Mathematical Programming (2022) 192:477–496
https://doi.org/10.1007/s10107-021-01704-4

FULL LENGTH PAPER
Series B

Single source unsplittable flows with arc-wise lower and upper bounds

Sarah Morell[1] · Martin Skutella[1]

Received: 18 June 2020 / Accepted: 11 August 2021 / Published online: 1 September 2021

Abstract

In a digraph with a source and several destination nodes with associated demands, an unsplittable flow routes each demand along a single path from the common source to its destination. Given some flow x that is not necessarily unsplittable but satisfies all demands, it is a natural question to ask for an unsplittable flow y that does not deviate from x by too much, i.e., $y_a \approx x_a$ for all arcs a. Twenty years ago, in a landmark paper, Dinitz et al. (Combinatorica 19:17–41, 1999) proved that there exists an unsplittable flow y such that $y_a \leq x_a + d_{\max}$ for all arcs a, where $d_{\max}$ denotes the maximum demand value. Our first contribution is a considerably simpler one-page proof for this classical result, based upon an entirely new approach. Secondly, using a subtle variant of this approach, we obtain a new result: There is an unsplittable flow y such that $y_a \geq x_a - d_{\max}$ for all arcs a. Finally, building upon an iterative rounding technique previously introduced by Kolliopoulos and Stein (SIAM J Comput 31:919–946, 2002) and Skutella (Math Program 91:493–514, 2002), we prove existence of an unsplittable flow that simultaneously satisfies the upper and lower bounds for the special case when demands are integer multiples of each other. For arbitrary demand values, we prove the weaker simultaneous bounds $x_a/2 - d_{\max} \leq y_a \leq 2x_a + d_{\max}$ for all arcs a.

Keywords Network flow · Unsplittable flow · Rounding · Flow augmentation

Mathematics Subject Classification 90C27 Combinatorial Optimization

Partially supported by DFG Priority Programme 1736 (grant SK 58/10-2).

✉ Sarah Morell
morell@math.tu-berlin.de

Martin Skutella
skutella@math.tu-berlin.de

[1] Combinatorial Optimization & Graph Algorithms Group, Institut für Mathematik, Technische Universität Berlin, Berlin, Germany

1 Introduction

Ever since the seminal work of Ford and Fulkerson [6] network flows belong to the most important and fundamental class of problems in combinatorial optimization and mathematical programming. We refer to the classical textbook [1] by Ahuja, Magnanti, and Orlin as well as the very recent new textbook [17] by Williamson on the topic.

1.1 Problem setting and notation

Let $D = (V, A)$ be a directed acyclic graph with source node $s \in V$ and k commodities with destination nodes $t_1, \dots, t_k \in V$ and associated demands $d_1, \dots, d_k \in \mathbb{R}_{>0}$. A flow $x \in \mathbb{R}^A_{\geq 0}$ *satisfies the given demands* if it simultaneously sends d_i units of flow from s to t_i, for all $i = 1, \dots, k$. That is, x must satisfy the following flow conservation constraints:

$$x(\delta^{\text{in}}(v)) - x(\delta^{\text{out}}(v)) = \begin{cases} d_i & \text{if } v = t_i \text{ for some } i \in \{1, \dots, k\}, \\ -\sum_{i=1}^{k} d_i & \text{if } v = s, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Here, $\delta^{\text{in}}(v)$ and $\delta^{\text{out}}(v)$ denote the set of incoming and outgoing arcs of node v, respectively; for $B \subseteq A$, let $x(B) := \sum_{a \in B} x_a$. In the following, whenever we refer to a *flow*, we mean a flow satisfying the given demands, i.e., Constraints (1), unless stated otherwise.

The following classical integrality property of network flows (see, e.g., [1, Theorem 9.10]) follows, for example, from the fact that the node-arc incidence matrix, which implicitly occurs on the left-hand side of (1), is totally unimodular.

Theorem 1 *If the demands $d_1, \dots, d_k$ are all integral, then any flow x can be written as a convex combination of integral flows such that each such integral flow $y \in \mathbb{Z}^A_{\geq 0}$ satisfies*

$$\lfloor x_a \rfloor \leq y_a \leq \lceil x_a \rceil \qquad \textit{for all } a \in A.$$

In particular, there exists an integral flow y obeying these upper and lower bounds.

1.2 Single source unsplittable flows

In 1996, Kleinberg [7] introduced *single source unsplittable flows*. A flow is called *unsplittable* if the entire demand of each commodity is routed along one path from the source to its destination node. That is, an unsplittable flow y can be specified as follows: for all $i = 1, \dots, k$, there is one s-t_i-path P_i^y in D such that

$$y_a = \sum_{i : a \in P_i^y} d_i \qquad \text{for all } a \in A.$$

In order to emphasize the fact that a particular flow x is not necessarily unsplittable, we sometimes refer to x as a *fractional* flow in this case. For the special case of unit demands, a flow is unsplittable if and only if it is integral.

1.3 Related literature

Single source unsplittable flows constitute a special case of more general unsplittable flows where each commodity has its own source and destination node. General unsplittable flows have been well studied in the literature as an interesting extension of disjoint paths. For instance, if we are given arc capacities and demands for each commodity and look for an unsplittable flow of minimum congestion, i.e., of minimum overload of arc capacities, Raghavan and Thompson [13,14] present an approximation algorithm based on their randomized rounding technique. We refer to the survey [10] by Kolliopoulos for an overview of results on general unsplittable flows.

The problem of finding a single source unsplittable flow in a directed graph with capacities on the arcs contains several well-known NP-complete problems as special cases, e.g., Partition, Bin Packing, or even scheduling parallel machines with makespan objective; we refer to Kleinberg's PhD thesis [7] for more details and other special cases.

Kleinberg [7], Dinitz et al. [4], Kolliopoulos and Stein [11], and Skutella [16] present approximation algorithms for various optimization versions of the single source unsplittable flow problem. Du and Kolliopoulos [5] have implemented and empirically tested several of those approximation algorithms.

Baier et al. [2] introduce the following interesting relaxation of unsplittable flows. For a given $k \geq 1$, a *k-splittable flow* must route each commodity along at most k paths. In particular, 1-splittable flows are unsplittable flows. It follows from the classical flow decomposition theorem that k-splittability is not a meaningful restriction for $k \geq |A|$. Single source k-splittable flows are studied, e.g., by Kolliopoulos [9], Koch, Skutella, and Spenke [8], as well as Salazar and Skutella [15].

The central result in the seminal paper of Dinitz et al. [4] is the following theorem on single source unsplittable flows, where the maximum demand value is denoted by $d_{\max} := \max\{d_1, \ldots, d_k\}$.

Theorem 2 *For a given flow x, there exists an unsplittable flow y such that*

$$y_a \;\leq\; x_a + d_{\max} \qquad \text{for all } a \in A. \tag{2}$$

The proof of this theorem given in [4] is in fact algorithmic, that is, the authors provide an efficient algorithm that turns the given flow x into an unsplittable flow y satisfying the arc-wise upper bounds (2). We give a rather short and intuitive sketch of their procedure.

Starting from the fractional flow x, the algorithm aims to iteratively move destination nodes towards the source node s by repeatedly augmenting flow along residual cycles featuring a special property. Intuitively, in order to find such a cycle, one starts at an arbitrary node in the graph and moves *forward* along arcs whenever possible, and only moves *backward* (i.e., in the opposite direction of arcs) if a destination node without outgoing arcs is reached. As soon as a cycle occurs, flow is decreased along the forward arcs of the cycle and increased along the backward arcs until one can either remove a forward arc with no more flow on it or move a destination node t_i towards s along an incoming (backward) arc with flow value at least d_i. This process is iterated until all destination nodes reach the source node s. The resulting unsplittable flow y sends each commodity i along the s-t_i-path through which its destination node t_i has moved towards the sink (in opposite direction).

