal{B}$ and $\mathcal{O}$ refer to the subsets of T of the satisfied requests, and sometimes to the actual k-systems that realize the satisfied requests. We say that a k-system $q \in \mathcal{B}$ is a *witness* for a k-system p if p and q share an edge. Obviously, a request with a small k-system in the optimal solution that was rejected by the k-BGA must have a witness in $\mathcal{B}$.

Let $\mathcal{O}' \subseteq \mathcal{O}$ denote the set of all k-systems in $\mathcal{O}$ that are larger than L and that correspond to requests *not* accepted by the k-BGA and that do *not* have a witness in $\mathcal{B}$. Then each k-system in $\mathcal{O} - \mathcal{O}'$ either has a witness or was accepted by the k-BGA. Since the k-systems in $\mathcal{O} - \mathcal{O}'$ are edge-disjoint, each request accepted by the k-BGA can be a witness to at most L requests in $\mathcal{O} - \mathcal{O}'$. Hence, $|\mathcal{O} - \mathcal{O}'| \leq (1 + L)|\mathcal{B}|$.

It remains to prove an upper bound on $|\mathcal{O}'|$. To achieve this, we transform the k-systems in $\mathcal{O}'$ into a set $\mathcal{P}$ of possibly overlapping but *small* k-systems. Since these small k-systems would have been candidates for the k-BGA but were not picked, each of them has at least one witness in $\mathcal{B}$. Then we show that the small k-systems in $\mathcal{P}$ do not overlap much and thus many k-systems from $\mathcal{B}$ are needed in order to provide a witness for every k-system in $\mathcal{P}$.

Note that the set $\mathcal{O}'$ of k-systems can be viewed as a feasible solution of relative flow value 1 to the set of requests $\mathcal{O}'$ of the concurrent multicommodity flow problem where each request has demand k. The Shortening lemma with parameter $\epsilon = 1/(2k)$ immediately implies the following fact.

Fact 2.2 *The k-systems in $\mathcal{O}'$ can be transformed into a set $\mathcal{R}$ of flow systems transporting the same amount of flow such that every flow path has a length of at most $5k \cdot F$. Furthermore, the congestion at every edge used by a k-system in $\mathcal{O}'$ is at most $1+1/(2k)$, and the congestion at every other edge is at most $1/(2k)$.*

This does not immediately provide us with short k-systems for the requests in $\mathcal{O}'$. However, it is possible to extract short k-systems out of the flow system $\mathcal{R}$.

Lemma 2.3 *For every request in $\mathcal{O}'$, a set of small k-systems can be extracted out of its flow system in $\mathcal{R}$ with a total flow value of at least $1/4$.*

Proof. Let (s_i, t_i) be a fixed request from $\mathcal{O}'$ and let E_i be the set of all edges that are traversed by the flow system for (s_i, t_i) in $\mathcal{R}$. Consider any set of $k-1$ edges in E_i. Since the edge congestion caused by $\mathcal{R}$ is at most $1+1/(2k)$, the total amount of flow in the flow system for (s_i, t_i) in $\mathcal{R}$ that traverses the $k-1$ edges is at most $(k-1)(1+1/(2k)) \leq k-1/2$. Thus, the minimal $s_i - t_i$-cut in the graph (V, E_i) consists of at least k edges. Hence, Menger's theorem [6] implies that there are k edge-disjoint paths between s_i and t_i in E_i. We take any such k paths and denote them as the k-system σ_1. We associate a *weight* (i.e., total flow) of $k \cdot \epsilon_1$ with σ_1, where ϵ_1 is the minimum flow from s_i to t_i through an edge in E_i belonging to the k-system σ_1.

Assume now that we have already found ℓ k-systems $\sigma_1, \sigma_2, \ldots, \sigma_\ell$ for some $\ell \geq 1$. If $\sum_{j=1}^{\ell} k \cdot \epsilon_j \geq \frac{1}{2}$ we stop the process of defining σ_j. Otherwise, the minimal $s_i - t_i$-cut in (V, E_i) must still be at least k, because the total flow along any $k-1$ edges in E_i is still less than the total remaining flow from s_i to t_i. Thus, we can apply Menger's theorem again. This allows us to find another k-system $\sigma_{\ell+1}$ between s_i and t_i and in the same way as above we associate with it a weight $\epsilon_{\ell+1}$. Let $\hat{\ell}$ be the number of k-systems at the end of the process.

So far there is no guarantee that any of the k-systems defined above will be small, neither that they will transport enough flow between the terminal pair s_i and t_i. However, after a simple procedure they will satisfy our needs.

According to Fact 2.2, all flow paths in $\mathcal{R}$ have a length of at most $5kF$. Hence, the total amount of edge capacity consumed by a flow system in $\mathcal{R}$ representing a request in $\mathcal{O}'$ is at most $5k^2F$. If there were k-systems in $\sigma_1, \cdots, \sigma_{\hat{\ell}}$ of total weight at least $1/4$ that use more than $20k^3F$ edges each, then they would not fit into the available edge capacity, because $20k^3F \cdot 1/(4k) = 5k^2F$. Thus, there exists a subset of the k-systems $\sigma_1, \cdots, \sigma_{\hat{\ell}}$ with total weight at least $1/4$ such that each of them is small, that is, each of them uses at most $20k^3F$ edges. □

Let $\mathcal{S}_i$ denote the set of small k-systems for request (s_i, t_i) given by Lemma 2.3, and let $\mathcal{S}$ be the set of all $\mathcal{S}_i$. A random experiment will finally help us to bound $|\mathcal{O}'|$ in terms of $|\mathcal{B}|$. Independently for each $(s_i, t_i) \in \mathcal{O}'$, choose exactly one of its k-systems in $\mathcal{S}$, where a k-system σ_j is picked with a probability proportional to its flow value. After the selection, each of the chosen k-systems is used to carry k units of flow, one unit along each of its paths. Let $\mathcal{P}$ denote the chosen k-systems with the k units of flow.

