SIAM J. COMPUT.
Vol. 39, No. 5, pp. 1856–1887

EDGE DISJOINT PATHS IN MODERATELY CONNECTED GRAPHS*

SATISH RAO† AND SHUHENG ZHOU‡

Abstract. We study the edge disjoint paths (EDP) problem in undirected graphs: Given a graph G with n nodes and a set $\mathcal{T}$ of pairs of terminals, connect as many terminal pairs as possible using paths that are mutually edge disjoint. This leads to a variety of classic NP-complete problems, for which approximability is not well understood. We show a polylogarithmic approximation algorithm for the undirected EDP problem in general graphs with a moderate restriction on graph connectivity; we require the global minimum cut of G to be $\Omega(\log^5 n)$. Previously, constant or polylogarithmic approximation algorithms were known for trees with parallel edges, expanders, grids, grid-like graphs, and, most recently, even-degree planar graphs. These graphs either have special structure (e.g., they exclude minors) or have large numbers of short disjoint paths. Our algorithm extends previous techniques in that it applies to graphs with high diameters and asymptotically large minors.

Key words. edge disjoint paths, polylogarithmic approximation, random sampling in cuts

AMS subject classifications. 68W25, 68R10, 90C27, 90C35

DOI. 10.1137/080715093

1. Introduction.

In this paper, we explore approximation for the edge disjoint paths (EDP) problem: Given a graph with n nodes and a set of terminal pairs, connect as many of the specified pairs as possible using paths that are mutually edge disjoint. The EDP problem has a multitude of applications in areas such as VLSI design, routing, and admission control in large-scale, high-speed, and optical networks. Moreover, the EDP problem and its variants have also been prominent topics in combinatorics and theoretical computer science for decades. For example, the celebrated theory of graph minors of Robertson and Seymour [33] gives a polynomial time algorithm for routing all the pairs given a constant number of pairs. However, varying the number of terminal pairs leads to a variety of classic NP-complete problems, for which approximability is an interesting problem. In a recent breakthrough [3], Andrews and Zhang showed an $\Omega(\log^{\frac{1}{3}-\epsilon} n)$ lower bound on the hardness of approximation for the undirected EDP problem.

In this work, we show a polylogarithmic approximation algorithm for the undirected EDP problem in general graphs with a moderate restriction on graph connectivity; we require that there are $\Omega(\log^5 n)$ edge disjoint paths between every pair of vertices; i.e., the global min cut is of size $\Omega(\log^5 n)$. If this moderately connected case holds, we can route $\Omega(\mathsf{OPT}/\operatorname{polylog} n)$ pairs using disjoint paths with congestion 1, where OPT is the maximum number of pairs that one can route edge disjointly for the given EDP instance. Previously, constant or polylogarithmic approximation algorithms were known for trees with parallel edges, expanders, grids, grid-like graphs,

*Received by the editors February 5, 2008; accepted for publication (in revised form) October 6, 2009; published electronically January 27, 2010. A preliminary version of this paper has appeared in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006)*.

http://www.siam.org/journals/sicomp/39-5/71509.html

†Computer Science Division, University of California, Berkeley, CA 94720 (satishr@EECS.berkeley.edu). This author's research was supported in part by NSF award CCF-0515304.

‡Seminar für Statistik, ETH Zürich, CH-8092 Zürich, Switzerland (zhou@stat.math.ethz.ch). This material is based on research sponsored in part by the Army Research Office, under agreement DAAD19-02-1-0389 and NSF grant CNF-0435382. This work was done while the author was visiting UC Berkeley.

and, most recently, even-degree planar graphs [23]. The results rely either on excluding a minor (or other structural properties) or on the fact that very short paths exist. Our algorithm extends previous techniques; for example, our graphs can have high diameters and contain very large minors. We are hopeful that this constraint on the global minimum cut can be removed if congestion on each edge is allowed to be $O(\log \log n)$. Formally, we have the following result.

THEOREM 1.1. *There is a* polylog n*-approximation algorithm for the EDP problem in a general graph $\mathcal{G}$ with minimum cut $\Omega(\log^5 n)$ with high probability.*

1.1. The approach. We begin with a fractional relaxation of the problem, where each terminal pair can route a real-valued amount of flow between 0 and 1, and this flow can be split fractionally across a set of distinct paths. This can be expressed as a linear program (LP) and can be solved efficiently. We denote the value of an optimal fractional LP solution as OPT^*. Our algorithm routes a polylogarithmic fraction of this value using integral edge disjoint paths. The algorithm proceeds by decomposing the graph into well-connected subgraphs, based on OPT^*, so that a subset of the terminal pairs that remain within each subgraph is "well connected," following a decomposition procedure of Chekuri, Khanna, and Shepherd [11]. Then, for each well-connected subgraph G, we construct an expander graph that can be embedded into G using its terminal set. We use a result by Khandekar, Rao, and Vazirani in [22], where they show that one can build an expander graph H on a set of nodes V by constructing $O(\log^2 n)$ perfect matchings $M_1, \ldots, M_{O(\log^2 n)}$ between $O(\log^2 n)$ sets of equal partitions of V in an iterative manner.

Our contribution along this line is to route each perfect matching M_t, $\forall t$, on one of the $O(\log^2 n)$ (edge disjoint) subgraphs of G. The "splitting procedure," motivated by Karger's theorem [20], simply assigns edges of G uniformly at random into $O(\log^2 n)$ subgraphs. Using Karger's arguments, we show that all cuts in each subgraph have approximately the correct size with high probability. Here we crucially use the polylogarithmic lower bound on the min-cut. We then route each matching M_t on a unique split subgraph using a max-flow computation with unit capacities. Thus, we can route all $O(\log^2 n)$ matchings edge disjointly in G and embed an expander graph H integrally with congestion 1 on G.

After we construct such an expander graph H for each G, we route terminal pairs in H greedily via short paths. This is effective since there are plenty of short disjoint paths in an expander graph [7, 24]. Since a node in H maps to a cluster of nodes in G that is connected by a spanning tree, we put a capacity constraint on $V(H)$: we allow only a single path to go through each node. We greedily connect a pair of terminals from G via a path in H while taking both nodes and edges along the chosen path away from H, until no short paths remain between any unrouted terminal pair. For the pairs we indeed route, we know the congestion is 1 in the original graph G, since we use each edge and node in H only once, and edges and nodes of H correspond to disjoint paths of G.

We use a lemma in [17] to show that such a greedy method ensures that we route a sufficiently large number of such pairs; we note that this method was proposed but analyzed somewhat differently by Kleinberg and Rubinfeld [24]. Our analysis is more like that of Obata [30] and yields somewhat stronger bounds. Our approximation factor is $O(\log^{10} n)$. (A breakdown of this factor is described in Theorem 3.4.) Finally, we note that it is possible to improve the approximation factor in this paper using the cut-finding procedures given by Orecchia et al. [31] recently to replace that of the KRV-FINDCUT procedure (cf. Figure 3.1) in our construction of the expander graph

H. Their procedure will result in an expander with a higher expansion factor, the details of which are beyond the scope of the current paper.

1.2. Related work. Much of the recent work on the EDP problem has focused on understanding the polynomial time approximability of the problem. Previously, constant or polylogarithmic approximation algorithms were known for trees with parallel edges [17], expanders [24, 29], grids, grid-like graphs [5, 6, 25, 26], and even-degree planar graphs [23]. For general graphs, the best approximation ratio for the EDP problem in directed graphs is $O(\min(n^{2/3}, \sqrt{m}))$ [8, 27, 28, 34, 35], where m denotes the number of edges in the input graph. This is matched by the $\Omega(m^{\frac{1}{2}-\epsilon})$-hardness of approximation result by Guruswami et al. [19]. For undirected and directed acyclic graphs, the upper bound has been improved to $O(\sqrt{n})$ [13]. For even-degree planar graphs, an $O(\log^2 n)$-approximation [23] was obtained recently.

A variant is the EDP with congestion (EDPwC) problem, where the goal is to route as many terminals as possible, such that at most ω demands can go through any edge in the graph. For the undirected EDPwC problem on planar graphs, for $\omega = 2$ and 4, $O(\log n)$ [10, 11] and constant [12] approximations have been obtained, respectively. For undirected graphs, the hardness results [1] are $\Omega(\log^{1/2-\epsilon} n)$ for the EDP problem and $\Omega(\log^{(1-\epsilon)/(\omega+1)} n)$ for the EDPwC problem. For the directed EDPwC problem with $\omega > 1$, $O(\omega n^{1/\omega})$-approximation algorithms based on randomized rounding of the multicommodity flow relaxation are shown in [34, 28]. The hardness result of $n^{\Omega(1/\omega)}$ is shown in Chuzhoy et al. [15] that for all integer-valued ω satisfying $1 \leq \omega \leq \alpha \log n / \log\log n$, where $\alpha > 0$ is an absolute constant.

