

Maximum Edge-Disjoint Paths in k -Sums of Graphs

Chandra Chekuri*

Guyslain Naves†

F. Bruce Shepherd‡

February 15, 2013

Abstract

We consider the approximability of the maximum edge-disjoint paths problem (MEDP) in undirected graphs, and in particular, the integrality gap of the natural multicommodity flow based relaxation for it. The integrality gap is known to be $\Omega(\sqrt{n})$ even for planar graphs [14] due to a simple topological obstruction and a major focus, following earlier work [17], has been understanding the gap if some constant congestion is allowed. In planar graphs the integrality gap is $O(1)$ with congestion 2 [23, 7]. In general graphs, recent work has shown the gap to be $\text{polylog}(n)$ [10, 11] with congestion 2. Moreover, the gap is $\log^{\Omega(c)} n$ in general graphs with congestion c for any constant $c \geq 1$ [1].

It is natural to ask for which classes of graphs does a constant-factor constant-congestion property hold. It is easy to deduce that for given constant bounds on the approximation and congestion, the class of “nice” graphs is minor-closed. Is the converse true? Does every proper minor-closed family of graphs exhibit a constant factor, constant congestion bound relative to the LP relaxation? We conjecture that the answer is yes. One stumbling block has been that such bounds were not known for bounded treewidth graphs (or even treewidth 3). In this paper we give a polytime algorithm which takes a fractional routing solution in a graph of bounded treewidth and is able to integrally route a constant fraction of the LP solution’s value. Note that we do not incur any edge congestion. Previously this was not known even for series parallel graphs which have treewidth 2. The algorithm is based on a more general argument that applies to k -sums of graphs in some graph family, as long as the graph family has a constant factor, constant congestion bound. We then use this to show that such bounds hold for the class of k -sums of bounded genus graphs.

1 Introduction

The disjoint paths problem is the following: given an undirected graph $G = (V, E)$ and node pairs $H = \{s_1 t_1, \dots, s_p t_p\}$, are there disjoint paths connecting the given pairs? We use NDP and EDP to refer to the version in which the paths are required to be node-disjoint or edge-disjoint. Disjoint path problems are cornerstone problems in combinatorial optimization. The seminal work on graph minors of Robertson and Seymour [21] gives a polynomial time algorithm for NDP (and hence also for EDP) when p is fixed; the algorithmic and structural tools developed for this have led to many other fundamental results. In contrast to the undirected case, the problem in directed graphs is NP-Complete for $p = 2$ [13]. Further, NDP and EDP are NP-Complete in undirected graphs when p is part of the input. The maximization versions of EDP and NDP have also attracted intense interest, especially in connection to its approximability. In the *maximum edge-disjoint path* (MAX EDP)

*Dept. of Computer Science, University of Illinois, Urbana, IL 61801. chekuri@illinois.edu. Work on this paper is partly supported by NSF grant CCF-1016684.

†Laboratoire d’informatique fondamentale, Faculté des sciences de Luminy, Marseille, France. guyslain.naves@lif.univ-mrs.fr.

‡Dept. Mathematics and Statistics, McGill University, Montreal. bruce.shepherd@mcgill.ca. Work is partly supported by NSERC Discovery/Accelerator Grants.

problem we are given an undirected (in this paper) graph $G = (V, E)$, and node pairs $H = \{s_1 t_1, \dots, s_p t_p\}$, called *commodities* or *demands*. MAX EDP asks for a maximum size subset $I \subseteq \{1, 2, \dots, p\}$ of commodities which is *routable*. A set I is routable if there is a family of edge-disjoint paths $(P_i)_{i \in I}$ where P_i has extremities s_i and t_i for each $i \in I$. In a more general setting, the edges have integer capacities $c : E(G) \rightarrow \mathbb{N}$, and instead of edge-disjoint paths, we ask that for each edge $e \in E(G)$, at most $c(e)$ paths of $(P_i)_{i \in I}$ contain e . For any demand $h = st \in H$, denote by \mathcal{P}_h the set of st -paths in G , and $\mathcal{P} = \bigcup_{h \in H} \mathcal{P}_h$. A natural linear programming relaxation of MAX EDP is then:

$$\begin{array}{ll} \max & \sum_{h \in H} z_h \\ & \sum_{P \in \mathcal{P}_h} x_P = z_h \leq 1 \quad (\text{for all } h \in H) \\ & \sum_{P \in \mathcal{P}, e \in P} x_P \leq c_e \quad (\text{for all } e \in E) \\ & x \geq 0 \end{array} \quad \text{subject to} \quad (1)$$

NP-Completeness of EDP implies that MAX EDP is NP-Hard. In fact, MAX EDP is NP-Hard in capacitated trees for which EDP is trivially solvable. This indicates that MAX EDP inherits hardness also from the selection of the subset of demands to route. As pointed out in [14], a grid example shows that the integrality gap of the multicommodity flow relaxation may be as large as $\Omega(\sqrt{n})$ even in planar graphs. However, the grid example is not robust in the sense that if we allow edge-congestion 2 (or equivalently, if we assume all capacities are initially at least 2), then the example only has a constant factor gap. This observation led Kleinberg-Tardos [17] to seek better approximations (polylog or constant factor) for planar graphs in the regime where some low congestion is allowed. With some work, this agenda proved fruitful: a constant-approximation with edge congestion 4 was proved possible in planar graphs [7]; this was improved to (an optimal) edge congestion 2 in [23].

In general graphs, Chuzhoy [10] recently obtained the first poly-logarithmic approximation with constant congestion (14). This was subsequently improved to the optimal congestion of 2 by Chuzhoy and Li [11]. It is also known that, in general graphs, the integrality gap of the flow LP is $\Omega(\log^{\Omega(1/c)} n)$ even if congestion c is allowed; the known hardness of approximation results for MAX EDP with congestion have similar bounds as the integrality gap bounds, see [1].

For any constants $\alpha, \beta \geq 1$, one may ask for which graphs does the LP for MAX EDP admit an integrality gap of α if edge congestion β is allowed. It is natural to require this for any possible collection of demands and any possible assignment of edge capacities. For fixed constants, it is easy to see that the class of such graphs is closed under minors. Is the converse true? That is, do all minor-closed graphs exhibit a *constant factor constant-congestion* (CFCC) integrality gap for MAX EDP? In fact we consider the following stronger conjecture with congestion 2.

Conjecture 1. Let \mathcal{G} be any proper minor-closed family of graphs. Then the integrality gap of the flow LP for MAX EDP is at most a constant $c_{\mathcal{G}}$ when congestion 2 is allowed.

The preceding conjecture is inherently a geometric question, but one would also anticipate a polytime algorithm for producing the routable sets which establish the gap. In attempting to prove Conjecture 1, one must delve into the structure of minor-closed families of graphs, and in particular the characterization given by Robertson and Seymour [21]. Two minor-closed families that form the building blocks for this characterization are (i) graphs embedded on surfaces of bounded genus (in particular planar graphs), and (ii) graphs with bounded treewidth. For MAX EDP, we have a constant factor integrality gap with congestion 2 for planar graphs. In [8] it is shown that the integrality gap of the LP for MAX EDP in graphs of treewidth at most k is $O(k \log k \log n)$; note that this is with congestion 1. Existing integrality gap results, when interpreted in terms of treewidth k , show that the integrality gap is $\Omega(k)$ for congestion 1 and $\Omega(\log^{O(1/c)} k)$ for congestion $c > 1$. It was asked in [8] whether the gap is $O(k)$ with congestion 1. In particular, the question of whether the gap

is $O(1)$ for $k = 2$ (this is precisely the class of series parallel graphs) was open. In this paper we show the following result.

Theorem 1.1. *The integrality gap of the flow LP for MAX EDP is $2^{O(k)}$ in graphs of treewidth at most k . Moreover, there is a polynomial-time algorithm that given a graph G , a tree decomposition for G of width k , and fractional solution to the LP of value opt , outputs an integral solution of value $\Omega(opt/2^{O(k)})$.*

The preceding theorem is a special case of a more general theorem that we prove below. Let \mathcal{G} be a family of graphs. For any integer $k \geq 1$, let \mathcal{G}_k denote the class of graphs obtained from \mathcal{G} by the k -sum operation. The k -sum operation is formally defined in Section 2.1; the structure theorem of Robertson and Seymour is based on the k -sum operation over certain classes of graphs.

Theorem 1.2. *Let \mathcal{G} be a minor-closed class of graphs such that the integrality gap of the flow LP is α with congestion β . Then the integrality gap of the flow LP for the class \mathcal{G}_k is $2^{O(k)}\alpha$ with congestion $\beta + 3$.*

The preceding theorem is effective in the following sense: there is a polynomial-time algorithm that gives a constant factor, constant congestion result for \mathcal{G}_k assuming that (i) such an algorithm exists for \mathcal{G} and (ii) there is a polynomial-time algorithm to find a tree decomposition over \mathcal{G} for a given graph $G \in \mathcal{G}_k$.

We give the following as a second piece of evidence towards Conjecture 1.

Theorem 1.3. *The integrality gap of the flow LP on graphs of genus $g > 0$ is $O(g \log^2(g + 1))$ with congestion 3.¹*

Theorems 1.2 and 1.3 imply that the class of graphs obtained as k -sums of graphs with genus g is CFCC when k and g are fixed constants. The bottleneck in extending our results to prove Conjecture 1 are planar graphs (or more generally bounded genus graphs) that have “vortices” which play a non-trivial role in the RS structure theorem.

A brief discussion of technical ideas and related work: The approximability of MAX EDP in undirected and directed graphs has received much attention in the recent years. We refer the reader to some recent papers [11, 10, 23, 1]. A framework based on well-linked decompositions [5] has played an important role in understanding the integrality gap of the flow relaxation in undirected graphs. It is based on recursively cutting the input graph along sparse cuts until the given instance is well-linked. However, this framework loses at least a logarithmic factor in the approximation. The work in [7] obtained a constant factor approximation for planar graphs by using a more refined decomposition that took advantage of the structure of planar graphs. For graphs of treewidth k , [8] used the well-linked decomposition framework to obtain an $O(k \log k \log n)$ -approximation and integrality gap. Our work here shows that one can bypass the well-linked decomposition framework for bounded treewidth graphs, and more generally for k -sums over families of graphs. The key high-level idea is to effectively reduce the (tree)width of one side of a sparse cut if the terminals cannot route to a small set of nodes. Making this work requires a somewhat nuanced induction hypothesis. For bounded-genus graphs, we adapt the well-linked decomposition to effectively reduce the problem to the planar graph case.

