Length-Bounded and Dynamic k-Splittable Flows*

Maren Martens and Martin Skutella

Universität Dortmund, Fachbereich Mathematik, Lehrstuhl V,
Vogelpothsweg 87, 44227 Dortmund, Germany,
{maren.martens,martin.skutella}@math.uni-dortmund.de,
http://www.mathematik.uni-dortmund.de/lsv

Summary. Classical network flow problems do not impose restrictions on the choice of paths on which flow is sent. Only the arc capacities of the network have to be obeyed. This scenario is not always realistic. In fact, there are many problems for which, e.g., the number of paths being used to route a commodity or the length of such paths has to be small. These restrictions are considered in the length-bounded k-splittable s-t-flow problem: The problem is a variant of the well known classical s-t-flow problem with the additional requirement that the number of paths that may be used to route the flow and the maximum length of those paths are bounded. Our main result is that we can efficiently compute a length-bounded s-t-flow which sends one fourth of the maximum flow value while exceeding the length bound by a factor of at most 2. We also show that this result leads to approximation algorithms for dynamic k-splittable s-t-flows.

1 Introduction

Problem Definition and Motivation

We consider generalizations of the classical maximum s-t-flow problem where flow must be sent through a given network (digraph) $G = (V, E)$ with arc capacities $c : E \to \mathbb{R}^+$ from a source $s \in V$ to a sink $t \in V$.

k-Splittable Flows. The NP-hard *maximum k-splittable s-t-flow problem* introduced by Baier, Köhler, and Skutella [3] asks for a maximum s-t-flow which can be decomposed into flow on at most k paths. Here k is either a fixed constant or part of the input[2]. A feasible solution to this problem is called *k-splittable s-t-flow*; it is specified by a collection $\mathcal{P} = (P_1, \ldots, P_k)$ of k paths from s to t with corresponding nonnegative flow values $f_1, \ldots, f_k$ such that arc capacities are obeyed: $\sum_{i:\, e\in P_i} f_i \leq c(e)$

* This work was partially supported by DFG Focus Program 1126, "Algorithmic Aspects of Large and Complex Networks", grant no. SK 58/4-1 and SK 58/5-3.

[2] Since every s-t-flow can be decomposed into flow on at most $|E|$ paths and cycles, we always assume that $k \leq |E|$.

for all $e \in E$. The value of this flow is $\sum_{i=1}^{k} f_i$. Notice that some of the values f_i can be zero such that less than k paths are actually used to send flow.

A *uniform k-splittable s-t-flow* is a k-splittable s-t-flow where every path carries the same amount of flow f, i.e., $f = f_1 = \cdots = f_k$. The s-t-paths in collection $\mathcal{P}$ are not necessarily distinct, that is, $\mathcal{P}$ may contain several copies of the same s-t-path such that the total amount of flow being sent along this path is a multiple of the common value f.

The notion of k-splittable flows is motivated by transportation problems where divisible goods have to be shipped through a network using a bounded number of containers and each container must be routed along some path through the network. In the more general context of multicommodity flows, k-splittable flows generalize the notion of unsplittable flows which were introduced by Kleinberg [6]. A natural restriction in the area of transportation is to bound the length of paths that might be used to ship some commodity from its source to its destination. We therefore consider a generalization of k-splittable s-t-flows by imposing bounds on the lengths of the paths in $\mathcal{P}$.

Length-Bounded Flows. In addition to the setting described above we assume that there are also *arc lengths* $\ell : E \to \mathbb{R}^+$. Then, an s-t-flow specified by a collection of s-t-paths $\mathcal{P} = (P_1, \ldots, P_k)$ and corresponding flow values $f_1, \ldots, f_k$ is called *L-length-bounded* for some $L \in \mathbb{R}^+$ if $\sum_{e \in P_i} \ell(e) \leq L$ for $i = 1, \ldots, k$, that is, no path in $\mathcal{P}$ is longer than L. Baier [2] gives an extensive survey of what is known for length-bounded flows. We consider the *maximum length-bounded k-splittable s-t-flow problem*: Given k and L, find a maximum k-splittable s-t-flow among the ones which are L-length-bounded. This constitutes a natural combination and generalization of k-splittable and length-bounded s-t-flows.