Since each k-system in $\mathcal{P}$ is small, it must have been a candidate for the

k-BGA. But it was rejected by the k-BGA and hence it must have a witness in $\mathcal{B}$. By the definition of $\mathcal{O}'$ this witnessing must be at an edge that is not used by any k-system in $\mathcal{O}'$. Hence, only edges outside of the edges used by $\mathcal{O}'$ can be potential witness edges. From Fact 2.2 we know that each of these edges can have a congestion of at most $1/(2k)$. Hence, after selecting a k-system for each request at random and shipping a demand of 1 along each of its paths, the expected congestion at every potential witness edge is at most 2. Thus, in expectation, every k-system from $\mathcal{B}$ can serve as a witness to at most $2 \cdot L$ k-systems from $\mathcal{P}$. We conclude that there exists a random choice for which the k-systems from $\mathcal{B}$ serve as witnesses to at most $2 \cdot L \cdot |\mathcal{B}|$ k-systems from $\mathcal{P}$ (cf. [13]). Since $|\mathcal{P}| = |\mathcal{O}'|$, the proof is completed. □ □

The above upper bound on the competitive ratio for the k-BGA with parameter $L = 20k^3F$ is the best possible, since a k-system of size $\Theta(k^3F)$ may prevent $\Theta(k^3F)$ other k-systems from being selected. An open question is whether it is possible to achieve a better competitive ratio with a stronger restriction on the size of the k-systems that are used by the k-BGA.

2.2 General online lower bound

Next we show there is a lower bound that holds for the competitive ratio of any deterministic online algorithm for the k-EDP problem which is not far away from the performance of the k-BGA.

Theorem 2.4 *For any n, k, and $F \geq \log_k n$ with $n \geq k^2 \cdot F$ there is a graph G of size $\Theta(n)$ with maximum degree $O(k)$ and flow number $\Theta(F)$ such that the competitive ratio of any deterministic online algorithm on G is $\Omega(k \cdot F)$.*

Proof. A basic building block of our construction is the following simple graph. Let D_k (*diamond*) denote the graph consisting of two bipartite graphs $K_{1,k}$ and $K_{k,1}$ glued naturally together at the larger sides. The two k-degree nodes in D_k are its *endpoints*. Let C (*chaplet*) denote the graph consisting of F diamond graphs attached one to the other at the endpoints, like in an open chaplet.

The core of the graph G consists of $m = n/(k \cdot F) \geq k$ disjoint copies of the chaplet graph C attached to the inputs of a k-ary multibutterfly (Figure 1). In addition, a node s is connected to the first k chaplet graphs and a node t is connected to the first k output nodes of the multibutterfly.

Let $s_{i,j}$ denote the first endpoint of a diamond j in a chaplet i, and let $t_{i,j} (= s_{i,j+1})$ denote the other endpoint. We will use the fact that a k-ary multibutterfly with n' inputs and outputs (which is a network of degree $O(k)$) can route any r-relation from the inputs to the outputs with edge congestion and dilation at most $O(\max[r/k, \log_k n'])$ [16].

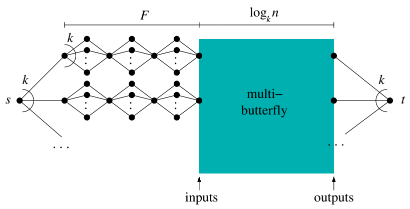

Figure 1: The graph for the lower bound.

First, we show that our graph G has a flow number of $\Theta(F)$. Since the diameter of G is $\Omega(F)$ it is sufficient to prove that a PMFP with $\pi(u) = c(u)/\Gamma$ for the given graph can be solved with congestion and dilation $O(F)$. Consider each node v of degree δ_v to consist of δ_v copies of nodes and let V' be the set of all of these copies. Then the PMFP reduces to the problem of sending a packet of size $1/N$ for any pair of nodes in V', where $N = |V'|$. Such a routing problem can be split into N permutations σ_i with $\sigma_i(v) = (v + i) \bmod N$ for all $i \in \{0, \ldots, N-1\}$ and $v \in V'$. Each such permutation represents a routing problem ρ in the original network where each node is the starting point and endpoint of a number of packets that is equal to its degree. We want to bound the congestion and dilation for routing such a problem.

In order to route ρ, we first move all packets to the inputs of the k-ary multibutterfly in such a way that every input node of the multibutterfly will have $O(kF)$ packets. This can clearly be done with edge congestion $O(F)$ and dilation $O(F)$. Next, we use the multibutterfly to send the packets to the rows of their destinations. Since every input has $O(k \cdot F)$ packets, this can also be done with congestion and dilation $O(F)$. Finally, all packets are sent to their correct destinations. This also causes a congestion and dilation of at most $O(F)$. Hence, routing ρ only requires a total congestion and dilation of $O(F)$.