There are two streams of questions comparing minimum cuts to maximum flows in graphs. First, the *flow-cut gap* measures the gap between a sparsest cut and a maximum concurrent flow of an instance. The second measures the *throughput-gap* by comparing the maximum throughput flow and the minimum multicut. These gap results have been of fundamental importance in algorithms starting with the seminal work of Leighton and Rao [19]. It is known that the gaps in general undirected graphs are $\Theta(\log n)$; see [24] for a survey. It

¹ We believe that the congestion bound in the preceding theorem can be improved to 2 with some additional technical work. We do not give a polynomial-time algorithm although we believe that it too is achievable with some (potentially messy) technical work.

is also conjectured [15] (the GNRS Conjecture) that the flow-cut gap is $O(1)$ for minor-closed families. This conjecture is very much open and is not known even for planar graphs or treewidth 3 graphs; see [18] for relevant discussion and known results. In contrast, the work of Klein, Plotkin and Rao [16] showed that the throughput-gap is $O(1)$ in any proper minor-closed family of graphs (formally shown in [25]). The focus of these works is on fractional flows, in contrast to our focus on integral routings. Conjecture 1 is essentially asking about the integrality gap of throughput flows. Given the $O(1)$ throughput-gap [16], it can also be viewed as asking whether the gap between the maximum *integer* throughput flow with congestion 2 is within an $O(1)$ factor of the minimum multicut. Analogously for flow-cut gaps, [9] conjectured that the gap between the maximum integer concurrent flow and the sparsest cut is $O(1)$ in minor-free graphs.

2 Preliminaries

Recall that an instance of MAX EDP consists of a graph G and demand pairs H . In general H can be a multiset, however it is convenient to assume that H is a matching on the nodes of G . Indeed we just have to attach the terminals to leaves created from new nodes. With this assumption we use X to denote the set of terminals (the endpoints of the demand pairs) and M the matching on X that corresponds to the demands. We call the triple (G, X, M) a matching instance of MAX EDP. Let \bar{x} be a feasible solution to the LP relaxation (1). For each node $v \in X$, we also use $x(v)$ to denote the value $\sum_{P \in \mathcal{P}_h} x_P$ where v is an endpoint of the demand h ; this is called the *marginal value* of v . We assume that all capacities c_e are 1; this does not affect the integrality gap analysis. Moreover, as argued previously (cf. [3]), at a loss of a factor of 2 in the approximation ratio, the assumption can be made for polynomial-time algorithms that are based on rounding a solution to the flow relaxation.

2.1 k -Sums and the structure theorem of Robertson and Seymour

Let G_1 and G_2 be two graphs, and C_i a clique of size k in G_i . The graph G obtained by identifying the nodes of C_1 one-to-one with those of C_2 , and then removing some of the edges between nodes of $C_1 = C_2$, is called a k -sum of G_1 and G_2 . For a class of graphs \mathcal{G} , we define the class \mathcal{G}_k of the graphs *obtained from \mathcal{G} by k -sums*, to be the smallest class of graphs such that: (i) \mathcal{G} is included in \mathcal{G}_k , and (ii) if G is a k -sum of $G_1 \in \mathcal{G}$ and $G_2 \in \mathcal{G}_k$, then $G \in \mathcal{G}_k$.

Fix a class of graphs \mathcal{G} . A tree \mathcal{T} is a *tree decomposition* over \mathcal{G} for a graph $G = (V, E)$, if each node A in \mathcal{T} is associated to a subset of nodes $X_A \subseteq V$, called a *bag*, and the following properties hold:

- (i) for each $v \in V(G)$, the set of nodes of \mathcal{T} whose bags contain v , form a non-empty sub-tree of \mathcal{T} ,
- (ii) for each edge $uv \in E(G)$, there is a bag with both u and v in it,
- (iii) for any bag X , the graph obtained from $G[X]$ by adding cliques over $X \cap Y$, for every adjacent bag Y , is in \mathcal{G} . We denote this graph by $G[[X]]$.

When \mathcal{G} is closed under taking minors, condition (iii) implies that $G[X]$ itself is in \mathcal{G} , as well as any graph obtained from $G[X]$ by adding edges in $X \cap Y$, for any adjacent bag Y . We sometimes identify the nodes of \mathcal{T} with their respective bags. We also denote by $V(\mathcal{T})$, the union of all bags, and so $V(\mathcal{T}) \subseteq V(G)$.

A set of nodes $X \cap Y$, for X and Y adjacent bags, is called a *separator*. When the tree decomposition \mathcal{T} is minimal (with respect to the number of bags), the separators are disconnecting node sets of G . Thus each edge e of \mathcal{T} identifies a separator, denoted by V_e . For convenience, we usually work with rooted tree decompositions, where an arbitrary node is chosen to be the root. Then, for bag X and its parent Y , we denote by S_X the separator $X \cap Y$.

The *width* of a tree decomposition \mathcal{T} is the maximum cardinality of a separator of \mathcal{T} . The *width* of a graph (relative to a graph class \mathcal{G}) is the smallest width of a tree decomposition for that graph. A graph of width k can thus be obtained by k -sums of graphs from \mathcal{G} . As a special case, the *treewidth* of a graph G is the smallest k such that G admits a decomposition of width k relative to the class of all graphs with at most $k + 1$ nodes.

Let \mathcal{T} be a tree decomposition of a graph G , rooted at a node R . For any edge e of the tree decomposition, let $\overline{\mathcal{T}}_e$ and \mathcal{T}_e be the subtrees obtained from \mathcal{T} by removing e , with $R \in \overline{\mathcal{T}}_e$. We denote by G_e the graph obtained from the induced subgraph of G on node set $V(\mathcal{T}_e)$, and then removing all edges in $V(\overline{\mathcal{T}}_e) \times V(\mathcal{T}_e)$. Note that \mathcal{T}_e is a tree decomposition of G_e .

We recall informally the graph structure theorem proved by Robertson and Seymour. For $k \in \mathbb{N}$, let \mathcal{L}^k be the graphs obtained in the following way.

- we start from a graph G embeddable on a surface of genus k ,
- then we add *vortices* of width k to at most k faces of G ,
- then we add at most k *apex* nodes. That is, each of these nodes can be adjacent to an arbitrary subset of nodes.

Then, we denote $\mathcal{L}^k = \mathcal{L}_k^k$. For a graph H , we denote by \mathcal{K}_H the graphs that do not contain an H -minor.

Theorem 2.1 (Robertson and Seymour [22]). *For any graph H , there is an integer $k > 0$ such that $\mathcal{K}_H \subseteq \mathcal{L}^k$.*

In order to prove Conjecture 1, one should be able to use the preceding decomposition theorem, proving that the CFCC property holds for bounded genus graph and is preserved by adding a constant number of vortices and apex nodes, and by taking k -sums. Apex nodes are easy to deal with. This paper provides a proof for bounded genus graphs and for k -sums. This leaves only the cases of vortices as the bottleneck in proving the conjecture.

3 Technical Ingredients

We rely on several technical tools and ingredients that are either explicitly or implicitly used in recent work on MAX EDP.

3.1 Moving Terminals

We first describe a general tool (ideas of which are leveraged also in previous work, cf. [7, 8]) that allows us to reduce a MAX EDP instance to a simpler one by moving the terminals to a specific set of new locations (nodes). The two instances are equivalent for MAX EDP, up to an additional constant congestion and constant factor approximation.

Lemma 3.1. *Suppose we have a (matching) instance (G, H) of MAX EDP with some solution \bar{x} to its LP relaxation. Let $\text{val}(\bar{x}) = \sum_P x_P$. Suppose that for some $S, R \subseteq V(G)$, there is a flow which routes $x(s)$ from each $s \in S$, and all flow terminates in R . Then there is another (matching) instance (G', H') with the following properties.*

1. *The new instance has a (fractional) solution of value at least $\text{val}(\bar{x})/5$,*
2. *If there is an integral solution for (G', H') of value val' and congestion c , then there is an integral solution for (G, H) of the same value and congestion $c + 2$,*

3. G', H' is obtained from G, H by hanging off pendant stars from some of the nodes.

Proof. Let T be a forest of G spanning all the nodes of S such that each component of T contains at least one node of R . We consider each component of T separately, so we assume here that T is a tree. Let $r \in R \cap V(T)$ and take this as a root of T .

We partition S into subsets S_1, \dots, S_ℓ with the following properties.

- (i) for each $i \in [1, \ell]$, $1 \leq x(S_i) \leq 2$ (except possibly S_1 may have $x(S_1) < 1$),
- (ii) to each S_i is associated a subtree T_i of T spanning S_i ,
- (iii) T_1, \dots, T_ℓ are edge-disjoint.

We achieve this by the following iterative scheme.

- If $x(T) \leq 2$, choose $S_1 = S$, $T_1 = T$, $\ell = 1$. Else:
- Find a deepest node v in T , such that the subtree T' rooted at v has $x(T') \geq 1$.
- let v_1, \dots, v_i be the children of v , and T'_1, \dots, T'_i be the subtrees rooted at v_1, \dots, v_i respectively. If $\sum_{j=1}^i x(T'_j) < 1$, then define $A := E(T')$ and $U := V(T') \cap S$. Otherwise, find the smallest i' such that $x(B) \geq 1$, where $B := \bigcup_{j=1}^{i'} V(T'_j) \cap S$. Then set $A := \bigcup_{j=1}^{i'} (E(T'_j) \cup \{vv_j\})$. In both cases, we get $1 \leq x(A) \leq 2$, and A induces a tree. (Note that in the latter case v was not placed in B but does lie in $V(A)$.)
- proceed inductively on $T - A$ and $S - B$, to find $S_1, \dots, S_{\ell'}$ and $T_1, \dots, T_{\ell'}$. Then $\ell := \ell' + 1$, $S_\ell := B$ and $T_\ell := A$.

Note that with this scheme, $x(S_1)$ might be less than 1, but in that case T_1 contains the root node from R . As the hypotheses $x(S_i) \geq 1$ is only used to prove the existence of edge-disjoint paths from the S_i 's to R , this does not pose a problem (we can simply take the trivial path at r for S_1).