A closely related problem is the congestion minimization problem: Given a graph and a set of terminal pairs, connect *all* pairs with integral paths while minimizing the maximum number of paths through any edge. Raghavan and Thompson [32] show that by applying a randomized rounding to a linear relaxation of the problem, one obtains an $O(\log n / \log\log n)$-approximation for both directed and undirected graphs. For hardness of approximation, Andrews and Zhang [2] show a result of $\Omega((\log\log^{1-\epsilon} m))$ for undirected and an almost-tight result [4] of $\Omega(\log^{1-\epsilon} m)$ for directed graphs, improving that of $\Omega(\log\log m)$ by Chuzhoy and Naor [16]; Most recently, Chuzhoy et al. [15] show an $\Omega(\log n / \log\log n)$ hardness result, so that the inapproximation and approximation factors are within constant factors of each other.

Finally, the all-or-nothing flow (ANF) problem [9, 11] is to choose a subset of terminal pairs such that for each chosen pair, one can fractionally route a unit of flow for all the chosen pairs. The hardness result for the undirected ANF problem and the ANF with congestion problem is the same as that of EDP and EDPwC [1]. Currently, there exists an $O(\log^2 n)$-approximation [11] for the ANF problem. Indeed, we build on the techniques developed in this approximation algorithm for the ANF problem.

2. Definitions and preliminaries. We work with graph $G = (V, E)$ with unit-capacity edges, where we allow parallel edges, unless we specify a capacity function for edges explicitly. For a capacitated graph $G = (V, E, c)$, where c is an integer capacity function on edges, one can replace each edge $e \in E$ with $c(e)$ parallel edges. An instance of a routing problem consists of a graph $\mathcal{G} = (V, E)$ and a set of terminal pairs $\mathcal{T} = \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$. Nodes in $\mathcal{T}$ are referred to as terminals.

We note that throughout this paper, we allow a single vertex in $\mathcal{G}$ to appear in at most $O(\log^5 n)$ pairs in $\mathcal{T}$. This restriction comes from our construction of H, an interesting aspect of which is its relation to elements that contribute to our approximation factor (cf. Theorem 3.1). Given an EDP instance $(\mathcal{G}, \mathcal{T})$ with k pairs of terminals, each with a unit demand satisfying the restriction above, we will use the LP relaxation as specified in (2.2a)–(2.2d) to obtain an optimal fractional solution:

$$\mathsf{OPT}^*(\mathcal{G},\mathcal{T}) \;:=\; \max\sum_{i=1}^{k}\sum_{p\in\mathcal{P}_i}\bar{f}(p), \tag{2.1}$$

where $\mathcal{P}_i, \forall i=1,\dots,k$, denotes the set of paths joining s_i and t_i in $\mathcal{G}$, and the optimal solution $\bar{f}(p)$, $\forall p\in\mathcal{P}_i, \forall i=1,\dots,k$ is obtained as follows:

$$\max\sum_{i=1}^{k}x_i \qquad \text{s.t.} \tag{2.2a}$$

$$x_i-\sum_{p\in\mathcal{P}_i}f(p)=0,\;\;\forall 1\le i\le k, \tag{2.2b}$$

$$\sum_{p:e\in p}f(p)\le 1,\;\;\forall e\in E, \tag{2.2c}$$

$$x_i, f(p)\in[0,1],\;\;\forall 1\le i\le k, \forall p. \tag{2.2d}$$

In the text, where we always refer to a single instance, we primarily use OPT^*.

For a cut $(S,\bar{S}=V\setminus S)$ in G, let $\delta_G(S)$, or simply $\delta(S)$ when it is clear, denote the set of edges with exactly one endpoint in S in G. Let $\mathsf{cap}(S,\bar{S})=|\delta_G(S)|$ denote the total capacity of edges in the cut.

DEFINITION 2.1. *A graph $G=(E,V)$ is an α-expander if for every set $S\subset V$, $|S|\le|V|/2$, we have $|\delta_G(S)|\ge\alpha\,|S|$.*

Given a nonnegative weight function $\pi: Y\to\mathbb{R}^+$ on a set of nodes Y in G, we use the following definitions from [11].

DEFINITION 2.2 (see [11]). *A set Y is π-cut-linked in G if $\forall S$ such that $\pi(S\cap Y)=\sum_{y\in S\cap Y}\pi(y)\le\pi(Y)/2$, $|\delta(S)|\ge\pi(S\cap Y)$; we also refer to (G,Y) as a π-cut-linked instance.*

DEFINITION 2.3 (see [11]). *A set Y is π-flow-linked in G if there is a feasible multicommodity flow for the problem with demand $\mathsf{dem}(u,v)=\pi(u)\pi(v)/\pi(Y)$ between every unordered pair of terminals $u,v\in Y$.*

Remark 2.4. Note that this is a product flow with $\mathsf{dem}(u,v)=w(u)w(v)$, where $w(u)=\pi(u)/\sqrt{\pi(Y)}$.

We have the following proposition immediately from the definitions above.

PROPOSITION 2.5 (see [11]). *If a set Y is π-flow-linked in G, then it is $\pi/2$-cut-linked. If Y is π-cut-linked in G, then it is $\pi/\beta(G)$-flow-linked, where $\beta(G)$ is the worst-case min-cut max-flow gap on product multicommodity flow instances on G.*

DEFINITION 2.6 (see [11]). *A set of nodes Y is well linked in G if $\forall S$ such that $|S\cap Y|\le|Y|/2$, $|\delta(S)|\ge|S\cap Y|$.*

Finally, we note that the entire set of important parameters and notation are listed in Table 7.1 at the end of section 7 for reference.

3. Decomposition and an outline of the routing procedure. In this section, we first present Theorem 3.1 regarding a preprocessing phase of our algorithm that decomposes and processes $(\mathcal{G},\mathcal{T})$ into a collection of cut-linked instances with a min-cut $\Omega(\log^3 n)$ in each subgraph. We then state our main theorem with a breakdown of the polylog n-approximation factor. Finally, we give an outline on how we route terminal pairs in each cut-linked instance (G,T); note that, from section 3.1

through the end of the paper, we use G to refer to a subgraph that we obtain through Theorem 3.1, while $\mathcal{G}$ refers to the original input graph.

THEOREM 3.1. *Suppose we are given an EDP instance* $(\mathcal{G}, \mathcal{T})$*, where* $\mathcal{G}$ *has a min-cut of size* $\Omega(\kappa \log^2 n)$*, where* $\kappa = \Omega(\log^3 n)$*, such that a solution* $\bar{f}$ *to the fractional EDP problem, with* x_i*,* $\forall i$*, being specified as in* (2.2a)–(2.2d)*, satisfies* $\forall u \in \mathcal{G}$*,* $\sum_{i:\{s_i=u \cup t_i=u\}} x_i \leq W\beta(\mathcal{G})\lambda(n) = O(\log^5 n)$*, where* $W = \Theta(\log^2 n)$ *and* β, λ *are defined below. Then there is a polynomial time decomposition algorithm, which produces a disjoint set of subgraphs* $G_1, G_2, \ldots$*, and a weight function* $\pi : V(\mathcal{G}) \to \mathbb{R}^+$ *on* $V(\mathcal{G})$*, for which the following hold:*

1. *there are* $\eta_1, \ldots, \eta_k$ *such that* $\forall u$ *in a subgraph* G_j*,*

$$\pi(u) = \sum_{i:s_i=u, t_i \in G_j} \eta_i x_i, \tag{3.1}$$

   which implies that $\forall (s_i, t_i) \in \mathcal{T}$*,* x_i *contributes the same amount of weight to* $\pi(s_i)$ *and* $\pi(t_i)$*; and* $\pi(G_j) := \sum_{u \in G_j} \pi(u)$*;*
2. *the set of nodes* $V(G_j)$ *in each subgraph* G_j *is* π*-cut-linked in* G_j*,* $\forall j$*;*
3. *each subgraph* G_j *has min-cut* $\kappa = \Omega(\log^3 n)$*;*
4. $\forall u$ *in a subgraph* G_j *such that* $\pi(G_j) \geq \Omega(\log^3 n)$*, we have for* $\beta(\mathcal{G}) = O(\log n)$ *as in Proposition* 2.5 *and* $\lambda(n) = 10\beta(\mathcal{G}) \log \mathsf{OPT}^*(\mathcal{G}, \mathcal{T}) = O(\log^2 n)$*,*

$$\pi(u) \leq \sum_{i:s_i=u, t_i \in G_j} x_i / \beta(\mathcal{G})\lambda(n) \leq W;$$

5. $\pi(\mathcal{G}) := \sum_{j=1,2,\ldots} \pi(G_j) = \Omega(\mathsf{OPT}^*/\beta(\mathcal{G})\lambda(n))$.

Remark 3.2. These two parameters are used throughout the paper: (a) $\beta(\mathcal{G}) = O(\log n)$ is the worst-case min-cut–max-flow gap on product commodity flow instances on $\mathcal{G}$; (b) $\lambda(n) = 10\beta(\mathcal{G}) \log \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$; see section 8.1 for details.

The decomposition essentially says that summing across all subgraphs G, a constant fraction of terminal pairs in $\mathcal{T}$ remains (conditions 4 and 5); indeed, we lose only a constant fraction of all pairs (s_i, t_i) in $\mathcal{T}$, for which a zero weight η_i are assigned in (3.1). In addition, each subgraph G is well connected with respect to Y, the set of induced terminals of $\mathcal{T}$ in G, in the sense that (G, Y) is a π-cut-linked instance. This decomposition is based on that of Chekuri, Khanna, and Shepherd [11]; we need to do some additional work to ensure that the min-cut condition holds. We prove a dual (flow-based) version of this result (Theorem 8.3) in section 8.1.

