

On Routing Disjoint Paths in Bounded Treewidth Graphs

Alina Ene¹, Matthias Mnich^{*2}, Marcin Pilipczuk^{†3}, and
Andrej Risteski⁴

1 Department of Computer Science and DIMAP, University of Warwick,
Warwick, United Kingdom

A.Ene@dcs.warwick.ac.uk

2 University of Bonn, Bonn, Germany

mmnich@uni-bonn.de

3 Institute of Informatics, University of Warsaw, Warsaw, Poland

malcin@mimuw.edu.pl

4 Department of Computer Science, Princeton University, Princeton, USA

risteski@princeton.edu

Abstract

We study the problem of routing on disjoint paths in bounded treewidth graphs with both edge and node capacities. The input consists of a capacitated graph G and a collection of k source-destination pairs $\mathcal{M} = \{(s_1, t_1), \dots, (s_k, t_k)\}$. The goal is to maximize the number of pairs that can be routed subject to the capacities in the graph. A routing of a subset \mathcal{M}' of the pairs is a collection \mathcal{P} of paths such that, for each pair $(s_i, t_i) \in \mathcal{M}'$, there is a path in \mathcal{P} connecting s_i to t_i . In the Maximum Edge Disjoint Paths (MaxEDP) problem, the graph G has capacities $\text{cap}(e)$ on the edges and a routing \mathcal{P} is *feasible* if each edge e is in at most $\text{cap}(e)$ of the paths of \mathcal{P} . The Maximum Node Disjoint Paths (MaxNDP) problem is the node-capacitated counterpart of MaxEDP.

In this paper we obtain an $\mathcal{O}(r^3)$ approximation for MaxEDP on graphs of treewidth at most r and a matching approximation for MaxNDP on graphs of pathwidth at most r . Our results build on and significantly improve the work by Chekuri et al. [ICALP 2013] who obtained an $\mathcal{O}(r \cdot 3^r)$ approximation for MaxEDP.

1998 ACM Subject Classification G.2.2 Graph algorithms

Keywords and phrases Algorithms and data structures, disjoint paths, treewidth

Digital Object Identifier 10.4230/LIPIcs.SWAT.2016.15

1 Introduction

In this paper, we study disjoint paths routing problems on bounded treewidth graphs. In this setting, we are given an undirected capacitated graph G and a collection of source-destination pairs $\mathcal{M} = \{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\}$. The goal is to select a maximum-sized subset $\mathcal{M}' \subseteq \mathcal{M}$ of the pairs that can be *routed* subject to the capacities in the graph. More precisely, a routing of \mathcal{M}' is a collection \mathcal{P} of paths such that, for each pair $(s_i, t_i) \in \mathcal{M}'$, there is a path in \mathcal{P} connecting s_i to t_i . In the Maximum Edge Disjoint Paths (MaxEDP)

* Research partially supported by ERC Starting Grant 306465 (BeyondWorstCase).

† Research done while at University of Warwick, partially supported by DIMAP and by Warwick-QMUL Alliance in Advances in Discrete Mathematics and its Applications.



problem, the graph G has capacities $\text{cap}(e)$ on the edges and a routing \mathcal{P} is *feasible* if each edge e is in at most $\text{cap}(e)$ of the paths of \mathcal{P} . The Maximum Node Disjoint Paths (MaxNDP) problem is the node-capacitated counterpart of MaxEDP.

Disjoint paths problems are fundamental problems with a long history and significant connections to optimization and structural graph theory. The decision versions of MaxEDP and MaxNDP ask whether all of the pairs can be routed subject to the capacities. Karp [18] showed that, when the number of pairs is part of the input, the decision problem is NP-complete. In undirected graphs, MaxEDP and MaxNDP are solvable in polynomial time when the number of pairs is constant; this is a deep result of Robertson and Seymour [22] that builds on several fundamental structural results from their graph minors project.

In this paper, we consider the optimization problems MaxEDP and MaxNDP when the number of pairs are part of the input. These problems are NP-hard and the main focus in this paper is on approximation algorithms for these problems in bounded treewidth graphs. Although they may appear to be quite specialized at first, MaxEDP and MaxNDP on capacitated graphs of small treewidth capture a surprisingly rich class of problems; in fact, as shown by Garg, Vazirani, and Yannakakis [16], these problems are quite interesting and general even on trees.

MaxEDP and MaxNDP have received considerable attention, leading to several breakthroughs both in terms of approximation algorithms and hardness results. MaxEDP is APX-hard even in edge-capacitated trees [16], whereas the decision problem is trivial on trees; thus some of the hardness of the problem stems from having to select a subset of the pairs to route. Moreover, by subdividing the edges, one can easily show that MaxNDP generalizes MaxEDP in capacitated graphs. However, node capacities pose several additional technical challenges and extending the results for MaxEDP to MaxNDP is far from immediate even in restricted graph classes and our understanding of MaxNDP is more limited.

In general graphs, the best approximation for MaxEDP and MaxNDP is an $\mathcal{O}(\sqrt{n})$ approximation [4, 19], where n is the number of nodes, whereas the best hardness for undirected graphs is only $\Omega((\log n)^{1/2-\epsilon})$ [2]. Bridging this gap is a fundamental open problem that seems quite challenging at the moment. There have been several breakthrough results on a relaxed version of these problems where congestion is allowed¹. This line of work has culminated with a polylog(n) approximation with congestion 2 for MaxEDP [14] and congestion 51 for MaxNDP [6]. In addition to the routing results, this work has led to several significant insights into the structure of graphs with large treewidth and to several surprising applications [5].

Most of the results for routing on disjoint paths use a natural multi-commodity flow relaxation as a starting point. A well-known integrality gap instance due to Garg et al. [16] shows that this relaxation has an integrality gap of $\Omega(\sqrt{n})$, and this is the main obstacle for improving the $\mathcal{O}(\sqrt{n})$ approximation in general graphs. The integrality gap example is an instance on an $n \times n$ grid that exploits a topological obstruction in the plane that prevents a large integral routing (see Fig. 2). Since an $n \times n$ grid has treewidth $\Theta(\sqrt{n})$, it suggests the following natural and tantalizing conjecture that was asked by Chekuri et al. [9].

► **Conjecture 1** ([9]). *The integrality gap of the standard multi-commodity flow relaxation for MaxEDP/MaxNDP is $\Theta(r)$ with congestion 1, where r is the treewidth of the graph.*

Recently, Chekuri et al. [10] showed that MaxEDP admits an $\mathcal{O}(r \cdot 3^r)$ approximation on graphs of treewidth at most r . This is the first approximation for the problem that is

¹ A collection of paths has an *edge* (resp. *node*) *congestion* of c if each edge (resp. node) is in at most $c \cdot \text{cap}(e)$ (resp. $c \cdot \text{cap}(v)$) paths.

independent of n and k , and the first step towards resolving the conjecture. One of the main questions left open by the work of Chekuri et al. [10]—that was explicitly asked by them—is whether the exponential dependency on the treewidth is necessary. In this paper, we address this question and we make a significant progress towards resolving Conjecture 1.

► **Theorem 2.** *The integrality gap of the multi-commodity flow relaxation is $\mathcal{O}(r^3)$ for MaxEDP in edge-capacitated undirected graphs of treewidth at most r . Moreover, there is a polynomial time algorithm that, given a tree decomposition of G of width at most r and a fractional solution to the relaxation of value OPT, constructs an integral routing of size $\Omega(\text{OPT}/r^3)$.*

As mentioned above, MaxNDP in node-capacitated graphs is more general than MaxEDP and it poses several additional technical challenges. In this paper, we give an $\mathcal{O}(r^3)$ approximation for MaxNDP on graphs of pathwidth at most r with arbitrary node capacities. This is the first approximation guarantee for MaxNDP that is independent of n and it improves the $\mathcal{O}(r \log r \log n)$ approximation of Chekuri et al. [9].

