

Flows with Unit Path Capacities and Related Packing and Covering Problems*

Maren Martens¹ and Martin Skutella²

¹ University of British Columbia, Sauder School of Business,
2053 Main Mall, Vancouver, BC V6T 1Z2, Canada, maren.martens@sauder.ubc.ca

² TU Berlin, Institut für Mathematik, MA 5–2, Str. des 17. Juni 136, 10623 Berlin, Germany,
skutella@math.tu-berlin.de

Abstract. Since the seminal work of Ford and Fulkerson in the 1950s, network flow theory is one of the most important and most active areas of research in combinatorial optimization. Coming from the classical maximum flow problem, we introduce and study an apparently basic but new flow problem that features a couple of interesting peculiarities. We derive several results on the complexity and approximability of the new problem. On the way we also discover two closely related basic covering and packing problems that are of independent interest. Starting from an LP formulation of the maximum s - t -flow problem in path variables, we introduce unit upper bounds on the amount of flow being sent along each path. The resulting (fractional) flow problem is NP-hard; its integral version is strongly NP-hard already on very simple classes of graphs. For the fractional problem we present an FPTAS that is based on solving the k shortest paths problem iteratively. We show that the integral problem is hard to approximate and give an interesting $O(\log m)$ -approximation algorithm, where m is the number of arcs in the considered graph. For the multicommodity version of the problem there is an $O(\sqrt{m})$ -approximation algorithm. We argue that this performance guarantee is best possible, unless P=NP.

1 Introduction

Problem Definition and Notation. The classical maximum s - t -flow problem has been studied from many different points of view. Numerous algorithms are known to solve the problem in polynomial time. Ford and Fulkerson [2] proved already in the 1950s that there always exists an integral optimal solution to the maximum s - t -flow problem provided that all arc capacities are integral. It is also well known that any s - t -flow can be decomposed into flow along paths and cycles. Omitting flow along cycles (which does not contribute to the flow value) yields an alternative LP formulation of the problem in path variables. In this paper we study a new network flow problem in which the flow on any path is bounded by 1, i.e., we add box constraints to the LP formulation of the maximum flow problem in path variables. We call the resulting problem the *maximum one-flow problem*. Our motivation for studying it is mainly academic,

* This work was partially supported by the DFG Research Center Matheon in Berlin, by the Graduate School of Production Engineering and Logistics, North Rhine-Westphalia, by the DFG Focus Program 1126 and the DFG grants SK 58/4-1 and SK 58/5-3, and by an NSERC Operating Grant. Part of this work was done while the authors were at Universität Dortmund.

but the problem is also well motivated when we think of applications in transportation/communication networks where every single path might be unreliable. In such situations it is reasonable to diversify a commodity/information among several different paths. This can be accomplished by forbidding to send more than a fixed amount of flow along a single path. A more formal definition of the problem is as follows: We are given a network (digraph) $D = (V, A)$ with arc capacities $u : A \rightarrow \mathbb{R}^+$ and two distinguished nodes $s, t \in V$. We assume that $u(a) \geq 1$, for all $a \in A$. If not stated otherwise, $m := |A|$ denotes the number of arcs in the network. Let \mathcal{P} be the set of simple directed s - t -paths in D . Then the *maximum one-flow problem (max-1FP)* and its dual can be formulated as follows, where the path variable x_P denotes the amount of flow sent along path $P \in \mathcal{P}$:

$$\begin{aligned}
\max \sum_{P \in \mathcal{P}} x_P & & \min \sum_{a \in A} u(a)y_a + \sum_{P \in \mathcal{P}} z_P \\
\text{s.t. } \sum_{P \ni a} x_P \leq u(a) \quad \forall a \in A & \quad (1) & \text{s.t. } z_P + \sum_{a \in P} y_a \geq 1 \quad \forall P \in \mathcal{P} \\
0 \leq x_P \leq 1 \quad \forall P \in \mathcal{P} & \quad (2) & z_P, y_a \geq 0 \quad \forall a \in A, P \in \mathcal{P}
\end{aligned}$$