Combining the fact that all packets are of size $1/N$ with the fact that we have N permutations σ_i, it follows that the congestion and dilation of routing the PMFP in the given graph is $O(F)$. Hence, its flow number is $\Theta(F)$.

Now consider the following two sequences of requests:

(1) (s,t), and

(2) $(s,t),(s_{1,1},t_{1,1}),(s_{1,2},t_{1,2}),\ldots,(s_{1,F},t_{1,F}),(s_{2,1},t_{2,1}),$
$\ldots,(s_{k,F},t_{k,F})$

Obviously, every deterministic online algorithm has to accept (s,t) to ensure a finite competitive ratio for the sequence (1). However, in this case none of the other requests in (2) can be satisfied. But the optimal solution for (2) is to reject (s,t) and to accept all other requests. Hence, the competitive ratio is $\Omega(k \cdot F)$. □

2.3 Managing requests with profits

In the *k edge-disjoint paths with profits problem (k-EDPP)* we are given an undirected graph $G = (V,E)$ and a set of requests T. Each request $r_i = (s_i,t_i)$ has a positive profit $b(r_i)$. The problem is to find a subset S of the pairs in T of maximum profit for which it is possible to select disjoint paths such that each pair is connected by k disjoint paths.

It turns out that a simple offline variant of the k-BGA gives the same approximation ratio for the k-EDPP as we have for the k-EDP. The algorithm involves sorting the requests in decreasing order of their profits and running the k-BGA on this sorted sequence. We call this algorithm the *sorted* k-BGA.

Theorem 2.5 *Given a network G of flow number F, the approximation ratio of the sorted k-BGA with parameter $L = 20k^3F$ is $O(k^3F)$ for the k-EDPP.*

Proof. The proof is almost identical to the proof of Theorem 2.1. The only additional observation is that, since the sorted k-BGA proceeds through the requests from the most profitable, every small k-system in $\mathcal{O} - \mathcal{O}'$ and in the modified set $\mathcal{P}$ has a witness in $\mathcal{B}$ with larger or equal profit. □

2.4 The multi-EDP

Another variant of the k-EDP our techniques can be applied to is the *multiple edge-disjoint paths problem* (multi-EDP) which is defined as follows:

given a graph G and a set of terminal pairs with integral demands d_i, $1 \le d_i \le \Delta$, find a maximum subset of the pairs for which it is possible to select disjoint paths so that every selected pair i has d_i disjoint paths. Let $d_{\max}$ denote the maximal demand over all requests.

A variant of the k-BGA, the *multi-BGA*, can be used here as well: Given a request with demand d_i, reject it if it is not possible to find d_i edge-disjoint paths between the terminal pairs of total length at most $20 d_i d_{\max}^2 F$. Otherwise, select any such d_i paths for it.

Theorem 2.6 *Given a network G of flow number F, the competitive ratio of the multi-BGA is $O(d_{\max}^3 F)$.*

Proof. The proof goes along the same lines as the proof of Theorem 2.1: first, the Shortening lemma with parameter $\epsilon = 1/(2d_{\max})$ is applied and, afterwards, the extraction procedure is used. The difference is that now we extract only d_i-systems for a request with demand d_i, not $d_{\max}$-systems. □

3 Algorithms for the k-DFP

Throughout this section we will assume that the maximal demand is at most k times larger than the minimal edge capacity, which is analogous to assumptions made in almost all papers about the UFP. We call this the *weak bottleneck assumption*. Moreover, we assume that all edge capacities are the same. Since F is invariant to scaling, we simply set all edge capacities to one. The minimal demand of a request will be denoted by $d_{\min}$. We first show how to solve the offline k-DFP, and then mention how to extend this solution to the online case.

To solve the offline k-DFP, we first sort the requests in decreasing order of their demands. On this sorted sequence of requests we use an algorithm that is very similar to the k-BGA: Let L be a suitably chosen parameter. Given a request with a demand of d, accept it if it is possible to find k edge-disjoint paths $p_1, p_2, \ldots p_k$ of flow value d/k between the terminal nodes of the request that fit into the network without violating the capacity constraints and whose total length $\sum_{i=1}^{k} |p_i|$ is at most L. Otherwise, reject it. This extension of the k-BGA will be called *k-flow BGA*.

The next theorem demonstrates that the performance of the k-flow BGA for the k-DFP is comparable to the performance of the k-BGA for the k-EDP. It is slightly worse due to a technical reason: it is much harder to use our technique for extracting short k-systems for the k-DFP than for the k-EDP.

Theorem 3.1 *Given a unit-capacity network G with flow number F, the approximation ratio of the k-flow BGA for the k-DFP with parameter $L = O(k^3 F \log(kF))$, when run on requests sorted in non-increasing order, is $O(k^3 F \log(kF))$.*

Proof. As usual, let $\mathcal{B}$ denote the set of k-systems for the requests accepted by the BGA and $\mathcal{O}$ be the set of k-systems in the optimal solution. Each k-system consists of k disjoint flow paths which we also call *streams*. For notational simplicity we will sometimes think about $\mathcal{B}$ and $\mathcal{O}$ also as a set of streams (instead of k-systems).