3.1. An overall routing algorithm in each decomposed subgraph G. We assume that we have the π-cut-linked subgraphs given by Theorem 3.1. We will treat each subgraph and its induced subproblem (G, T) independently. We use $\pi(G)$ to denote $\pi(V(G))$ in the following sections. Let Y be the set of terminals of T that is assigned a positive weight by function π in instance G. We further assume that $\pi(G) = \Omega(\log^7 n)$. If not, we just route an arbitrary pair of terminals in T; otherwise, we use Procedure EMBEDANDROUTE (G, T, π) in Figure 3.1 to route. We now state Theorem 3.3, which we prove throughout the remainder of the paper until section 7. We first summarize parameters that are related to EMBEDANDROUTE.

Parameters and conditions related to an induced subproblem (G, T)*.*

- $\omega \log^2 n$ is the number of matchings in Figure 3.1, where ω is a large enough constant to guarantee the success probability in Theorem 6.2;
- min-cut $\kappa \geq \frac{(d+2)(\ln n)(\omega \log^2 n+1)}{\epsilon^2}$, where $d \geq 4$ and $0 < \epsilon < 1$;

Given graph G with min-cut $\Omega(\log^3 n)$ and a weight function $\pi : V(G) \to \mathbb{R}^+$
$\{G^1, \ldots, G^Z\} = \textsc{Split}(G, Z, \pi)$
$\{\mathcal{X}, \mathcal{C}\} = \textsc{Clustering}(G^Z, \pi)$, where $\mathcal{X} = \{X_1, \ldots, X_r\}$ and $\mathcal{C} = \{C_1, \ldots, C_r\}$
Given a set of superterminals $\mathcal{X}$ of size r
Let $\mathcal{X}$ map to vertex set $V(H)$ of expander H
For $t = 1$ to $\omega \log^2 n$
    $(S, \bar{S} = \mathcal{X} \setminus S) = \text{KRV-}\textsc{FindCut}(\mathcal{X}, \{M_k : k < t\})$ s.t. $|S| = |\bar{S}| = r/2$
    Matching $M_t = \textsc{FindMatch}(S, \mathcal{X} \setminus S, G^t)$ s.t. M_t is routable in G^t
Combine $M_1, \ldots, M_{\omega \log^2 n}$ to form the edge set F on vertices $V(H)$
$\textsc{ExpanderRoute}(H, T, X)$
End

Fig. 3.1. *Procedure* $\textsc{EmbedAndRoute}(G, T, \pi)$.

- sampling probability $q = 1/(\omega \log^2 n + 1)$;
- the number of split subgraphs $Z = 1/q = \omega \log^2 n + 1$;
- $W = Z/(1-\epsilon) = (\omega \log^2 n + 1)/(1-\epsilon)$ for some $0 < \epsilon < 1$;
- $r \geq \max\{1, (\pi(G) - (W-1))/(2W-1)\}$, such that $\forall i \in [1, \ldots, r], 2W - 1 \geq \pi(X_i) = \sum_{v \in X_i} \pi(v) \geq W$, and $\pi(\mathcal{X}) \geq \pi(G) - (W-1)$; i.e., at most $W-1$ unit of weight is not counted in $\mathcal{X}$.

Theorem 3.3. *Given an induced instance (G, T) with min-cut κ of G being $\Omega(\log^3 n)$ and a weight function $\pi : V(G) \to [0, W]$ such that Y is π-cut-linked in G and $\pi(G) = \Omega(W \log^5 n)$, where $W = \Theta(\log^2 n)$. Procedure* $\textsc{EmbedAndRoute}$ *routes at least* $\max\{1, \Omega(\pi(G)/W \log^5 n)\}$ *pairs of T in G edge disjointly, with probability at least $1 - O(\log^2 n/n^{d-1})$, where $d \geq 4$. The dependence of κ on d is shown in* (4.5).

Combining Theorems 3.3 and 3.1 proves Theorem 3.4.

Theorem 3.4. *Given an EDP instance $(\mathcal{G}, \mathcal{T})$, where $\mathcal{G}$ has a min-cut $\Omega(\lambda(n)\kappa)$, let the approximation factor be $O(\lambda(n)\beta(\mathcal{G})W \log^5 n)$. Then with probability at least $1 - O(\log^2 n/n^{d-2})$, where $d \geq 4$, we can route $\Omega(\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})/g)$ terminal pairs edge disjointly in $\mathcal{G}$. The dependence of κ on d is shown in* (4.5).

Proof. By the union bound, the approximation statement in Theorem 3.3 holds for all node disjoint subgraphs $G_1, G_2, \ldots, G_\ell$ simultaneously with probability at least $1 - O\left(\log^2 n/n^{d-2}\right)$, where $d \geq 4$, given the trivial bound of $\ell \leq n$ and the probability of failure as bounded in Theorem 3.3 for a single graph G_j, $\forall j = 1, \ldots, \ell$. Now the bound and decomposition of the approximation factor follows from Theorem 3.3, the definition of $\pi(\mathcal{G})$, and its lower bound, as stated in condition 5 of Theorem 3.1. ∎

4. Obtaining Z split subgraphs of G.

In this section, we analyze a procedure that splits a graph G, with min-cut $\kappa = \Omega(\log^3 n)$, into Z subgraphs, where $Z = \omega \log^2 n + 1$, by extending a uniform sampling scheme from Karger [20]. We thus obtain a set of cut-linked instances as in Lemma 4.1, which follows immediately from Theorem 4.2. Theorem 4.2 says that with high probability, all cuts can be preserved in all split graphs $G^1, \ldots, G^Z$ of G we thus obtain. We prove Theorem 4.2 in section 4.1.

Procedure Split (G, Z, π): Given a graph $G = (V, E)$ with min-cut $\kappa = \Omega(\log^3 n)$, a weight function $\pi : V(G) \to \mathbb{R}^+$, a set of terminals Y in G such that (G, Y) is a π-cut-linked instance, and probability $q = 1/Z$.

Output: A set of randomized split subgraphs $G^1, \ldots, G^Z$ of G.

Each split subgraph $G^j, \forall j = 1, \ldots, Z$ inherits the same set of vertices of G; edges of G are placed independently and uniformly at random into the Z subgraphs; and each

$e = (u, v) \in E$ is placed between the same endpoints u, v in the chosen subgraph. We retain the same weight function π for all nodes in V in each split subgraph G^j, $\forall j$.

LEMMA 4.1. *With probability* $1 - O(\log^2 n/n^3)$, Y *is* $\frac{(1-\epsilon)\pi}{Z}$*-cut-linked in* G^j, $\forall j$, *for some* $0 < \epsilon < 1$.

Proof. Since Y is π-cut-linked in G, then $|\delta(S)| \geq \pi(S \cap Y)$, $\forall S$ such that $\pi(S \cap Y) \leq \pi(Y)/2$ in G. Let $\delta_j(S)$ denote the size of cut $(S, V \setminus S)$ in G^j. With probability $1 - O(\log^2 n/n^3)$, we have $|\delta_j(S)| \geq (1-\epsilon)q\,|\delta(S)| \geq \frac{(1-\epsilon)\pi(S\cap Y)}{Z}$, for all S such that $\pi(S \cap Y) \leq \pi(Y)/2$ and all j, as shown in Theorem 4.2 for $d \geq 4$. Hence Y is $(1-\epsilon)\pi/Z$-cut-linked in $G^j, \forall j$. $\square$

Recall for $S \in V$, $|\delta_G(S)|$ denote the size of $(S, V \setminus S)$ in G. For the same cut $(S, V \setminus S)$ we have, in G^j,

$$\mathbf{E}[|\delta_{G^j}(S)|] = q\,|\delta_G(S)|\,, \quad \forall G^j, \tag{4.1}$$

where q is the probability that an edge $e \in E$ is placed in G^j, $\forall j$.

THEOREM 4.2. *Let* $G = (V, E)$ *be any graph with unit-weight edges and min-cut* $\kappa = \Omega(\log^3 n)$, *where* $n \geq |V(G)| := n_1$. *Let* $\epsilon = \sqrt{3(d+2)(\ln n)/q\kappa}$, *where* $d \geq 4$. *If* $\epsilon \leq 1$, *then with probability* $1 - O(\log^2 n/n^{d-1})$, *every cut* $(S, V \setminus S)$ *in every subgraph* $G^1, G^2, \ldots, G^Z$ *of* G *has value between* $(1-\epsilon)$ *and* $(1+\epsilon)$ *times its expected value* $q\,|\delta_G(S)|$, *as shown in* (4.1).

Remark 4.3. It is clear that a large enough min-cut (which is allowed to depend on ϵ) ensures that $\epsilon \leq 1$; see (4.5) below. We emphasize here that $n = |V(\mathcal{G})| \geq |V(G)|$ as we are working with a single piece due to the decomposition of the original graph $\mathcal{G}$ as in Theorem 3.1; hence Theorem 4.2 allows us to bound the probability of failure in the sense of the theorem across all subgraphs of $\mathcal{G}$, as the total number of such node disjoint subgraphs G can trivially be bounded by n.