The proof implies the following strengthening of Theorem 2 which can also be found in [4].

Corollary 1 *For a given flow x, there is an unsplittable flow y such that, for every arc $a \in A$, the sum of all but one of the demands routed along a is at most x_a.*

1.4 Contribution and outline

In Sect. 2 we present a considerably simpler one-page proof of Theorem 2 and Corollary 1, based upon an entirely new approach. Dinitz, Garg, and Goemans start with the fractional flow $y := x$ and then iteratively modify y, always maintaining Property (2), until y is unsplittable. In contrast, we start with an arbitrary unsplittable flow y violating Property (2) and then iteratively modify y, always maintaining an unsplittable flow, until y meets Property (2).

In Sect. 3, using a similar approach, we derive the following new covering analogue of the packing result in Theorem 2:

Theorem 3 *For a given flow x, there exists an unsplittable flow y such that*

$$y_a \geq x_a - d_{\max} \quad \text{for all } a \in A. \tag{3}$$

To prove this result, we again iteratively turn an arbitrary unsplittable flow into one that meets Property (3). In contrast to the packing result, however, the proof of the covering bounds in (3) turns out to be somewhat more intricate, requiring several additional insights and arguments.

Similar in spirit to Corollary 1, our proof implies the following strengthening of Theorem 3.

Corollary 2 *For a given flow x, there exists an unsplittable flow y such that $y_a \geq \tilde{x}_a - \max_i d_i$ for all $a \in A$, where the maximum is only taken over those commodities i with arc a lying on some s-t_i-path.*

Section 4 considers upper bounds (2) and lower bounds (3) simultaneously. Using techniques introduced by Kolliopoulos and Stein [11], Skutella [16], and Martens et al. [12], we obtain the following generalization of Theorem 1:

Theorem 4 *If the demand values are integer multiples of each other, i.e., $d_k \mid d_{k-1} \mid \ldots \mid d_1$, then any flow x can be written as a convex combination of unsplittable flows such that each such unsplittable flow y satisfies*

$$x_a - d_{\max} \leq y_a \leq x_a + d_{\max} \quad \text{for all } a \in A. \tag{4}$$

In particular, for arbitrary cost $c = (c_a)_{a \in A}$ on the arcs, there exists an unsplittable flow y obeying Property (4) such that $c(y) \leq c(x)$.

Finally, for arbitrary demand values, we obtain the following slightly weaker bounds:

Theorem 5 *For a given flow x, there exists an unsplittable flow y such that*

$$\frac{x_a}{2} - d_{\max} \ \leq \ y_a \ \leq \ 2\,x_a + d_{\max} \quad \textit{for all } a \in A. \tag{5}$$

We conclude in Sect. 5 by pointing out several interesting open problems and stating a stronger version of Goemans' unsplittable flow conjecture.

1.5 Preliminaries

We assume throughout this paper that, without loss of generality, each node $v \in V$ lies on an s-t_i-path for some $i \in \{1, \ldots, k\}$. Paths are considered to be subsets of the given arc set A. For a path Q and two nodes $v, w \in V$ lying on path Q (with v being visited first), the v-w-subpath of Q is denoted by $Q|_{[v,w]}$. Finally, for a subset of nodes $X \subseteq V \backslash \{s\}$, let $d(X) := \sum_{i: t_i \in X} d_i$ denote the total demand of sinks in X.

2 A short proof of the Dinitz–Garg–Goemans Theorem

Our novel proof of Theorem 2 relies on a simple augmentation step, called *upper bound preserving (UBP)* augmentation step. For a given flow x and an arbitrary unsplittable flow y, we say that a node v is *UBP-reachable w.r.t. y* if there exists an s-v-path Q such that

$$y_a \ \leq \ x_a \qquad \text{for all } a \in Q.$$

A *UBP augmentation step* for an unsplittable flow y is defined as follows: Given a node v that is UBP-reachable w.r.t. y along path Q and a commodity i such that node v lies on path P_i^y, reroute commodity i from s to v along path Q. This results in a new unsplittable flow y' using a new s-t_i-path $P_i^{y'}$ with $P_i^{y'}|_{[s,v]} = Q$; see Fig. 1 for an illustration. Notice that

$$y'_a \leq x_a + d_i \qquad \text{for all } a \in P_i^{y'}|_{[s,v]},$$

which explains why the augmentation step is called *upper bound preserving*.

If an unsplittable flow y' results from y by a finite sequence of UBP augmentation steps, we write $y \overset{\text{UBP}}{\rightsquigarrow} y'$. We say that a node v is *eventually UBP-reachable (eUBP-reachable) w.r.t. y* if there exists an unsplittable flow y' with $y \overset{\text{UBP}}{\rightsquigarrow} y'$ such that v is UBP-reachable w.r.t. y'.

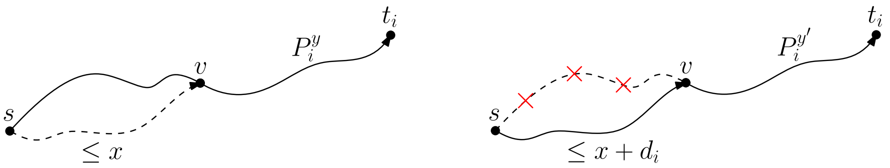

Fig. 1 For a given unsplittable flow y, let v be UBP-reachable w.r.t. y along a path Q (illustrated dashed on the left) and let i be a commodity such that v lies on path P_i^y. A UBP augmentation step reroutes commodity i from s to v along path Q. The resulting unsplittable flow y' is illustrated on the right

Lemma 1 *For any unsplittable flow y, all nodes in V are eUBP-reachable w.r.t. y.*

Proof Assume by contradiction that there exists an unsplittable flow y such that the set X_y of eUBP-reachable nodes w.r.t. y is a proper subset of V, i.e., $X_y \subsetneq V$. Applying UBP augmentation steps to y cannot enlarge the set of eUBP-reachable nodes. Hence, if we choose y such that X_y is inclusion-wise minimal, then $X_y = X_{y'} =: X$ for any unsplittable flow y' with $y \overset{\text{UBP}}{\rightsquigarrow} y'$. We prove three important properties of y and X:

(P1) $y_a > x_a$ for all $a \in \delta^{\text{out}}(X)$.
Indeed, if there is an arc $a = (v, w) \in \delta^{\text{out}}(X)$ with $y_a \leq x_a$, let y' be an unsplittable flow resulting from y by a shortest possible sequence of UBP augmentation steps such that v is UBP-reachable w.r.t. y'. As long as node v is not UBP-reachable, flow on arc a cannot increase during a UBP augmentation step. Hence, $y'_a \leq y_a \leq x_a$ which implies that not only v but also w is UBP-reachable w.r.t. y'. Hence, $w \in X$, a contradiction.

(P2) $y(\delta^{\text{in}}(X)) = \sum_{a \in \delta^{\text{in}}(X)} y_a > 0$.
By assumption, there is a path from source $s \in X$ to any node in $V \backslash X$. Hence, $\delta^{\text{out}}(X) \neq \varnothing$. Both flows x and y satisfy the same set of demands. In particular, $y(\delta^{\text{out}}(X)) - y(\delta^{\text{in}}(X)) = d(V \backslash X) = x(\delta^{\text{out}}(X)) - x(\delta^{\text{in}}(X))$. But, by Property (P1), $y(\delta^{\text{out}}(X)) > x(\delta^{\text{out}}(X))$. Therefore, $y(\delta^{\text{in}}(X)) > 0$.