For each stream $q \in \mathcal{B}$ or $q \in \mathcal{O}$, let $f(q)$ denote the flow along that stream. If q belongs to the request (s_i, t_i) with demand d_i, then $f(q) = d_i/k$. For a set $\mathcal{Q}$ of streams let $||\mathcal{Q}|| = \sum_{q \in \mathcal{Q}} f(q)$. Also, for an edge $e \in E$ and a stream q, let $F(e,q)$ denote the sum of flow values of all streams in $\mathcal{B}$ passing through e whose flow is at least as large as the flow of q, i.e., $F(e,q) = ||\{p \mid p \in \mathcal{B},\ e \in p,\ f(p) \geq f(q)\}||$. A stream $p \in \mathcal{B}$ is a *witness* for a stream q if $f(p) \geq f(q)$ and p and q intersect in an edge e with $F(e,q) + f(q) > 1$. For each edge e let $\mathcal{W}(e, \mathcal{B})$ denote the set of streams in $\mathcal{B}$ that serve as witnesses on e. Similarly, for each edge e let $\mathcal{V}(e, \mathcal{Q})$ denote the set of streams in $\mathcal{Q}$ that have witnesses on e. We also say that a k-system has a witness on an edge e if any of its k streams has a witness on e. We start with a simple observation.

Claim 3.2 *For any stream q and edge e, if q has a witness on e then $||\mathcal{W}(e, \mathcal{B})|| \geq 1/2$.*

Proof. Let p be a witness of q on e. Assume, by contradiction, that $F(e,q) < 1/2$. It easily follows that $f(p) < 1/2$. Since $f(q) \leq f(p)$ and $F(e,q) + f(q) > 1$ by the definition of a witness, we have a contradiction. □

Let $\mathcal{O}' \subset \mathcal{O}$ be the set of k-systems that are larger than L and that correspond to requests *not* accepted by the k-flow BGA and that do *not* have a witness in $\mathcal{B}$. The next two bounds on $||\mathcal{O} \setminus \mathcal{O}'||$ and $||\mathcal{O}'||$ complete the proof.

Lemma 3.3 $||\mathcal{O} \setminus \mathcal{O}'|| \leq (1 + 2L) \cdot ||\mathcal{B}||$.

Proof. We partition $\mathcal{O} \setminus \mathcal{O}'$ into two sets. Let $\mathcal{O}_1 \subseteq \mathcal{O} \setminus \mathcal{O}'$ consist of all the k-systems corresponding to requests accepted by the BGA and let $\mathcal{O}_2 = (\mathcal{O} \setminus \mathcal{O}') \setminus \mathcal{O}_1$. Obviously, $||\mathcal{O}_1|| \leq ||\mathcal{B}||$. Note that each k-system in $\mathcal{O}_2$ must have a witness in $\mathcal{B}$. Let $E' \subseteq E$ denote the set of all edges on which some k-system from $\mathcal{O}_2$ has a witness. We then have

$$||\mathcal{O}_2|| \leq \sum_{e \in E'} k||\mathcal{V}(e, \mathcal{O}_2)|| \leq \sum_{e \in E'} k \leq \sum_{e \in E'} k \cdot 2||\mathcal{W}(e, \mathcal{B})||$$

For the first inequality note that a k-system of demand d_i in $\mathcal{O}_2$ may only have a witness at a single edge, and this edge can only be traversed by a flow of d_i/k belonging to that k-system. The second inequality holds due to the unit capacities and the last one follows from Claim 3.2.

Since all k-systems in $\mathcal{B}$ are of length at most L, we have

$$\begin{aligned} \sum_{e \in E'} ||\mathcal{W}(e, \mathcal{B})|| &\leq \sum_{\text{streams } p \in \mathcal{B}} |p| \cdot f(p) \\ &\leq \sum_{k-\text{systems } s \in \mathcal{B}} L \cdot d(s)/k \leq L \cdot ||\mathcal{B}||/k \ . \end{aligned}$$

This completes the proof of Lemma 3.3. □

In the next lemma we bound $||\mathcal{O}'||$ by first transforming the large k-systems in $\mathcal{O}'$ into a set $\mathcal{S}$ of small k-systems and then bounding $||\mathcal{S}||$ in terms of $||\mathcal{B}||$.

Lemma 3.4 $||\mathcal{O}'|| = O(L \cdot ||\mathcal{B}||)$.

Proof. In order to prove the lemma, we will transform the k-systems in $\mathcal{O}'$ into a set of k-systems $\mathcal{S}$ in which each k-system has a length at most L and therefore must have a witness in $\mathcal{B}$. To achieve this, we perform a sequence of transformations:

1. First, we scale the demands and edge capacities so that each edge in G has a capacity of $C = \lceil 3k/d_{min} \rceil$ and all requests have demands that are integral multiples of k. More precisely, the demand of each request of original demand d is set to $d' = k \cdot \lceil C \cdot d/k \rceil$. Since $d'/C \in [d, (1 + 1/3)d]$, this slightly increases the demands and therefore also the flows along the streams so that the total flow along an edge is now at most $(1 + 1/3)C$. Note that slightly increasing the demands only increases $||\mathcal{O}'||$ and therefore only makes the bound on the relationship between $||\mathcal{O}'||$ and $||\mathcal{B}||$ more pessimistic.

2. Next, we replace each request (s_i, t_i) in $\mathcal{O}'$ by d_i'/k *elementary* requests of demand k each, shipped along the same k-system as for (s_i, t_i). For every k-system of such a request, we only keep the first $8c \cdot kF$ and the last $8c \cdot kF$ nodes along each of its k streams, for some $c = O(\log(kF))$. The resulting set of (possibly disconnected) streams of a k-system will be called a *k-core*. As shown in Claim 3.5, we can distribute the elementary requests into C/c sets $S_1, \ldots, S_{C/c}$ so that the congestion caused by the k-cores within each set is at most $2c$ at each edge.