4.1. Proof of Theorem 4.2. In order to prove Theorem 4.2, we need to introduce a definition by Karger [20] regarding a uniform random sampling scheme on an unweighted graph $G = (V, E)$, from which Lemma 4.5 immediately follows. We then state the Chernoff bound that we need in order to derive Lemma 4.7, which shows a large deviation bound for a particular cut $(S, V \setminus S)$ of G in a randomly sampled subgraph, whose expected value is given in (4.1). Theorem 4.2 follows from the union bound, by summing up probabilities of the large deviation events across all split graphs, which are small due to the min-cut condition as stated in the theorem.

DEFINITION 4.4 (see [21]). *A* q*-skeleton of* G *is a random subgraph* $G(q)$ *constructed on the same vertices of* G *by placing each edge* $e \in E$ *in* $G(q)$ *independently with probability* q.

LEMMA 4.5. *Every randomized subgraph* G^j, $\forall j$, *is a* q*-skeleton of* G.

Proof. Recall the construction of a random subgraph G^j, $\forall j$, of G: on the same set of vertices as G, each edge $e \in E$ of the original graph G is placed in G^j independently with probability q. Hence, G^j, $\forall j$, is a q-skeleton of G by Definition 4.4. $\square$

We now define indicator variables I_e^j, $\forall j$, $\forall e \in E$, such that $I_e^j = 1$ when e is placed in G^j, and 0 otherwise; hence I_e^j is a Bernoulli random variable with success probability q, $\forall j, \forall e$. Note that random variables I_e^j, $\forall j = 1, \ldots, 1/q$, are not independent; in fact, $\sum_{j=1}^{1/q} I_e^j = 1 \;\forall\, e$.

Now consider a cut $(S, V(G) \setminus S)$ of size c in G. Let $I_1^j, I_2^j, \ldots, I_c^j$ be the indicator variables that signal whether the unit-weight edges $e_1, e_2, \ldots, e_c$ of the cut $(S, V(G) \setminus$

S) appear in a random subgraph G^j; thus we have, for all $j = 1, \ldots, Z$,

$$|\delta_{G^j}(S)| := \sum_{y=1}^{c} I_y^j. \tag{4.2}$$

It is clear that $I_1^j, I_2^j, \ldots, I_c^j$ are independently and identically distributed random variables whose common distribution is the Bernoulli distribution with parameter q, by the construction of a random subgraph G^j, $\forall j$, as shown in Lemma 4.5. One can now apply Lemma 4.6 to obtain a large-deviation bound for $|\delta_{G^j}(S)|$ as stated in Lemma 4.7, given (4.1) and (4.2).

LEMMA 4.6 (see Chernoff [14]). *Let Y be a sum of m independent Bernoulli random variables with success probability $q_1, \ldots, q_m$ and expected value $\mu = \sum_{i=1}^m q_i$. Then for $0 \le \sigma \le 1$, $\mathbf{Pr}[|Y - \mu| \ge \sigma\mu] \le 2e^{-\sigma^2\mu/3}$.*

LEMMA 4.7. *Consider a cut $(S, V(G) \setminus S)$ of size c in G. We have for all G^j, where $j = 1, \ldots, Z$, where $Z = \omega \log^2 n + 1$,*

$$\mathbf{Pr}[||\delta_{G^j}(S)| - qc| \ge \epsilon qc] \le 2e^{-\epsilon^2 qc/3}. \tag{4.3}$$

We need the following lemma to bound the number of every set of "small" events.

LEMMA 4.8 (see [21, cf. Theorem 2.1]). *In an undirected graph G with minimum cut κ and $|V(G)| \le n_1 \le n$, the number of cuts of value less than $\xi\kappa$ for $\xi \ge 1$ is less than $n_1^{2\xi}$.*

Proof of Theorem 4.2. Let $\tau = 2^{n_1} - 2$ be the number of cuts in graph G with n_1 nodes (and hence also in $G^1, \ldots, G^Z$). Let $c_1, \ldots, c_z$ be the expected values of the τ cuts in a q-skeleton listed in nondecreasing order so that $q\kappa = c_1 \le \cdots \le c_z$. Note that for the uniform sampling scheme that we use, it is clear that the original cut also follows this ordering: $c_1/q \le \cdots \le c_r/q$. Given a split graph G^j, $\forall j$, let $\mathcal{E}_i^j$, $\forall j$, $\forall i$ be the event that the value of a cut $\delta_{G^j}(S, V/S)$ in G^j deviates from its expectation c_i by more than ϵc_i. First by Lemma 4.7, we have

$$\mathbf{Pr}\Big[\mathcal{E}_i^j\Big] \le 2e^{-\epsilon^2 c_i/3}, \ \ \forall r \text{ cuts in } G^j, \forall j = 1, \ldots, Z. \tag{4.4}$$

Now given that every random split subgraph G^j, $\forall j$, is a q-skeleton of G by Lemma 4.5, we essentially apply Karger's theorem [21] (cf. Theorem 2.1) to each subgraph G^j (with a small alteration on q), whose conclusion holds by summing up probabilities of all z large-deviation events as bounded in (4.4); this is shown in Lemma 4.9 below. We first define the following parameters for a given $1 > \epsilon > 0$ and for $d \ge 4$:

$$q = \frac{3(d+2)\ln n}{\epsilon^2 \kappa}, \quad \text{where} \quad \kappa = \frac{3(d+2)\ln n(\omega \log^2 n + 1)}{\epsilon^2}. \tag{4.5}$$

Formally, we have the following.

LEMMA 4.9 (see [21, cf. Theorem 2.1]). *$\forall G^j$ with $n_1 \le n$ nodes, we have for $d \ge 4$, q as in (4.5), and $\tau = 2^{n_1} - 2$,*

$$\sum_{i=1}^{\tau} \mathbf{Pr}\Big[\mathcal{E}_i^j\Big] \le \frac{2}{n^{d-1}}. \tag{4.6}$$

We can now use the union bound to sum up the probabilities of bad events across all split subgraphs $G^1, \dots, G^Z$ of G, which yields following:

$$\sum_{j=1}^{Z}\sum_{i=1}^{\tau}\mathbf{Pr}\left[\mathcal{E}_i^j\right] \le \frac{2(\omega \log^2 n + 1)}{n^{d-1}}.$$

The theorem thus follows. ∎

Remark 4.10. Lemma 4.9 can be tightened up by at least a factor of $\frac{1}{n}$ using a slighter longer argument in [21]. We include a shorter proof next for self-containment.

Note that $\mathcal{E}_i^j$, $\forall j = 1, \dots, Z$, are not independent, since the indicator random variables that contribute to value of $|\delta_{G^j}(S, V/S)|$ are not at all independent across all subgraphs. However, we use only a union bound that does not assume anything about dependency among events.

Finally, we show the proof for Lemma 4.9, as our case here is slightly different from the original setting due to the small alteration on q, as $\ln n$ depends on $|V(\mathcal{G})|$ rather than n_1, the size of $|V(G)|$ of the current graph G we are working on. This alteration allows us to sum up bad events bounded in (4.6) across all G^j, $\forall j = 1, \dots, Z$ and all decomposed subgraphs G of $\mathcal{G}$ as given by Theorem 3.1; see also Remark 4.3.

Proof of Lemma 4.9. For a cut c of size $\xi\kappa$ in G, its expected value is $c_i := \xi q \kappa$ in a q-skeleton of G, where $\xi \ge 1$ and $r \ge k \ge 1$; thus we have by (4.4) and (4.5)

$$\begin{aligned}\mathbf{Pr}\left[\mathcal{E}_i^j \ s.t.\ c_i = \xi q \kappa\right] &=: \mathbf{Pr}[\xi] \\ &\le 2e^{-\epsilon^2 c_i/3} = 2e^{-\epsilon^2 \xi q \kappa/3} \\ &= 2e^{-\xi(d+2)\ln n} \\ &= \frac{1}{n^{\xi(d+2)}}. \end{aligned} \tag{4.7}$$

Now by taking a sequence of $\xi = 3/2, 2, 5/2, \dots$ and applying (4.7) and Lemma 4.8,

$$\begin{aligned}\sum_{i=1}^{\tau}\mathbf{Pr}\left[\mathcal{E}_i^j\right] &\le \sum_{\xi=3/2,2,\dots}^{\infty} n_1^{2\xi}\cdot \mathbf{Pr}[\xi - 1/2] \\ &\le \sum_{\xi=3/2,2,\dots}^{\infty} n_1^{2\xi}\frac{1}{n^{(\xi-1/2)(d+2)}} \\ &\le \sum_{\xi=3/2,2,\dots}^{\infty} n^{d/2+1}\frac{1}{n^{d\xi}} \\ &\le \frac{2}{n^{d-1}} \le 2/n^3\end{aligned}$$

for $d \ge 4$; hence the lemma holds. ∎

5. Forming superterminals that are well linked. The procedure in this section constructs superterminals as follows. Without loss of generality, we pick G^Z for forming edge disjoint connected components C in G^Z, where $\pi(C) = \Omega(\log^2 n)$, each connecting a subset of terminals. Note that G^Z is a connected graph with a min-cut of $\Omega(\log n)$ with high probability, by Theorem 4.2. Roughly, the idea is that these clustered terminals are better connected than individual terminals. They are well linked in the sense that any cut that splits off K superterminals as one entity contains at least K edges in G^j, $\forall j = 1, \dots, Z-1$. This allows us to compute congestion-free maximum flows in section 6.1.