(P3) $y'(\delta^{\text{in}}(X)) \leq y(\delta^{\text{in}}(X))$ for each unsplittable flow y' with $y \overset{\text{UBP}}{\rightsquigarrow} y'$.
It is sufficient to consider the case that y' is obtained from y by one UBP augmentation step, which reroutes commodity i from source s to some node $v \in X$ along an s-v-path Q. By definition of X, path Q must remain within X. Moreover, $P_i^y|_{[s,v]}$ may either remain within X, implying that the total in-flow of y does not change; or it may exit and re-enter X, implying that $y(\delta^{\text{in}}(X))$ decreases by (a multiple of) demand d_i.

Since there are only finitely many different unsplittable flows and in view of Property (P3), we choose y with $y(\delta^{\text{in}}(X))$ minimal. By Property (P2), there exists an arc $a = (w, v) \in \delta^{\text{in}}(X)$ such that $y_a > 0$. Since $v \in X$, there is an unsplittable flow y' with $y \overset{\text{UBP}}{\rightsquigarrow} y'$ such that v is UBP-reachable w.r.t. y' via an s-v-path Q; see Fig. 2.

Flow on arc a remains unchanged, i.e., $y'_a = y_a$, and there exists a commodity i with $a \in P_i^{y'}$. Rerouting commodity i from s to v along Q decreases $y'(\delta^{\text{in}}(X)) = y(\delta^{\text{in}}(X))$, contradicting its minimality. ◻

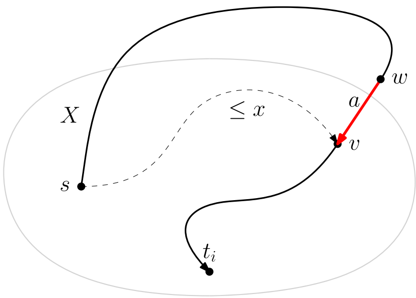

Fig. 2 Arc $a = (w, v) \in \delta^{\text{in}}(X)$ is used by path $P_i^{y'}$, illustrated solid, for some commodity i. Its demand d_i can be rerouted from s to v along an s-v-path Q, illustrated dashed

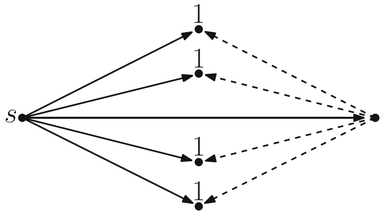

Fig. 3 An instance with k demands of value 1. Flow x is given as follows: solid arcs have flow value $\frac{k}{k+1}$ and dashed arcs carry flow value $\frac{1}{k+1}$. Notice that any unsplittable flow sends zero flow on some solid arc

In order to prove Theorem 2, we start with an arbitrary unsplittable flow y_0. By Lemma 1, each of the destination nodes $t_i, i = 1, \ldots, k$ is eUBP-reachable w.r.t. any unsplittable flow, i.e., t_i is UBP-reachable w.r.t. some unsplittable flow via an s-t_i-path Q. We apply a UBP augmentation step, rerouting demand d_i along path Q. The resulting flow y_i satisfies the desired arc-wise upper bound on all arcs lying on the new s-t_i-path $P_i^{y_i} = Q$.

Whenever a commodity i is routed properly, meaning that y_i satisfies the arc-wise upper bound on all arcs, it remains so, independently of which UBP augmentation steps may follow. Hence, if we apply this procedure successively to each of the destination nodes $t_i, i = 1, \ldots, k$, the commodities $1, \ldots, i$ are properly routed in the resulting flow y_i. Therefore, the final flow y_k satisfies the arc-wise upper bound on all arcs. This concludes the proof of Theorem 2.

Notice that for every arc $a \in A$ and for the commodity j being routed along a at last, the final unsplittable flow y satisfies $y(a) - d_j \leq x(a)$. This implies that Corollary 1 can also be deduced from the previously presented proof of Theorem 2.

3 Unsplittable flows with arc-wise lower bounds

Notice that the lower bound (3) in Theorem 3 is tight in the following sense: For each $\varepsilon > 0$, there exists a digraph D together with a fractional flow x such that no unsplittable flow y satisfies $y_a \geq x_a - d_{\max} + \varepsilon$, for all $a \in A$; see the instance depicted in Fig. 3 with $k = \lceil 1/\varepsilon \rceil$.

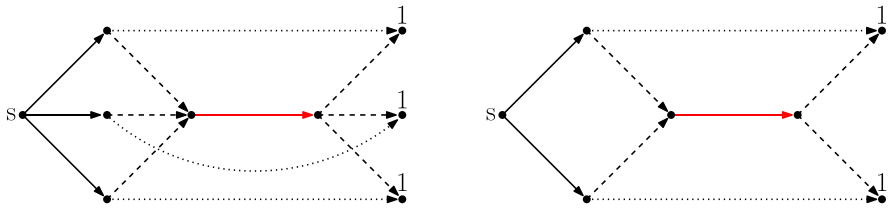

Fig. 4 Let the digraph have three commodities with demand 1, as illustrated on the left, and let the fractional flow x be given as follows: $x_{\text{solid}} = 1, x_{\text{dashed}} = 1 - \epsilon, x_{\text{dotted}} = \epsilon$, and $x_{\text{red}} = 3 - 3\epsilon$ for some $\epsilon > 0$. The algorithm in [4] first creates an alternating cycle including the red arc as a forward arc. Hence, we decrease flow on the red arc by $1 - \epsilon$ and move one demand towards source s along its corresponding dotted arc. The remaining instance with two commodities and $x'_{\text{red}} = 2 - 2\epsilon$ is illustrated on the right. Any further step again decreases flow on the red arc by $1 - \epsilon$, hence violating the desired lower bound on the red arc

We point out that the techniques provided in [4] are not adaptable for handling arc-wise lower bounds, as illustrated in Fig. 4. By adding commodities and expanding the given graph, the violation of lower bounds can be arbitrarily large.

There does not seem to be any analogous alternative operation for lower bounds. Consequently, even though the two problems regarding arc-wise upper and lower bounds seem similar in spirit, we need to develop new tools in order to solve the above mentioned problem.

3.1 A proof for unsplittable flows satisfying arc-wise lower bounds

The proof of Theorem 3 is based on a similar approach as our proof of the Dinitz-Garg-Goemans Theorem in Sect. 2, yet turns out to be somewhat more intricate. Since we are no longer interested in upper bounds but in lower bounds, we now use *lower bound preserving (LBP)* augmentation steps: for an unsplittable flow y, we say that node v is *LBP-reachable w.r.t.* y if there is a commodity i whose s-t_i-path P_i^y passes through node v and the s-v-subpath $P_i^y|_{[s,v]}$ satisfies

$$y_a \geq x_a \qquad \text{for all } a \in P_i^y|_{[s,v]}.$$

To emphasize the role of commodity i, we also say that node v is *LBP-reachable for commodity* i *w.r.t.* y in this case.

An LBP augmentation step for an unsplittable flow y is defined as follows: Given a node v that is LBP-reachable for commodity i w.r.t. y, reroute commodity i from source s to node v along an arbitrary s-v-path Q. This results in a new unsplittable flow y' using a new s-t_i-path $P_i^{y'}$ with $P_i^{y'}|_{[s,v]} = Q$; see Fig. 5 for an illustration.