Notice that omitting the constraints $x_P \leq 1$ yields the classical maximum s - t -flow problem. An s - t -flow fulfilling (1) and (2) is called a *one-flow*. In an *integral one-flow* each s - t -path sends either 0 or 1 unit of flow. To emphasize that a certain one-flow is not necessarily integral, we sometimes call it *fractional*. Note that in general the encoding size of a maximum one-flow is not polynomial in the input size of the problem since one might want to send flow along exponentially many s - t -paths. Therefore, the best one can expect in terms of complexity are algorithms with running time polynomially bounded in the input plus output size.

The integer version of the dual can be interpreted as a special minimum cut problem, where each s - t -path must be destroyed and this can be done by deleting either a single arc on the path or the path itself. The deletion of an arc a is in general more expensive than that of a whole path ($u(a)$ instead of 1), but can also destroy more than one path simultaneously. The dual separation problem of the classical maximum s - t -flow problem is a shortest path problem. It is not difficult to observe that the dual separation problem of the max-1FP can be solved by computing the k shortest s - t -paths with respect to the dual arc lengths y_a , where k is the number of paths $P \in \mathcal{P}$ with $z_P > 0$ plus 1.

Related Results from the Literature. To the best of our knowledge, the one-flow problem (1FP) is studied for the first time here. However, there is some literature dealing with problems related to it. The problem to compute the number of different (simple) s - t -paths in a network is a special case of the max-1FP. (Consider the case when all arc capacities are infinite.) Valiant [10] shows that this problem is #P-complete under polynomial-time reductions. Similar to the problem of counting paths in a graph is the edge-disjoint paths problem (EDP) that is the same as the integral multicommodity 1FP when we fix all capacities in the considered network to 1. Overviews on the EDP can, e.g., be found in [3–5]. For more general packing problems also many results have been obtained. One that is of particular interest for the present paper is obtained by Plotkin, Shmoys, and Tardos [8] and will be introduced later on. In [7] the authors have already

considered a certain network flow problem with path capacities, where the flow of any commodity is restricted to at most k paths whose flow values may not exceed given bounds (capacities). The general k shortest paths problem (for a single source and a single sink) is of great interest for the problem considered in this paper. To solve the Lagrange relaxation of the max-1FP that is obtained by penalizing the violation of arc capacities in the objective function, we make use of a result by Lawler [6] who shows how to compute k shortest (simple) paths in a digraph in $O(kn^3)$ time.

Contribution of this Paper. As mentioned above, the problem of computing the number of different simple s - t -paths in a network is #P-complete and a special case of the max-1FP. It therefore follows immediately that computing the maximum one-flow value is NP-hard. This holds for the fractional as well as for the integral 1FP. We prove that the integral max-1FP is strongly NP-hard already on very simple acyclic networks consisting of a chain of parallel arcs. Even worse, the integral max-1FP is APX-hard, even in networks where the number of s - t -paths is polynomially bounded in the size of the network. One interesting consequence of these hardness results is the following: It is NP-hard to decide whether a given integral s - t -flow has an integral path decomposition such that each path carries at most one unit of flow. In Section 2 we establish a close relation between two interesting new combinatorial problems and the special case of the max-1FP on networks consisting of a chain of parallel arcs. The first problem is to cover the edges of a complete graph by cuts of bounded size where the size of a cut is the cardinality of the smaller of the two vertex subsets. The second problem is a packing problem: Consider a set where each element has a given integral weight and find a pre-specified number of different subsets such that the number of subsets containing an element is bounded by the element's weight. The two problems are equivalent and, maybe surprisingly, strongly NP-hard. This also yields the strong NP-hardness of the integral max-1FP on chains of parallel arcs. Moreover, we show that already on a chain of parallel arcs of length 3 the max-1FP has an integrality gap. We also prove that it might happen that each arc in a network carries an integral amount of flow in a maximum one-flow but no maximum one-flow is integral. In Section 3 we show that the approach of Plotkin et al. [8] yields an FPTAS for the fractional max-1FP. The core of the algorithm consists of iteratively solving k shortest paths problems on the given network with varying arc lengths. In Section 4 we derive several approximation algorithms for the integral max-1FP. Our main result is a randomized approximation algorithm with performance ratio $O(\log m)$. Finally, in Section 5 we study multicommodity versions of the 1FP. We show that the FPTAS from Section 3 can be generalized to the fractional multicommodity 1FP. For the integral maximum multicommodity 1FP we present a randomized $O(\sqrt{m})$ -approximation algorithm and show that, unless P=NP, no better approximation is possible. Moreover, we present an $O(\log m)$ -approximation algorithm for the problem to find an integral multicommodity one-flow with minimum congestion. This extended abstract leaves out most proofs.