► **Theorem 3.** *The integrality gap of the multi-commodity flow relaxation is $\mathcal{O}(r^3)$ for MaxNDP in node-capacitated undirected graphs of pathwidth at most r . Moreover, there is a polynomial time algorithm that, given a path decomposition of G of width at most r and a fractional solution to the relaxation of value OPT, constructs an integral routing of size $\Omega(\text{OPT}/r^3)$.*

The study of routing problems in bounded treewidth graphs is motivated not only by the goal of understanding the integrality gap of the multi-commodity flow relaxation but also by the broader goal of giving a more refined understanding of the approximability of routing problems in undirected graphs. Andrews et al. [2] have shown that MaxEDP and MaxNDP in general graphs cannot be approximated within a factor better than $(\log n)^{\Omega(1/c)}$ even if we allow a constant congestion $c \geq 1$. Thus in order to obtain constant factor approximations one needs to use additional structure. However, this seems challenging with our current techniques and there are only a handful of results in this direction.

One of the main obstacles for obtaining constant factor approximations for disjoint paths problems is that most approaches rely on a powerful pre-processing step that reduces an arbitrary instance of MaxEDP/MaxNDP to a much more structured instance in which the terminals² are *well-linked*. This reduction is achieved using the well-linked decomposition technique of Chekuri, Khanna, and Shepherd [7], which necessarily leads to an $\Omega(\log n)$ loss even in very special classes of graphs such as bounded treewidth graphs. Chekuri, Khanna, and Shepherd [8] showed that the well-linked decomposition framework can be bypassed in planar graphs, leading to an $\mathcal{O}(1)$ approximation for MaxEDP with congestion 4 (the congestion was later improved by Séguin-Charbonneau and Shepherd [24] from 4 to 2). This result suggests that it may be possible to obtain constant factor approximations with constant congestion for much more general classes of graphs. In particular, Chekuri et al. [10] conjecture that this is the case for the class of all minor-free graphs.

► **Conjecture 4** ([10]). *Let \mathcal{G} be any proper minor-closed family of graphs. Then the integrality gap of the multi-commodity flow relaxation for MaxEDP is at most a constant $c_{\mathcal{G}}$ when congestion 2 is allowed.*

² The vertices participating in the pairs \mathcal{M} are called *terminals*.

A natural approach is to attack Conjecture 4 using the structure theorem for minor-free graphs given by Robertson and Seymour [21, 23] that asserts that every such graph admits a tree decomposition where the size of every adhesion (the intersection of neighboring bags) is bounded, and after turning the adhesions into cliques, every bag induces a structurally simpler graph: one of bounded genus, with potentially a bounded number of apices and vortices. Thus in some sense, in order to resolve Conjecture 4, one needs to understand the base graph class (bounded genus graphs with apices and vortices) and how to tackle bounded width tree decompositions.

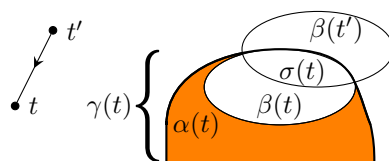
The recent work of Chekuri et al. [10] has made a significant progress toward resolving Conjecture 4 by providing a toolbox for the latter issue, and the only ingredient that is still missing is an algorithm for planar and bounded genus graphs with a constant number of vortices (in the disjoint paths setting, apices are very easy to handle). However, one of the main drawbacks of their approach is that it leads to approximation guarantees that are *exponential* in the treewidth. Our work strengthens the approach of Chekuri et al. [10] and it gives a much more graceful *polynomial* dependence in the approximation ratio.

► **Theorem 5.** *Let \mathcal{G} be a minor-closed class of graphs such that the integrality gap of the multi-commodity flow relaxation is α with congestion β . Let \mathcal{G}_ℓ be the class of graphs that admit a tree decomposition where, after turning all adhesions into cliques, each bag induces a graph from \mathcal{G} , and each adhesion has size at most ℓ . Then the integrality gap of the relaxation for the class \mathcal{G}_ℓ is $\mathcal{O}(\ell^3) \cdot \alpha$ with congestion $\beta + 3$.*

We also revisit the well-linked decomposition framework of Chekuri et al. [7] and we ask whether the $\Omega(\log n)$ loss is necessary for very structured graph classes. For bounded treewidth graphs, we give a well-linked decomposition framework that reduces an arbitrary instance of MaxEDP to node-disjoint instances of MaxEDP that are *well-linked*. The loss in the approximation for our decomposition is only $\mathcal{O}(r^3)$, which improves the guarantee of $\mathcal{O}(\log r \log n)$ from Chekuri et al. [9] when r is much smaller than n .

It is straightforward to obtain the improved well-linked decomposition from our algorithm for MaxEDP. Nevertheless, we believe it is beneficial to have such a well-linked decomposition, given that well-linked decompositions are one of the technical tools at the heart of the recent algorithms for routing on disjoint paths, integral concurrent flows [3], and flow and cut sparsifiers [13]. In particular, we hope that such a well-linked decomposition will have applications to finding flow and cut sparsifiers with Steiner nodes for bounded treewidth graphs. A sparsifier for a graph G with k source-sink pairs is a significantly smaller graph H containing the terminals (and potentially other vertices, called Steiner nodes) that approximately preserves multi-commodity flows or cuts between the terminals. Such sparsifiers have been extensively studied and several results are known both in general graphs and in bounded treewidth graphs (see Andoni et al. [1] and references therein).

A different question one could ask for problems in bounded treewidth graphs is whether additional computational power beyond polynomial-time running time can help with MaxEDP or MaxNDP. It is a standard exercise to design an $n^{\mathcal{O}(r)}$ -time dynamic programming algorithm (i.e., polynomial for every constant r) for MaxNDP in uncapacitated graphs of treewidth r , while the aforementioned results on hardness of MaxEDP in capacitated trees [16] rule out similar results for capacitated variants. Between the world of having r as part of the input, and having r as a fixed constant, lies the world of *parameterized complexity*, that asks for algorithms (called *fixed-parameter algorithms*) with running time $f(r) \cdot n^c$, where f is any computable function, and c is a constant independent of the parameter. It is natural to ask whether allowing such running time can lead to better approximation algorithms. As a



■ **Figure 1** Notations used for a node t with parent t' in a tree decomposition (\mathcal{T}, β) . The shaded part defines $\alpha(t)$.

first step towards resolving this question, we show a hardness for MaxNDP parameterized by *treedepth*, a much more restrictive graph parameter than treewidth (cf. [20]).

► **Theorem 6.** *MaxNDP parameterized by the treedepth of the input graph is $W[1]$ -hard, even with unit capacities.*

Consequently, the existence of an *exact* fixed-parameter algorithm is highly unlikely. We remark that our motivation for the choice of treedepth as parameter stems from the observation that a number of algorithms using the Sherali-Adams hierarchy to approximate a somewhat related problem of **Nonuniform Sparsest Cut** in bounded treewidth graphs [12, 17] in fact implicitly uses a rounding scheme based on treedepth rather than treewidth.

Due to space constraints, we defer the proof of Theorem 6 to the full version of this paper.