Notice that for each $a \in P_i^y|_{[s,v]}$ we have $y'_a \geq x_a - d_i$, which explains why the augmentation step is called *lower bound preserving*. If an unsplittable flow y' results from y by a finite sequence of LBP augmentation steps, we write $y \overset{\text{LBP}}{\rightsquigarrow} y'$. We say that node v is *eventually LBP-reachable (eLBP-reachable) for commodity* i *w.r.t.* y if there is an unsplittable flow y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$ such that v is LBP-reachable (for commodity i) w.r.t. y'.

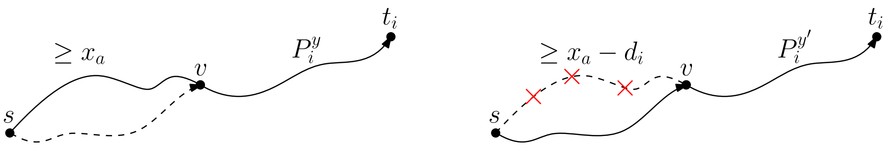

Fig. 5 For a given unsplittable flow y, let v be LBP-reachable for some commodity i w.r.t. y. An LBP augmentation step reroutes commodity i from s to v along an arbitrary s-v-path Q (illustrated dashed on the left). The resulting unsplittable flow y' is illustrated on the right

Proposition 1 *Let y be an unsplittable flow and v a node on path P_i^y for some commodity i. If v is eLBP-reachable w.r.t. y' for all y' with $y \overset{LBP}{\rightsquigarrow} y'$, then v is eLBP-reachable* for commodity *i w.r.t. y.*

Proof Assume by contradiction that v is not eLBP-reachable for commodity i w.r.t. y. Then any y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$ constitutes another counterexample. In particular, node v is on path $P_i^{y'}$ and the s-v-subpath $P_i^{y'}|_{[s,v]}$ contains at least one arc a such that $y'_a < x_a$. Let $a^{y'} = (u^{y'}, w^{y'})$ be such arc closest to v. Let $\gamma^{y'}$ be the number of arcs on $P_i^{y'}|_{[w^{y'},v]}$.

Choose counterexample y such that the following two criteria are met in the given order:

(i) γ^y is maximal,
(ii) y_{a^y} is maximal.

The situation is depicted in Fig. 6.

Consider a sequence of unsplittable flows $y = y_0, y_1, \ldots, y_q$, where each y_j results from y_{j-1} via an LBP augmentation step, such that there is a node w on $P_i^y|_{[w^y,v]}$ that is LBP-reachable w.r.t. y_q but no node on $P_i^y|_{[w^y,v]}$ is LBP-reachable w.r.t. y_j for any $j < q$. Such a sequence exists since node v is eLBP-reachable w.r.t. y. As no node on $P_i^y|_{[w^y,v]}$ is LBP-reachable w.r.t. y_j for any $j < q$, flow on the arcs of $P_i^y|_{[u^y,v]}$ is never decreased during these LBP augmentation steps and $P_i^y|_{[u^y,v]} = P_i^{y_j}|_{[u^y,v]}$ for all $j \leq q$. In particular, due to (i) and (ii), $\gamma^{y_j} = \gamma^y$, $a^{y_j} = a^y$, and the flow on arc a^y remains unchanged for all $j \leq q$.

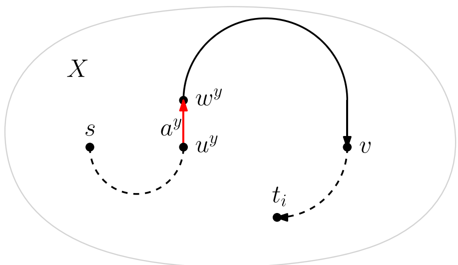

Fig. 6 Arc $a^y = (u^y, w^y)$ satisfies $y_{a^y} < x_{a^y}$ and is closest to v among all such arcs on path $P_i^y|_{[s,v]}$. This implies that all arcs a on $P_i^y|_{[w^y,v]}$, illustrated solid, carry flow $y_a \geq x_a$. The number of arcs on $P_i^y|_{[w^y,v]}$ is denoted by γ^y

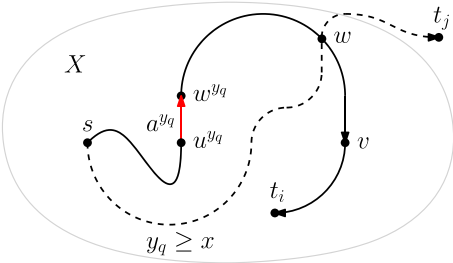

Fig. 7 Node w is LBP-reachable for some commodity j w.r.t. y_q; path $P_j^{y_q}$ is illustrated dashed. The demand d_j can be rerouted along the s-w-subpath of $P_i^{y_q}$, now illustrated solid

Finally, node w is LBP-reachable for some commodity w.r.t y_q; see Fig. 7. Rerouting that commodity along $P_i^y|_{[s,w]}$ strictly increases flow on a^y, contradicting the choice of y in terms of (i), (ii). □

Lemma 2 *For any unsplittable flow y, all nodes in V are eLBP-reachable for some commodity w.r.t. y.*

Proof Assume by contradiction that there exists an unsplittable flow y such that the set X_y of eLBP-reachable nodes w.r.t. y is a proper subset of V, i.e., $X_y \subsetneq V$. Applying LBP augmentation steps to y cannot enlarge the set of eLBP-reachable nodes. Hence, if we choose y such that X_y is inclusion-wise minimal, then $X_y = X_{y'} =: X$ for any unsplittable flow y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$. Notice that an LBP augmentation step cannot decrease the outgoing flow $y(\delta^{\text{out}}(X))$. Indeed, if a node $v \in X$ is LBP-reachable for some commodity i w.r.t. y, then all nodes on the s-v-subpath $P_i^y|_{[s,v]}$ are LBP-reachable for commodity i w.r.t y. Hence, $P_i^y|_{[s,v]}$ remains within node set X_y and rerouting commodity i cannot decrease $y(\delta^{\text{out}}(X))$. We can thus conclude that $y'(\delta^{\text{out}}(X)) \geq y(\delta^{\text{out}}(X))$ for each unsplittable flow y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$. Since there are only finitely many different unsplittable flows and in view of this property, choose y with $y(\delta^{\text{out}}(X))$ maximal. We prove two important properties of y and X:

(P1') $\delta^{\text{in}}(X) = \varnothing$.
Indeed, if there is an arc $(w, v) \in \delta^{\text{in}}(X)$, let y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$ be such that node $v \in X$ is LBP-reachable for some commodity i w.r.t. y'. Rerouting commodity i along an arbitrary s-v-path Q with $(w, v) \in Q$ increases flow on (w, v) as well as on some arc in $\delta^{\text{out}}(X)$, contradicting the maximality of $y(\delta^{\text{out}}(X))$.

(P2') There is an arc $a \in \delta^{\text{out}}(X)$ with $y_a > 0$ and $y_a \geq x_a$.
Consider a node $v \in V \setminus X$. By assumption, there is a v-t_i-path in D for some commodity i. By Property (P1'), node t_i must also lie in $V \setminus X$. Therefore, again by Property (P1), we have

$$y(\delta^{\text{out}}(X)) = x(\delta^{\text{out}}(X)) = d(V \setminus X) \geq d_i > 0.$$

This implies the existence of arc a with the properties stated above.