Each S_i is called a *cluster*. As each cluster sends at least one unit of flow to R (with the exception of S_1 already mentioned), there is a family of edge-disjoint paths P_1, \dots, P_ℓ , where each P_i goes from a node $s_i \in S_i$ to some node $r_i \in R$. This can be seen by adding dummy source nodes (one for each S_i) adjacent to nodes in each S_i , and a single dummy sink node adjacent from each r_i (a detailed proof is found in [7]).

We now define a new instance of MAX EDP G', H' . G' is obtained from G by adding ℓ new nodes x_1, \dots, x_ℓ with degree one, where x_i is adjacent to r_i . The capacity of a new edge $r_i x_i$ is 1, and we re-define P_i as extending to x_i . We identify each terminal in S with the r_i associated with its cluster as follows. Let $\phi(s) := s$ if $s \notin S$ and $\phi(s) = x_i$ if $s \in S_i$. Then let $\phi(H) := \{\phi(s)\phi(t) : st \in H\}$. These demands do not yet form a matching, so H' is obtained from $\phi(H)$ by simply deporting each of the terminals in S to new nodes forming leaves.

We show how to transform \bar{x} into a fractional flow \bar{x}' in G', H' with congestion 5, such that \bar{x}' has the same value as \bar{x} . For that, we only extend the flow paths for the demands in $H' \setminus H$. Let $st \in H$ be such a demand and $s't' = \phi(s)\phi(t)$ its image. For any st -path P with value x_P , let P' be the path obtained from P by:

- if $s \in S_i$ for some i , concatenate P_i and the unique ss_i -path of T_i ,
- similarly if $t \in S_j$ for some j , concatenate P_j and the unique ss_j -path of T_j .

Then, set $x'_{P'} := x_P$. Then \bar{x}' has the same value as \bar{x} by construction, but has higher congestion. Any of the edges $r_i x_i$ (or additional leaves from an x_i) have congestion at most 2 by construction; thus it is enough

to focus on edges within G . The original flow paths incur congestion of at most 1 on any edge, so we address the added congestion from extending the flow paths. The edges of any P_i are charged by at most 2 units (by terminals within S_i) and each T_i is also charge by at most 2 units. As an edge may be contained in at most one P_i and at most one T_i the extra congestion is bounded by 4. Hence the total congestion of x' is at most 5, and in particular, this implies that the fractional optimal solution in G' is at least $\frac{1}{5}\text{OPT}$.

Suppose now that we have an integral solution to MAX EDP for G', H' with congestion c . We show how to transform it into a solution for G, H of the same value with congestion $c + 2$. Since we can assume the flow paths used are simple, we only need to address the flow paths for demands in $H' \setminus H$. Let P be any path in the solution satisfying a commodity associated with a node $s' = \phi(s)$, where s is in S_i . Then we extend P by concatenating P_i and the unique $s_i s$ -path of T_i to it. We may also shortcut this to obtain a simple path. Again, this clearly defines a solution of same value to the original problem. Since the capacity of $r_i x_i$ is one, we use each P_i at most once, and a path in T_i is used for only one such $s \in S_i$. As the paths P_i are disjoint, and the subtrees are disjoint, each edge is used at most $c + 2$ times: c for the original routing, 1 for the P_i paths, and 1 for the paths inside the T_i 's. \square

3.2 Sparsifiers

Let $G = (V, E)$ be a (multi) graph and let $S \subset V$. We are interested in creating a graph H only on the node set S that acts as a proxy for routing between nodes in S in the original graph G . The notion of sparsifiers, introduced in [20], has many possible formulations depending on the various applications. For instance, a Gomory-Hu Tree can be viewed as a sparsifier which encodes pairwise maximum flows in a graph. We are interested in the following model. We say that $H = (S, E_H)$ is a (σ, ρ) -sparsifier for S in G if the following properties are true:

- any feasible (fractional) multicommodity flow in G with the endpoints in S is (fractionally) routable in H with congestion at most σ
- any integer multicommodity flow in H is integrally routable in G with congestion ρ .

Existing sparsifier results mostly focus on fractional routing (or cut preservation) while we need *integer* sparsifiers in the sense of the second point above. Chuzhoy [10] developed an integer sparsifier result but it uses Steiner nodes and has limitations that preclude its direct use in our setting. Instead, a simple argument based on splitting-off gives the following weak sparsifier result that suffices for our purposes.

Theorem 3.2. *Let $G = (V, E)$ be a graph and $S \subset V$. There is a $(|S|^2, 2)$ -sparsifier for S in G .*

Proof. First, by standard T -join theory, G contains a subset E' of edges, such that if we add an extra copy of each such edge, we obtain an Eulerian graph G' . We may now apply splitting off repeatedly at the (even degree) nodes of $V - S$. Each operation preserves the minimum cut between any pair of nodes $u, v \in S$. This ultimately results in a (multi) graph $H = (S, F)$ on S . We claim that H is the desired sparsifier.

Note that any integral routing in H can easily mapped to an integral routing in G' since the edges in F map to edge-disjoint paths in G' . Since G' had potentially an extra copy of an edge from G we see that $\rho = 2$.

Now consider any fractional multicommodity flow in G between nodes in S . Say $d(uv)$ flow is routed between $u, v \in S$. Then $d(u, v) \leq \lambda_G(u, v)$ where $\lambda_G(u, v)$ is the capacity of a min u - v cut in G . Since the splitting-off operation preserved connectivity, $\lambda_H(u, v) \geq \lambda_G(u, v)$, hence we can route $d(u, v)$ flow between u and v in H . However, we have $|S|(|S| - 1)/2$ distinct pairs of nodes in S and routing their flows simultaneously in H can result in a congestion of at most $|S|(|S| - 1) \leq |S|^2$ since each individual flow can be feasibly routed in H . This shows that $\sigma \leq |S|^2$. \square

Remark 1. The proof of the preceding theorem shows that the congestion parameter ρ can be chosen to be an additive 1 if G is a capacitated graph.

3.3 Routings through a small set of nodes

We use as a black box the following result from Section 3.1 in [6].

Proposition 3.3. Let G, H be a MAX EDP instance and let \bar{x} be a fractional solution such that there is a node v that is contained in every flow path with positive flow. Then there is a polynomial time algorithm that routes at least $\frac{1}{12} \sum_i x_i$ pairs from H on edge-disjoint paths.

Remark 2. The bound of $1/12$ in the preceding proposition is not explicitly stated in [6] but can be inferred from the arguments.

Now suppose that instead of a single node v , there is a subset S that intersects every flow path in a fractional solution \bar{x} . It is then easy to see that there is a node v that intersects flow paths of total value at least $\sum_i x_i / |S|$. We can then apply the preceding proposition to claim that we can route $\frac{1}{12|S|} \sum_i x_i$ pairs. We combine this with a simple re-routing argument that is relevant to our algorithm to obtain the following.

Proposition 3.4. Let G, H be a matching instance of MAX EDP and let \bar{x} be a feasible fractional solution for it. Suppose that there is also a second flow that routes at least x_i/α flow from each terminal to some $S \subseteq V$ where $\alpha \geq 1$. Then there is an integral routing of at least $\frac{1}{36\alpha|S|} \sum_i x_i$ pairs.

Proof. Let $v \in S$ be the terminal which receives the most flow. Clearly this is of value at least $\frac{\sum_i x_i}{\alpha|S|}$. Consider a pair $s_i t_i$ such that one of the end points, say s_i sends $y_i \leq x_i/\alpha$ flow to v . The other end point t_i may send less than y_i (or no flow) to v . We may then create a y_i flow from t_i to v by using the x_i -flow between s_i, t_i and the flow from s_i to v . It is easy to see that overlaying all of the flows will cause capacities to be violated by a factor of at most 3; we scale down the flows by a factor of 3 to satisfy the capacity constraints. Via this process we can find a new fractional solution \bar{x}' such that (1) all the flow paths contain v and (2) $\sum_i x'_i \geq \frac{\sum_i y_i}{3} \geq \frac{\sum_i x_i}{3\alpha|S|}$. The result now follows by applying Proposition 3.3. \square

4 MAX EDP in k -sums over a family \mathcal{G}

The goal of this section is to prove Theorem 1.2. Throughout, we assume \mathcal{G} is a minor closed family, and we wish to prove bounds for the family \mathcal{G}_k obtained by k -sums. In particular, we assume that every subgraph on k nodes is included in \mathcal{G} .

Let \mathcal{A} be an algorithm/oracle that has the following property: given an MAX EDP instance on a graph $G \in \mathcal{G}$ it integrally routes h pairs with congestion β where h is at least a $1/\alpha$ fraction of the value of an optimum fractional solution to that instance. We call \mathcal{A} an (α, β) -oracle. We describe an algorithm using \mathcal{A} to approximate MAX EDP on \mathcal{G}_k . The proof is via induction on the width of a decomposition and the number of nodes. One basic step is to take a sparse cut S , lose all the flow crossing that cut, and recurse on both sides. We need to make our recursion on treewidth on the side S to which we charge the flow lost by cutting the graph. On the other side, $V \setminus S$, we simply recurse on the number of nodes. The main difficulty is to show how the treewidth is decreased on the S side. The trick is a trade-off between the main treewidth parameter and some connectivity properties in parts of the graph with higher treewidth. To drive this we need a more refined induction hypothesis rather than basing it only on the width of a tree decomposition.

Given $k \leq p$, a p -degenerate k -tree decomposition over \mathcal{G} of a connected graph G is a rooted tree decomposition \mathcal{T} where some leaves (nodes of degree 1) of the tree are labelled *degenerate* and:

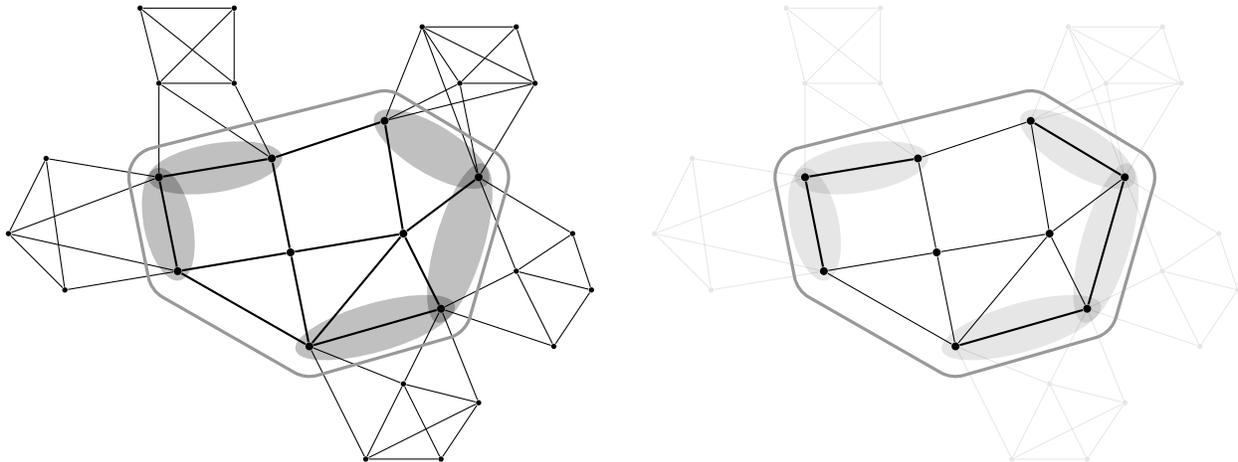


Figure 1: Illustration of the base case with $k = 0$ and $p \geq 1$: the graph corresponding to each degenerate leaf is replaced by a sparsifier on the associated separator.