3. Afterwards, we consider each S_i separately. We will reconnect disconnected streams in each k-core in S_i with flow systems derived from the flow number. The reconnected k-cores will not yet consists of k disjoint streams. We will show in Claim 3.6 how to extract k-systems of length at most L from each reconnected k-core.

4. Once we have found the short k-systems, we will be able to compare $\|\mathcal{O}'\|$ with $\|\mathcal{B}\|$ with the help of witnesses.

Next we present two vital claims. The proof of the first claim requires the use of the Lovász Local Lemma, and the proof of the second claim is similar to the proof of Theorem 2.1.

Claim 3.5 *The elementary requests can be distributed into C/c sets $S_1, \ldots, S_{C/c}$ for some $c = O(\log(kF))$ so that for each set S_i the edge congestion caused by its k-cores is at most $2c$.*

Proof. We first prove the claim for $c = O(\log(kCF))$ and afterwards demonstrate how to get to $c = O(\log(kF))$.

Consider the random experiment of assigning to each elementary request a number $i \in \{1, \ldots, C/c\}$ uniformly and independently at random, and let S_i be the set of all requests that choose number i. For every edge e let the random variable $X_{e,i}$ denote the number of streams assigned to S_i that traverse e. Since the maximal edge congestion is at most $4C/3$, we have $\mathrm{E}[X_{e,i}] \le 4c/3$ for every edge e. Every edge e can be used by at most one stream of any k-core. Hence, a k-core can contribute a value of at most 1 to $X_{e,i}$ and the contributions of different k-cores are independent. We can use the Chernoff bound to derive

$$\Pr[X_{e,i} \ge (1 + 1/3) \cdot 4c/3] \le \mathrm{e}^{-(1/3)^2 \cdot (4c/3)/3} = \mathrm{e}^{-4c/3^4} .$$

For every edge e and every $i \in \{1, \ldots, C/c\}$ let $A_{e,i}$ be the event that $X_{e,i} > 2c$. Since $(4/3)^2 \le 2$, the above probability estimate bounds the probability that the event $A_{v,i}$ appears. Our aim is to show, with the help of the LLL, that it is possible in the random experiment to assign numbers to the requests so that none of these events appears, which would yield our claim. To apply the LLL we have to bound the dependencies among the events $A_{e,i}$.

Each edge e can be used by at most $2C$ k-cores and these are the only k-cores that affect the values $X_{e,i}$, $i \in \{1, \ldots, C/c\}$. Realizing that each of the k-cores contains at most $2k(8c \cdot kF)$ edges and that the k-cores choose their sets S_i independently at random, we conclude that the event $A_{e,i}$ depends on at most $32ck^2CF$ other events $A_{f,j}$.

To be able to use the LLL, we now olny have to choose the value c so that

$$\mathrm{e} \cdot \mathrm{e}^{-4c/3^4}(32ck^2CF + 1) \leq 1 \ .$$

This can certainly be achieved by setting $c = \Theta(\ln(kCF))$ large enough.

The above procedure is sufficient if $C = (kF)^{O(1)}$. If $C = (kF)^{\Omega(1)}$ a more involved technique will be used. The k-cores will be distributed into the sets S_i not in a single step but in a sequence of refinements (a similar approach was used, e.g., by Leighton et al. [15] and Scheideler [16]). In the first refinement, our aim is to show that for $c_1 = O(\ln^3 C)$ the k-cores can be distributed into the sets $S_1, \ldots, S_{C/c_1}$ so that the edge congestion in each S_i is at most $(1 + O(1/\ln C))4c_1/3$. For this we use the same random experiment as for c above. It follows that $\mathrm{E}[X_{e,i}] = 4c_1/3$ and that

$$\begin{aligned} \Pr[X_{e,i} \geq (1 + 1/\sqrt[3]{c_1}) \cdot 4c_1/3] &\leq \mathrm{e}^{-(1/\sqrt[3]{c_1})^2 \cdot (4c_1/3)/3} \\ &= \mathrm{e}^{-4\sqrt[3]{c_1}/9} \ . \end{aligned}$$

Hence, to be able to use the LLL, we have to choose the value c_1 so that

$$\mathrm{e} \cdot \mathrm{e}^{-4\sqrt[3]{c_1}/9}(32c_1k^2CF + 1) \leq 1 \ .$$

This can certainly be achieved by setting $c_1 = \Theta(\ln^3 C)$ large enough, which completes the first refinement step.

In the second refinement step, each S_i is refined separately. Consider some fixed S_i. Our aim is to show that for $c_2 = O(\ln^3 c_1)$ the k-cores in S_i can be distributed into the sets $S_{i,1}, \ldots, S_{i,c_1/c_2}$ so that the edge congestion in each $S_{i,j}$ is at most $(1 + 1/\sqrt[3]{c_2})(1 + 1/\sqrt[3]{c_1})4c_2/3$. The proof for this follows exactly the same lines as for c_1. Thus, overall C/c_2 sets $S_{i,j}$ are produced in the second step, with the corresponding congestion bound.