2 Preliminaries

Tree and path decompositions. In this paper all tree decompositions are rooted; that is, a tree decomposition of a graph G is a pair (\mathcal{T}, β) where \mathcal{T} is a rooted tree and $\beta : V(\mathcal{T}) \rightarrow 2^{V(G)}$ is a mapping such that (i) for every $e \in E(G)$ there is a node $t \in V(\mathcal{T})$ with $e \subseteq \beta(t)$, and (ii) for every $v \in V(G)$ the set $\{t \mid v \in \beta(t)\}$ is non-empty and connected in \mathcal{T} .

For a node $t \in V(\mathcal{T})$, we call the set $\beta(t)$ the *bag* at node t , while for an edge $st \in E(\mathcal{T})$, the set $\beta(t) \cap \beta(s)$ is called an *adhesion*. For a non-root node $t \in V(\mathcal{T})$, by $\text{parent}(t)$ we denote the parent of t , and by $\sigma(t) := \beta(t) \cap \beta(\text{parent}(t))$ the adhesion on the edge to the parent of t , called henceforth *the parent adhesion*; for the root node $t_0 \in V(\mathcal{T})$ we put $\sigma(t_0) = \emptyset$. For two nodes $s, t \in V(\mathcal{T})$, we denote by $s \preceq t$ if s is a descendant of t , and put $\gamma(t) := \bigcup_{s \preceq t} \beta(s)$, $\alpha(t) := \gamma(t) \setminus \sigma(t)$, and $G(t) := G[\gamma(t)] \setminus E(G[\sigma(t)])$.

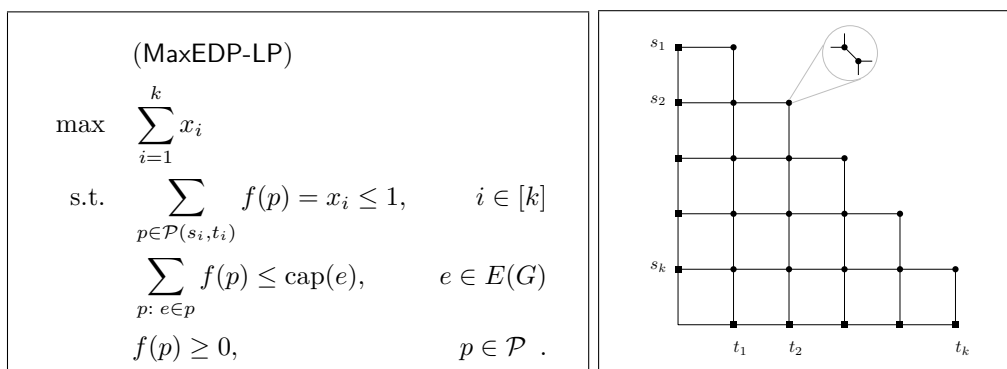
A *torso* at node t is a graph obtained from $G[\beta(t)]$ by turning every adhesion for an edge incident to t into a clique.

We say that (A, B) is a *separation* in G if $A \cup B = V(G)$ and there does not exist an edge of G with an endpoint in $A \setminus B$ and the other endpoint in $B \setminus A$. We use the following well-known property of a tree decomposition.

► **Lemma 7** ([15, Lemma 12.3.1]). *Let (\mathcal{T}, β) be a tree decomposition of a graph G . Then for each $t \in V(\mathcal{T})$ the pair $(\gamma(t), V(G) \setminus \alpha(t))$ is a separation of G , and $\gamma(t) \cap (V(G) \setminus \alpha(t)) = \sigma(t)$.*

A *path decomposition* is a tree decomposition where \mathcal{T} is a path, rooted at one of its endpoints.

The width of a tree or path decomposition (\mathcal{T}, β) is defined as $\max_t |\beta(t)| - 1$. To ease the notation, we will always consider decompositions of width *less* than r , for some integer r , so that every bag is of size at most r .



■ **Figure 2** The multi-commodity flow relaxation for MaxEDP. The instance on the right is the $\Omega(\sqrt{n})$ integrality gap example for MaxEDP with unit edge capacities [16]. Any integral routing routes at most one pair whereas there is a multi-commodity flow that sends $1/2$ units of flow for each pair (s_i, t_i) along the canonical path from s_i to t_i in the grid.

Problem definitions. The input to MaxEDP is an undirected graph G with edge capacities $\text{cap}(e) \in \mathbb{Z}_+$ and a collection $\mathcal{M} = \{(s_1, t_1), \dots, (s_k, t_k)\}$ of vertex pairs. A *routing* for a subset $\mathcal{M}' \subseteq \mathcal{M}$ is a collection \mathcal{P} of paths in G such that, for each pair $(s_i, t_i) \in \mathcal{M}'$, \mathcal{P} contains a path connecting s_i to t_i . The routing is *feasible* if every edge e is in at most $\text{cap}(e)$ paths. In the Maximum Edge Disjoint Paths problem (MaxEDP), the goal is to maximize the number of pairs that can be feasibly routed. The Maximum Node Disjoint Paths problem (MaxNDP) is the node-capacitated variant of MaxEDP in which each node v has a capacity $\text{cap}(v)$ and in a feasible routing each node appears in at most $\text{cap}(v)$ paths.

We refer to the vertices participating in the pairs \mathcal{M} as *terminals*. It is convenient to assume that \mathcal{M} form a matching on the terminals; this can be ensured by making several copies of a terminal and attaching them as leaves.

Multicommodity flow relaxation. We use the following standard multicommodity flow relaxation for MaxEDP (there is an analogous relaxation for MaxNDP). We use $\mathcal{P}(u, v)$ to denote the set of all paths in G from u to v , for each pair (u, v) of nodes. Since the pairs \mathcal{M} form a matching, the sets $\mathcal{P}(s_i, t_i)$ are pairwise disjoint. Let $\mathcal{P} = \bigcup_{i=1}^k \mathcal{P}(s_i, t_i)$. The LP has a variable $f(p)$ for each path $p \in \mathcal{P}$ representing the amount of flow on p . For each pair $(s_i, t_i) \in \mathcal{M}$, the LP has a variable x_i denoting the total amount of flow routed for the pair (in the corresponding IP, x_i denotes whether the pair is routed or not). The LP imposes the constraint that there is a flow from s_i to t_i of value x_i . Additionally, the LP has capacity constraints that ensure that the total amount of flow on paths using a given edge (resp. node for MaxNDP) is at the capacity of the edge (resp. node).

It is well-known that the relaxation MaxEDP-LP can be solved in polynomial time, since there is an efficient separation oracle for the dual (alternatively, one can write a compact relaxation). Let (f, \mathbf{x}) denote a feasible solution to MaxEDP-LP for an instance (G, \mathcal{M}) of MaxEDP. For each vertex v , let $x(v)$ denote the *marginal value* of v in the multi-commodity flow f ; thus, $x(v)$ is the amount of flow routed for each terminal v .

3

 Algorithm for MaxEDP in Bounded Treewidth Graphs

We give a polynomial time algorithm for MaxEDP that achieves an $\mathcal{O}(r^3)$ approximation for graphs with treewidth less than r . Our algorithm builds on the work of Chekuri et al. [10],

and it improves their approximation guarantee from $\mathcal{O}(r \cdot 3^r)$ to $\mathcal{O}(r^3)$. We use the following routing argument as a building block.

► **Proposition 8** ([11, Proposition 3.4]). *Let (G, \mathcal{M}) be an instance of MaxEDP and let (f, \mathbf{x}) be a feasible fractional solution for the instance. If there is a set $S \subseteq V(G)$, a value $\alpha \geq 1$ and a flow g that for each $v \in V(G)$ routes at least $x(v)/\alpha$ units of flow to some vertex in S , then there is an integral routing of at least $\binom{|f|}{36\alpha|S|}$ pairs.*

We will later apply Proposition 8 by letting S be a subset of a bag in a tree decomposition of G .