Given split subgraphs $G^1, \dots, G^Z$ of G, each with the same weight function π on its vertex set $V(G^j) = V$, $\forall j$, that we obtain through PROCEDURE SPLIT(G, Z, π), we aim to find a set $\mathcal{X} = \{X_1, \dots, X_r\}$ of node disjoint "superterminals," where each superterminal $X_i \in \mathcal{X}$ consists of a subset of terminals in Y and each X_i gathers a weight between W and $2W-1$. In addition, we want to find an edge disjoint set of clusters $\mathcal{C} = \{C_1, \dots, C_r\}$, where $C_i = (V_i, E_i)$, such that $X_i \subseteq V_i$ and C_i is a connected component, where terminals in X_i are connected through E_i. We use the following procedure to accomplish these goals.

Procedure Clustering $(\boldsymbol{G^Z}, \boldsymbol{\pi})$: Given a split subgraph G^Z and a weight function $\pi : V(G^Z) \to \mathbb{R}^+$ and $\pi(V(G^Z)) = \pi(G) \geq W$.

Output: $\mathcal{X} = \{X_1, \dots, X_r\}$ and $\mathcal{C} = \{C_1, \dots, \mathcal{C}_r\}$ as specified in Lemma 5.1.

We group subsets of vertices of V in an edge disjoint manner, following a procedure from [9], by choosing an arbitrary rooted spanning tree of G^Z and greedily partitioning the tree into a set $\mathcal{C}$ of edge disjoint connected components of G^Z.

LEMMA 5.1 (see [9]). *Let G^Z be a connected graph with a weight function $\pi : V(G^Z) \to [0, W]$ such that $\pi(V(G^Z)) \geq W$. We can find r edge disjoint connected components $C_1 = (V_1, E_1), \dots, C_r = (V_r, E_r)$, where*

$$\max\left\{1, \frac{\pi(G) - (W-1)}{2W-1}\right\} \leq r \leq \max\left\{\frac{\pi(G)}{W}, n\right\} \tag{5.1}$$

such that there exist vertex disjoint subsets $X_1, \dots, X_r$ and for each i, (a) *$X_i \subseteq V_i$;* (b) *$2W - 1 \geq \sum_{v \in X_i} \pi(v) \geq W$.*

Remark 5.2. It is clear that $E_1, \dots, E_r$ are disjoint set of edges by construction; however, it is worth pointing out that although sets $X_1, \dots, X_r$ are disjoint, $V_1, \dots, V_r$ are not. Hence a terminal $x \in Y$ may belong to two clusters of $\mathcal{C}$ while it can belong only to one subset in $\mathcal{X}$, and hence its weight contributes only to one cluster. For example, the spanning tree T_i for connecting terminals in X_i in C_i, as constructed in Theorem 6.1, may traverse some node in cluster C_j, where $i \neq j$.

Result. To get an intuition of the purpose of forming such clusters, consider a cut $(U, V \setminus U)$ in a split subgraph $G^j, \forall j$. Let U be a subset of $V(G)$ such that $\pi(U) = \sum_{x \in U \cap Y} \pi(x) \leq \pi(Y)/2$. Let K be the number of superterminals that are contained in U. We now show that superterminals are "well linked," with a hint of Definition 2.6.

LEMMA 5.3. *For all split subgraphs $G^1, \dots, G^Z$, where $Z = \omega \log^2 n + 1$, and $\forall U \subset V(G)$ such that $\pi(U) \leq \pi(Y)/2$, it holds with probability $1 - O(\log^2 n / n^{d-1})$, where $d \geq 4$, that $|\delta_{G^j}(U)| \geq K$, where $K := |\{X_i \in \mathcal{X} : X_i \subseteq U\}|$.*

Proof. With high probability, Y is $\frac{(1-\epsilon)\pi}{(\omega \log^2 n + 1)}$-cut-linked in $G^1, \dots, G^Z$, as shown in Lemma 4.1. Thus $\forall G^j$, by Definition 2.2 and that of K, we have

$$\begin{aligned} |\delta_{G^j}(U)| &\geq \frac{1-\epsilon}{\omega \log^2 n + 1} \sum_{x \in U} \pi(x) \\ &\geq \frac{1-\epsilon}{\omega \log^2 n + 1} \sum_{i : X_i \subseteq U} \sum_{x \in X_i} \pi(x) \\ &\geq \frac{(1-\epsilon) K W}{(\omega \log^2 n + 1)} = K, \end{aligned}$$

where the last line is due to the lower bound of W on $\sum_{x \in X_i} \pi(x)$. $\square$

6. Constructing and embedding an expander H in G. In this section, we use the superterminals from the previous section as nodes in an expander H that we embed in G. The edges of H are defined using a technique in [22] that builds an expander using $O(\log^2 n)$ matchings. We embed this expander in G by routing each matching in one of the split graphs using a maximum flow computation. This allows us to embed H into G with no congestion. The following procedure restates this outline. Theorem 6.1 is a main technical contribution of this paper.

---

0. Given a set of points $V(H)$ of size r
1. for $j = 1$ to $\omega \log^2 n$
2. $\quad (S, \bar{S} = V(H) \setminus S) =$ KRV-FindCut$(V(H), \{M_k : k < j\})$ s.t. $|S| = |\bar{S}| = r/2$
3. $\quad M_j =$ FindMatch$(S, \bar{S})$ s.t. M_j is a matching between S and $\bar{S}$
4. Combine $M_1, \ldots, M_{\omega \log^2 n}$ to form the edge set F on vertices $V(H)$
5. End

Fig. 6.1. *KRV-Procedure* Constructing an α-Expander H.

Procedure EmbedExpander ($G^1, \ldots, G^{\omega \log^2 n}, \mathcal{X}$):
Output: An expander $H = (V', F)$ routable in G such that (a) $|V'| := r$ satisfies (5.1), and $\forall i \in V'$; (b) $\pi(i) = \pi(X_i)$ and $\pi(H) = \pi(\mathcal{X})$; (c) F consists of $M_1, \ldots, M_{\omega \log^2 n}$, and thus $|F| = \frac{\omega \log^2 n}{2}$.
We use Steps 3–8 of Procedure EmbedAndRoute in Figure 3.1, where we substitute Procedure FindMatch with Figure 6.2 while relying on an existing Procedure KRV-FindCut [22]. At each round j, we use KRV-FindCut to generate an equal-sized partition $(S, \mathcal{X} \setminus S = \bar{S})$; we then find a matching M_j between S and $\bar{S}$ by computing a single-commodity max-flow using FindMatch$(S, \bar{S}, G^j)$ in G^j, which we add to F as edges.

Theorem 6.1. *With probability at least* $1 - 1/n^3$, (a) EmbedExpander *constructs a* $1/4$*-expander* $H = (V', F)$. (b) *In addition, H is embedded into G as follows: Each node i of H corresponds to a superterminal X_i in $\mathcal{X}$ in G such that all superterminals are mutually node disjoint and each superterminal is connected by a spanning tree T_i in G; Each edge (i, j) in H corresponds to a path P_{ij} from a terminal node in X_i to a terminal node in X_j; all paths P_{ij} and trees T_i are mutually edge disjoint in G.*

Khandekar, Rao, and Vazirani [22] show that the procedure in Figure 6.1 produces an expander graph H with high probability, as stated in Theorem 6.2.

Theorem 6.2 (see KRV 06 [22]). *Given a set of nodes $V(H)$ of size r, $\exists$ a* KRV-FindCut *procedure such that given any* FindMatch *procedure, the* KRV*-Procedure in Figure* 6.1 *produces an α-expander graph H, for $\alpha \geq 1/4$, with probability* $1 - \frac{1}{n^C}$, *where $C \geq 3$ given a large enough constant ω.*

Remark 6.3. The same argument as in Remark 4.3 for bounding the probability of failure across all decomposed graphs of $\mathcal{G}$ for events as described in Theorem 4.2 applies here; thus, with probability at least $1 - O(1/n^2)$, for each decomposed subgraph G of $\mathcal{G}$, we obtain an α-expander H that is embedded into G as in Theorem 6.1.

In the rest of this section, we first describe the FindMatch procedure as in Figure 6.2, which we shall plug into the KRV-Procedure as described in Figure 6.1. We then prove Theorem 6.1 in section 6.2.

0. Given an equal partition $(S, \bar{S})$ of $\mathcal{X}$, we form a flow graph G' from G^j by adding auxiliary nodes and directed unit-capacity edges:
1. Add a special source node s_0 and sink node t_0
2. Add nodes $s_1, \ldots, s_{r/2}$ and an edge from s_0 to s_k, $\forall k = 1, \ldots, r/2$;
3. Add nodes $t_1, \ldots, t_{r/2}$; from each t_k, $\forall k = 1, \ldots, r/2$, add an edge to t_0
4. From each s_k, add an edge to each terminal $x \in X_{i_k}$ s.t. $X_{i_k} \in S$
5. To each node t_k, add an edge from each terminal $x \in X_{j_k}$ s.t. $X_{j_k} \in \bar{S}$
6. Route a max-flow from s_0 to t_0
7. Decompose the flow to obtain a matching between S and $\bar{S}$
8. End

FIG. 6.2. *Procedure* FINDMATCH$(S, \bar{S}, G^j)$.