In general, in the $(\ell+1)$st refinement step, each set S established in refinement ℓ is refined separately, using $c_{\ell+1} = O(\ln^3 c_\ell)$, until $c_{\ell+1} = O(\ln(kF))$ for the first time. Note that in this case, $c_\ell = \omega(\ln(kF))$ and $c_\ell = (kF)^{O(1)}$. At this point we use the method presented at the beginning of the proof for the parameter c to obtain C/c' sets $S_1, \ldots, S_{C/c'}$ for some $c' = O(\ln(kF))$ with a congestion of at most

$$\left(\prod_{j=1}^{\ell}(1 + 1/\sqrt[3]{c_j})\right) \cdot (4/3)^2 \cdot c'$$

where l is the total number of refinement steps. Using the facts that $1+x \leq \mathrm{e}^x$ for all $x \geq 0$ and that $\mathrm{e}^x \leq 1 + 2x$ for all $0 \leq x \leq 1/2$, it holds for the product that

$$\prod_{j=1}^{\ell}(1 + 1/\sqrt[3]{c_j}) \leq \mathrm{e}^{\sum_{j=0}^{\ell} 1/\sqrt[3]{c_j}} \leq \mathrm{e}^{\epsilon} \leq 1 + 2\epsilon$$

for a constant $0 < \epsilon \leq 1/2$ that can be made arbitrarily small by making sure that c_ℓ is above a certain constant value depending on ϵ. Hence, it is possible to select the values $c_1, \ldots, c_\ell, c'$ so that the congestion in each S_i at the end is at most $2c'$. □

Claim 3.6 *For every set S_i, every elementary request in S_i can be given k-systems of total flow value at least $1/4$ such that each of them consists of at most L edges. Furthermore, the congestion of every edge used by an original k-system in S_i is at most $2c + 1/(2k)$, and the congestion of every other edge is at most $1/(2k)$.*

Proof. For an elementary request r let $p_1^r, \ldots, p_{\ell_r}^r$ be all the disconnected streams in its k-core, $1 \leq \ell_r \leq k$. Let the first $8c \cdot kF$ nodes in p_i^r be denoted by $a_{i,1}^r, \ldots, a_{i,8c\cdot kF}^r$ and the last$8c \cdot kF$ nodes in p_i^r be denoted by $b_{i,1}^r, \ldots, b_{i,8c\cdot kF}^r$. Consider the set of pairs

$$\mathcal{L} = \bigcup_{r\in S_1} \bigcup_{i=1}^{\ell_r} \bigcup_{j=1}^{8c\cdot kF} \{(a_{i,j}^r, b_{i,j}^r)\}\ .$$

Due to the congestion bound in Claim 3.5, a node v of degree δ can be a starting point or endpoint of at most $2c\delta$ pairs in $\mathcal{L}$. From Lemma 1.2

we know that for any network G with flow number F and any instance I of the BMFP on G there is a feasible solution for I with congestion and dilation at most $2F$. Hence, it is possible to connect all of the pairs in $\mathcal{L}$ by flow systems of length at most $2F$ and flow value $f(p_i^r)$ so that the edge congestion is at most $2c \cdot 2F$. Let the flow system between $a_{i,j}^r$ and $b_{i,j}^r$ be denoted by $f_{i,j}^r$. For each elementary request $r = (s,t)$ and each $1 \leq i \leq \ell_r$ and each $1 \leq j \leq 8c \cdot kF$, we define a flow system $g_{i,j}^r$: first, it moves from s to $a_{i,j}^r$ along p_i^r, then from $a_{i,j}^r$ to $b_{i,j}^r$ along $f_{i,j}^r$, and finally from $b_{i,j}^r$ to t along p_i^r, and we assign to it a flow value of $f(p_i^r)/(8c \cdot kF)$. This ensures that a total flow of $f(p_i^r)$ is still being shipped for each p_i^r. Furthermore, this allows us to reduce the flow along $f_{i,j}^r$ by a factor of $1/(8c \cdot kF)$. Hence, the edge congestion caused by the $f_{i,j}^r$ for all r, i, j reduces to at most $4c \cdot F/(8c \cdot kF) = 1/(2k)$. Therefore, the additional congestion at any edge is at most $1/(2k)$, which proves the congestion bounds in the claim.

Now consider any given elementary request $r = (s,t)$. For any set of $k-1$ edges, the congestion caused by the flow systems for r is at most $(k-1)(1+1/(2k)) \leq k - 1/2$. Hence, according to Menger's theorem there are k edge-disjoint flows in the system from s to t. Continuing with the same arguments as in Theorem 2.1, we obtain a set of k-systems for r with as properties stated in the claim. □

Now that we have short k-systems for every elementary request, we combine them back into the original requests. For a request with demand d this results in a set of k-systems of size at most L each and total flow value at least $d/(4k)$. Let the set of all these k-systems for all requests be denoted by $\mathcal{S}$. Since every k-system has a size at most L, it could have been a candidate for the BGA. Thus, each of these k-systems must have a witness. Crucially, every edge that has witnesses for these k-systems must be an edge that is not used by *any* of the original k-systems in $\mathcal{O}'$. (This follows directly from the definition of $\mathcal{O}'$.) According to the proof of Claim 3.6, the amount of flow from $\mathcal{S}$ traversing any of these edges is at most $1/(2k)$. Let E' be the set of all witness edges.