Our starting point is a tree decomposition (\mathcal{T}, β) for G of width less than r and a fractional solution (f, \mathbf{x}) to the multicommodity flow relaxation for MaxEDP given in Section 2, that is, the flow f routes $\mathbf{x}(v)$ units of flow for each vertex $v \in V$. We let $|f|$ denote the total amount of flow routed by f , i.e., $|f| = \binom{1}{2} \sum_{v \in V} \mathbf{x}(v)$.

The following definitions play a key role in our algorithm.

► **Definition 9** (Safe node). A node $t \in V(\mathcal{T})$ is *safe* with respect to (f, \mathbf{x}) if there is a second multicommodity flow g in $G(t)$ that satisfies the edge capacities of $G(t)$ and, for each vertex $z \in \gamma(t)$, g routes at least $\binom{1}{4r} \cdot \mathbf{x}(z)$ units of flow from z to the adhesion $\sigma(t)$. The node t is *unsafe* if it is not safe.

► **Definition 10** (Good node). A node $t \in V(\mathcal{T})$ is *good* with respect to (f, \mathbf{x}) if every flow path in the support of f that has an endpoint in $\gamma(t)$ also intersects $\sigma(t)$; in other words, no flow path is completely contained in $G[\alpha(t)]$. A node is *bad* if it is not good.

► **Remark.** If a node t is good then it is also safe, as shown by the following multicommodity flow g in $G(t)$. For each path p in the support of f that originates in $\gamma(t)$, let p' be the smallest prefix of p that ends at a vertex of $\sigma(t)$ (since p intersects $\sigma(t)$, there is such a prefix); we set $g(p') = f(p)$. The resulting flow g is a feasible multicommodity flow in $G(t)$ that routes $\mathbf{x}(z)$ units of flow from z to $\sigma(t)$ for each vertex $z \in \gamma(t)$. Therefore, t is safe.

Our approach is an inductive argument based on the maximum size of a parent adhesion that is bad or unsafe. More precisely, we prove the following:

► **Theorem 11.** *Let (G, \mathcal{M}) be an instance of MaxEDP and let (f, \mathbf{x}) be a fractional solution for (G, \mathcal{M}) , where f is a feasible multicommodity flow in G for \mathcal{M} with marginal values \mathbf{x} . Let (\mathcal{T}, β) be a tree decomposition for G of width less than r . Let ℓ_1 be the maximum size of a parent adhesion of an unsafe node, and let ℓ_2 be the maximum size of a parent adhesion of a bad node. There is a polynomial time algorithm that constructs an integral routing of size at least $\binom{1}{144r^3} \cdot \left(1 - \binom{1}{r}\right)^{\ell_1 + \ell_2} \cdot |f|$.*

Proof. We start with a bit of preprocessing. If $|f| = 0$, then we return an empty routing. Otherwise, the root node of \mathcal{T} is always unsafe and bad, and the integers ℓ_1 and ℓ_2 are well-defined. By considering every connected component of G independently (with inherited tree decomposition from (\mathcal{T}, β)), we assume that G is connected; note that in this step all safe or good adhesions remain safe or good for every connected component. Furthermore, we delete from (\mathcal{T}, β) all nodes with empty bags; note that the connectivity of G ensures that the nodes with non-empty bags induce a connected subtree of \mathcal{T} . In this step, the root of \mathcal{T} may have moved to a different node (the topmost node with non-empty bag), but the parent-children relation in the tree remains unchanged.

Once G is connected and no bag is empty, the only empty parent adhesion is the one for the root node. We prove Theorem 11 by induction on $\ell_1 + \ell_2 + |V(G)|$.

Base case. In the base case, we assume that $\ell_1 = \ell_2 = 0$. Since every parent adhesion of a non-root node is non-empty, that implies that the only bad node is the root t_0 , that is, every flow path in f passes through $\beta(t_0)$, which is of size at most r . By applying Proposition 8 with $S = \beta(t_0)$ and $\alpha = 1$, we construct an integral routing of size at least $\frac{1}{36r}|f| \geq \frac{1}{144r^3}|f|$.

In the inductive step, we consider two cases, depending on if $0 \leq \ell_1 < \ell_2$ or $0 < \ell_1 = \ell_2$.

Inductive step when $0 \leq \ell_1 < \ell_2$. Let $\{t_1, t_2, \dots, t_p\}$ be the topmost bad nodes of \mathcal{T} with parent adhesions of size ℓ_2 , that is, it is a minimal set of such bad nodes such that for every bad node t with parent adhesion of size ℓ_2 , there exists an $i \in \{1, \dots, p\}$ with $t \preceq t_i$. For $i = 1, \dots, p$, let f_i^{inside} be the subflow of f consisting of all paths that are completely contained in $G[\alpha(t_i)]$. Furthermore, since $\ell_1 < \ell_2$, the node t_i is safe; let g_i be the corresponding flow, i.e., a flow that routes $\frac{1}{4r}x(v)$ from every $v \in \gamma(t_i)$ to $\sigma(t_i)$ in $G(t_i)$. By applying Proposition 8, there is an integral routing \mathcal{P}_i in $G(t_i)$ that routes at least $\binom{1}{144r^2}|f_i^{\text{inside}}|$ pairs. Since the subgraphs $\{G(t_i) : 1 \leq i \leq p\}$ are edge-disjoint, we get an integral routing $\mathcal{P} := \bigcup_i \mathcal{P}_i$ of size at least $\binom{1}{144r^2} \sum_{i=1}^p |f_i^{\text{inside}}|$.

If $\sum_{i=1}^p |f_i^{\text{inside}}| > \frac{1}{r}|f|$, then we can return the routing \mathcal{P} as the desired solution. Otherwise, we drop the flows f_i^{inside} , that is, consider a flow $f' := f - \sum_{i=1}^p f_i^{\text{inside}}$. Clearly, $|f'| \geq (1 - \frac{1}{r})|f|$. Furthermore, by definition of f_i^{inside} , every node t_i is good with respect to f' . Since deleting a flow path cannot turn a good node into a bad one nor a safe node into an unsafe one, and all descendants of a good node are also good, we infer that every unsafe node with respect to f' has parent adhesion of size at most ℓ_1 , while every bad node with respect to f' has parent adhesion of size *less* than ℓ_2 . Consequently, by the induction hypothesis we obtain an integral routing of size at least $\frac{1}{144r^3} (1 - \frac{1}{r})^{\ell_1 + \ell_2 - 1} |f'| \geq \frac{1}{144r^3} (1 - \frac{1}{r})^{\ell_1 + \ell_2} |f|$.

Inductive step when $0 < \ell_1 = \ell_2$. In this case, we pick a node t° to be the lowest node of \mathcal{T} that is unsafe and has parent adhesion of size ℓ_1 . By the definition of an unsafe node and Menger's theorem, there exists a set $U \subseteq \alpha(t^\circ)$ such that $\text{cap}(\delta(U)) < \frac{1}{4r}\mathbf{x}(U)$. With a bit more care, we can extract a set U with one more property:

► **Lemma 12.** *In polynomial time we can find a set $U \subseteq \alpha(t^\circ)$ for which (i) $\text{cap}(\delta(U)) < \frac{1}{4r}\mathbf{x}(U)$, and (ii) for every non-root node t , if $\sigma(t) \subseteq U$, then $\gamma(t) \subseteq U$.*

Proof. Consider an auxiliary graph G' , obtained from $G[\gamma(t^\circ)]$ by adding a super-source s^* , linked for every $v \in \gamma(t^\circ)$ by an arc (s^*, v) of capacity $\frac{1}{4r}\mathbf{x}(v)$, and a super-sink t^* , linked for every $v \in \sigma(t^\circ)$ by an arc (v, t^*) of infinite capacity. Let U be such a set that $\delta(U \cup \{s^*\})$ is a minimum s^* - t^* cut in this graph. Clearly, since U is unsafe, $\text{cap}(\delta_{G'}(U \cup \{s^*\})) < \frac{1}{4r}\mathbf{x}(\gamma(t^\circ)) = \text{cap}(\delta_{G'}(s^*))$, so $U \neq \emptyset$. Also, $U \subseteq \alpha(t^\circ)$, as each node in $\sigma(t^\circ)$ is connected to t^* with an infinite-capacity arc.