- for every node X of \mathcal{T} , either $G[[X]] \in \mathcal{G}$ or X is a degenerate leaf (in which case $G[[X]]$ may be arbitrary),
- the separator corresponding to any edge $uv \in \mathcal{T}$ is of size at most k , unless it is incident to a degenerate leaf, in which case it may be up to size p .

A pendant leaf is not necessarily degenerate but if it is not, then it corresponds to a graph in \mathcal{G} . We use (k, p) -tree decomposition as a shorthand notation. We call a multiflow in such a graph G *flush* with the decomposition if every flow path that terminates at some node in a degenerate leaf L , also intersects S_L (we recall that S_L is the separator $V(L) \cap V(X)$ where X is the parent node of L).

We may think of graphs with such flush / degenerate decompositions as having an *effective treewidth* of size k . In effect, we can ignore that some separators can be larger, because we have the separate property of flushness which we can leverage via results such as Proposition 3.3.

The next theorem describes the main algorithm for converting an LP solution on some graph in \mathcal{G}_p , into an integral routing. As we process this solution, the tree decomposition becomes degenerate with the value of p fixed, but we gradually reduce k to 0.

Theorem 4.1. *Let \mathcal{A} be an (α, β) oracle for MAX EDP in \mathcal{G} . Let G be graph with a (k, p) -tree decomposition \mathcal{T} and suppose that G, H is an instance of MAX EDP with a fractional solution f that is flush with \mathcal{T} . Then there is an algorithm with oracle \mathcal{A} , which computes an integral multicommodity flow with congestion $\beta + 3$ and value $\gamma = \frac{\sum_i x_i}{360 \cdot \alpha (p+1)^2 12^k}$. Moreover, this algorithm can be used to obtain the following:*

1. an LP-based approximation algorithm with ratio $O(\alpha(p+1)^2 12^p)$ and congestion $\beta + 3$ for MAX EDP in \mathcal{G}_p
2. an algorithm with approximation ratio $O((p+1)12^p)$ and congestion 1 for the class of graphs of treewidth p .

The proof of the preceding theorem is somewhat long and technical and occupies the rest of the section. To help the exposition we break it up into several components. The proof proceeds by induction on k and the number of nodes. The base case with $k = 0$ is non-trivial and we treat it first.

The base case: We can assume without loss of generality that G is connected. If $k = 0$ and $p = 0$, then every separator from a decomposition would be empty. Since G is connected, the tree is a single (non-degenerate) node and hence lies in \mathcal{G} ; the result follows via the existence of \mathcal{A} . So throughout we assume a fixed $p \geq 1$ and consider a decomposition together with a flush fractional routing f as described. Let $x : V \rightarrow \mathbb{R}$ be the marginal values of f . Again we use x_i to denote the common value $x(s_i) = x(t_i)$. Hence $|f| = \sum_i x_i = \frac{1}{2} \sum_{v \in V(G)} x(v)$ where we use the notation $|f|$ to denote the value of the flow f .

Now assume that $k = 0$ and $p > 1$. We may assume that \mathcal{T} has more than one node. Since G is connected, each separator corresponding to an edge of \mathcal{T} must be non-trivial. Since $k = 0$, each edge of \mathcal{T} must be incident to a degenerate leaf. It follows that \mathcal{T} is a single edge between two degenerate leaves or a star whose leaves are all degenerate. If \mathcal{T} is a single edge between two degenerate leaves, all flow paths intersect the separator of size p associated with the edge; hence by Proposition 3.3, there is an integral routing of value at least $(\sum_i x_i)/(12p)$. We therefore restrict our attention to the case when \mathcal{T} is a star with ℓ leaves. Let $G^* = G[X]$ where X is the bag at the center/root; we observe that $G^* \in \mathcal{G}$. Let X_i be the bag at the i 'th leaf. We let G_i denote the graph obtained from $G[X_i]$ after removing the edges between the separator nodes $S_i = X \cap X_i$.

The base case of theorem assertion (2) on treewidth p graphs holds as follows. By flushness, all flow paths intersect G^* , which has at most $p + 1$ nodes. Hence there is some node which is receiving at least $(\sum_i x_i)/(p + 1)$ of this flow. Hence by Proposition 3.3, there is a (congestion 1) integral routing of value at least $(\sum_i x_i)/(12(p + 1))$.

Now consider the general case with a minor-closed class \mathcal{G} and \mathcal{T} a star whose center is $G^* \in \mathcal{G}$. We proceed in the following steps:

1. Move the terminals in each G_i to the separator S_i that is contained in G^* (using Lemma 3.1).
2. Replace each G_i by a $(p^2, 2)$ sparsifier on S_i to obtain a graph G' . Via the sparsifier property, scale the flow in G down by a factor of p^2 to make it feasible in G' .
3. Apply the algorithm \mathcal{A} on the new instance in G' .
4. Transfer the routing in G' to a routing in G with additive $+1$ via the sparsifier property.
5. Use the second part of Lemma 3.1 to convert the routing in G into a solution for our original instance before the terminals were moved (incur an additional $+2$ additive congestion).

We describe the steps in more detail. We observe that the graphs G_i are edge-disjoint. The first step is a simple application of Lemma 3.1 where for each i , we move any terminals in $G_i - S_i$ to the separator S_i via clustering. This is possible because of the flushness assumption; if P is a flow path with an endpoint in $G_i - S_i$ then that path intersects S_i . This incurs a factor 5 loss in the value of the new flow we work with (and it incurs an additive 2 congestion when we convert back to an integral solution for our original instance). To avoid notational overload we let f be the flow for the new instance which is at least $\frac{1}{5}$ th of the original flow.

In step (2), we can simultaneously replace each G_i by a $(p^2, 2)$ -sparsifier F_i on S_i — see Theorem 3.2. Call the new instance G' and note that since we only added edges to the separators S_i , G' is a subgraph of $G[[X]]$ and hence $G' \in \mathcal{G}$. At this step, we also need to convert our flows in G_i 's to be flows in G' . The sparsifier guarantees that any multicommodity flow on S_i that is feasible in G_i can be routed in F_i with congestion p^2 . Hence, scaling the flow down by p^2 guarantees its feasibility in G' .

We now work with the new flow f' in the graph G' and apply \mathcal{A} to obtain a routing of size $|f'|/\alpha \geq |f|/(5p^2\alpha)$ with congestion β . We must now convert this integral routing to one in G . Again, for each i , there is an embedded integral routing in F_i which will be re-routed in G_i . We incur an additive 1 congestion for

this. Finally we apply the second part of Lemma 3.1 to route the pairs before they moved, incurring an additive congestion of 2.

Thus the total number of pairs routed is at least $|f|/(5\alpha p^2)$ and the overall congestion of the routing is $\beta + 3$. This proves the base case when $k = 0$.

The induction step: Henceforth, we assume that $p \geq k > 0$ and that \mathcal{T} contains at least one edge e with the associated separator V_e (the intersection of the bags at the two end points of e) of size equal to k ; otherwise we either have a degenerate star again, or we may use $k - 1$. We consider an easy setting when there is a flow g routing $x(v)/10$ from each vertex v to V_e . (Note that checking the existence of the desired flow to V_e can be done by a simple maximum-flow computation.) We then obtain an integral (congestion 1) flow of size $(\sum_i x_i)/(360k)$ via Proposition 3.4 which is sufficient to establish the induction step for k .

Assume now that there is no such flow g . Then there is a cut $U \subset V \setminus V_e$ with $c(U) := c(\delta(U)) < \frac{1}{10}x(U)$. We may assume that U is minimal and central ($G[U]$ and $G[V \setminus U]$ are connected). Such a cut can be recovered from the maximum flow computation. We now work with a reduced flow f' obtained from f by eliminating any flow path that intersects $\delta(U)$. We also let x' be the marginals for f' . Obviously we have

$$|f| - |f'| \leq c(U) < \frac{x(U)}{10}.$$

Let $f_U, f_{\bar{U}}$ be the flow vectors obtained from f' , where f_U only uses the flow paths contained in U , and $f_{\bar{U}}$ uses the flow paths contained in $V \setminus U$. The idea is that we recurse on $G[U]$ and $G[V \setminus U]$. We modify the instance on $G[U]$ to ensure that it has a $(k - 1, p)$ -tree decomposition and charge the lost flow to this side. The recursion on $G[V \setminus U]$ is based on reducing the number of nodes, the width is not reduced. Reducing the width on the U side and ensuring the flushness property is not immediate; it requires us to modifying f_U and in the process we may lose further flow. We explain this process before analyzing the number of pairs routed by the algorithm.

We note that $G[U]$ and $G[V \setminus U]$ easily admit (k, p) -tree decompositions, by intersecting the nodes of \mathcal{T} with U and $V \setminus U$ respectively, and removing the empty nodes; recall $G[U]$ and $G[V \setminus U]$ are connected. Denote these by \mathcal{T}_U and $\mathcal{T}_{\bar{U}}$ respectively. Degenerate leaves of \mathcal{T}_U and $\mathcal{T}_{\bar{U}}$ are the same as the degenerate leaves of \mathcal{T} . Some of these are “split” by the cut, otherwise they are simply assigned to either \mathcal{T}_U or $\mathcal{T}_{\bar{U}}$. If split, the two “halves” go to appropriate sides of the decomposition. The flows $f_U, f_{\bar{U}}$ will be flush with each such degenerate leaf (if the leaf is split, any flow path that crosses the cut is removed).