6.1. Finding a matching through a max-flow construction. We now show that given an arbitrary equal partition $(S, \bar{S})$ of the set $\mathcal{X} = \{X_1, \ldots, X_r\}$ that we obtain through Procedure CLUSTERING(G^Z, π), we can use the following procedure to route a max-flow of size $r/2$, such that the integral flow paths that we obtain through flow decomposition induce a perfect matching between S and $\bar{S}$. Let $S = \{X_{i_1}, \ldots, X_{i_{r/2}}\}$ and $\bar{S} = \{X_{j_1}, \ldots, X_{j_{r/2}}\}$, where $\{i_1, \ldots, i_{r/2}, j_1, \ldots, j_{r/2}\}$ is understood to be a permutation of the original sequence $\{1, \ldots, r\}$ returned by Procedure KRV-FINDCUT as in Figure 6.1.

LEMMA 6.4. *In each sampled graph G^t, $\forall t = 1, \ldots, Z-1$,* FINDMATCH *produces a perfect matching M_t between an equal partition $(S, \bar{S})$ of $\mathcal{X}$ such that for each edge in $e = (i, j) \in M_t$, there is an integral unit-flow path P_{ij} from a terminal in $X_i \in S$ to a terminal in $X_j \in \bar{S}$. All paths P_{ij}, such that $(i, j) \in M_t$ are edge disjoint in G^t.*

We first prove the following lemma.

LEMMA 6.5. *Every $s_0 - t_0$ cut has size at least $r/2$ in the flow graph G' as in Figure* 6.2 *with probability* $1 - O(\log^2 n / n^{d-1})$*, where $d \geq 4$.*

Proof. Let $(U, \bar{U})$ be a cut in the flow graph that separates s_0 from t_0; without loss of generality, let U be a subset such that $\pi(U \cap Y) \leq \pi(Y)/2$, and let $s_0 \in U$ (otherwise, we can just rename all the auxiliary nodes and the two subsets S and $\bar{S}$).

Consider any superterminal $X \in \mathcal{X}$ that we obtained through Lemma 5.1; if X is contained either in U or in $\bar{U}$, we call such a superterminal X uncut; otherwise, we say that X is cut by $(U, \bar{U})$.

1. Let $K_c^s = \left|\{X \in S : X \cap U, X \cap \bar{U} \neq \emptyset\}\right|$ denote the number of superterminals in S that are cut by $(U, \bar{U})$.
2. Let $\overline{K_{uc}^s} = \left|\{X \in S : X \subseteq \bar{U}\}\right|$ be the number of superterminals in S that are contained in $\bar{U}$.
3. Let $K_{uc}^s = |\{X \in S : X \subseteq U\}|$ denote the number of superterminals in S that are contained in U; hence $K_{uc}^s + \overline{K_{uc}^s} + K_c^s = r/2$, where $r = |\mathcal{X}|$.
4. Let $K_c^t = \left|\{X \in \bar{S} : X \cap U, X \cap \bar{U} \neq \emptyset\}\right|$ denote the number of superterminals in $\bar{S}$ that are cut.
5. Let $K_{uc}^t = \left|\{X \in \bar{S} : X \subseteq \bar{U}\}\right|$ denote the number of superterminals in $\bar{S}$ that are contained in U.

Given that G is π-cut-linked, we know that the sampled graph G^j is $(1-\epsilon)\pi/(\omega \log^2 n+1)$-cut-linked with high probability by Lemma 4.1. Recall that in our clustering scheme, the total weight of all terminals in one superterminal is at least $W = \frac{\omega \log^2 n+1}{1-\epsilon}$. Note that there is at least one directed auxiliary edge crossing the cut for all superterminals except those in S that are contained in U or those in $\bar{S}$ that are contained in $\bar{U}$. Thus we have

$$\begin{aligned}|\delta_{G'}(U)| &\geq |\delta_{G^j}(U)| + K^t_{uc} + \overline{K^s_{uc}} + K^s_c + K^t_c \\ &\geq \frac{(1-\epsilon)\sum_{x\in U}\pi(x)}{\omega \log^2 n + 1} + K^t_{uc} + \overline{K^s_{uc}} + K^s_c + K^t_c \\ &\geq \frac{(1-\epsilon)(K^s_{uc} + K^t_{uc})W}{\omega \log^2 n + 1} + K^t_{uc} + \overline{K^s_{uc}} + K^s_c + K^t_c \\ &\geq K^s_{uc} + \overline{K^s_{uc}} + K^s_c \geq r/2.\end{aligned}$$

Hence we have shown that the size of every cut $(U, \bar{U})$ in the flow graph G' has size at least $r/2$. ☐

Proof of Lemma 6.4. By Lemma 6.5, and the fact that there exists an $s_0 - t_0$ cut of size $r/2$, (e.g., $(\{s_0\}, V(G') \setminus \{s_0\})$) we know the $s_0 - t_0$ min-cut is $r/2$. Hence by the max-flow min-cut theorem, we know that there exists a max-flow of size $r/2$ from s_0 to t_0. We next decompose the max-flow into $r/2$ integer flow paths (hence edge disjoint), which induce a perfect matching M_t between S and $\bar{S}$ as follows.

Consider an integral flow path P_k, $\forall k = 1, \ldots, r/2$. Let the directed path P_k start with s_0 and go through s_k and some terminal $x \in X_{i_k} \in S$; and let P_k end with some terminal $y \in X_{j_{k'}} \in \bar{S}$, $t_{k'}$ and then t_0 for some $k' \in [1, \ldots, r/2]$. No other path in the max-flow can go through the same pair of superterminals X_{i_k}, $X_{j_{k'}}$ due to the capacity constraints on edges (s_0, s_k) and $(t_{k'}, t_0)$. Hence $M_t = \{(i_k, j_{k'}),\ k \in [1, \ldots, r/2],$ where $k' \in [1, \ldots, r/2]\}$ is a perfect matching between S and $\bar{S}$. ☐

6.2. Proof of Theorem 6.1. The expander property (a) follows from Theorem 6.2. Each edge $e = (i, j)$ in the matching M_t maps to an integral flow path that connects X_i and X_j in G^t; all such flow paths can be simultaneously routed in G^t edge disjointly due to the max-flow computation, as we show in Lemma 6.4. Since each matching M_t is on a unique split subgraph G^t, the entire set of edges in $M_1, \ldots, M_{\omega \log^2 n}$, which comprise the edge set F of H, corresponds to edge disjoint paths in $G^1, \ldots, G^{Z-1}$, where $Z = \omega \log^2 n + 1$. Finally, all spanning trees T_i, $\forall i$, are constructed using a disjoint set of edges in the last split graph G^Z as in Lemma 5.1.

7. Routing on an expander *H* node disjointly. In this section, we show that the following greedy algorithm routes $\Omega(r/\log^5 n)$ pairs of terminals in $H = (V(H), F)$, where $r = |V(H)| = \Theta(\pi(G)/W)$ as shown in (5.1) and $|F| = \frac{r\omega \log^2 n}{2}$ due to the construction in section 6.

Procedure ExpanderRoute (H, T, X): Given an uncapacitated expander H with at least $512 \log^5 n$ nodes, with node degree $\omega \log^2 n$. While there is a pair (s, t) in $T \subseteq \mathcal{T}$ whose path length is strictly less than ν in $H = (V, E)$, where $\nu = a_3 \omega \log^3 n$ and $a_3 = 32$, remove both nodes and edges from H, along a path through which we connect a pair of terminals in T.

Since we take away both nodes and edges as we route a path across the expander H due to the node capacity constraints on $V(H)$, routing the set P of pairs via integral paths on H induces no congestion in G by Theorem 6.1. Hence we need only to argue that $|P|$ is large to finish our proof. Formally, we show the following.

THEOREM 7.1. *Given a degree-*$(\omega \log^2 n)$ *expander* $H = (V, E)$, *where* $|V| =: r \geq 512 \log^5 n$, *the procedure above routes* $\Omega(r/\log^5 n)$ *pairs, node disjointly, in* H.

Let H' be the remaining graph of expander $H = (V, E)$, after we take away nodes and edges along the paths used to route terminal pairs in D. Note that all pairs

$T' \subseteq T$ that remain in H' must have distance at least ν. This is the main condition that allows us to prove Theorem 7.1. Let us also define a multicut L as a set of edges whose removal separates all pairs in T' that remain in $H' = (V', E')$.

Recall that any node-balanced cut in H must have at least $\Omega(r)$ edges. Now suppose that we can find a node-balanced cut $(U, \bar{U})$ in H such that at most half of its edges remain in H' and hence $\Omega(r)$ edges have been removed when routing D. Since routing each pair in D removes at most $\nu\omega \log^2 n$ edges, where $\nu = a_3\omega \log^3 n$ for a properly chosen constant a_3, we conclude $|D|$ must be $\Omega(r/\log^5 n)$, where $r = |V(H)|$.