Consider an arc $a = (v, w) \in \delta^{\text{out}}(X)$ as in (P2') and some commodity i with $a \in P_i^y$. Since $w \in V \setminus X$ and $\delta^{\text{in}}(X) = \varnothing$ by (P1'), it is impossible for any sequence of LBP augmentations to add or delete flow on arc a. In particular, $y'_a = y_a \geq x_a$ and $a \in P_i^{y'}$ for each unsplittable flow y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$. By Proposition 1, node v is eLBP-reachable for commodity i w.r.t. y. Hence node w is eLBP-reachable for commodity i w.r.t. y, a contradiction to $w \notin X$. □

Lemma 3 *For any unsplittable flow y and any arc a, there exists an unsplittable flow y' with $y \overset{LBP}{\rightsquigarrow} y'$ such that $y'_a \geq x_a$.*

Proof Assume by contradiction that there exists an unsplittable flow y and an arc $a = (v, w) \in A$ such that $y'_a < x_a$ for any y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$. Notice that flow on arc a can never be decreased by an LBP augmentation step. We may choose y in such a way that y_a is maximal. By Lemma 2, node w is LBP-reachable for some commodity i w.r.t. an unsplittable flow y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$. Rerouting commodity i along any s-w-path Q with $a \in Q$ increases the flow on arc a, hence contradicting its maximality. □

We can now prove Theorem 3: Let y be an arbitrary unsplittable flow. By Lemma 3, for each arc $a \in A$ there exists an unsplittable flow y' with $y \overset{\text{LBP}}{\rightsquigarrow} y'$ such that $y'_a \geq x_a$. Notice that any flow resulting from y' by a sequence of LBP augmentation has a flow value on a of at least $x_a - d_{\max}$. Going through all arcs successively leads to a final unsplittable flow with the desired properties. This concludes the proof of Theorem 3.

In order to prove Corollary 2, notice that for every arc $a \in A$ and for the last commodity j being removed from a in an LBP augmentation step, the final unsplittable flow y satisfies $y(a) + d_j \geq x(a)$. This implies that

$$y(a) \;\geq\; x(a) - d_j \;\geq\; x_a - \max_i d_i,$$

where the maximum is only taken over those commodities i such that arc a lies on some s-t_i-path.

3.2 Problem variants for unsplittable flows satisfying arc-wise lower bounds

In the context of unsplittable flows respecting arc-wise upper bounds, several interesting problem variants have been considered in the literature; see, e.g., [7]. Theorem 2 immediately implies approximation results for the *minimum congestion problem* whose objective is to bound the violation of given upper bounds (arc capacities). Another prominent problem, the *minimum number of rounds problem*, asks for a partition of the set of commodities into a minimum number of subsets (rounds) such that each subset can be routed unsplittably without violating given arc capacities. Finally, the *maximum routable demand problem* asks for a feasible (w.r.t. arc capacities) unsplittable routing of a subset of commodities of maximum total demand. We refer to [4] for further details.

With a view to these optimization problems, the (fractional) flow x in Theorem 2 plays the role of a solution to a fractional relaxation obeying given arc capacities.

Similarly, Theorem 3 is relevant for unsplittable flow problems with lower capacities on the arcs. If we assume that a given (fractional) flow x obeys lower arc capacities, a meaningful question in this context is, how many copies of the commodities are needed such that one can find an unsplittable routing y of *all* copies such that $y_a \geq x_a$ for all arcs $a \in A$.

Corollary 3 *Given a fractional flow x, let $\alpha \geq 1$ be such that $\alpha \cdot x_a \geq \max_i d_i$ for all arcs $a \in A$, where the maximum is taken over all i with arc a lying on an s-t_i-path. Then, at most $\lceil 1+\alpha \rceil$ copies of commodities are necessary in order to find an unsplittable routing y of all those copies such that $y_a \geq x_a$ for all $a \in A$.*

Proof Define $\tilde{x}$ by $\tilde{x}_a = \lceil 1+\alpha \rceil \cdot x_a$ for each $a \in A$. Hence $\tilde{x}$ satisfies $\lceil 1+\alpha \rceil$ copies of each demand d_i, for $i = 1, \ldots, k$. By Corollary 2, there exists an unsplittable flow y satisfying $\lceil 1+\alpha \rceil$ copies of each demand d_i, $i = 1, \ldots, k$, such that $y_a \geq \tilde{x}_a - \max_i d_i$ for all $a \in A$, where the maximum is taken over all commodities i with arc a lying on an s-t_i-path. Since $\alpha \cdot x_a \geq \max_i d_i$ for all $a \in A$, and by definition of $\tilde{x}$, we get $y_a \geq \lceil 1+\alpha \rceil \cdot x_a - \alpha \cdot x_a \geq x_a$, for all $a \in A$. □

4 Combining lower and upper bounds

In this section we obtain results on unsplittable flows that simultaneously obey arc-wise upper and lower bounds with respect to the given fractional flow x. We first consider the special case where demands are multiples of each other, i.e., $d_k \mid d_{k-1} \mid \ldots \mid d_1$, and show how Theorem 4 can be obtained via methods introduced by Kolliopoulos and Stein [11], Skutella [16], and Martens et al. [12]. To make the terminology precise, for $a, b \in \mathbb{R}_{>0}$ we write $a \mid b$ if there is an integer $c \in \mathbb{Z}$ such that $a \cdot c = b$. In this case we say that b is *a-integral*.

Proof of Theorem 4 In the considered case, the demands are all d_k-integral for the minimum demand value d_k. By scaling the demand values and the given flow x accordingly, we may assume that $d_k = 1$ such that all demands are integral. Therefore, by Theorem 1, the given fractional flow x is a convex combination of d_k-integral flows whose flow values on the arcs differ from x by (strictly) less than d_k. It therefore suffices to show that any d_k-integral flow can be written as a convex combination of unsplittable flows whose flow values differ by at most $d_1 - d_k$. The proof of the following proposition thus concludes the proof of Theorem 4. □

Proposition 2 *If $d_k \mid d_{k-1} \mid \ldots \mid d_1$, then any d_k-integral flow x can be written as a convex combination of unsplittable flows such that each such unsplittable flow y satisfies*

$$x_a - (d_1 - d_k) \ \leq \ y_a \ \leq \ x_a + (d_1 - d_k) \quad \text{for all } a \in A.$$

Proof We use induction on the number of commodities k. For the case $k = 1$, the d_k-integral flow x is, in fact, an unsplittable flow and we are done. Thus assume that $k > 1$ and the proposition holds for the case of $k - 1$ commodities.