We proceed in $\mathcal{T}_{\bar{U}}$ by induction on the number of nodes. However, since we charge the lost flow to the cut (i.e., to \mathcal{T}_U) we modify \mathcal{T}_U to obtain a $(k - 1, p)$ -tree decomposition. We state a lemma that accomplishes this.

Lemma 4.2. *For the residual instance on $G[U]$ with flow f_U we can either route $\frac{1}{360k} \cdot |f_U|/2$ pairs integrally or find a $(k - 1, p)$ -tree decomposition \mathcal{T}'_U and a reduced flow vector f'_U that is flush with \mathcal{T}'_U with $|f'_U| \geq |f_U|/2$.*

We postpone the proof of the above lemma and proceed to finish the recursive analysis. We apply the induction hypothesis for k on $\mathcal{T}_{\bar{U}}$ with the number of nodes reduced; hence the algorithm routes at least $\frac{|f_{\bar{U}}|}{\alpha(p+1)^2 360 \cdot 12^k}$ pairs in $G[V \setminus U]$ with congestion at most $\beta + 3$. For $G[U]$ we consider two cases based on the preceding lemma. In the first case the algorithm directly routes $\frac{1}{360k} \cdot |f_U|/2$ pairs integrally in $G[U]$. In the second case we recurse on $G[U]$ with the flow f'_U that is flush with respect to the $(k - 1, p)$ -tree decomposition \mathcal{T}'_U ; by the induction hypothesis the algorithm routes at least $\frac{|f'_U|}{\alpha(p+1)^2 360 \cdot 12^{k-1}}$ pairs with congestion $\beta + 3$. Since the number of pairs routed in this second case is less than in the first case, we may focus on it, as we now show that the total number of pairs routed in $G[U]$ and $G[V \setminus U]$ satisfies the induction hypothesis for k . We first observe that $|f| \leq |f_U| + |f_{\bar{U}}| + c(U)$. Moreover, $c(U) \leq x(U)/10$, and since at least $x(U)/2$ flow originated in U , and we

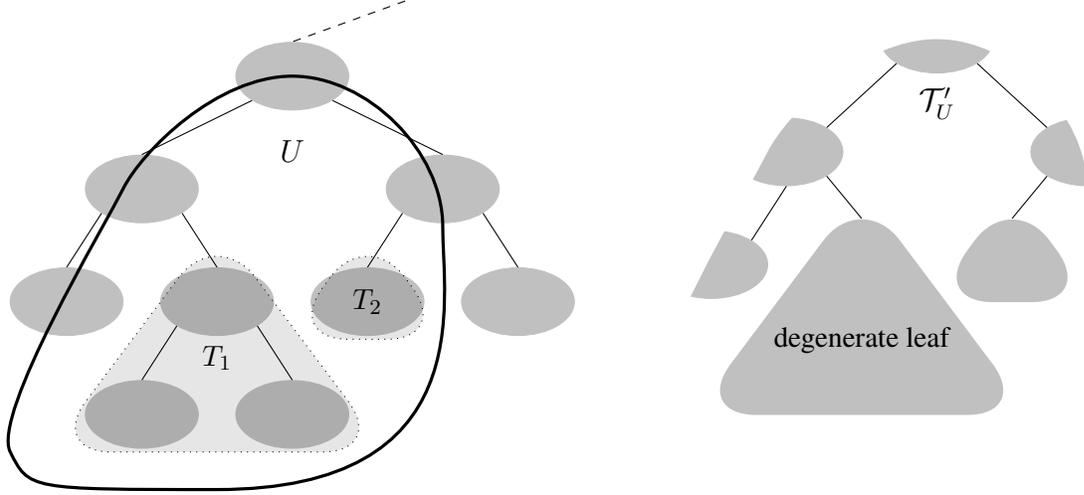


Figure 2: Induction step: reducing the effective treewidth by creating degenerate leaves. T_1, T_2 are maximal subtrees rooted at separators of size k , converted to degenerate leaves.

lost at most $c(U)$ of this flow, we have $|f_U| \geq 4c(U)$. The total number of pairs routed is at least

$$\begin{aligned}
\frac{|f'_U|}{360\alpha(p+1)^2 \cdot 12^{k-1}} + \frac{|f_{\bar{U}}|}{360\alpha(p+1)^2 \cdot 12^k} &\geq \frac{12|f'_U| + |f_{\bar{U}}|}{360\alpha(p+1)^2 \cdot 12^k} \\
&\geq \frac{6|f_U| + |f_{\bar{U}}|}{360\alpha(p+1)^2 \cdot 12^k} \\
&\geq \frac{|f_U| + |f_{\bar{U}}| + c(U)}{360\alpha(p+1)^2 \cdot 12^k} \\
&\geq \frac{|f|}{360\alpha(p+1)^2 \cdot 12^k},
\end{aligned}$$

which establishes the induction step for k . A very similar analysis shows the slightly stronger bound for treewidth p graphs — we simply have to use the stronger induction hypothesis in the preceding calculations and observe that the base case analysis also proves the desired stronger hypothesis.

Proof of Lemma 4.2: Recall that \mathcal{T}_U is a (k, p) -degenerate decomposition for $G[U]$ and that f_U is flush with respect to \mathcal{T}_U . However, we wish to find a $(k-1, p)$ -degenerate decomposition. Recall that U is disjoint from V_e , the separator of size k associated with an edge e of \mathcal{T} . We can assume that \mathcal{T} is rooted and without loss of generality that \mathcal{T}_U is a sub-tree of $\mathcal{T}(e)$. The reason that \mathcal{T}_U may not be a $(k-1, p)$ -tree decomposition is that it may contain an edge e' not incident to a degenerate leaf such that $|V_{e'}| = k$. $V_{e'}$ was then also a separator of G since \mathcal{T} is a (k, p) -tree decomposition. By centrality of U , every node that $V_{e'}$ separates from V_e must then be in U ; therefore, letting $U' = V(\mathcal{T}_{e'})$ (the union of all bags contained in $\mathcal{T}_{e'}$), we have $U' \subset U$. We claim that we can route $x(v)/10$ from each $v \in U'$ to $V_{e'}$ in $G' = G_{e'}$ (this is the graph induced by $G[U']$ but edges between the separator nodes $V_{e'}$ removed); this follows from the minimality of U since a cutset induced by $W \subset U' \setminus V_{e'}$ is also a cutset of G . We define a $(k-1, p)$ -tree decomposition \mathcal{T}'_U , by contracting every maximal subtree of \mathcal{T}_U rooted at such a separator of size k . Each such subtree identifies a new degenerate leaf in \mathcal{T}'_U . However, the flow f_U may not be flush with \mathcal{T}'_U due to the creation of new degenerate leaves. We try to amend this by dropping flow paths with end points in a degenerate leaf L that does not intersect the separator S_L . Let f'_U be the residual flow; observe that by definition f'_U is flush with respect to \mathcal{T}'_U . Two cases arise.

- If $|f'_U| \geq |f_U|/2$ then we have the desired degenerate $(k - 1, p)$ -decomposition of $G[U]$.
- Else at least $|f_U|/2$ of the flow is being routed completely within the new degenerate leaves L . Moreover, these graphs and flows are edge-disjoint. Note also that for each terminal v in such an L , we have that the terminals involved can route $x(v)/10$ to its S_L . We can thus obtain a constant fraction of the profit by applying the method from Proposition 3.4, to every new leaf. In particular, we route at least $\frac{1}{360k} \cdot |f_U|/2$ of the pairs. In this case, we no longer need to recurse on \mathcal{T}_U .

This finishes the proof of the lemma.

Finally, the main inductive claim implies the claimed algorithmic results since we can start with a proper p -decomposition \mathcal{T} of $G \in \mathcal{G}_p$ (viewed as a (p, p) -tree decomposition with no degenerate leaves) and an arbitrary multiflow on its support (since there are no degenerate leaves the flushness is satisfied) in order to begin our induction. This finishes the proof. \square

5 MEDP in Bounded Genus Graphs

In this section we consider MAX EDP and obtain an approximation ratio that depends on the genus of the given graph. Throughout we use g to denote the genus of the given graph G ; we assume that $g > 0$ since we already understand planar graphs. We assume that the given instance of MAX EDP is a matching instance in which the terminals are degree 1 leaves and that each terminal participates in exactly one pair. An instance of MAX EDP with this restriction is characterized by a tuple (G, X, M) where G is the graph and X is the set of terminals and M is a matching on the terminals corresponding to the given pairs. We also assume that the degree of each node is at most 4; this can be arranged without changing the genus by replacing a high-degree node by a grid (see [3] for the description for planar graphs which generalizes easily for any surface). Let \bar{x} be a feasible fractional solution to the multicommodity flow based linear programming relaxation. We use the terminology $\text{val}(x)$ to denote $\sum_i x_i$, the fractional amount of flow routed by x . We let β_g denote the flow-cut gap for product-multicommodity flow instances in a graph of genus g ; this is known to be $O(\log(g + 1))$ [18]. We implicitly assume that β_g is an effective upper bound on the flow-cut gap in that there is a polynomial-time algorithm that outputs a sparse cut no worse than β_g times the maximum concurrent flow for a given product multicommodity instance on a genus g graph. The main result is the following.

Theorem 5.1. *Let \bar{x} be a feasible fractional solution to an MAX EDP instance in a graph of genus $g > 0$. Then $\Omega(\text{val}(\bar{x})/\gamma_g)$ pairs can be routed with congestion 3 where $\gamma_g = O(g \log^2(g + 1))$.*

As we remarked previously we do not have currently have a polynomial-time algorithm for the guarantee that we establishes in the preceding theorem. There are three high-level ingredients in establishing the preceding theorem.

- A constant factor approximation with congestion 2 for planar graphs based on the LP relaxation [7, 23].
- An $O(g)$ -approximation with congestion 3 for graphs with genus g when the terminals are well-linked. This extends the result in [5] for planar graphs to bounded genus graphs via the use of grid minors.
- A modification of the well-linked decomposition of [5] that terminates the decomposition when the graph is planar even if the terminals are not well-linked.

We first give some formal definitions on well-linked sets; the material follows [5] closely.