Dynamic Flows. A crucial characteristic of network flows occurring in real-world applications is flow variation over time and the fact that flow does not travel instantaneously through a network but requires a certain amount of time (*transit time*) to travel through each arc. Both characteristics are captured by *dynamic flows* which specify a flow rate for each arc and each point in time. The *quickest s-t-flow problem* is to send a given amount of flow from s to t such that the last unit of flow arrives at the sink t as early as possible, i.e., within minimum time T. We consider the *quickest k-splittable s-t-flow problem* where, as in the static setting described above, the number of s-t-paths used to send flow is bounded by k. This dynamic flow problem is NP-hard since already its 'static' counterpart is NP-hard [3].

Related Results from the Literature

As mentioned above k-splittable flows are introduced in [3]. Among other results it is shown there that a maximum uniform k-splittable s-t-flow can be computed in polynomial time by a variant of the classical augmenting path algorithm. In contrast, it is NP-hard to find a maximum k-splittable s-t-flow. The value of a maximum k-splittable s-t-flow is at most twice as large as the value of a maximum uniform k-splittable s-t-flow. That is, computing a maximum uniform k-splittable s-t-flow yields a $1/2$-approximation algorithm for the maximum k-splittable s-t-flow problem. Other results on k-splittable flows have been found, e.g., by Bagchi [1]. Ford and Fulkerson [5] introduce dynamic s-t-flows. It follows from their work that the quickest s-t-flow problem can be solved in polynomial time. Fleischer and Skutella [4] show that certain NP-hard generalizations of the quickest s-t-flow problem (with multiple commodities or costs) can efficiently be approximated with constant performance guarantees via static length-bounded flow computations.

Contribution of this Paper

In Section 2 we present the following bicriteria approximation result for computing maximum length-bounded k-splittable s-t-flows.

Theorem 1. *There is a polynomial-time algorithm that computes a $2L$-length-bounded k-splittable s-t-flow whose flow value is at least one fourth of the value of a maximum L-length-bounded k-splittable s-t-flow.*

In Section 3 we apply a variant of this result in order to obtain the following approximation for the quickest k-splittable s-t-flow problem.

Theorem 2. *There is a $(3+2\sqrt{2})$-approximation algorithm for the quickest k-splittable s-t-flow problem.*

Due to space limitations, we only give an intuitive outline of the proof of Theorem 2.We conclude by presenting an interesting open problem.

2 Length-Bounded k-Splittable Flows

In this section we derive a simple combinatorial algorithm with the property stated in Theorem 1. Throughout this section, lengths of arcs are also interpreted as cost coefficients. We first show that a k-splittable s-t-flow obeying the given length bound L only on average can be found in polynomial time.

Lemma 1. *For given k and L, a maximum uniform k-splittable s-t-flow with average path length at most L can be computed in polynomial time.*

The proof of Lemma 1 is similar to the proof of [3, Theorem 6]. It is based on the insight that a uniform k-splittable s-t-flow with flow value kf is an f-integral s-t-flow (a flow is called *f-integral* for some $f \in \mathbb{R}^+$ if the flow value on each arc is an integral multiple of f). Moreover, any f-integral s-t-flow of value kf induces a uniform k-splittable s-t-flow of the same value by constructing an f-integral decomposition into paths and cycles and ignoring the cycles.

Proof (of Lemma 1). Consider a maximum uniform k-splittable s-t-flow with average path length at most L. There exists at least one arc $e \in E$ with a tight capacity constraint since otherwise a better solution can be obtained by increasing the common flow value f on all k paths. Hence, f is equal to the capacity $c(e)$ of arc e divided by the number of paths using the arc. Thus, $f = c(e)/i$ for some arc $e \in E$ and some $i \in \{1, \dots, k\}$. Based on this insight we formulate an algorithm: For all $e \in E$ and $i \in \{1, \dots, k\}$, compute a $c(e)/i$-integral min-cost s-t-flow of value $F_{e,i} := kc(e)/i$ or find out that no such flow exists. Among all computed flows whose total cost is at most $F_{e,i}L$ output one with largest flow value. If no such flow exists then output the zero flow. The running time of this algorithm is dominated by $k|E|$ min-cost s-t-flow computations. □

As discussed above, the flow described in Lemma 1 is an f-integral s-t-flow of value kf and cost at most kfL for some $f \in \mathbb{R}^+$. It can be turned into a $2L$-length-bounded k-splittable s-t-flow while decreasing the flow value only by a factor $1/2$.