2 Interesting related problems

In this section we study the max-1FP on a restricted class of networks that are given by chains of parallel arcs. In order to obtain a better understanding of the max-1FP on

this particular class of networks we consider two equivalent combinatorial optimization problems, one of which is a covering and the other a packing problem. Although these two problems are easy to formulate and seem quite natural, they have not appeared in the literature before to the best of our knowledge.

We consider networks that consist of $n + 1$ vertices v_0, v_1, \dots, v_n and $2n$ arcs ($n \in \mathbb{N}$) such that there are two parallel arcs from v_{i-1} to v_i , for $i = 1, \dots, n$. Vertex v_0 is the source and v_n is the sink. We call one arc of each pair of parallel arcs the *upper* and the other one the *lower arc*. All lower arcs have infinite capacity. The capacity of the i th upper arc is $c_i \in \mathbb{N}$, for $i = 1, \dots, n$. We call such a capacitated network a *chain of parallel arcs*. An integral one-flow is given by a set of s - t -paths that are pairwise distinct. Notice that two s - t -paths in the considered network are different if and only if there is a pair of arcs where one path uses the upper arc and the other path uses the lower arc. In particular, the i th pair of arcs can distinguish a subset of at most c_i paths from all other paths. This motivates the following problem.

Bounded Cut Cover Problem

GIVEN: k numbers $c_1, \dots, c_k \in \mathbb{N}$ and a number $q \in \mathbb{N}$.

TASK: Find k subsets $M_1, \dots, M_k \subseteq \{1, \dots, q\}$ with $|M_i| \leq c_i$, for $i = 1, \dots, k$, such that for any pair $j, \ell \in \{1, \dots, q\}$ with $j \neq \ell$ there is some $i \in \{1, \dots, k\}$ with $|M_i \cap \{j, \ell\}| = 1$; or decide that no such family of subsets exists.

The name that we choose for this problem stems from the following graph-theoretic interpretation: Consider a complete undirected graph with vertex set $\{1, \dots, q\}$. The question is whether the edges of the complete graph can be covered by k cuts where, for $i = 1, \dots, k$, the i th cut partitions the vertex set into two subsets the smaller of which has cardinality at most c_i .

Lemma 1. *The bounded cut cover problem has a solution if and only if there exists an integral one-flow of value q in a chain of parallel arcs of length k where the capacities of the upper arcs are c_1, \dots, c_k .*

Fractional Bounded Cut Cover Problem

GIVEN: k numbers $c_1, \dots, c_k \in \mathbb{N}$ and a number $q \in \mathbb{R}^+$.

TASK: For some $r \geq q$, find weights $x_1, \dots, x_r \in [0, 1]$ with $\sum_{j=1}^r x_j = q$ and determine k subsets $M_1, \dots, M_k \subseteq \{1, \dots, r\}$ with $\sum_{j \in M_i} x_j \leq c_i$, for $i = 1, \dots, k$, such that for any pair $j, \ell \in \{1, \dots, r\}$ with $j \neq \ell$ there is some $i \in \{1, \dots, k\}$ with $|M_i \cap \{j, \ell\}| = 1$; or decide that this is not possible.