Well-linked Sets: Let $X \subseteq V$ be a set of nodes and let $\pi : X \rightarrow [0, 1]$ be a weight function on X . We say X is π -flow-well-linked in G if there is a feasible multicommodity flow in G for the following demand matrix: between every unordered pair of terminals $u, v \in X$ there is a demand $\pi(u)\pi(v)/\pi(X)$ (other node pairs have zero demand). We say that X is π -cut-well-linked in G if $|\delta(S)| \geq \pi(S \cap X)$ for all S such that $\pi(S \cap X) \leq \pi(X)/2$. It can be checked easily that if a set X is π -flow-well-linked in G , then it is $\pi/2$ -cut-linked. If $\pi(u) = \alpha$ for all $u \in X$, we say that X is α -flow(or cut)-well-linked. If $\alpha = 1$ we simply say that X is well-linked. Given $\pi : X \rightarrow [0, 1]$ one can check in polynomial time whether X is π -flow-well-linked or not via linear programming.

One can efficiently find an approximate sparse cut if X is not π -flow-well-linked via the algorithmic aspects of the product flow-cut gap; the lemma below follows from [18].

Lemma 5.2. *Let $G = (V, E)$ be a graph of genus at most $g > 1$. Let $X \subseteq V$ and $\pi : X \rightarrow [0, 1]$. There is a polynomial-time algorithm that given G, X and π decides whether X is π -flow-well-linked in G and if not outputs a set $S \subseteq V$ such that $\pi(S) \leq \pi(V \setminus S)$ and $|\delta(S)| \leq \beta_g \cdot \pi(S)$ where $\beta_g = O(\log(g + 1))$.*

We now formally state the theorems that correspond to the high-level ingredients. The first is a constant factor approximation for routing in planar graphs.

Theorem 5.3 ([7, 23]). *Let \bar{x} be a feasible fractional solution to an MAX EDP instance in a planar graph. Then there is a polynomial-time algorithm that routes $\Omega(\text{val}(\bar{x}))$ pairs with congestion 2.*

The second ingredient is an $O(g)$ -approximation if the terminals are well-linked. More precisely we have the following theorem which we prove in Section 5.2.

Theorem 5.4. *Let \bar{x} be a feasible fractional solution to an MAX EDP instance in a graph of genus $g > 0$. Moreover, suppose the terminal set X is π -flow-well-linked in G where $\pi(v) = \rho \cdot x(v)$ for some scalar $\rho \leq 1$. Then there is an algorithm that routes $\Omega(\rho \cdot \text{val}(\bar{x})/g)$ pairs with congestion 3.*

The last ingredient is the following adaptation of the well-linked decomposition from [5].

Theorem 5.5. *Let (G, X, M) be an instance of MAX EDP in a graph of genus at most g and let \bar{x} be a feasible fractional solution. Then there is a polynomial-time algorithm that decomposes the given instances into several instances $(G_1, X_1, M_1), \dots, (G_h, X_h, M_h)$ with the following properties.*

- *The graphs G_1, G_2, \dots, G_h are node-disjoint subgraphs of G .*
- *For $1 \leq j \leq h$, $M_j \subseteq M$ and the end points of M_j are in G_j .*
- *For $1 \leq j \leq h$, there is a feasible fractional solution \bar{x}^j for the instance (G_j, X_j, M_j) such that $\sum_j \text{val}(\bar{x}^j) = \Omega(\text{val}(\bar{x}))$.*
- *For $1 \leq j \leq h$, G_j is planar or the terminals X_j are $\bar{x}^j/(10\beta_g \log(g + 1))$ -flow-well-linked in G_j .*

Assuming the preceding three theorems we finish the proof of Theorem 5.1. Let G be a graph of genus $\leq g$. We apply the decomposition given by Theorem 5.4 which reduces the original problem instance (G, X, M) to a collection of separate instances $(G_1, X_1, M_1), \dots, (G_h, X_h, M_h)$. Note that pairs in these new instances are from the original instance and routings in these instances can be combined into a routing in the original graph G since the G_1, G_2, \dots, G_h are node-disjoint (and hence also edge-disjoint). The total fractional solution value in the new instances is $\Omega(\text{val}(\bar{x})/\beta_g \log(g + 1))$. If G_j is planar we use Theorem 5.3 to route $\Omega(\text{val}(\bar{x}^j))$ pairs from M_j in G_j with congestion 2. If G_j is not planar, then X_j is $\bar{x}^j/(10\beta_g \log(g + 1))$ -flow-well-linked. Then, via Theorem 5.4, we can route $\Omega(\text{val}(\bar{x}^j)/(g\beta_g \log(g + 1)))$ pairs from M_j in G_j with congestion 4. Thus we route in total $\Omega(\sum_j \text{val}(\bar{x}^j)/(g\beta_g \log(g + 1)))$ pairs. Since $\sum_j \text{val}(\bar{x}^j) = \Omega(\text{val}(\bar{x}))$, we route a total of $\Omega(\text{val}(\bar{x})/(g\beta_g \log(g + 1))) = \Omega(\text{val}(\bar{x})/(g \log^2(g + 1)))$ pairs from M in G with congestion 3.

5.1 Proof of Theorem 5.5

We start with a well-known fact.

Proposition 5.6. Let $G = (V, E)$ be a connected graph of genus g . Let $S \subset V$ be such that $G_1 = G[S]$ and $G_2 = G[V \setminus S]$ are both connected. Then $g_1 + g_2 \leq g$ where g_1 is the genus of G and g_2 is the genus of G_2 .

The algorithm below is an adaptation of the well-linked decomposition algorithm from [5] that recursively partitions a graph if the terminals are not well-linked (with a certain parameter). In the adaptation below we stop the partitioning if the graph becomes planar even if the terminals are not well-linked. We start with an instance (G, X, M) and an associated fractional solution \bar{x} . We fix a particular selection of flow paths for the solution \bar{x} . The algorithm recursively cuts G into subgraphs by removing edges. The flow paths that use the removed edges are lost and each connected component retains flow corresponding to those flow paths which are completely contained in that component.

Decomposition Algorithm:

1. If $\text{val}(\bar{x}) < 10\beta_g \log(g+1)$ or if G is a planar graph stop and output (G, X, M) with fractional solution \bar{x} .
2. Else if X is $\bar{x}/(10\beta_g \log(g+1))$ flow-well-linked in G then stop and output (G, X, M) with fractional solution \bar{x} .
3. Else find a sparse cut $\delta(S)$ such that $x(S) \leq \text{val}(\bar{x})/2$ and $|\delta(S)| \leq 1/(10 \log(g+1)) \cdot x(S)$. Let $G_1 = G[S]$ and $G_2 = G[V \setminus S]$. (Assume wlog that G_1, G_2 are connected). Let (G_1, X_1, M_1) and (G_2, X_2, M_2) be the induced instances on G_1 and G_2 with fractional solutions \bar{x}^1 and \bar{x}^2 respectively. Recurse separately on (G_1, X_1, M_1) and (G_2, X_2, M_2) .

We now prove that the above algorithm outputs a decomposition as stated in Theorem 5.5. The only property that is non-trivial to see is the one about the total flow retained in the decomposition. As in [5] it is easier to upper bound the total number of edges cut by the algorithm which in turn upper bounds the total amount of flow from the original solution \bar{x} that is lost during the decomposition. We write a recurrence for this as follows. Let $L(a, r)$ be total number of edges cut by the algorithm if a is the amount of flow in the graph and $r \leq g$ is the genus. The base case is $L(a, 0) = 0$ since the algorithm stops when the graph is planar. The algorithm cuts and recurses only when the terminals are not $\bar{x}/(10\beta_g \log(g+1))$ flow-well linked. Suppose H is the current graph with flow value $a = \sum_v x(v)/2$ and genus r , and H is partitioned into H_1 and H_2 . Let a_1 and a_2 be the total flow values in H_1 and H_2 and let r_1 and r_2 be their genus respectively. We have $a_1 + a_2 \leq a$ and by Proposition 5.6 we have $r_1 + r_2 \leq r$. We claim that the number of edges cut is at most $\frac{1}{4 \log(g+1)} \min\{a_1, a_2\}$. To see this, suppose $\delta(S)$ is the sparse cut in H that resulted in H_1 and H_2 with $H_1 = H[S]$ and $H_2 = H[V \setminus S]$ and $a_1 = \min\{a_1, a_2\}$. Let \bar{x} be the fractional solution in H and \bar{x}' the solution after the partitioning. We have $a = x(V(H))/2$. Since the terminals are not $\bar{x}/(10\beta_g \log(g+1))$ flow-well-linked in H , by Lemma 5.2, $|\delta(S)| \leq \beta_g \cdot x(S)/(10\beta_g \log(g+1)) \leq x(S)/(10 \log(g+1))$. We have $a_1 = x'(S)/2$. Moreover, $x'(S) \geq x(S) - 2|\delta(S)|$ since a flow path p crossing the cut with flow f_p can contribute at most $2f_p$ to the reduction of the marginal values of $x(S)$ at its end points. Thus $2a_1 = x'(S) \geq x(S) - 2|\delta(S)| \geq (10 \log(g+1) - 2)|\delta(S)| \geq 8|\delta(S)|$ since $g \geq 1$, which implies that $|\delta(S)| \leq a_1/4$, as claimed.

Therefore we have the following recurrence for the total number of edges cut in the overall decomposition:

$$L(a, r) \leq L(a_1, r_1) + L(a_2, r_2) + \frac{1}{4 \log(g+1)} \min\{a_1, a_2\}.$$

We prove by induction on r and number of nodes of G that for $r \leq g$, $L(a, r) \leq \frac{\log(r+1)}{2 \log(g+1)} \cdot a$. We had already seen the base case with $r = 0$ since $L(a, r) = 0$. By induction $L(a_1, r_1) \leq \frac{\log(r_1+1)}{2 \log(g+1)} \cdot a_1$ and $L(a_2, r_2) \leq \frac{\log(r_2+1)}{2 \log(g+1)} \cdot a_2$. Since $r_1 + r_2 \leq r$, $\min\{r_1 + 1, r_2 + 1\} \leq (r + 1)/2$. It is not hard to see that $a_1 \log(r_1 + 1) + a_2 \log(r_2 + 1) \leq (a_1 + a_2) \log(r + 1) - \min\{a_1, a_2\} \log 2$. Using the recurrence.

$$\begin{aligned}
L(a, r) &\leq L(a_1, r_1) + L(a_2, r_2) + \frac{1}{4 \log(g+1)} \min\{a_1, a_2\} \\
&\leq \frac{\log(r_1 + 1)}{2 \log(g+1)} \cdot a_1 + \frac{\log(r_2 + 1)}{2 \log(g+1)} \cdot a_2 + \frac{1}{4 \log(g+1)} \min\{a_1, a_2\} \\
&\leq \frac{\log(r + 1)}{2 \log(g+1)} (a_1 + a_2) - \frac{\log 2}{2 \log(g+1)} \min\{a_1, a_2\} + \frac{1}{4 \log(g+1)} \min\{a_1, a_2\} \\
&\leq \frac{\log(r + 1)}{2 \log(g+1)} \cdot a.
\end{aligned}$$

Thus $L(\text{val}(\bar{x}), g) \leq \frac{\log(g+1)}{2 \log(g+1)} \text{val}(\bar{x}) \leq \text{val}(\bar{x})/2$. Hence the total flow that remains after the decomposition is at least $\text{val}(\bar{x})/2$.