Lemma 2. *Given an f-integral s-t-flow of value kf and cost at most kfL, a $2L$-length-bounded uniform k-splittable s-t-flow of value $kf/2$ can be found in polynomial time.*

Proof. The algorithm works as follows. First, the given flow is made acyclic by repeatedly canceling flow on cycles. Notice that this step does not increase cost since all cost coefficients (arc lengths) are nonnegative. Next, we cancel $f/2$ units of flow along the currently longest flow-carrying s-t-path and repeat this step k times. The resulting s-t-flow is $f/2$-integral and has flow value $kf/2$. Moreover, the length of any flow-carrying s-t-path is at most $2L$. Otherwise, all paths on which flow was canceled have length strictly larger than $2L$. Since $kf/2$ flow units were deleted from these paths, the cost of the initial flow must have been strictly larger than kfL—a contradiction. □

Notice that the computed s-t-flow is not only $2L$-length-bounded but has the stronger property that the length of *any* flow-carrying path is at most $2L$. This means that *any* path decomposition of this flow has the nice property of being $2L$-length-bounded. It is easy to come up with examples showing that this does not hold for arbitrary length-bounded s-t-flows.

We can now state the bicriteria approximation algorithm mentioned in Theorem 1: In the first step, a maximum uniform k-splittable s-t-flow with average path length at most L is computed (see Lemma 1). The second step turns this flow into a $2L$-length-bounded uniform k-splittable s-t-flow (Lemma 2). It remains to show the performance guarantee $1/4$ for the value of the computed flow. This follows from Lemma 2 and the following result.

Lemma 3. *The value of a maximum L-length-bounded k-splittable s-t-flow is at most twice as large as the value of a maximum L-length-bounded uniform k-splittable s-t-flow.*

The proof of this result is identical to the proof of [3, Theorem 12] and therefore omitted. Since the problem solved in Lemma 1 is a relaxation of the maximum L-length-bounded uniform k-splittable s-t-flow problem, the value of the flow computed in the first step of our algorithm is at least half as large as the optimum. Since the second step decreases the flow value by another factor $1/2$, this concludes the proof of Theorem 1.

Using the same technique as in the proof of Lemma 2 one can show for any ϵ with $0 < \epsilon < 1$ that given an f-integral s-t-flow of value kf and cost at most kfL, a $(1/\epsilon)L$-length-bounded uniform k-splittable s-t-flow of value $(1-\epsilon)kf$ can be found in polynomial time. This result can be used in order to show that there is a polynomial-time algorithm that computes a $(1/\epsilon)L$-length-bounded k-splittable s-t-flow whose flow value is at least $(1-\epsilon)/2$ times the value of a maximum L-length-bounded k-splittable s-t-flow.

A Note on the Complexity of Length-Bounded k-Splittable Flows

To emphasize that the maximum length-bounded k-splittable s-t-flow problem is indeed harder than the usual maximum length-bounded s-t-flow problem, we want to point out that it is possible to find a maximum length-bounded s-t-flow in polynomial time, if all arc lengths are equal to 1 (see, e.g., [2]). It is also shown in [2] that it is NP-complete to decide whether a digraph has a given number of length-bounded arc-disjoint s-t-paths with respect to unit arc lengths. This implies the following remark.