There is again a graph-theoretic interpretation of the problem. The task is to find a complete graph with weights on the vertices such that the weight x_i of every vertex i is between 0 and 1 and the weights sum up to q . Moreover, the edges of the complete graph must be covered by k cuts such that the i th cut partitions the vertex set into two subsets the lighter of which has total weight at most c_i . Associating the weighted nodes of the complete graph with s - t -paths of corresponding flow value yields the following observation.

Lemma 2. *The fractional bounded cut cover problem has a solution if and only if there exists a (fractional) one-flow of value q in a chain of parallel arcs of length k where the capacities of the upper arcs are c_1, \dots, c_k .*

It is natural to ask whether the fractional bounded cut cover problem allows for larger values of q with feasible solutions than the non-fractional version. By Lemmas 1 and 2, this is equivalent to the question whether for chains of parallel arcs there always exists a maximum one-flow that is integral. In the fractional bounded cut cover problem the price of the additional degree of freedom given by the possibility to assign fractional weights to the nodes is an increase in the number of nodes (since the node weights still have to sum up to q). On the one hand, a larger number of nodes makes it more difficult to cover all edges of the complete graph. On the other hand, fractional weights on the vertices allow for more balanced cuts that contain more edges. We show below that there exist instances with a larger feasible value of q in the fractional version of the problem than in the integral version. Before we discuss this issue in more detail, we present another equivalent packing problem.

In a chain of parallel arcs, every s - t -path is uniquely determined by the subset of upper arcs contained in the path. Therefore, computing an integral one-flow of value q corresponds to finding a family of q pairwise distinct subsets of $\{1, \dots, k\}$ such that $i \in \{1, \dots, k\}$ is contained in at most c_i of these subsets.

Capacitated Set Packing Problem

GIVEN: k numbers $c_1, \dots, c_k \in \mathbb{N}$ and a number $q \in \mathbb{N}$.

TASK: Find q pairwise distinct subsets of $\{1, \dots, k\}$ such that element $i \in \{1, \dots, k\}$ is contained in at most c_i of these subsets, for $i = 1, \dots, k$; or decide that no such family of subsets exists.

Fractional Capacitated Set Packing Problem

GIVEN: k numbers $c_1, \dots, c_k \in \mathbb{N}$ and a number $q \in \mathbb{R}^+$.

TASK: For some $r \geq q$, find pairwise distinct subsets N_1, \dots, N_r of $\{1, \dots, k\}$ with weights $x_1, \dots, x_r \in [0, 1]$ such that $\sum_{j=1}^r x_j = q$ and $\sum_{j:i \in N_j} x_j \leq c_i$, for $i = 1, \dots, k$; or decide that this is not possible.

Lemma 3. *The (fractional) capacitated set packing problem has a solution if and only if there exists an integral (fractional) one-flow of value q in a chain of parallel arcs of length k where the capacities of the upper arcs are c_1, \dots, c_k . In particular, the (fractional) capacitated set packing problem is equivalent to the (fractional) bounded cut cover problem.*

The following instance shows that the fractional capacitated set packing problem in general allows for strictly larger values of q with feasible solutions than the non-fractional version. Due to Lemma 3, the same holds for the bounded cut cover problem. Let $k = 3$ and $c_1 = c_2 = c_3 = 2$. It is not difficult to check that $q = 5$ is the largest value of q with a feasible solution to the non-fractional version of the problem: Choose for example the subsets \emptyset , $\{1\}$, $\{2\}$, $\{3\}$, and $\{1, 2, 3\}$. But there is a solution to the fractional version of the problem with $q = 5.5$: Choose subsets \emptyset , $\{1\}$, $\{2\}$, and $\{3\}$ all with weight 1. In addition choose subsets $\{1, 2\}$, $\{1, 3\}$, and $\{2, 3\}$ all with weight $1/2$.