For each request we now choose independently at random one of its k-systems, with probability proportional to the flow values of the k-systems. This will result in a set of k-systems $\mathcal{P}$ in which each request has exactly one k-system and in which the expected amount of flow traversing any edge in E' is at most $1/(2k)$. Next, we assign the original demand of the request to each of these k-systems. This causes the expected amount of flow that traverses any edge in E' to increase from at most $1/(2k)$ to at most $4k \cdot 1/(2k) = 2$.

We are now ready to bound $\|\mathcal{P}\|$ in terms of $\|\mathcal{B}\|$. For every k-system $h \in \mathcal{S}$, let the indicator variable X_h take the value 1 if and only if h is chosen to be in $\mathcal{P}$. We shall look upon $\|\mathcal{P}\|$ as a random variable (though it always has the same value) and bound its value by bounding its expected value $E[\|\mathcal{P}\|]$. In the following we assume that $f(h)$ is the flow along a stream of the k-system h and $d(h)$ is the demand of the request corresponding to h. Also, recall that the total flow value of k-systems in $\mathcal{S}$ belonging to a request with demand d is at least $d/(4k)$.

$$\begin{aligned}
\mathrm{E}[\|\mathcal{P}\|] &\leq \mathrm{E}\left[\sum_{e \in E'} k \cdot \|\mathcal{V}(e,\mathcal{P})\|\right] \\
&\leq \sum_{e \in E'} k \cdot \mathrm{E}\left[\sum_{p \in \mathcal{S}:\, e \in p} X_p \cdot \frac{d(p)}{k}\right] \\
&\leq \sum_{e \in E'} k \cdot \sum_{p \in \mathcal{S}:\, e \in p} \frac{k \cdot f(p)}{d(p)/(4k)} \cdot \frac{d(p)}{k} \\
&\leq \sum_{e \in E'} k \cdot 4k \sum_{p \in \mathcal{S}:\, e \in p} f(p) \\
&\leq \sum_{e \in E'} 4k^2 \cdot \frac{1}{2k} \leq \sum_{e \in E'} 2k \\
&\leq 4k \sum_{e \in E'} \|\mathcal{W}(e,\mathcal{B})\| \leq \ldots \leq 4L \cdot \|\mathcal{B}\|
\end{aligned}$$

where the last calculations are done in the same way as in the proof of Lemma 3.3. □

Combining the two lemmas proves the theorem. □

We note that if the minimum demand of a request, $d_{\min}$, fulfills $d_{\min} \geq k/\log(kF)$, then one would not need Claim 3.5. In particular, if $d_{\min}$ were known in advance, then the k-flow BGA could choose $L = O(k^3F/(d_{\min}/k))$ to achieve an approximation ratio of $O(k^3F/(d_{\min}/k))$. This would allow a smooth transition from the bounds for the k-EDP (where $d_{\min} = k$) to the k-DFP.

3.1 Online algorithms for the k-DFP

In this section we present a randomized online algorithm for the k-DFP. This algorithm, which we shall call the *randomized* k-flow BGA, is an extension of the k-flow BGA algorithm for the offline k-DFP. The technique we present for making offline algorithms online has been used before [2, 14].

Consider, first, the set $\mathcal{O}$ of k-systems for requests accepted by the optimal algorithm. Let $\mathcal{O}_1 \subseteq \mathcal{O}$ consist of k-systems each with demand at least $k/2$, and let $\mathcal{O}_2 = \mathcal{O} \setminus \mathcal{O}_1$. Exactly one of the following events is true: (1) $||\mathcal{O}_1|| \geq 1/2 \cdot ||\mathcal{O}||$ or (2) $||\mathcal{O}_2|| > 1/2 \cdot ||\mathcal{O}||$.

The randomized k-flow BGA begins by *guessing* which of the two events above will happen. If it guesses the former, it ignores all requests with demand less than $k/2$ and runs the regular k-flow BGA on the rest of the requests. If it guesses the latter, it ignores all requests with demand at least $k/2$ and runs the k-flow BGA on the rest.

Theorem 3.7 *Given a unit-capacity network G with flow number F, the expected competitive ratio of the randomized k-flow BGA for the online k-DFP is $O(k^3 F \log(kF))$ when run with parameter $L = O(k^3 F \log(kF))$.*

Proof. The proof runs along exactly the same lines as the proof for Theorem 3.1, but we have to prove Lemma 3.2 for the changed situation. Note that the original proof for Lemma 3.2 relies on the fact that requests are sorted in a non-decreasing order before being considered. That need not be true here. Let $\mathcal{B}$ denote, as usual, the k-systems for requests accepted by the randomized k-flow BGA.

Consider the case when the algorithm guesses that $||\mathcal{O}_1|| \geq 1/2 \cdot ||\mathcal{O}||$. We claim that for any stream $q \in \mathcal{O}_1$ and edge e, if q has a witness on e then $\|\mathcal{W}(e, \mathcal{B})\| \geq 1/2$. Let q be witnessed by p, a stream in $\mathcal{B}$. Now, since the algorithm only considers requests with demand at least $k/2$, $f(p) \geq 1/2$. The claim follows since $\|\mathcal{W}(e, \mathcal{B})\| \geq f(p)$. Following the rest of the proof for Theorem 3.1, substituting $\mathcal{O}_1$ for $\mathcal{O}$, shows that in this case the randomized k-flow BGA will have a competitive ratio of $O(k^3 F \log(kF))$.