Proposition 1. *Even in a network with unit arc lengths it is NP-complete to decide whether there exists a length-bounded k-splittable s-t-flow of given value.*

Proof. It is easy to see that the problem is in NP. We reduce the NP-complete length-bounded arc-disjoint s-t-paths problem to it. One can decide whether a digraph has a given number M of length-bounded arc-disjoint s-t-paths with respect to unit arc lengths by checking if there exists a length-bounded M-splittable s-t-flow of value M in the network based upon this digraph with unit capacities. □

3 Dynamic *k*-Splittable Flows

The approximation algorithm in Theorem 1 can be used to construct an approximation algorithm for the quickest k-splittable s-t-flow problem. An instance of this problem consists of the same input as an instance of the maximum k-splittable s-t-flow problem. In addition, we are given transit times $\tau : E \to \mathbb{R}^+$ on the arcs and a prescribed demand value D. The task is to send D units of flow from the source s to the sink t on at most k paths within minimal time horizon T. For an exact definition of dynamic s-t-flows we refer to [5, 4]. We sometimes use the notion 'static flow' in order to emphasize that some flow is not dynamic. It follows from the work of Fleischer and Skutella [4] that a dynamic (k-splittable) s-t-flow of value D with time horizon T yields a static T-length-bounded (k-splittable) s-t-flow of value D/T (here we interpret transit times of arcs also as lengths). This static flow can be obtained by essentially averaging the dynamic flow over time. In particular, if the dynamic flow sends flow along at most k paths, then the same holds for the resulting static flow. On the other hand, a static T-length-bounded (k-splittable) s-t-flow of value d can be transformed into a dynamic (k-splittable) s-t-flow of value D

with time horizon $T + D/d$. The underlying transformation sends flow according to the given static flow pattern into the network for D/d time units. Then one has to wait for another T time units until the last unit of flow (traveling on a path of length, i.e., transit time, at most T) has arrived at the sink. Notice that the resulting dynamic flow uses exactly the same s-t-paths as the underlying static flow. For further details we refer to [4].

We can now prove Theorem 2. The time horizon of an optimum solution to the quickest k-splittable s-t-flow problem is denoted by T^*. Thus, there exists a static T^*-length-bounded k-splittable s-t-flow of value D/T^*. If T^* was known, one could compute a $2T^*$-length-bounded k-splittable s-t-flow of value at least $D/(4T^*)$; see Theorem 1. By slightly modifying the algorithm presented in Section 2 we can find $T \leq T^*$ and a $2T$-length-bounded k-splittable s-t-flow of value at least $D/(4T)$ in polynomial time. (We omit further details due to space limitations.) Applying the result of [4] thus yields a dynamic k-splittable s-t-flow of value D with time horizon $2T + 4T \leq 6T^*$.

Analogously we can use the general bicriteria approximation for length-bounded k-splittable s-t-flows in order to obtain a $(1+\epsilon)/(\epsilon-\epsilon^2)$-approximation algorithm for the dynamic k-splittable flow problem for every ϵ with $0 < \epsilon < 1$. Optimizing over ϵ we obtain a minimum for $\epsilon = \sqrt{2}-1$ which yields a $(3+2\sqrt{2})$-approximation with $3 + 2\sqrt{2} \approx 5.828$. This concludes the proof of Theorem 2.

Concluding Remark

We conclude by presenting a challenging open problem. Given a network with capacities and lengths on the arcs, a single source node s, and k sink nodes $t_1, \ldots, t_k$ with demand values $d_1, \ldots, d_k$. It is NP-hard to find an unsplittable flow that sends d_i units of flow from s to t_i along a single path of length at most L for $i = 1, \ldots, k$. It is an open problem to find a bicriteria approximation algorithm which sends a constant fraction of each demand d_i along a single s-t_i-path of length $O(L)$.

References

1. A. Bagchi. *Efficient Strategies for Topics in Internet Algorithmics*. PhD thesis, The Johns Hopkins University, October 2002.
2. G. Baier. *Flows with Path Restrictions*. PhD thesis, TU Berlin, 2003.
3. G. Baier, E. Köhler, and M. Skutella. On the k-splittable flow problem. *Algorithmica*, 42:231–248, 2005.
4. L. Fleischer and M. Skutella. The quickest multicommodity flow problem. In *Proceedings of the 9th Conference on Integer Programming and Combinatorial Optimization*, pages 36–53, 2002.
5. L. R. Ford and D. R. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research, 6*, pages 419–433, 1958.
6. J. M. Kleinberg. *Approximation Algorithms for Disjoint Path Problems*. PhD thesis, Massachusetts Institute of Technology, May 1996.