We can also show that in general there is no integral optimal solution to the dual problem of the max-1FP. Consider the chain of parallel arcs corresponding to the instance of the capacitated set packing problem introduced above. An optimal dual solution destroys the path using all lower arcs and half of each path that uses exactly one upper arc; further, one half of each upper arc is deleted. The following result underlines the discrepancy between the fractional and the integral one-flow problem even more.

Proposition 1. *The existence of a maximum one-flow where the flow value on each arc is integral does in general not imply the existence of an integral path decomposition where each path carries at most one unit of flow.*

In contrast to the classical set packing problem and many similar problems known from the literature, the capacitated set packing problem allows to choose arbitrary subsets that do not have to belong to a given family of subsets. This might make the problem seem to be easier. However, we can prove the following somewhat surprising theorem.

Theorem 1. *The capacitated set packing problem is strongly NP-hard. If q is polynomially bounded in k , the problem is strongly NP-complete.*

One can reduce the strongly NP-hard 3-PARTITION problem to the capacitated set packing problem. Let A with $|A| = 3\ell =: k$ be the ground set of a given 3-PARTITION instance. The weights of elements in A sum up to ℓB for some $B \in \mathbb{N}$. The main idea of the reduction is to identify A with the set $\{1, \dots, k\}$ and define c_1, \dots, c_k and q such that all subsets of A with total weight less than B and the subsets of a 3-Partition of A must be chosen to end up with the desired q subsets of A . The following is an immediate implication of this reduction.

Corollary 1. *For an integral one-flow given in arc variables, it is NP-hard to compute an integral path decomposition.*

A reduction to 3-SAT proves that the situation is even worse: The integral max-1FP is APX-hard, even in networks where the number of s - t -paths is polynomially bounded in the size of the network. As an immediate consequence of Theorem 1 we can state the following hardness results.

Theorem 2. *(i) The bounded cut cover problem is strongly NP-hard. If q is polynomially bounded in k , then the problem is strongly NP-complete. (ii) The problem of finding an integral one-flow of maximum value for a chain of parallel arcs is strongly NP-hard, even if the maximum flow value is polynomially bounded in the size of the network (i.e., number of vertices).*

It follows that, in contrast to the problem to count the number of s - t -paths in a digraph, the integral max-1FP is already strongly NP-hard in acyclic networks. Further, the strong NP-hardness of the capacitated set packing problem immediately implies that it is even strongly NP-hard to compute only the value of a maximum integral one-flow in those networks. Note that in the integral 1FP the flow value bounds the number of paths that are used to route a flow. Thus, we can derive the following from Theorem 2 (ii).

Corollary 2. *Unless $P = NP$, there is no algorithm for the integral max-1FP on chains of parallel arcs whose runtime is pseudo-polynomial in input plus output size.*

Theorem 3. *It is NP-hard to decide whether a given (integral) s - t -flow can be decomposed into integral flows along paths and cycles such that no path carries more than one unit of flow.*

3 An FPTAS for the Fractional Max-1FP

Theorem 4. *For any $\epsilon > 0$ and any instance of the max-1FP with maximum flow value F^* , it is possible to compute a maximum one-flow of value $(1 - \epsilon)F^*$ in time polynomial in the input size, ϵ^{-1} , and F^* .*