We claim that U satisfies the desired properties. The first property is immediate:

$$\text{cap}(\delta_G(U)) = \text{cap}(\delta_{G'}(U \cup \{s^*\})) - \frac{1}{4r}\mathbf{x}(\gamma(t^\circ) \setminus U) < \frac{1}{4r}(\mathbf{x}(\gamma(t^\circ)) - \mathbf{x}(\gamma(t^\circ) \setminus U)) = \frac{1}{4r}\mathbf{x}(U).$$

For the second property, pick a non root node t with $\sigma(t) \subseteq U$. Since $\sigma(t) \subseteq U \subseteq \alpha(t^\circ)$, we have $t \preceq t^\circ$, $t \neq t^\circ$, and $\gamma(t) \subseteq \alpha(t^\circ)$. Let $U' := U \cup \gamma(t)$. By Lemma 7, $\delta_{G'}(U') \subseteq \delta_{G'}(U)$, and hence $\delta_{G'}(U' \cup \{s^*\}) \subseteq \delta_{G'}(U \cup \{s^*\})$. However, since $\delta_{G'}(U \cup \{s^*\})$ is a minimum cut, we have actually $\delta_{G'}(U') = \delta_{G'}(U)$. Since G is connected, this implies that $U = U'$, and thus $\gamma(t) \subseteq U$. As the choice of t was arbitrary, U satisfies the second property. ◀

Using the cut U , we split the graph G and the flow f into two pieces as follows. Let $G_1 = G[U]$ and $G_2 = G - U$. Let f_i be the restriction of f to G_i , i.e., the flow consisting of

only flow paths that are contained in G_i . Let \mathbf{x}_i be the marginals of f_i and let \mathcal{M}_i be the subset of \mathcal{M} consisting of all pairs (s, t) such that $\{s, t\} \subseteq V(G_i)$; note that $x_i(s) = x_i(t)$ for each pair $(s, t) \in \mathcal{M}_i$ and thus (f_i, \mathbf{x}_i) is a fractional routing for the instance (G_i, \mathcal{M}_i) . Let (\mathcal{T}, β_1) and (\mathcal{T}, β_2) be the restriction of (\mathcal{T}, β) to the vertices of G_1 and G_2 , respectively; we define mappings σ_i, γ_i , and α_i naturally. In what follows, we consider separately two instances $\mathcal{I}_i := \langle (G_i, \mathcal{M}_i), (f_i, \mathbf{x}_i), (\mathcal{T}, \beta_i) \rangle$ for $i = 1, 2$.

An important observation is the following:

► **Lemma 13.** *Every node $t \in V(\mathcal{T})$ that is good in the original instance (i.e., as a node of \mathcal{T} , with respect to (f, \mathbf{x})) is also good in \mathcal{I}_i with respect to (f_i, \mathbf{x}_i) .*

Proof. Note that every flow path in f_i is also present in f , and therefore intersects the parent adhesion of f if t is a good node in the original instance. ◀

Consequently, every node $t \in V(\mathcal{T})$ with $|\sigma(t)| > \ell_2$ is good in the instance \mathcal{I}_i , and the maximum size of a parent adhesion of a bad node in instance \mathcal{I}_i is at most ℓ_2 . Hence, both \mathcal{I}_1 and \mathcal{I}_2 satisfy the assumptions of Theorem 11 with not larger values of ℓ_1 and ℓ_2 . Furthermore, note that $|V(G_i)| < |V(G)|$ for $i = 1, 2$.

For \mathcal{I}_2 , the above reasoning allows us to simply just apply inductive step, obtaining an integral routing \mathcal{P}_2 of size at least

$$|\mathcal{P}_2| \geq \binom{1}{144r^3} \left(1 - \binom{1}{r}\right)^{\ell_1 + \ell_2} \cdot |f_2|. \quad (1)$$

For \mathcal{I}_1 , we are going to obtain a larger routing via an inductive step with better bounds.

► **Lemma 14.** *The size of the largest parent adhesion of an unsafe node in \mathcal{I}_1 is less than ℓ_1 .*

Proof. Assume the contrary, let $t \in V(\mathcal{T})$ be an unsafe adhesion with $|\sigma_1(t)| \geq \ell_1$. If $|\sigma(t)| > \ell_1$, then t is good in the original instance, and by Lemma 13 it remains good in \mathcal{I}_1 . Consequently, $|\sigma(t)| = |\sigma_1(t)| = \ell_1$; in particular, $\sigma(t) = \sigma_1(t) \subseteq U$.

By Lemma 12(ii) we have $\gamma(t) \subseteq U$. Consequently, t is safe in the original instance if and only if it is safe in \mathcal{I}_1 . Since $t \preceq t^\circ, t \neq t^\circ$, but $|\sigma(t)| = \ell_2$, by the choice of t° it holds that t is safe in the original instance, a contradiction. ◀

Lemma 14 allows us to apply the inductive step to \mathcal{I}_1 and obtain an integral routing \mathcal{P}_1 of size at least

$$|\mathcal{P}_1| \geq \binom{1}{144r^3} \left(1 - \binom{1}{r}\right)^{\ell_1 - 1 + \ell_2} \cdot |f_1|. \quad (2)$$

Let us now estimate the amount of flow lost by the separation into \mathcal{I}_1 and \mathcal{I}_2 , i.e., $g = f - f_1 - f_2$. As every flow path in g passes through $\delta(U)$, we have $|g| \leq \text{cap}(\delta(U)) < \frac{1}{4r} \mathbf{x}(U)$. Since $|f_1| + |g| \geq \frac{1}{2} \mathbf{x}(U)$ (no flow path in f_2 originates in U), we have that $|g| \leq \frac{1}{4r} \cdot 2 \cdot (|f_1| + |g|)$. Hence,

$$|g| \leq \frac{1}{2r} \cdot \left(1 - \frac{1}{2r}\right)^{-1} |f_1| \leq \frac{1}{r} |f_1|. \quad (3)$$

By putting up together (1), (2), and (3), we obtain that

$$\begin{aligned} |\mathcal{P}_1| + |\mathcal{P}_2| &\geq \frac{1}{144r^3} \left(1 - \frac{1}{r}\right)^{\ell_1 + \ell_2} \left(|f_2| + \left(1 - \frac{1}{r}\right)^{-1} |f_1|\right) \\ &\geq \frac{1}{144r^3} \left(1 - \frac{1}{r}\right)^{\ell_1 + \ell_2} (|f_2| + |f_1| + |g|) = \frac{1}{144r^3} \left(1 - \frac{1}{r}\right)^{\ell_1 + \ell_2} |f|. \end{aligned}$$

15:10 On Routing Disjoint Paths in Bounded Treewidth Graphs

This concludes the proof of Theorem 11. Since $\ell_1, \ell_2 \leq r$, while $(1 - \frac{1}{r})^{2r} = \Omega(1)$, Theorem 11 immediately implies the promised $\mathcal{O}(r^3)$ -approximation algorithm. ◀

► **Remark.** We conclude with observing that the improved approximation ratio of $\mathcal{O}(r^3)$ directly translates to the more general setting of k -sums of graph from some minor closed family \mathcal{G} , as discussed by Chekuri et al. [10]. That is, if we are able to α -approximate MaxEDP with congestion β in graphs from \mathcal{G} , we can have $\mathcal{O}(\alpha r^5)$ -approximation algorithm with congestion $(\beta + 3)$ in graphs admitting a tree decomposition of maximum adhesion size at most r , and the torso of every bag being from the class \mathcal{G} .