Any d_k-integral flow x can be easily interpreted to route commodity k unsplittably: choose a flow-carrying s-t_k-path P_k, decrease the flow along P_k by d_k, and delete commodity k from the instance. This leaves us with a d_k-integral flow x' satisfying the remaining demands $d_{k-1}, \dots, d_1$, which are all d_{k-1}-integral. Notice that it suffices to show that x' can be written as a convex combination of unsplittable flows satisfying demands $d_{k-1}, \dots, d_1$ such that each such unsplittable flow y satisfies

$$x'_a - (d_1 - d_k) \;\le\; y_a \;\le\; x'_a + (d_1 - d_k) \quad \text{for all } a \in A.$$

By scaling the demand values $d_k, \dots, d_1$ and flow x' accordingly, we may assume $d_{k-1} = 1$ such that all remaining demands $d_{k-1}, \dots, d_1$ are integral. Thus, by Theorem 1 and since $d_k \mid 1$, the d_k-integral flow x' can be written as a convex combination of integral flows with each such integral flow x'' satisfying

$$x'_a - (1 - d_k) \;\le\; x''_a \;\le\; x'_a + (1 - d_k) \quad \text{for all } a \in A. \tag{6}$$

By induction, each such integral flow x'' can again be written as a convex combination of unsplittable flows y with

$$x''_a - (d_1 - 1) \;\le\; y_a \;\le\; x''_a + (d_1 - 1) \quad \text{for all } a \in A. \tag{7}$$

In summary, x' can be written as a convex combination of unsplittable flows y such that, due to (6) and (7),

$$x'_a - (d_1 - d_k) \;\le\; y_a \;\le\; x'_a + (d_1 - d_k) \quad \text{for all } a \in A.$$

This concludes the proof. □

It is not difficult to see that there is an efficient algorithm that computes the convex combinations in Theorem 4 and Proposition 2. We refer to [12,16] for further details. Moreover, analogously to [16], Theorem 4 can be slightly strengthened:

Corollary 4 *If $d_k \mid d_{k-1} \mid \dots \mid d_1$, then any flow x can be written as a convex combination of unsplittable flows such that each such unsplittable flow y satisfies*

$$x_a - d_{\max} \;\le\; y_a \;\le\; x_a + \max_{i : a \in P_i^y} d_i \quad \textit{for all } a \in A.$$

We finally turn to the proof of Theorem 5. The basic idea is to round down the demand values such that the rounded demands satisfy the conditions of Theorem 4. More precisely, the rounded demand values are of the form $\bar{d}_i = 2^{\lfloor \log(d_i/d_{\min}) \rfloor} d_{\min}$ where $d_{\min} := \min_{i=1,\dots,k} d_i$. The flow x has to be modified accordingly in a careful way. Then, we apply Theorem 4 to the modified flow $\bar{x}$ which yields an unsplittable flow $\bar{y}$ satisfying the rounded demands. Finally, we increase flow on the paths of $\bar{y}$ to create an unsplittable flow y meeting the original demand values. Further details are provided in Algorithm 1. Theorem 5 then follows from the next lemma.

Algorithm 1:

Input : flow x on D satisfying demands $d_1, \ldots, d_k$;
Output: unsplittable flow y given by an s-t_i-path for $i = 1, \ldots, k$;
for $i = 1, \ldots, k$, set $\bar{d}_i := d_{\min} \cdot 2^{\lfloor \log(d_i/d_{\min}) \rfloor}$;
compute a flow $\bar{x}$ satisfying demands $\bar{d}_1, \ldots, \bar{d}_k$ with $\frac{x_a}{2} \leq \bar{x}_a \leq x_a$ for all $a \in A$;
apply Theorem 4 to $\bar{x}$, yielding an unsplittable flow $\bar{y}$ for demands $\bar{d}_1, \ldots, \bar{d}_k$;
return unsplittable flow y for original demands with $P_i^y = P_i^{\bar{y}}$, for $i = 1, \ldots, k$;

Lemma 4 *Algorithm 1 computes an unsplittable flow y satisfying* (5).

Proof To prove correctness of Algorithm 1, we need to argue that there is a flow $\bar{x}$ satisfying demands $\bar{d}_1, \ldots, \bar{d}_k$ with $\frac{x_a}{2} \leq \bar{x}_a \leq x_a$ for all $a \in A$.

By definition, $\bar{d}_i \leq d_i < 2\bar{d}_i$ and, thus, $d_i - \bar{d}_i < \frac{d_i}{2}$ for all $i = 1, \ldots, k$. Therefore, since the flow $\frac{x}{2}$ satisfies demands $\frac{d_1}{2}, \ldots, \frac{d_k}{2}$, the well known cut condition for network flows (see, e.g., [1]) implies that there is a flow x' satisfying demands $d_1 - \bar{d}_1, \ldots, d_k - \bar{d}_k$ with $x'_a \leq \frac{x_a}{2}$ for all $a \in A$. This implies that $\bar{x} := x - x'$ has the desired properties stated above.

It remains to show that y satisfies the lower and upper bounds (5). Applying Corollary 4 (or Theorem 4) to $\bar{x}$ yields an unsplittable flow $\bar{y}$ such that

$$\bar{x}_a - \bar{d}_{\max} \;\leq\; \bar{y}_a \;\leq\; \bar{x}_a + \bar{d}_{\max} \quad \text{for all } a \in A.$$

By construction of flow y, we obtain the lower bounds

$$y_a \;\geq\; \bar{y}_a \;\geq\; \bar{x}_a - \bar{d}_{\max} \;\geq\; \frac{x_a}{2} - d_{\max} \quad \text{for all } a \in A.$$

In order to prove the upper bounds, let d_{i_a} be the maximum demand value routed across arc $a \in A$. Corollary 4 yields $\bar{y}_a \leq \bar{x}_a + \bar{d}_{i_a}$ for all $a \in A$. Therefore,

$$y_a = \sum_{\substack{i:a\in P_i,\\ i\neq i_a}} d_i + d_{i_a} \;\leq\; 2 \sum_{\substack{i:a\in P_i,\\ i\neq i_a}} \bar{d}_i + d_{i_a} \;\leq\; 2\,\bar{x}_a + d_{i_a} \;\leq\; 2\,x_a + d_{\max}.$$

This concludes the proof. □

5 Conclusion

We conclude by pointing out interesting open problems and conjectures related to the results presented in this paper.

While the original proof of Theorem 2 in [4] comes with a *polynomial-time* algorithm for computing an unsplittable flow y satisfying (2), our proof seems to only give rise to an exponential time algorithm. The number of augmentation steps needed to turn an arbitrary unsplittable flow into an unsplittable flow y satisfying (2) is in $O\big((k+2)^{n-1}\big)$, see Appendix A for further details. We conjecture, however, that there always exists a sequence of UBP augmentation steps leading to an unsplittable flow y satisfying (2) whose length is polynomially bounded. We also conjecture that a polynomially bounded sequence of LBP augmentation steps exists in the context of Theorem 3.

With respect to the combination of upper and lower bounds discussed in Sect. 4, we conjecture that the bounds given in Theorem 5 can be strengthened as follows:

Conjecture 1 For a given flow x, there exists an unsplittable flow y such that

$$x_a - d_{\max} \;\le\; y_a \;\le\; x_a + d_{\max} \quad \text{for all } a \in A.$$

In [4], it has already been pointed out that one main application of the unsplittable flow problem is a parallel machine scheduling problem with makespan objective. In this context, Lars Rohwedder (personal communication, February 2020) pointed out a more general connection between unsplittable flows and machine scheduling. Proving Conjecture 1 in a constructive manner (i.e., giving an efficient algorithm that turns x into y), would imply a constant-factor approximation algorithm for the problem of minimizing the maximum flow-time in a setting of unrelated machines. The best known approximation factor for this problem is $O(\log n)$, due to Bansal and Kulkarni [3]. Further details can be found in Appendix B.

We conjecture that any flow x can be written as a convex combination of unsplittable flows y satisfying (2) and (3). This can be equivalently stated as follows (cf., e.g., [12]):

Conjecture 2 Given arbitrary cost $c = (c_a)_{a\in A}$ on the arcs and a (fractional) flow x, there exists an unsplittable flow y such that $c(y) \le c(x)$ and

$$x_a - d_{\max} \;\le\; y_a \;\le\; x_a + d_{\max} \quad \text{for all } a \in A.$$

This conjecture is a strengthening of a famous, yet still unresolved conjecture of Goemans (see [16]) which does not take the lower bounds on y into account. Theorem 4 implies that Conjecture 2 is true for the special case of demands that are multiples of each other (cp. Theorem 1 for the case of unit demands).