First we show how to compute a one-flow of a given value F that does not violate arc capacities by more than a factor $(1 + \epsilon)$, for some $\epsilon > 0$, or decide that no valid flow of value F exists. A flow violating all arc capacities by at most a factor $(1 + \epsilon)$ is called $(1 + \epsilon)$ -approximate. Plotkin, Shmoys, and Tardos [8] developed an appropriate algorithm for the general fractional set packing problem. In that problem sets of capacitated elements are given and the task is to search for a packing, i.e., a selection of sets such that each element is contained in at most as many sets as its capacity permits. To compute $(1 + \epsilon)$ -approximate packings of a given size, Plotkin et al. use a Lagrange relaxation that penalizes the violation of the capacity constraints in the objective function. Iteratively, they choose reasonable Lagrange multipliers, compute a solution to the relaxed problem, and combine this with the current solution. This algorithm runs in time polynomial in the input size and ϵ^{-1} . It can be adapted to the fractional 1FP. For given Lagrange multipliers $\lambda : A \rightarrow \mathbb{R}^+$ and $\lambda_P := \sum_{a \in P} \lambda(a)$ for all s - t -paths $P \in \mathcal{P}$, the Lagrange relaxation is $\min\{\sum_{P \in \mathcal{P}} \lambda_P x_P \mid \sum_{P \in \mathcal{P}} x_P \geq F, 0 \leq x_P \leq 1 \forall P \in \mathcal{P}\}$. It can be solved in time polynomial in the input size and $\lceil F \rceil$ by computing the $\lceil F \rceil$ shortest paths according to the length function λ (see, e.g., [6]). The $\lceil F \rceil$ shortest paths must carry one unit of flow ($x_P = 1$) and the $(\lfloor F \rfloor + 1)$ -shortest path gets a flow value of $F - \lfloor F \rfloor$. To obtain a one-flow that obeys all arc capacities and approximates the maximum flow value F^* within a factor $(1 - \epsilon)$, we embed the algorithm by Plotkin et al. in a binary search. That binary search can be implemented to run in time polynomial in the input size, ϵ^{-1} , and F^* .

4 Approximating the Integral Max-1FP

We assume that all arc capacities are integral and start with the observation that, for the integral max-1FP, the additive integrality gap is at most m . This result follows from basic linear programming theory. With the FPTAS from the previous section we obtain Corollary 3 and Theorem 5 as immediate consequences.

Proposition 2. *The difference of the value F_F^* of a maximum fractional one-flow and the value F_I^* of a maximum integral one-flow is less than m .*

Corollary 3. For any $\epsilon > 0$, an integral one-flow of value at least $(1 - \epsilon)F_F^* - m$ can be computed in time polynomial in the input size and F_I^* .

Theorem 5. There exists a constant factor approximation algorithm for the integral max-IFP whose runtime is polynomial in input plus output size if we restrict to instances whose maximum fractional flow value is larger than some constant $c > 1$ times its number of arcs.

Subsequently we develop a randomized $O(\log m)$ -approximation algorithm for the integral max-IFP that works for arbitrary instances. We start with a simple observation.

Lemma 4. Applying Raghavan and Thompson's [9] randomized rounding method to a fractional solution computed by the FPTAS from Section 3, we obtain a constant factor approximation algorithm for the integral max-IFP if the minimum arc capacity is at least $\Omega(\log m)$.

In order to obtain a randomized $O(\log m)$ -approximation algorithm for arbitrary instances of the integral max-IFP, we compute an approximate maximum integral one-flow in a modified network by giving a special treatment to arcs whose capacity is less than $\log m$. We call such arcs *thin*, whereas an arc with capacity at least $\log m$ is called *thick*. A path is called *thick* if all its arcs are thick; otherwise it is called *thin*.

For a given instance of the max-IFP, we compute an approximate solution to the fractional problem using the FPTAS from Section 3 for some constant $\epsilon > 0$. If the total flow value along thick paths is at least half of the total flow value (and thus at least a constant fraction of the maximum integral flow value), we can use randomized rounding as explained above in order to obtain a constant factor approximation. Otherwise we can use the algorithm described in the following which computes an $O(\log m)$ -approximation from the flow that is routed along thin paths.