5.2 Proof of Theorem 5.4

We start with a grouping technique from [5] that boosts the well-linkedness.

Theorem 5.7. *Let \bar{x} be a feasible fractional solution to an MAX EDP instance (G, X, M) . Moreover, suppose the terminal set X is π -flow-well-linked in G where $\pi(v) = \rho \cdot x(v)$ for some scalar $\rho \leq 1$. Then there is a polynomial-time algorithm that routes $\Omega(\rho \cdot \text{val}(\bar{x}))$ pairs of M edge-disjointly or outputs a new instance (G, X', M') where $M' \subset M$ and X' is well-linked and $|M'| = \Omega(\rho|M|)$.*

Using the preceding theorem we assume that we are working with an instance (G, X, M) where X is well-linked. Recall that G has genus at most g and degree of each node is at most 4. We use the observation below (see [3]) that relates the size of a well-linked set and the treewidth.

Lemma 5.8. *Let G be a graph with maximum degree Δ and let $X \subseteq V$ be a well-linked set in G . Then the treewidth of G is $\Omega(|X|/\Delta)$.*

Thus we can assume that G has treewidth $\Omega(|X|)$. Demaine et al. [12] showed the following theorem on the size of a grid minor in graphs of genus g following the work of Robertson and Seymour.

Theorem 5.9 ([12]). *Let G be a graph of genus at most g . Then G has a grid minor of size $\Omega(h/g)$ where h is the treewidth of G .*

Following the scheme from [5], one can use the grid minor as a cross bar to route a large number of pairs from M provided we can route $\Omega(|X|/g)$ terminals to the “interface” of the grid-minor of size $\Omega(|X|/g)$ that is guaranteed to exist in G . We can view the grid minor as rows and columns. In our current context we take every other node in the first row of the grid as the interface of the grid-minor. Each node v of the minor corresponds to a subset of connected nodes A_v in original graph G that are contracted to form v . To simplify notation we say that $S \subset V$ is the interface of a grid-minor in G if $|S \cap A_v| = 1$ for each interface node v of the grid-minor. The following lemma is essentially implicit in previous work, in particular [4] used in the context of planar graphs.

Lemma 5.10. *Suppose $G = (V, E)$ contains a $h \times h$ grid as a minor. Let $S \subseteq V$ be the interface of the grid-minor. Then S is well-linked in G . Moreover, any matching M on S is routable in G with congestion 2.*

The key technical difficulty is to ensure that G contains a large grid-minor whose interface is reachable from the terminals X . In a sense, the grid-minor’s existence is shown via the well-linkedness of X and hence there should be such a “reachable” grid-minor. The following theorem formalizes the existence of the desired grid-minor.

Theorem 5.11. *Let G be a graph of genus $g > 0$. Suppose X is a well-linked set in G . Then G contains a $h \times h$ grid-minor with interface S such that $h = \Omega(|X|/g)$ and at least $h/8$ edge-disjoint paths in G from X to S .*

Proving the preceding theorem formally requires work. In [4] an argument tailored to planar graphs was used to prove a similar theorem, and in fact a polynomial-time algorithm was given to find the desired grid-minor. In that same paper a general deletable edge lemma [2] was announced (though never published); in the appendix we give a streamlined proof based on the original manuscript. We postpone the proof of the preceding theorem and outline how to complete the proof of Theorem 5.4. We start with our well-linked set X , and via Theorem 5.11, find a grid-minor of size $h = \Omega(|X|/g)$ such that there are $h/8$ edge-disjoint paths from X to the interface S . Let $X' \subset X$ be $h/8$ terminals that are the end points of these edge-disjoint paths. Recall that we are interested in routing a given matching M on X . If X' contains a “large” sub-matching $M' \subset M$ then we can use the grid-minor to route M' with congestion 3 as follows. Let $S' \subset S$ be end points of the paths from X' to S . Clearly M' induces a matching on S' and the grid-minor can route this matching with congestion 2. Patching this routing with the paths from X' to S' gives the desired congestion 3 routing of M' in G . However, it may be the case that $|M'|$ is very small, even zero, because X' only contains one end point from every edge in M . However, here, we can use the fact that X and S are well-linked in G to argue that we can route any desired subset of X of sufficiently large size to S . Thus, we can assume that indeed $|M'|$ is a constant fraction of $|X'|$. This was essentially done in [3] and details are also included in the appendix (see Lemma A.3). This shows the number of pairs from M that can be routed with congestion 3 in G is a constant fraction of h , the size of the grid-minor. Since $h = \Omega(|X|/g)$ we route $\Omega(|X|/g)$ pairs from M . This finishes the proof of Theorem 5.4.

Now we come to the proof of Theorem 5.11. This is done in Section A via the following high-level approach. Suppose X is a well-linked set in G . We are guaranteed that G has a grid-minor of size $h = \Omega(|X|/g)$. Let S be its interface. If there are $h/8$ edge-disjoint paths from X to S then we are done. Otherwise the claim is that there is an edge e such that X is well-linked in $G - e$. This is established in Theorem A.4. In other words, if we start with a graph which is edge-minimal subject to X being well-linked, then following the procedure above yields a grid-minor that is reachable from X . The reason that this does not immediately lead to a polynomial-time algorithm to find such a grid-minor is the fact that checking whether a given set X is not well-linked is NP-Complete. Note that we can check if X is flow-well-linked but the deletable edge lemma we have works with the notion of cut-well-linkedness which is hard to check. In [4] a poly-time deletable edge lemma was obtained for the special case of planar graphs and we believe that it can be extended to the case of genus g graphs but the technical details are quite involved.

6 Open Problems and Concluding Remarks

Resolving Conjecture 1 is the main open problem that arises from this work. In particular, does a planar graph with a single vortex satisfy the CFCC property? This is the key technical obstacle.

Theorem 4.1 loses an approximation factor that is exponential in k . Is this necessary? In particular, the known integrality gap for the flow LP on graphs of treewidth k is only $\Omega(k)$; this comes from the grid example.

Moreover, the congestion we obtain for \mathcal{G}_k is $\beta + 3$ where β is the congestion guaranteed for the class \mathcal{G} . Can this be improved further, say to β or $\beta + 1$?

We believe that Theorem 5.1 can be made algorithmic and also conjecture that the congestion bound can be improved to 2. The key bottleneck is to find an algorithmic proof of Theorem 5.11.

Acknowledgments: We thank Sanjeev Khanna for allowing us to include Section A in the paper for the sake of completeness.

References

- [1] M. Andrews, J. Chuzhoy, V. Guruswami, S. Khanna, K. Talwar, and L. Zhang. Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs. *Combinatorica*, 30(5):485–520, 2010.
- [2] C. Chekuri, S. Khanna, and F. B. Shepherd. A deletable edge lemma for general graphs. Manuscript, 2004.
- [3] C. Chekuri, S. Khanna, and F. B. Shepherd. Edge-disjoint paths in planar graphs. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 71–80, 2004.
- [4] C. Chekuri, S. Khanna, and F.B. Shepherd. Edge-disjoint paths in planar graphs. In *Proc. of IEEE FOCS*, pages 71–80, 2004.
- [5] C. Chekuri, S. Khanna, and F.B. Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 183–192. ACM, 2005.
- [6] C. Chekuri, S. Khanna, and F.B. Shepherd. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory of Computing*, 2(7):137–146, 2006.
- [7] C. Chekuri, S. Khanna, and F.B. Shepherd. Edge-disjoint paths in planar graphs with constant congestion. *SIAM Journal on Computing*, 39:281–301, 2009.
- [8] C. Chekuri, S. Khanna, and F.B. Shepherd. A note on multiflows and treewidth. *Algorithmica*, 54(3):400–412, 2009.
- [9] Chandra Chekuri, F. Bruce Shepherd, and Christophe Weibel. Flow-cut gaps for integer and fractional multiflows. *Journal of Combinatorial Theory, Series B*, 103(2):248 – 273, 2013.
- [10] J. Chuzhoy. Routing in undirected graphs with constant congestion. In *Proceedings of the 44th Symposium on Theory of Computing*, pages 855–874. ACM, 2012.
- [11] J. Chuzhoy and S. Li. A polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2. In *Proc. of IEEE FOCS*, 2012.
- [12] Erik D Demaine, Fedor V Fomin, MohammadTaghi Hajiaghayi, and Dimitrios M Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *J. ACM*, 52(6):866–893, 2005.
- [13] Steven Fortune, John Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111 – 121, 1980.

- [14] 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.
- [15] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, trees and ℓ_1 -embeddings of graphs. *Combinatorica*, 24(2):233–269, 2004.
- [16] Philip Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 682–690, New York, NY, USA, 1993. ACM.
- [17] J. Kleinberg and E. Tardos. Approximations for the disjoint paths problem in high-diameter planar networks. *Journal of Computers and System Sciences*, 57(1):61–73, 1998.
- [18] J.R. Lee and A. Sidiropoulos. Genus and the geometry of the cut graph. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 193–201. Society for Industrial and Applied Mathematics, 2010.
- [19] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.
- [20] A. Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 3–12. IEEE, 2009.
- [21] N. Robertson and P.D. Seymour. Graph minors: a survey. *Surveys in combinatorics*, 103:153–171, 1985.
- [22] Neil Robertson and P.D Seymour. Graph minors. xvi. excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43 – 76, 2003.
- [23] L. Seguin-Charbonneau and F.B. Shepherd. Maximum edge-disjoint paths in planar graphs with congestion 2. In *Proc. of IEEE FOCS*, pages 200–209, 2011.
- [24] David Shmoys. *Approximation Algorithms for NP-Hard Problems*, chapter Cut problems and their application to divide-and-conquer. PWS series in computer science. PWS Pub. Co., 1997.
- [25] Éva Tardos and Vijay V. Vazirani. Improved bounds for the max-flow min-multicut ratio for planar and k , r -free graphs. *Information Processing Letters*, 47(2):77 – 80, 1993.