We hope that the new techniques and methods presented in Sects. 2 and 3 will turn out to be of further use and stimulate progress towards these open problems. They might also be useful in the context of k-splittable flows considered, e.g., in [9,15].

Acknowledgements We would like to thank Lars Rohwedder for pointing out the intriguing connection between unsplittable flows and scheduling. In particular, we would like to thank him for allowing us to outline his observations in Appendix B. Our thanks go to the anonymous referees for insightful comments and to one referee in particular for indicating an improvement in Appendix B. Furthermore, we would like to thank Rico Zenklusen as well as Mohammed Majthoub Almoghrabi and Philipp Warode for interesting discussions on the topic of this paper.

Funding Open Access funding enabled and organized by Projekt DEAL.

A An upper bound on the number of UBP augmentation steps

In this section we prove an upper bound on the number of UBP augmentation steps that are necessary to turn an arbitrary unsplittable flow into one that satisfies (2). To this end, we first derive an upper bound on the number of UBP augmentation steps needed to make nodes UBP-reachable.

Lemma 5 *For $1 \leq q \leq n$, let T_q be the minimum number such that for any unsplittable flow y there exists a sequence of UBP augmentation steps and corresponding unsplittable flows $y = y^0, y^1, \ldots, y^{T_q}$ with*

$$\left|\left\{v \in V \mid \exists\, \ell \in [T_q] : \text{ } v \textit{ is UBP-reachable w.r.t. } y^\ell\right\}\right| \geq q.$$

Then, $T_1 = T_2 = 0$, and for any $2 \leq q \leq n-1$ it holds that $T_{q+1} \leq (k+2)(T_q+1)$.

Proof Notice that $T_1 = T_2 = 0$ since s is UBP-reachable w.r.t. any unsplittable flow y and there is always an arc $a = (s, v) \in \delta^{\text{out}}(s)$ with $y_a \leq x_a$ such that also node v is UBP-reachable w.r.t. y. It remains to prove that $T_{q+1} \leq (k+2)(T_q+1)$ for $2 \leq q \leq n-1$.

Starting from unsplittable flow y_0, we construct a sequence of at most $(k+2)(T_q+1)$ UBP augmentation steps, structured into at most $k+2$ phases $0, 1, \ldots, k, k+1$, each consisting of at most T_q+1 UBP augmentation steps: Phase 0 consists of exactly T_q UBP augmentation steps and produces a sequence of unsplittable flows $y^0, y^1, \ldots, y^{T_q}$ with

$$X = \left\{v \in V \mid \exists\, \ell \in [T_q] : \text{ } v \text{ is UBP-reachable w.r.t. } y^\ell\right\} \quad \text{and} \quad |X| \geq q;$$

by definition, $s \in X$. If $|X| \geq q+1$, we are done. Otherwise, as soon as some node in $V \setminus X$ becomes UBP-reachable w.r.t. an unsplittable flow occurring in one of the next k phases $1, \ldots, k$, we are done. In what follows, we thus assume that this does not happen, and construct these phases in such a way that some node $w \in V \setminus X$ then becomes UBP-reachable in phase $k+1$.

Notice that, as long as only nodes $v \in X$ become UBP-reachable, commodities are only rerouted along s-v-paths lying within X, and no flow is ever added to an arc that does not lie within X.

For $i = 1, \ldots, k$, the purpose of phase i is to ensure that commodity i is routed along a path that does *not* contain an arc in $\delta^{\text{in}}(X)$. Phase i works as follows. Let P_i be commodity i's path at the beginning of phase i, and let v_i be the last node on P lying in X. Since we assume that no node in $V \setminus X$ becomes UBP-reachable in phase i, we can choose a sequence of T_q UBP augmentation steps such that all nodes in X eventually become UBP-reachable. In particular, node v_i becomes UBP-reachable along some s-v_i-path Q after at most T_q UBP augmentation steps. At this point, we terminate the phase with one last UBP augmentation step that reroutes commodity i along Q.

Thus, when phase k terminates with unsplittable flow $\bar{y}$, no commodity is routed along a path containing an arc in $\delta^{\text{in}}(X)$. In particular, $\bar{y}\big(\delta^{\text{in}}(X)\big) = 0$ such that

$$\bar{y}\big(\delta^{\text{out}}(X)\big) = \sum\nolimits_{i:t_i \in V \setminus X} d_i \leq x\big(\delta^{\text{out}}(X)\big).$$

Thus, there is an arc $a = (v, w) \in \delta^{\text{out}}(X)$ with $\bar{y}_a \leq x_a$. Phase $k+1$ then consists of a sequence of at most T_q UBP augmentation steps until v or some node in $V \setminus X$ becomes UBP-reachable. As soon as v becomes UBP-reachable along an s-v-path Q, also w is UBP-reachable along the s-w-path that we obtain by adding arc $a = (v, w)$ to Q. □

Corollary 5 *Using the terminology of Lemma 5, $T_n \in O\big((k+2)^{n-2}\big)$. In particular, using the general strategy described in Sect. 2, any unsplittable flow can be turned into one that satisfies the arc-wise upper bounds (2) by a sequence of at most $O\big((k+2)^{n-1}\big)$ UBP augmentation steps.*

Notice that the upper bound $O\big((k+2)^{n-1}\big)$ is asymptotically much smaller than the trivial upper bound given by the number of unsplittable flows, which may be as large as $2^{k(n-2)}$ since there can be up to 2^{n-2} different s-t_i-path for each commodity $i = 1, \dots, k$.

B Unsplittable flows and scheduling

As mentioned in Sect. 5, proving Conjecture 1 via an efficient algorithm would imply a constant-factor approximation algorithm for the problem of minimizing the maximum flow-time in a setting with unrelated machines. In what follows, we describe this connection between unsplittable flows and the maximum flow-time problem, which is due to Lars Rohwedder (personal communication, February 2020). An anonymous referee pointed out a slight improvement that yields performance ratio 3, as described below.

We are given a set J of n jobs and a set M of m machines. Each job j is specified by its release date r_j and a processing time p_j, which does not depend on the chosen machine. We assume the release times to be distinct, i.e., $r_1 < \cdots < r_n$, which can be achieved by permitting infinitesimally small perturbations. We consider the non-preemptive case where jobs cannot be interrupted once started. Furthermore, each job j can only be processed on a restricted subset of machines $M_j \subseteq M$. The objective is to minimize the maximum flow-time of all jobs, where the flow-time of job j is defined as the amount of time job j spends in the system, i.e., its completion time minus its release date.

Let OPT be the maximum flow-time of an optimal schedule of the jobs. To find a schedule such that the maximum flow-time is provably within $O(OPT)$, we proceed as follows:

(S1) Solve the following linear programming relaxation of the problem:

$$\min \quad \overline{\text{OPT}}$$

$$\text{s.t.} \quad \sum_{j: r_j \in [t,t']} x_{ij} \;\leq\; (t'-t) + \overline{\text{OPT}} \qquad \text{for all } i \in M, t, t' \in \{r_1, \ldots, r_n\}, t \leq t' \tag{8}$$

$$\sum_{i \in M} x_{ij} \;=\; p_j \quad \text{for all } j \in J, \tag{9}$$

$$x_{ij} \;\geq\; 0 \quad \text{for all } j \in J, i \in M, \tag{10}$$

$$x_{ij} \;=\; 0 \quad \text{for all } j \in J, i \notin M_j. \tag{11}$$

Here, variable x_{ij} for $j \in J$ and $i \in M$ corresponds to the amount of time job j is being processed on machine i. For Constraint (8), notice that any job released during the interval $[t, t']$ must be completed by time $t' + \overline{\text{OPT}}$. It suffices to consider intervals $[t, t']$ such that t and t' are release dates of jobs as this corresponds to the tightest constraints. Constraint (9) ensures that each job is completely processed. Constraints (10) and (11) force values x_{ij} to be non-negative, with the possibility of being positive only if job j may be processed on machine i, i.e., only if $i \in M_j$. We refer to [3] for more details.
Since $\overline{\text{OPT}}$ is the optimal value of the relaxed version, we have $\overline{\text{OPT}} \leq \text{OPT}$.