The algorithm works as follows. First we delete the flow routed along thick paths. From now on we consider only the part of the underlying graph which is used by thin paths. For each thin arc (v, w) insert a new node \tilde{v} , delete the arc (v, w) , insert the arcs (v, \tilde{v}) and (\tilde{v}, w) and assign the capacity of (v, w) to them. (The flow is adjusted adequately using arcs (v, \tilde{v}) and (\tilde{v}, w) instead of (v, w) .) The resulting network is denoted by $D = (V, A)$, the set of newly inserted nodes by U . Next, we make a copy $D' = (V', A')$ of D . From each node in U we insert an arc to its copy in D' . The resulting graph is denoted by $\bar{D} = (\bar{V}, \bar{A})$. We define v' to be the clone of $v \in V$ in V' and a' to be the clone of $a \in A$ in A' . An arc connecting a node $u \in U$ with u' is denoted by a_u . For $u \in U$ that was inserted to divide an arc a of the original digraph, the capacity of a_u is the same as that of a . We modify the considered fractional flow by rerouting all its paths from D to D' along the last thin arc at which a rerouting is possible. More precisely, this works as follows. Consider any path P that is used in the original fractional flow and let $(v, w) \in P$ be the last thin arc on P . (This arc does not exist in D .) Then the analogon to P in \bar{D} uses the adjusted path P in D until it reaches v , then uses (v, \tilde{v}) and is rerouted to D' along $a_{\tilde{v}}$. In D' the new path uses the arc from \tilde{v}' to w' and then the arcs corresponding to the ones P used in D after (v, w) . Let the resulting s - t' -flow be denoted by \bar{x} . Note that the value $|\bar{x}|$ of \bar{x} is still only some constant factor smaller than the value of a maximum one-flow in the original network. We choose

integral capacities $u(a)$ and $u(a')$ for a thick arc $a \in A$ and its clone a' as follows. If $\lceil \bar{x}(a) \rceil + \lceil \bar{x}(a') \rceil$ is not larger than the original capacity of a , we set $u(a) = \lceil \bar{x}(a) \rceil$ and $u(a') = \lceil \bar{x}(a') \rceil$. Otherwise, we choose the capacities by rounding the smaller flow value up and the larger one down. (The sum of the resulting values is not larger than the original capacity of a , because this was assumed to be integral.) For all thin arcs $a \in \bar{A}$, the capacity $u(a)$ is set to 1. It is easy to prove that $u(a) > \bar{x}(a)/\log m$, for all $a \in \bar{A}$. It follows immediately that a (usual) maximum s - t' -flow in \bar{D} with capacities u has flow value at least $|\bar{x}|/\log m$, because $\bar{x}/\log m$ is a feasible s - t' -flow in that network. By network flow theory, an integral maximum s - t' -flow can be computed in polynomial time. Since the value of this flow is at least $|\bar{x}|/\log m$, it is only by a factor $O(\log m)$ smaller than the value of a maximum one-flow in the original network. If we reroute this flow to D , i.e., do not let it pass over from D to D' and let it use the corresponding arcs in D instead, it is still feasible, because the sum of arc capacities $u(a)$ and $u(a')$ is at most the original capacity of $a \in A$. Further, the resulting flow does not send more than one unit of flow along each path, because each path uses at least one thin arc (otherwise it would not have been able to get from D to D') whose capacity is now 1. This gives us the result in Theorem 6. It also follows immediately from our analysis that the (multiplicative) integrality gap of the max-1FP is $O(\log m)$.

Theorem 6. *There exists a randomized $O(\log m)$ -approximation algorithm for the integral max-1FP whose runtime is polynomial in input plus output size.*

5 Multicommodity One-Flows

In this section we consider the multicommodity version of the 1FP in that we have several source-sink-pairs. We still have a digraph $D = (V, A)$ with arc capacities $u : A \rightarrow \mathbb{R}^+$. Instead of a single source-sink-pair we now have *requests* $(s_i, t_i) \in V \times V$ for $i = 1, \dots, K$, where $K \in \mathbb{N}$ denotes the total number of such pairs. Different optimization problems for multicommodity flows have been considered in the literature. Among them are maximization of the total flow sent through a network and minimization of the congestion of a flow that satisfies a given demand for each request. (The congestion measures the relative overload on an arc—a detailed definition follows later.) We consider both such optimization problems in the context of one-flows. We use \mathcal{P}_i to denote the set of s_i - t_i -paths in D , for all $i = 1, \dots, K$, and $\mathcal{P} := \bigcup_{i=1}^K \mathcal{P}_i$.