To see this, observe that the only place when our algorithm uses that the *bags* are of bounded size (as opposed to *adhesions*) is the base case, where all flow paths pass through the bag $\beta(t_0)$ of the root node t_0 . However, in this case we can proceed exactly as Chekuri et al. [10]: using the flow paths, move the terminals to $\beta(t_0)$, replace connected components of $G - \beta(t_0)$ with their $(r^2, 2)$ -sparsifiers, and apply the algorithm for the class \mathcal{G} . In addition to the $\mathcal{O}(r^3)$ approximation factor of our algorithm, the application of the algorithm for \mathcal{G} incurs an approximation ratio of α and congestion of β , the use of sparsifiers adds a factor of r^2 to the approximation ratio and an additive constant $+1$ to the congestion, while the terminal move adds an additional amount of 2 to the final congestion.

4 Algorithm for MaxNDP in Bounded Pathwidth Graphs

In this section we develop an $\mathcal{O}(r^3)$ -approximation algorithm for MaxNDP in graphs of *pathwidth* less than r . We follow the outline of the MaxEDP algorithm from the previous section, with few essential changes.

Most importantly, we can no longer use Proposition 8, as it refers to edge disjoint paths, and the proof of its main ingredient by Chekuri et al. [4] relies on a clustering technique that stops to work for node disjoint paths. We fix this by providing in Sect. 4.1 a node-disjoint variant of Proposition 8, using the more involved clustering approach of Chekuri et al. [7].

Then, in Sect. 4.2 we revisit step-by-step the arguments for MaxEDP, pointing out remaining differences. We remark that the use of pathwidth instead of treewidth is only essential in the inductive step for the case $\ell_1 < \ell_2$: if we follow the argument for MaxEDP for bounded-treewidth graphs, the graphs $G(t_i)$ may not be node disjoint (but they are edge disjoint), breaking the argument. Note that for bounded pathwidth graphs, there is only one such graph considered, and the issue is nonexistent.

4.1 Routing to a small adhesion in a node-disjoint setting

► **Proposition 15.** *Let (G, \mathcal{M}) be an instance of MaxNDP and let (f, \mathbf{x}) be a feasible fractional solution for the instance. Suppose that there is also a second (feasible, i.e., respecting node capacities) flow that routes at least $x(v)/\alpha$ units of flow for each v to some set $S \subseteq V$, where $\alpha \geq 1$. Then there is an integral routing of $\Omega(|f|/(\alpha|S|))$ pairs.*

Proof. Without loss of generality, we may assume that the terminals of \mathcal{M} are pairwise distinct and of degree and capacity one: we can always move a terminal from a vertex t to a newly-created degree-1 capacity-1 neighbour of t .

Let g be the second flow mentioned in the statement. In what follows, we modify and simplify the flows f and g in a number of steps. We denote by f_1, f_2, \dots and g_1, g_2, \dots flows after subsequent modification steps; for the flow f_i , by \mathbf{x}_i we denote its marginals.

Symmetrizing the flow g . In the first step, we construct flows f_1 and g_1 with the following property: for every terminal pair $(s, t) \in \mathcal{M}$, for every $v \in S$, g_1 sends the same amount of flow from s to v as from t to v . To obtain this goal, we first take the flow $g/3$, and then for every $(s, t) \in \mathcal{M}$ redirect the flow originating at s to first go along the commodity for the pair (s, t) in flow $f/(3\alpha)$ to the vertex t , and then go to S in exactly the same manner as the flow originating at t does. It is easy to see that g_1 consists of three feasible flows scaled down by at least $1/3$, thus it is feasible. Finally, we set $f_1 := f/3$, so that g_1 again sends an amount of $\mathbf{x}_1(v)/\alpha$ flow from every vertex v to S . Note that $|f_1| = |f|/3$.

Restricting to single vertex of S . To construct flows f_2 and g_2 , pick a vertex $u \in S$ that receives the most flow in g_1 . Take g_2 to be the flow g_1 , restricted only to flow paths ending in u . Then, restrict f_1 to obtain f_2 as follows: for every terminal pair $(s, t) \in \mathcal{M}$, reduce the amount of flow from s to t to α times the total amount of flow sent from s to u by g_2 ; note that, by the previous step, it is also equal α times the total amount of flow sent from t to u by g_2 . By this step, we maintain the invariant that g_2 sends $\mathbf{x}_2(v)/\alpha$ flow from every $v \in V(G)$, and we have $|f_2| \geq |f_1|/|S| \geq |f|/(3|S|)$.

Rounding to a half-integral flow. In the next step, we essentially repeat the integral rounding procedure by Chekuri et al. [4, Section 3.2]. We use the following operation as a basic step in the rounding.

► **Lemma 16** ([4, Theorem 2.1]). *Let G be a directed graph with edge capacities. Given a flow h in G that goes from set $X \subseteq V(G)$ to a single vertex $u \in V(G)$, such that for every $v \in X$ the amount of flow originating in v is $\mathbf{z}(v)$, and a vertex $v_0 \in X$ such that $\mathbf{z}(v_0)$ is not an integer, one can in polynomial time compute a flow h' in G , sending $\mathbf{z}'(v)$ amount of flow from every $v \in X$ to u , such that $|h'| \geq |h|$, $\mathbf{z}'(v) = \mathbf{z}(v)$ for every $v \in X$ where $\mathbf{z}(v)$ is an integer, and $\mathbf{z}'(v_0) = \lceil \mathbf{z}(v_0) \rceil$.*

Since a standard reduction reduces flows in undirected node-capacitated graphs to directed edge-capacitated ones³, Lemma 16 applies also to undirected graphs with node capacities.

Split g_2 into two flows h_s and h_t : for every terminal pair $(s, t) \in \mathcal{M}$, we put the flow originating in s into h_s , and the flow originating in t into h_t . We perform a sequence of modifications to the flows h_s and h_t , maintaining the invariant that the same amount of flow originates in s in h_s as in t in h_t for every $(s, t) \in \mathcal{M}$. Along the process, both h_s and h_t are feasible flows, but $h_s + h_t$ may not be.

In a single step, we pick a terminal pair $(s, t) \in \mathcal{M}$ such that the amount of flow in h_s originating in s is not integral (and stop if no such pair exists). We apply Lemma 16 separately to s in h_s and to t in h_t , obtaining flows h'_s and h'_t . Finally, if for some terminal pair (s', t') , the amount of flow originating in s' in h'_s and in t' in h'_t differ, we restrict one of the flows so that both route the same amount of flow (being the minimum of the flows routed by h'_s from s' and by h'_t from t').

Since the rounding algorithm of Lemma 16 never modifies a source that already has an integral flow, this procedure stops after at most $|\mathcal{M}|$ steps. Furthermore, if in one step the flow from s has been increased from z to $\lceil z \rceil$, the total loss of flow to other pairs is $2(\lceil z \rceil - z)$.