Now consider the case when the algorithm guesses $||\mathcal{O}_2|| \geq 1/2 \cdot ||\mathcal{O}||$. We claim that even in this case for any stream $q \in \mathcal{O}_2$ and edge e, if q has a witness on e then $\|\mathcal{W}(e, \mathcal{B})\| \geq 1/2$. From the definition of witnessing, we have $F(e, q) + f(q) > 1$. Next, from the definition of $\mathcal{O}_2$, $f(q) < 1/2$. The claim follows as $\|\mathcal{W}(e, \mathcal{B})\| \geq F(e, q)$. As in the previous case, the rest of the proof for Theorem 3.1 applies here too; substitute $\mathcal{O}_2$ for $\mathcal{O}$.

The competitive ratio in both cases is $O(k^3 F \log(kF))$. Note that the algorithm may guess incorrectly which event shall be true. But that just reduces the expected competitive ratio by a factor of 2. $\square$

3.2 Comparison with other flow problems

In this section we demonstrate that the k-DFP may be harder to approximate than other related flow problems because of the requirement that the k paths for every request must be disjoint.

The k-splittable flow problem and the integral splittable flow problem have been defined in the introduction. As already mentioned there, previous proof techniques [14] imply the following result under the no-bottleneck assumption (i.e., the maximal demand is at most equal to the minimal edge capacity).

Theorem 3.8 *For a uniform-capacity network G with flow number F, the approximation ratio of the 1-BGA with parameter $L = 4F$ for the k-SFP and for the ISF, when run on requests ordered according to their demands starting from the largest, is $O(F)$.*

Proof. The crucial point is that in the analysis of the BGA algorithm for the UFP problem in the previous work [14] the solution of the BGA is compared with an optimal solution of a relaxed problem, namely the *fractional* maximum multicommodity flow problem, and this problem is also a relaxation for both the ISF and the k-SFP. It follows that the approximation guarantee $O(F)$ of the BGA proved for the UFP problem holds for the k-SFP and the ISF problems as well. $\square$

Using the standard techniques mentioned earlier, the algorithm can be converted into a randomized online algorithm with the same expected competitive ratio. If there is a guarantee that the ratio between the maximal and the minimal demand is at most 2 (or some other constant) or that the maximal demand is at most 1/2 (or some other constant smaller than 1, the edge capacity), the online algorithm can be made even deterministic with the same competitive ratio (cf. [13]). Taking into account the online lower bound of Theorem 2.4, this indicates that the k-SFP and the ISF are indeed simpler problems than the k-DFP.

The techniques of the current paper imply results for the ISF even when the no-bottleneck assumption does not hold and only the weak bottleneck assumption is guaranteed (i.e., the maximal demand is at most k times larger than the minimal edge capacity). In this case we use a variant of the k-flow BGA, *vary-BGA*, that looks for d_i disjoint paths of total length $O(d_i d_{\max}^2 F \log(kF))$ for a request with demand d_i. A consequence of the Theorem 3.1 is the following corollary:

Corollary 3.9 *Given a uniform-capacity network G with flow number F, the approximation ratio of the vary-BGA for the ISF under the weak bottleneck assumption, when run on requests ordered according to their demands starting from the largest, is $O(d_{\max}^3 F \log(kF))$.*

In this case the algorithm can also be converted into a randomized online algorithm.

4 Conclusions

In this paper we presented upper and lower bounds for the k-EDP and the k-DFP and related problems. Many problems remain open. For example, what is the best competitive ratio a deterministic algorithm can achieve for the k-EDP? We suspect that it is $O(k \cdot F)$, but it seems very hard to prove. Concerning the k-DFP, is it possible to simplify the proof and improve the upper bound? We suspect that it should be possible to prove an $O(k \cdot F)$ upper bound as well. Even an improvement of the $O(k^3 F \log(kF))$ bound k-DFP to $O(k^3 F)$ would be interesting.

A bunch of other problems arises for networks with nonuniform edge capacities: the k-flow BGA algorithm can be used on them as well but is it possible to prove the same performance bounds?

5 Acknowledgements

We would like to thank Rakesh Sinha for bringing these problems to our attention and Alan Frieze for helpful insights.

References

[1] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive online routing. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 32–40, 1993.

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

[3] G. Baier, E. Koehler, and M. Skutella. On the k-splittable flow problem. In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA)*, 2002. To appear.

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

[5] A. Bley. On the complexity of vertex-disjoint length-restricted path problems. Technical Report SC-98-20, Konrad-Zuse-Institut Berlin, 1998.

[6] B. Bollobás. *Modern Graph Theory.* New York: Springer, 1998.

[7] A. Chakrabarti, C. Chekuri, A. Gupta, and A. Kumar. Approximation algorithms for the unsplittable flow problem. Unpublished manuscript, 2001.

[8] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization.* John Wiley, New York, 1997.

[9] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal, R. Rado, and V. Sós, editors, *Infinite and Finite Sets (to Paul Erdős on his 60th birthday)*, pages 609–627. North-Holland, Amsterdam, 1975.

[10] 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 *Proceedings of the 31st ACM Symposium on Theory of Computing (STOC)*, pages 19–28, 1999.

[11] W. Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, 58:13–30, 1963.

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

[13] P. Kolman and C. Scheideler. Simple, routing-based on-line algorithms for maximum disjoint paths problem. In *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 38–47, 2001.

[14] P. Kolman and C. Scheideler. Improved bounds for the unsplittable flow problem. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2002.

[15] 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.

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