The proof of Theorem 7.1 therefore involves primarily finding such a balanced cut in H given L. Before we go on, we first state Lemma 7.2 regarding the existence of a small multicut L in H'. In fact, following the construction of [18], one can find such a multicut.

LEMMA 7.2 (see [18]). *If all remaining terminal pairs in $T' \subseteq T$ have distances at least ν in H', then there exists a multicut L in $H' = (V', E')$ of size $\frac{|E'| \log n}{\nu}$ in H' that separates every source and sink pair $(s_i, t_i) \in T'$.*

Applying Lemma 7.2 to H', we immediately have the following bound on $|L|$:

$$|L| \leq \frac{r\omega \log^3 n}{2\nu} = \frac{r}{2a_3}, \tag{7.1}$$

given that $|E'| \leq |E| = r\omega \log^2 n/2$.

7.1. Proof of Theorem 7.1. We prove Theorem 7.1 by first noting that condition 1 of Theorem 3.1 implies that any multicut of the terminals in H' ensures that no piece in H' separated by L contains more than half the weight of all terminals in H according to π. We now alter π to obtain a new weight function $\pi'(v)$, $\forall v \in X_i \in V(H')$, so that we can make a stronger claim about the weight of each cluster separated by L. We then use this fact to show that clusters separated by L can be rearranged to find a weight-balanced cut $(U', \bar{U}')$ in H' according to π'.

Procedure Alter (π, π'): Recall that for a pair of terminals $(s, t) \in T$, the same amount of weight w_{st}, according to their flow in $\bar{f}$, is contributed to both $\pi(s)$ and $\pi(t)$ as specified in (3.1). Now suppose that s is removed from H while routing the D subset of terminal pairs (as $s \in X_i \in V(H) \setminus V(H')$), but t remains in H'; we remove w_{st} from $\pi(t)$. We repeat this for all $t \in X_i \in V(H')$ and define this updated weight as π' and let $\pi'(H') = \sum_{u \in X_i \in V(H')} \pi'(u)$.

Recall that initially $\pi(H) = \pi(\mathcal{X}) \geq \pi(G) - (W-1)$, since at most $W-1$ of $\pi(G)$ is not assigned to any node in H, and each node in H has weight between W and $2W-1$ as shown in proof of Lemma 5.1. Hence the total weight taken away from π by routing $|D|$ terminal pairs of distance at most ν is at most $2\nu|D|(2W-1)$. Thus by Procedure ALTER(π, π'), we have $\pi'(X_i) \leq \pi(X_i) \leq 2W-1$ and

$$\pi'(H') \geq \pi(G) - (W-1) - 2\nu|D|(2W-1). \tag{7.2}$$

Now it is clear that only the remaining pairs $(s, t) \in T'$ contribute a positive weight to $\pi'(H')$ according to their flow in $\bar{f}$ as in (3.1). Let L be the multicut that separates all remaining terminals pairs $T' \subseteq T$ in H'. Thus L cuts the graph H' and group nodes in $V(H')$ into clusters such that each cluster (a connected component in H') has a weight of at most $\pi'(H')/2$, since each individual s_i, t_i in a pair in T', each contributing the same amount of weight to $\pi'(H')$ according to their flow x_i in $\bar{f}$, must belong to different clusters.

We then use L to find a weight-balanced cut $(U', V(H') \setminus U')$ in H' such that each side has weight at least $\pi'(H')/4$, where $\pi'(H') \geq \pi(G) - (W-1) - 2(2W-1)\nu\,|D|$. Given such a weight-balanced cut $(U', \bar{U}')$ in H', it is straightforward to verify that any partition $(U, V(H) \setminus U)$ in H, such that $U' \subseteq U$ and $(V(H') \setminus U') \subseteq (V(H) \setminus U)$, is node-balanced in H, as we show now in Lemma 7.3.

LEMMA 7.3. *Let $(U', V(H') \setminus U')$ be a $(1/4, 3/4)$-weight-balanced cut in H'. Let $r \geq 512 \log^5 n$. Consider any cut $(U, V(H) \setminus U)$ in H, such that $U' \subseteq U$ and $(V(H') \setminus U') \subseteq (V(H) \setminus U)$ before routing the terminals in D:*

$$\min(|U|, |V \setminus U|) \geq \frac{r}{8} - \nu|D|/2. \tag{7.3}$$

Proof. If U is the smaller side, we have $|U| \geq |U'|$; otherwise, we have $|V(H) \setminus U| \geq |V(H') \setminus U'|$. Now given that both $\pi'(U')$ and $\pi'(V(H') \setminus U')$ are at least $\pi'(H')/4$, the upper bound $2W-1$ on $\pi'(X_i) \leq \pi(X_i)$, and (7.2), we have

$$\begin{aligned}
\min(|U|, |V \setminus U|) &\geq \min\{|U'|, |V(H') \setminus U'|\} \\
&\geq \frac{\pi'(H')}{4(2W-1)} \\
&\geq \frac{\pi(G) - (W-1)}{4(2W-1)} - \frac{\nu|D|}{2} \\
&\geq \frac{\pi(G)}{8W} + \frac{\pi(G)}{16W^2} - \frac{1}{8} - \frac{\nu|D|}{2} \\
&\geq \frac{r}{8} - \nu|D|/2
\end{aligned}$$

given that $\frac{\pi(G)}{16W^2} = \Omega(\log^3 n) \geq \frac{1}{8}$ and $\frac{\pi(G)}{W} \geq r$ by (5.1). $\square$

Proof of Theorem 7.1. We build a $(1/4, 3/4)$-weight-balanced partition of H' from L as bounded in (7.1) as follows: we start with two empty sides A and B and then add the connected components (after removing the multicut L) of H' to the smaller side repeatedly. Each component contains at most $\pi'(H')/2$ weight due to (3.1) and Procedure ALTER; in the end neither side can contain more than $3\pi'(H')/4$ of weight; indeed, consider the step where, without loss of generality, side A was put over $3/4$ of $\pi'(H')$ by adding a component d: in that step, d could not have been added to A, since $\pi'(A) \geq \pi'(H')/4 \geq \pi'(B)$ before d was added, given that $d \leq \pi'(H')/2$.

By the construction of $(U', V(H') \setminus U')$ and by (7.1), we have

$$|\delta_{H'}(U')| \leq |L| \leq \frac{r}{2a_3}, \tag{7.4}$$

while for any cut $(U, V(H) \setminus U)$ in H, such that $U' \subseteq U$ and $(V(H') \setminus U') \subseteq (V(H) \setminus U)$, we have by Lemma 7.3

$$|\delta_H(U)| \geq \alpha \min(|U|, |V \setminus U|) \geq \alpha\left(\frac{r}{8} - \nu|D|/2\right), \tag{7.5}$$

as H is an α-expander. Note that the number of edges taken away from the balanced cut $(U, V(H) \setminus U)$ for routing the set D of unit flows is at most $\nu|D|\omega \log^2 n$, as each flow is of length at most $\nu - 1$ (hence taking away ν nodes), and each node in H has degree $\omega \log^2 n$. Combining this bound with (7.5) and (7.4), we have

$$\begin{aligned}
\alpha\left(\frac{r}{8} - \nu|D|/2\right) &\leq |\delta_H(U)| \\
&\leq |\delta_{H'}(U')| + \nu|D|\omega \log^2 n \\
&\leq \frac{\omega r}{2a_3} + \nu|D|\omega \log^2 n.
\end{aligned} \tag{7.6}$$

TABLE 7.1
Parameters related to the original and decomposed EDP instances.