³ Replace every edge with two infinite-capacity arcs in both directions, and then split every vertex into two vertices, connected by an edge of capacity equal to the capacity of the vertex, with all in-edges connected to the first copy, and all out-edges connected to the second copy.

15:12 On Routing Disjoint Paths in Bounded Treewidth Graphs

Therefore, if h_s° and h_t° are the final integral flows, we have $|h_s^\circ| + |h_t^\circ| \geq (|h_s| + |h_t|)/2 = |g_2|/2 = |f_2|/\alpha \geq |f|/(3\alpha|S|)$. We define $g_3 := (h_s^\circ + h_t^\circ)/2$; note that g_3 is a feasible flow.

Clustering a node-flow-linked set. Note that for every $(s, t) \in \mathcal{M}$, the flow g_3 routes either 0 or $1/2$ flow from both s and t to u . Let \mathcal{M}' be the set of pairs with flow value $1/2$, and let X' be the set of terminals in \mathcal{M}' . Note that $|\mathcal{M}'| = |g_3|/2 \geq |f|/(6\alpha|S|)$.

Using the flow g_3 , we now find a multicommodity flow that for every $(a, b) \in X' \times X'$ routes $\frac{1}{4|X'|}$ amount of flow from a to b . First, we use a flow $\frac{1}{2}g_3$ to send, for every $a \in X'$, a tuple of $|X'|$ portions of $\frac{1}{4|X'|}$ flow each from a to u . Second, we use a reversed flow $\frac{1}{2}g_3$ to send, for every $b \in X'$, a tuple of $|X'|$ portions of $\frac{1}{4|X'|}$ flow each from u to b . For every $(a, b) \in X' \times X'$, we combine one portion sent from a to u with one portion sent from u to b to obtain the commodity from a to b . We obtain the desired multicommodity flow, and we infer that X' is $\frac{1}{4}$ -node-flow-linked. This allows us to apply the following clustering result:

► **Lemma 17** ([7, Lemma 2.7]). *If X is α -node-flow-linked in a graph G with unit node capacities, then for any $h \geq 2$ there exists a forest F in G of maximum degree $\mathcal{O}(\frac{1}{\alpha} \log h)$ such that every tree in F spans at least h nodes from X .*

Since we can assume that no capacity in G exceeds $|\mathcal{M}|$, we can replace every vertex v of capacity $\text{cap}(v)$ with its $\text{cap}(v)$ copies. To such unweighted graph G' we apply Lemma 17 for X' , $\alpha = 1/4$ and $h = 3$, obtaining a forest F' ; recall that the terminals X' are of capacity 1, thus they are kept unmodified in G' . By standard argument we split the forest F' into node-disjoint trees T'_1, T'_2, \dots, T'_p , such that every tree T'_i contains at least three, and at most $d = \mathcal{O}(1)$ terminals of X' . By projecting the trees T'_i back onto G , we obtain a sequence of trees T_1, T_2, \dots, T_p , such that every vertex $v \in V(G)$ is present in at most $\text{cap}(v)$ trees T_i . Furthermore, since terminals are of capacity one, every terminal belongs to at most one tree, and every tree T_i contains at least three and at most d terminals.

In a greedy fashion, we chose a set $\mathcal{M}'' \subseteq \mathcal{M}'$ of size at least $|\mathcal{M}'|/d^2$, such that for every tree T_i , at most one terminal pair of \mathcal{M}'' has at least one terminal in T_i . A pair $(s, t) \in \mathcal{M}''$ is *local* if both s and t lie in the same tree T_i , and *distant* otherwise. If at least half of the pairs of \mathcal{M}'' are local, we can route them along trees T_i , obtaining a desired routing of size at least $|\mathcal{M}''|/2 \geq |\mathcal{M}'|/(2d^2) = \Omega(|f|/(\alpha|S|))$ and terminate the algorithm. Otherwise, we obtain a flow g_4 as follows: for every terminal t in a distant pair in \mathcal{M}'' , we take the tree T_i it lies on, route $3/5$ amount of flow along T_i equidistributed to three arbitrarily chosen terminals t^1, t^2, t^3 on T_i from \mathcal{M}' (i.e., every terminal t^j receives $1/5$ amount of flow), and then route the flow along the flow $\frac{2}{5}g_3$ to u . Since every tree T_i routes $3/5$ amount of flow, and g_3 is a feasible flow, the flow g_4 is a feasible flow that routes $3/5$ amount of flow from every terminal of \mathcal{M}'' to u . Furthermore, since at least half terminal pairs in \mathcal{M}'' is distant, we have $|g_4| \geq \frac{1}{2} \cdot 2|\mathcal{M}''| = \Omega(|f|/(\alpha|S|))$.

Final rounding of the flow. Let X'' be the set of all terminals of \mathcal{M}'' . Since the flow g_4 routes *more than* $1/2$ amount of flow for every terminal in X'' , we can conclude with simple rounding the flow g_4 in the same manner as it is done by Chekuri et al. [4, Section 3]. Construct an auxiliary graph G' by adding a super-source s^* of infinite capacity, adjacent to all terminals of \mathcal{M}'' . Extend g_4 in the natural manner, by routing every flow path first from s^* to an appropriate terminal. The extended flow g_4 is now a single source single sink flow from s^* to u in a graph with integer capacities, thus there exists an integral flow g_5 of no smaller size: $|g_5| \geq |g_4| = \frac{3}{5}|X''| = \frac{6}{5}|\mathcal{M}''|$. Hence, for at least $1/5$ of the pairs $(s, t) \in \mathcal{M}''$, the flow g_5 routes a single unit of flow both from s and from t to u . By

combining these paths into a single path from s to t , we obtain an integral routing of size at least $\frac{1}{5}|\mathcal{M}''| = \Omega(|f|/(\alpha|S|))$. This finishes the proof of Proposition 15. \blacktriangleleft

4.2 Details of the algorithm

Equipped with Proposition 15, we can now proceed to the description of the approximation algorithm. Assume we are given an MaxNDP instance (G, \mathcal{M}) and a path decomposition (\mathcal{T}, β) of G of width less than r ; recall that \mathcal{T} rooted in one of its endpoints. Let (f, \mathbf{x}) be a fractional solution to the multicommodity flow relaxation for MaxNDP, as in Sect. 2.

The definitions of safe and good node, as well as the induction scheme, are analogous.

► **Definition 18 (Safe node).** A node $t \in V(\mathcal{T})$ is *safe* with respect to (f, \mathbf{x}) if there is a second multicommodity flow g in $G(t)$ that satisfies the node capacities of $G(t)$ and, for each vertex $z \in \gamma(t)$, g routes at least $\binom{1}{4r} \cdot \mathbf{x}(z)$ units of flow from z to the adhesion $\sigma(t)$. The node t is *unsafe* if it is not safe.

► **Definition 19 (Good node).** A node $t \in V(\mathcal{T})$ is *good* with respect to (f, \mathbf{x}) if every flow path in the support of f that has an endpoint in $\gamma(t)$ also intersects $\sigma(t)$; in other words, no flow path is completely contained in $G[\alpha(t)]$. A node is *bad* if it is not good.

► **Theorem 20.** *Let (G, \mathcal{M}) be an instance of MaxNDP and let (f, \mathbf{x}) be a fractional solution for the instance, where f is a feasible multicommodity flow in G for the pairs \mathcal{M} with marginals \mathbf{x} . Let (\mathcal{T}, β) be a path decomposition for G of width less than r . Let ℓ_1 be the maximum size of a parent adhesion of an unsafe node, and let ℓ_2 be the maximum size of a parent adhesion of a bad node. There is a constant c and a polynomial time algorithm that constructs an integral routing of size at least $\binom{1}{cr^3} \cdot \left(1 - \binom{1}{r}\right)^{\ell_1 + \ell_2} \cdot |f|$.*

Proof. As in the case of MaxEDP, we can assume that the considered graph G is connected and that no bag is empty, and thus the only empty adhesion is the parent adhesion of the root.