(S2) Construct a digraph $D = (V, A)$ as follows (see Fig. 8 for an illustration): Add a single source node s and, for each job $j \in J$, a sink node t_j with demand $d_j := p_j$, which is the processing time of job j. For each machine $i \in M$, we add one copy of nodes $r_1^i, \ldots, r_n^i$.
Then, add arcs (s, r_1^i), $(r_1^i, r_2^i), \ldots, (r_{n-1}^i, r_n^i)$, and, for each job $j \in J$, add arc (r_j^i, t_j) if and only if $i \in M_j$. Notice that, for each job $j \in J$ and each machine $i \in M_j$, there exists a unique path using nodes $r_1^i, \ldots, r_j^i$.

(S3) The fractional solution computed in (S1) can now be translated into a fractional flow $x \in \mathbb{R}^A_{\geq 0}$ satisfying the demands in the digraph constructed in Step (S2). For each job $j \in J$ and each machine $i \in M_j$, we route the amount x_{ij} along the unique path using nodes $r_1^i, \ldots, r_j^i$. Notice that for job $j \in J$ with $j \neq 1$, machine $i \in M$, and arc $a = (r_{j-1}^i, r_j^i)$,

$$x(a) \;=\; \sum_{\ell \geq j} x_{i\ell}. \tag{12}$$

For the sake of the argument, we assume that there is an efficient algorithm that, for a given flow x, always produces an unsplittable flow y with the properties stated in Conjecture 1. For the case of the scheduling digraph illustrated in Fig. 8, this unsplittable flow y satisfies

$$x(a) - p_{\max} \;\leq\; y(a) \;\leq\; x(a) + p_{\max} \quad \text{for all arcs } a, \tag{13}$$

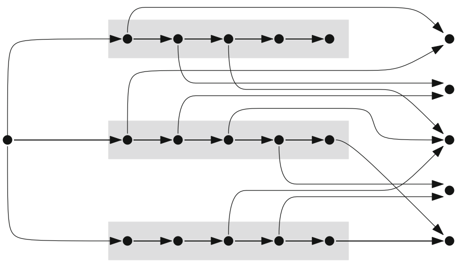

Fig. 8 We are given a set $J = \{j_1, \dots, j_5\}$ of five jobs and a set $M = \{m_1, m_2, m_3\}$ of three machines. Each job can only be processed on a restricted number of machines: $M_1 = \{m_1, m_2\}$, $M_2 = \{m_1, m_2\}$, $M_3 = \{m_1, m_2, m_3\}$, $M_4 = \{m_2, m_3\}$, $M_5 = \{m_2, m_3\}$. The construction of the corresponding digraph, as explained in (S2), is illustrated above

where $p_{\max}$ is the maximum processing time of jobs in J. The unsplittable flow y can be naturally translated into an assignment $(y_{ij})_{i \in M, j \in J}$ of jobs to machines: job j is assigned to machine $i \in M_j$, if and only if demand p_j is routed along nodes $r_1^i, \dots, r_j^i$. Then, for all $i \in M$, $j, k \in \{1, \dots, n\}$, $j \leq k$,

$$\begin{aligned}
\sum_{r_\ell \in [r_j, r_k]} y_{i\ell} &\overset{(12)}{=} y\big((r_{j-1}^i, r_j^i)\big) - y\big((r_k^i, r_{k+1}^i)\big) \\
&\overset{(13)}{\leq} \Big(x\big((r_{j-1}^i, r_j^i)\big) + p_{\max}\Big) - \Big(x\big((r_k^i, r_{k+1}^i)\big) - p_{\max}\Big) \\
&\overset{(12)}{=} \sum_{r_\ell \in [r_j, r_k]} x_{i\ell} + 2p_{\max} \\
&\overset{(12)}{\leq} \big(r_k - r_j\big) + \overline{\mathrm{OPT}} + 2p_{\max} \\
&\overset{(S1)}{\leq} \big(r_k - r_j\big) + 3 \cdot \mathrm{OPT},
\end{aligned}$$

where we use the fact that the maximum processing time $p_{\max}$ is a lower bound on OPT. We conclude that the maximum flow-time of distribution $(y_{ij})_{i \in M, j \in J}$ is at most $3 \cdot \mathrm{OPT}$ which means that it is within $O(\mathrm{OPT})$.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall Inc, Upper Saddle River (1993)
2. Baier, G., Köhler, E., Skutella, M.: On the k-splittable flow problem. Algorithmica **42**, 231–248 (2005)
3. Bansal, N., Kulkarni, J.: Minimizing flow-time on unrelated machines. In: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, pp. 851–860 (2015)
4. Dinitz, Y., Garg, N., Goemans, M.X.: On the single source unsplittable flow problem. Combinatorica **19**, 17–41 (1999)
5. Du, J., Kolliopoulos, S.: Implementing approximation algorithms for the single-source unsplittable flow problem. J. Exp. Algorithmics **10**(2), 3 (2005)
6. Ford, L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press (1962)
7. Kleinberg, J.M.: Approximation algorithms for disjoint paths problems. PhD thesis, M.I.T (1996)

8. Koch, R., Skutella, M., Spenke, I.: Maximum k-splittable s, t-flows. Theory Comput. Syst. **43**, 56–66 (2008)
9. Kolliopoulos, S.G.: Minimum-cost single-source 2-splittable flow. Inf. Process. Lett. **94**, 15–18 (2005)
10. Kolliopoulos, S.G.: Edge-disjoint paths and unsplittable flow. In: Gonzalez, T.F. (ed.) Handbook of Approximation Algorithms and Metaheuristics. Chapman and Hall/CRC (2007)
11. Kolliopoulos, S.G., Stein, C.: Approximation algorithms for single-source unsplittable flow. SIAM J. Comput. **31**, 919–946 (2002)
12. Martens, M., Salazar, F., Skutella, M.: Representing single source multicommodity flows as convex combinations of unsplittable flows. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) Algorithms – ESA '07. Lecture Notes in Computer Science, vol. 4698, pp. 395–406. Springer (2007)
13. Raghavan, P.: Probabilistic construction of deterministic algorithms: approximating packing integer programs. J. Comput. Syst. Sci. **37**, 130–143 (1988)
14. Raghavan, P., Thompson, C.D.: Randomized rounding: a technique for provably good algorithms and algorithmic proofs. Combinatorica **7**, 365–374 (1987)
15. Salazar, F., Skutella, M.: Single-source k-splittable min-cost flows. Oper. Res. Lett. **37**, 71–74 (2009)
16. Skutella, M.: Approximating the single source unsplittable min-cost flow problem. Math. Program. **91**, 493–514 (2002)
17. Williamson, D.P.: Network Flow Algorithms. Cambridge University Press (2019)