Maximum Multicommodity One-Flows. With the previous definitions we can describe the *maximum multicommodity one-flow problem (max-mc-1FP)* by the linear program in Section 1. Since solving the min-cost LP from Section 3 can easily be adjusted to the new situation, we can use the given FPTAS for the max-mc-1FP without any changes.

Theorem 7. *For any $\epsilon > 0$ and any instance of the max-mc-1FP with maximum flow value F^* , it is possible to compute a maximum multicommodity one-flow of value $(1 - \epsilon)F^*$ in time polynomial in the input size, ϵ^{-1} , and F^* .*

Proposition 2 still holds in the multicommodity case giving us a result similar to the one stated in Theorem 5. A result by Guruswami et al. [3] shows that, in general,

it is NP-hard to approximate the integral max-mc-IFP within a factor $O(m^{1/2-\epsilon})$, for any $\epsilon > 0$. However, randomized rounding yields an $O(\sqrt{m})$ -approximation algorithm for this problem. This can be proven using a more detailed analysis of the randomized rounding method given in [1].

Theorem 8. *There exists a randomized $O(\sqrt{m})$ -approximation algorithm for the integral max-mc-IFP whose runtime is polynomial in input plus output size.*

Minimizing Congestion. Here, each request (s_i, t_i) ($i = 1, \dots, K$) has a corresponding positive demand d_i which has to be satisfied by a one-flow. We look for a solution of minimum congestion. The *congestion* of a flow $(x_P)_{P \in \mathcal{P}}$ is the minimum μ such that $\sum_{P \ni a} x_P \leq \mu u(a)$, for all $a \in A$. The FPTAS from Section 3 can be adapted with the following result.

Theorem 9. *For any $\epsilon > 0$ and any instance of the min-cong-IFP with minimum congestion μ^* , it is possible to compute a multicommodity one-flow of congestion at most $(1 + \epsilon)\mu^*$ in time polynomial in the input size, ϵ^{-1} , and $d_{\max} := \max_i d_i$.*

Using Raghavan and Thompson's [9] randomized rounding method for the integral min-cong-IFP we obtain an $O(\log m)$ -approximation algorithm. Since the congestion version of the considered problem does not give any strict restrictions on the arc capacities, the algorithm can be derandomized by the method of conditional probabilities.

Theorem 10. *Applying randomized rounding to a nearly optimal fractional one-flow yields an $O(\log m)$ -approximation to the integral min-cong-IFP.*

References

1. A. Baveja, A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research*, 25:255–280, 2000.
2. L. R. Ford, D. R. Fulkerson. Maximal flow through a network. *Canadian J. of Math.*, 8:399–404, 1956.
3. V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. In *Proc. 31st Annual ACM Symposium on Theory of Computing*, 19–28, 1999.
4. J. M. Kleinberg. *Approximation Algorithms for Disjoint Path Problems*. PhD thesis, MIT, May 1996.
5. S. G. Kolliopoulos. Edge-disjoint paths and unsplittable flow. Chapter 57 in the *Handbook of Approximation Algorithms and Metaheuristics*, T. F. Gonzalez, Chapman-Hall/CRC Press, 2007.
6. E. L. Lawler. A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem. *Management Science*, 18:401–405, 1972.
7. M. Martens, M. Skutella. Flows on few paths: Algorithms and lower bounds. *Networks*, 48(2):68–76, 2006.
8. S. A. Plotkin, D. B. Shmoys, E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20:257–301, 1995.
9. P. Raghavan, C. D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.
10. L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. on Computing*, 8(3):410–421, 1979.