A A Deletable Edge Lemma

The following results are from the unpublished note [2]. We have streamlined the original proofs and include it for completeness.

Throughout this section we use well-linked to mean cut-well-linked. We say that $S \subseteq V$ is *routable* to $T \subseteq V$ in G if there are $|S|$ edge-disjoint paths from S to T in G such that each node in $S \cup T$ is the end point of at most one of the paths. We allow paths of length 0 if for example a node belongs to $S \cap T$. If S can be routed to T then of course $|S| \leq |T|$.

Given a graph G , two sets of nodes A and B , we define an *auxiliary graph* $G(A, B)$ obtained by attaching a node s_A (or just s) and a node t_B (or just t). The node s has an edge to each node in A and t has an edge to each node in B . The max s - t flow in $G(A, B)$ identifies the maximum routable set from A to B .

Proposition A.1. If H is well-linked in G then for any $X, Y \subset H$ and $|X| \leq |Y|$, X is routable to Y in G .

Proof. Consider the auxiliary graph $G' := G(X, Y)$. It is sufficient to prove that the s - t mincut in G' is $|X|$. Let $\delta(S')$ be an s - t minimum cut in G' with $s \in S'$. Let $S = S' - \{s\}$. We assume wlog that $|S \cap H| \leq |H|/2$, otherwise we can work with $V(G') \setminus S$. Since H is well-linked it follows that $|\delta_G(S)| \geq |S \cap H|$. We have that $|\delta_{G'}(S')| \geq |X \setminus S| + |\delta_G(S)| \geq |X \setminus S| + |S \cap H| \geq |X \setminus S| + |S \cap X| \geq |X|$. \square

Lemma A.2. *Let H_1 and H_2 be two disjoint well-linked sets in G . Suppose $A \subset H_1$ is routable to $B \subset H_2$ and $|A| \leq |H_1|/2$. Then given any $A' \subset H_1$ with $|A'| \leq |A|/2$, A' is routable to B .*

Proof. Consider the auxiliary graph $G' := G(A', B)$. Let $\delta_{G'}(S')$ be an s - t minimum cut in G' with $s \in S'$. Let $S = S' - \{s\}$. We argue that $|\delta_{G'}(S')| \geq |A'|$ which proves the lemma. Let $a = |A|$. We have the equality that

$$|\delta_{G'}(S')| = |\delta_G(S)| + |A' \setminus S| + |B \cap S|. \quad (2)$$

We consider two cases. In the first case $|S \cap A| \geq a/2$. Since A is routable to B it follows that $|\delta_G(S)| \geq a/2 - |S \cap B|$. Hence from Equation 2 we have that $|\delta_{G'}(S')| \geq a/2 \geq |A'|$.

In the second case, $|S \cap A| < a/2$ which implies that $|(V - S) \cap A| \geq a/2$. Let $Y = (V - S) \cap A$. Therefore $|Y| \geq a/2$. By Proposition A.1 we have that $S \cap A'$ is routable to Y . It follows that $|\delta_G(S)| \geq |S \cap A'|$. From Equation 2 $|\delta_{G'}(S')| \geq |S \cap A'| + |A' \setminus S| = |A'|$. \square

Lemma A.3. *Let H_1 and H_2 be two disjoint well-linked sets in G . Suppose $A \subset H_1$ is routable to $B \subset H_2$. Then given any $A' \subset H_1$ and $B' \subset H_2$ with $|A'| = |B'| \leq |A|/3$, A' is routable to B' .*

Proof. Consider the auxiliary graph $G' := G(A', B')$. Let $\delta_{G'}(S')$ be an s - t minimum cut in G' with $s \in S'$. Let $S = S' - \{s\}$ and $T = V(G) - S$. We argue that $|\delta_{G'}(S')| \geq |A'|$ which proves the lemma. Let $a = |A|$. We have the equality that

$$|\delta_{G'}(S')| = |\delta_G(S)| + |A' \setminus S| + |B' \cap S|. \quad (3)$$

We can assume that $|B| = |A|$. Consider a fixed routing from A to B . We call $u \in A, v \in B$ a pair if u and v are joined by a path in the fixed routing. Suppose the cut $\delta_G(S)$ separates at least $a/3$ pairs. Then clearly $|\delta_G(S)| \geq a/3$ and we are done. Therefore either S contains at least $a/3$ pairs or T contains $a/3$ pairs. If S contains at least $a/3$ pairs, then S contains $a/3$ nodes from H_2 . Then by the well-linkedness of H_2 we see that $|\delta_G(S)| \geq |T \cap B'|$ and hence $|\delta_{G'}(S')| \geq |T \cap B'| + |B' \cap S| = |B'|$. If T contains at least $a/3$ pairs, then we can use the well-linkedness of H_1 to argue that $|\delta_{G'}(S')| \geq |A'|$. \square

We note that Lemmas A.2 and A.3 are tight.

Let H_1 and H_2 be two disjoint well-linked sets in G . Let γ be the size of a min-cut s - t cut in the auxiliary graph induced by H_1 and H_2 . Thus there is a subset A of H_1 , and subset B of the H_2 such that $|A| = |B| = \gamma$ and A is routable to B .

The main result in this section is the following:

Theorem A.4. *If $\gamma < |H_2|/8$, then there is an edge e in G such that H_1 is well-linked in $G - e$.*

For simplicity we assume that each node in H_1 has degree one in G . This is without loss of generality. For each node $v \in H_1$ we can add a dummy node v' and attach v' to v by a single edge; the set $H'_1 = \{v' | v \in H_1\}$ is easily seen to be well-linked and the routability of H'_1 to H_2 is the same as that from H_1 to H_2 . Further, none of the edges (v', v) is deletable and hence any edge that we show is deletable in the modified instance is an edge in the original graph.

Let $k = |H_2|$. To prove the above theorem we consider an s - t minimum cut $\delta(S')$ in the auxiliary graph $G' := G(H_1, H_2)$. Let $S = S' - s$ and $T = V(G) - S$. We can evidently choose such a cut so that $|T \cap H_1| = 0$ since each node in H_1 is a leaf. We also have that $|T \cap H_2| \geq |H_2| - \gamma \geq |H_2|/2$. Given the existence of T , we may now choose a γ -cut $\delta(M)$ in G with M minimal subject to satisfying the property that $|M \cap H_2| \geq |H_2|/2 = k/2$ and $|M \cap H_1| = 0$. Note that M is not a stable set for otherwise $|\delta(M)| \geq k/2$ but by definition $|\delta(M)| = \gamma$.

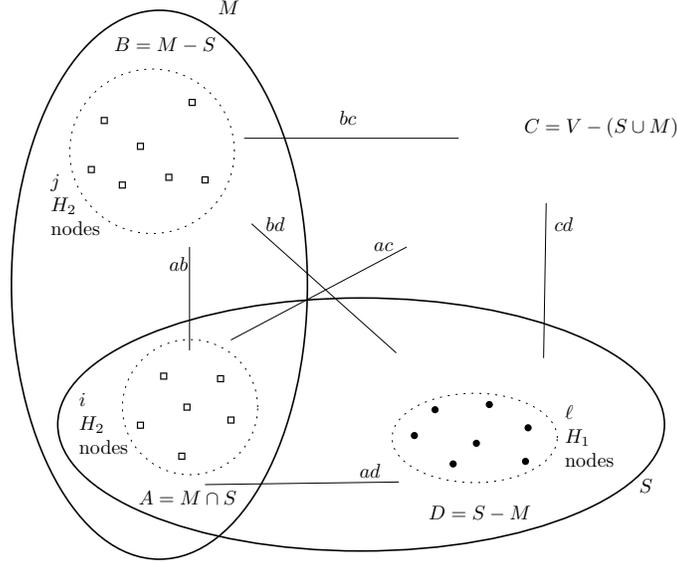


Figure 3: Illustration for proof of removable edge from M .

Lemma A.5. *Any edge with both ends in M is deletable.*

Proof. Let e be an edge inside M . Suppose e is not deletable. Then there is a light set S with respect to H_1 that is tight and e crosses S . That is, $|S \cap H_1| = \ell \leq |H_1|/2$ and $|\delta(S)| = \ell$ and $e \in \delta(S)$. Let i and j be the number of H_2 nodes in $M \cap S$ and $M - S$ respectively. Recall that M does not contain any H_1 nodes. See Fig 3.

We first observe that $(i+j) \geq k - \gamma \geq 7k/8$, otherwise more than γ nodes are in $V - M$ and $|\delta(V - M)| = |\delta(M)| = \gamma$; this would violate the well-linkedness of H_2 . Second, by submodularity and symmetry of the function $|\delta_G| : 2^V \rightarrow \mathbb{Z}_+$, we have,

$$|\delta(M)| + |\delta(S)| \geq |\delta(M \cap S)| + |\delta(M \cup S)|,$$

and

$$|\delta(M)| + |\delta(S)| \geq |\delta(M - S)| + |\delta(S - M)|.$$

We have $|\delta(M)| = \gamma$ and $|\delta(S)| = \ell$. Both $M \cup S$ and $S - M$ have exactly ℓ nodes of H_1 ; since H_1 is well-linked, $|\delta(M \cup S)| \geq \ell$ and $|\delta(S - M)| \geq \ell$. Thus, from the above submodularity inequalities, we get a contradiction if we can prove that $|\delta(M \cap S)| > \gamma$ or $|\delta(S - M)| > \gamma$. If $i \geq k/2$ we have $|\delta(M \cap S)| > \gamma$ for otherwise $M \cap S$ contradicts the minimality of M . Similarly if $j \geq k/2$ we have $|\delta(M - S)| > \gamma$ for otherwise $M - S$ contradicts the minimality of M .

We now consider the case that $i < k/2$ and $j < k/2$. Since $(i+j) \geq 7k/8$ we have $7k/16 \leq \max\{i, j\} < k/2$. If $i = \max\{i, j\}$ then by well-linkedness of H_2 , $|\delta(M \cap S)| \geq i \geq 7k/16 > \gamma$. If $j = \max\{i, j\}$

then again by well-linkedness of H_2 , $|\delta(M - S)| \geq j \geq 7k/16 > \gamma$. In both cases we get the desired contradiction. \square