Base case. In the base case $\ell_1 = \ell_2 = 0$ nothing changes compared to MaxEDP: all flow paths pass through the root bag, and Proposition 15 allows to route integrally $\Omega(|f|/r)$ paths.

Inductive step when $0 \leq \ell_1 < \ell_2$. Since we are considering now a path decomposition (as opposed to tree decomposition in the previous section), there exists a single topmost bad node t° with parent adhesion of size ℓ_2 . Let f^{inside} be the subflow of f consisting of all flow paths completely contained in $G[\alpha(t^\circ)]$. Since $\ell_1 < \ell_2$, the node t° is safe, and the flow witnessing it together with Proposition 15 allows to integrally route $\Omega(|f^{\text{inside}}|/r^2)$ terminal pairs. If $|f^{\text{inside}}| > |f|/r$, then we are done. Otherwise, we drop the flow f^{inside} from f , making t° and all its descendants good (thus decreasing ℓ_2 in the constructed instance), while losing only $1/r$ fraction of the flow f , and pass the instance to an inductive step.

Inductive step when $0 < \ell_1 = \ell_2$. Here again we take t° to be the lowest node of \mathcal{T} that is unsafe and has parent adhesion of size ℓ_1 . By the definition of an unsafe node and Menger's theorem, there exists a set $U \subseteq \alpha(t^\circ)$ such that $\text{cap}(N(U)) < \frac{1}{4r} \mathbf{x}(U)$. Using the same argument as in the proof of Lemma 12, we can ensure property 12, that is that if U contains an adhesion $\sigma(t)$, it contains as well the entire set $\gamma(t)$.

As in the case of MaxEDP, we split into instances \mathcal{I}_1 and \mathcal{I}_2 by taking $G_1 = G[U]$ and $G_2 = G - N[U]$, with inherited tree decompositions from (\mathcal{T}, β) . Since all nodes with parent

adhesions of size larger than $\ell_1 = \ell_2$ are good, there are also good in instances \mathcal{I}_i (i.e., Lemma 13 holds here as well) and we can again apply the inductive step to every connected component of the instance \mathcal{I}_2 with the same values of ℓ_1 and ℓ_2 , obtaining a routing \mathcal{P}_2 of size as in (1) (with 144 replaced by a constant c).

We analyse the instance \mathcal{I}_1 , without breaking it first into connected components. That is, we argue that in \mathcal{I}_1 the value of ℓ_1 dropped, that is, all nodes t satisfying $|\sigma(t)| = |\sigma_1(t)| = \ell_1$ are safe; note that they will remain safe once we consider every connected component separately. However, this fact follows from property 12 of the set U (Lemma 12): if for some node t we have $|\sigma(t)| = |\sigma_1(t)|$, it follows that $\sigma(t) \subseteq U$ hence $\gamma(t) \subseteq U$ and the notion of safeness for t is the same in \mathcal{I}_1 and in the original instance. However, $\sigma(t) \subseteq U \subseteq \alpha(t^\circ)$ implies $t \preceq t^\circ$ and $t \neq t^\circ$, hence t is safe in the original instance.

Consequently, an application of inductive step for every connected component of \mathcal{I}_1 uses strictly smaller value of ℓ_1 , and we obtain an integral routing \mathcal{P}_1 in \mathcal{I}_1 of size as in (2) (again with 144 replaced by a constant c). The remainder of the analysis from the previous section does not change, concluding the proof of Theorem 20. ◀

References

- 1 Alexandr Andoni, Anupam Gupta, and Robert Krauthgamer. Towards $(1+\varepsilon)$ -approximate flow sparsifiers. In *Proc. SODA 2014*, pages 279–293, 2014.
- 2 Matthew Andrews, Julia Chuzhoy, Venkatesan Guruswami, Sanjeev Khanna, Kunal Talwar, and Lisa Zhang. Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs. *Combinatorica*, 30(5):485–520, 2010.
- 3 Parinya Chalermsook, Julia Chuzhoy, Alina Ene, and Shi Li. Approximation algorithms and hardness of integral concurrent flow. In *Proc. STOC 2012*, pages 689–708, 2012.
- 4 C. Chekuri, S. Khanna, and F.B. Shepherd. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory Comput.*, 2(7):137–146, 2006.
- 5 Chandra Chekuri and Julia Chuzhoy. Large-treewidth graph decompositions and applications. In *Proc. STOC 2013*, pages 291–300, 2013.
- 6 Chandra Chekuri and Alina Ene. Poly-logarithmic approximation for maximum node disjoint paths with constant congestion. In *Proc. SODA 2013*, pages 326–341, 2013.
- 7 Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *Proc. STOC 2005*, pages 183–192, 2005.
- 8 Chandra Chekuri, Sanjeev Khanna, and F Bruce Shepherd. Edge-disjoint paths in planar graphs with constant congestion. *SIAM J. Comput.*, 39(1):281–301, 2009.
- 9 Chandra Chekuri, Sanjeev Khanna, and F Bruce Shepherd. A note on multiflows and treewidth. *Algorithmica*, 54(3):400–412, 2009.
- 10 Chandra Chekuri, Guylain Naves, and F. Bruce Shepherd. Maximum edge-disjoint paths in k -sums of graphs. In *Proc. ICALP 2013*, volume 7965 of *LNCS*, pages 328–339, 2013.
- 11 Chandra Chekuri, Guylain Naves, and F. Bruce Shepherd. Maximum edge-disjoint paths in k -sums of graphs. Technical report, ArXiv, 2013. URL: <http://arxiv.org/abs/1303.4897>.
- 12 Eden Chlamtac, Robert Krauthgamer, and Prasad Raghavendra. Approximating sparsest cut in graphs of bounded treewidth. In *Proc. APPROX-RANDOM 2010*, volume 6302 of *LNCS*, pages 124–137, 2010.
- 13 Julia Chuzhoy. On vertex sparsifiers with Steiner nodes. In *Proc. STOC 2012*, pages 673–688, 2012.
- 14 Julia Chuzhoy and Shi Li. A polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2. In *Proc. FOCS 2012*, pages 233–242, 2012.
- 15 Reinhard Diestel. *Graph theory*, volume 173. Springer, third edition, 2005.

- 16 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.
- 17 Anupam Gupta, Kunal Talwar, and David Witmer. Sparsest cut on bounded treewidth graphs: algorithms and hardness results. In *Proc. STOC 2013*, pages 281–290, 2013.
- 18 R.M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.
- 19 S.G. Kolliopoulos and C. Stein. Approximating disjoint-path problems using packing integer programs. *Math. Prog.*, 99(1):63–87, 2004.
- 20 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity – Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012.
- 21 Neil Robertson and Paul D Seymour. Graph minors. V. Excluding a planar graph. *J. Combinatorial Theory, Ser. B*, 41(1):92–114, 1986.
- 22 Neil Robertson and Paul D Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combinatorial Theory, Ser. B*, 63(1):65–110, 1995.
- 23 Neil Robertson and Paul D Seymour. Graph minors. XVI. Excluding a non-planar graph. *J. Combinatorial Theory, Ser. B*, 89(1):43–76, 2003.
- 24 Lïc Séguin-Charbonneau and F Bruce Shepherd. Maximum edge-disjoint paths in planar graphs with congestion 2. In *Proc. FOCS 2011*, pages 200–209, 2011.