Parameters	Meaning
α	expansion factor of H
$\beta(\mathcal{G})$	mincut-maxflow gap on product commodity flows on $\mathcal{G}$
$\delta_G(S)$	the set of edges with exactly one endpoint in S in G
ϵ	a constant suitably chosen in $(0,1)$, used throughout this table
κ	min-cut of G_j; for $0<\epsilon<1$ and $d\geq 4$, $\sim\geq \frac{(d+2)(\ln n)(\omega\log^2 n+1)}{\epsilon^2}$
η_i	$\forall i=1,\dots,k$ factor for x_i's contribution to $\pi(s_i),\pi(t_i)$
$\lambda(n)$	$\lambda(n)=10\beta(\mathcal{G})\log \mathsf{OPT}^*(\mathcal{G},\mathcal{T})$; see section 8.1 for details
$\nu-1$	the maximum length of an integral flow path allowed to route in H
τ	$\sim:=2^{n_1}-2$, the number of cuts in a subgraph G with n_1 nodes
ω	a constant that specifies the number of matchings $\omega\log^2 n$
a_3	a constant defined in section 7
$\mathsf{cap}(S,\bar{S})$	$\sim:=\lvert\delta_G(S)\rvert$; the total capacity of edges in the cut $(S,V\setminus S)$ in G
$\bar{f}$	an optimal fractional solution to (2.2a)–(2.2d)
g	approximation factor for $(\mathcal{G},\mathcal{T})$, $\sim=O(\lambda(n)\beta(\mathcal{G})W\log^5 n)$
k	the number of unit-demand pairs (s_i,t_i) in $\mathcal{T}$
n	the number of nodes in $\mathcal{G}$, the original input graph
p	an individual path in $\mathcal{P}$
q	sampling probability, for $d\geq 4$, $\sim=\frac{(d+2)(\ln n)}{\epsilon^2\kappa}=\frac{1}{\omega\log^2 n+1}$
r	the number of nodes in Expander H as bounded in (5.1)
x_i	amount of flow in OPT^* for (s_i,t_i); see (2.2a)–(2.2d)
C_i	$\forall i=1,\dots,r$, connected components connecting nodes in X_i
D	a set of terminal pairs in T that are routed greedily in H
F	a set of $\omega\log^2 n$ matchings $M_1,\dots,M_{\omega\log^2 n}$ that comprise the set of edges for H: $\lvert F\rvert=r\omega\log^2 n/2$
$\mathcal{G}$	the original input graph with min-cut $\Omega(\kappa\log^2 n)$
$(\mathcal{G},\mathcal{T})$	the original EDP instance
G,G_j	$\forall j=1,\dots,\ell$, a subgraph of $\mathcal{G}$ obtained through Theorem 3.1
(G,T)	an induced EDP instance in one of the disjoint set of subgraphs G of $\mathcal{G}$ in Theorem 3.1
H	an α-expander constructed in Theorem 6.1
M_i	$\forall i=1,\dots,\omega\log^2 n$, matchings in H
$\mathcal{P}_i$	$\forall i=1,\dots,k$, set of paths joining s_i,t_i in $\mathcal{G}$
$\mathcal{T}$	a set of terminals pairs $\{(s_1,t_1),(s_2,t_2),\dots,(s_k,t_k)\}$ in $\mathcal{G}$
Y	the set of induced terminals of $\mathcal{T}$ in G
X,X_i	superterminal: a subset of terminals in Y s.t. X_i are node disjoint
$\mathcal{X}$	a set of node-disjoint superterminals $\{X_1,\dots,X_r\}$
W	minimum weight for $X_i\in\mathcal{X}$, $\sim=\frac{\omega\log^2 n+1}{1-\epsilon}$, $0<\epsilon<1$
Z	the number of split subgraphs $\sim=1/p=\omega\log^2 n+1$
$\pi(v)$	weight of $v\in G$ assigned in Theorem 3.1
$\pi(X_i)$	$\sim:=\sum_{v\in X_i}\pi(v)\in[W,2W-1]$, $\forall i=1,\dots,r$
$\pi(\mathcal{X})$	$\sim\geq \pi(G)-(W-1)$

Now by taking $\alpha=1/4$ and $a_3=32$, we have by (7.6)

$$|D|\geq \frac{r\,(\alpha/8-1/2a_3)}{a_3\omega\log^3 n(\omega\log^2 n+\alpha/2)}\geq \frac{r}{2048\omega^2\log^5 n},$$

where ω is a constant defined in Figure 3.1. ∎

8. The decomposition procedure for Theorem 3.1. In this section, we first sketch a proof of Theorem 8.3, which states a more refined and stronger version of Theorem 3.1. The full proof of Theorem 8.3 is shown in section 10. A list of parameters and notation used from this section until section 10 is summarized in Table 10.1.

8.1. The CKS flow-linked decomposition theorem. We first transform $(\mathcal{G}, \mathcal{T})$ to a set of flow-linked instances by following a decomposition procedure in [11], the outcome of which is summarized in the following theorem.

THEOREM 8.1 (see [11]). *Let $\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$ be a solution to the LP for a given instance $(\mathcal{G}, \mathcal{T})$ of the EDP problem in an input graph $\mathcal{G}$. One can efficiently compute a partition of $\mathcal{G}$ into node disjoint induced subgraphs $G_1, G_2, \ldots, G_\ell$ and weight functions $\rho_i : V(G_i) \to \mathbb{R}^+$ with the following properties. Let $\mathcal{T}_i$ be the induced pairs of $\mathcal{T}$ in G_i and let Y_i be the set of terminals of $\mathcal{T}_i$.*

1. *Y_i is ρ_i-flow-linked in G_i.*
2. *$\sum_{i=1}^{\ell} \rho_i(Y_i) = \Omega(\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})/\lambda(n))$, where $\lambda(n) = 10\beta(\mathcal{G}) \log \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$.*

Remark 8.2. Although the original statement in the Chekuri–Khanna–Shepherd (CKS) decomposition theorem [11] (cf. Theorem 2.1) assumes that each node u belongs to only a single terminal pair in $\mathcal{T}$, which guarantees that $\rho_i(u) = \rho_i(v)$ holds for all $(u, v) \in \mathcal{T}_i$, their decomposition procedure and analysis apply to the general case that we consider in this paper; in particular, conditions 1 and 2 do not depend on such an assumption.

Before we go on, let us define the following notation that appears in the proof of Theorem 8.1 as in [11]. Let $G_1, G_2, \ldots, G_\ell$ be the node disjoint subgraphs of $\mathcal{G}$ produced by the CKS decomposition procedure in Theorem 8.1. Recall that $\mathcal{P}$ refers to the entire set of paths from the original flow decomposition as in (2.2a)–(2.2d). Let

- $\gamma(\mathcal{G}) = \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$, as in (2.1);
- $\gamma(G_i) = \sum_{p \in \mathcal{P}: p \in G_i} \bar{f}(p)$ denote the total flow induced in G_i by the original flow $\bar{f}$; it counts flow only on flow paths $\bar{f}(P)$ from the the original flow path decomposition that are completely contained in G_i;
- $\gamma(u, G_i)$ denote the flow in G_i for u; and hence
- $\gamma(G_i) = \frac{1}{2} \sum_{u \in G_i} \gamma(u, G_i)$ by definition.

Thus there exists at least one flow path with a positive amount of flow between a pair of terminals $(u, v) \in \mathcal{T}_i$ according to the original flow path decomposition of $\bar{f}$, which is entirely contained in G_i, if the contribution from the pair (u, v) to $\gamma(u)$ and $\gamma(v)$ (and hence $\rho_i(u)$ and $\rho_i(v)$ in Theorem 8.1) is positive.

All subgraphs $G_1, G_2, \ldots, G_\ell$ produced by the CKS decomposition procedure satisfy one of the following conditions:

1. The flow is sufficiently small, in that $\gamma(G_i) \leq \lambda(n)/10$. In this case, let $\rho_i(u) = \rho_i(v) = 1$ for some pair $(u, v) \in \mathcal{T}_i$ with positive flow in G_i; and $\rho_i(y) = 0$ for $y \neq u, v$. Hence one can just route a unit flow between the chosen pair $(u, v) \in \mathcal{T}_i$ along an integral path; such a path exists since G_i is a connected component.
2. Else, for $\gamma(G_i) > \lambda(n)/10$, Y_i is ρ_i-flow-linked in G_i, where Y_i is the set of terminals of $\mathcal{T}_i$, and ρ_i is defined as follows for G_i:

$$\rho_i(u) = \gamma(u, G_i) = 0, \quad \forall u \notin Y_i, \tag{8.1a}$$

$$\rho_i(u) = \frac{\gamma(u, G_i)}{\lambda(n)}, \quad \forall u \in Y_i, \quad \text{and hence} \tag{8.1b}$$

$$\rho_i(G_i) = \rho_i(Y_i) := \sum_{x \in Y_i} \rho_i(x) = \sum_{x \in Y_i} \frac{\gamma(x, G_i)}{\lambda(n)} = \frac{2\gamma(G_i)}{\lambda(n)}. \tag{8.1c}$$

For both cases, the CKS weight function on $V(G_i)$ satisfies $\rho_i(G_i) = \Omega(\frac{\gamma(G_i)}{\lambda(n)})$. From now on, we refer to both $(G_i, \mathcal{T}_i)$ and (G_i, Y_i) as ρ_i-flow-linked instances without differentiation.

Throughout the rest of the paper, we use $\mathsf{remaining\text{-}flow}$ to keep track of the total remaining flow of $\bar{f}$ between terminal pairs in $\mathcal{T}_i$, across all i, where $\mathcal{T}_i$ are the induced pairs of $\mathcal{T}$ in G_i. By the end of the CKS flow decomposition, we lose at most half of $\bar{f}$, where $|\bar{f}| = \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$, as the number of edges that were cut during flow decomposition is at most $\mathsf{OPT}^*/2$ (cf. the proof of Theorem 8.1 in [11]); hence

$$\mathsf{remaining\text{-}flow} := \sum_{i=1}^{\ell} \gamma(G_i) \geq \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})/2. \tag{8.2}$$

Note that $\mathsf{remaining\text{-}flow}$ is the lower bound on $\sum_i |\mathcal{T}_i|$. Thus condition 2 of Theorem 8.1 also holds, given (8.2), and hence

$$\sum_{i=1}^{\ell} \rho_i(Y_i) \geq \sum_{i=1}^{\ell} \frac{2\gamma(G_i)}{\lambda(n)} = \Omega(\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})/\lambda(n)). \tag{8.3}$$

The proof of the theorem appears in [11]. They use this procedure as the first step in a two-step transformation from the optimal multicommodity flow solution $\bar{f}$ to obtain sets of well-linked terminal sets, which eventually leads to an $O(\log^2 K)$-approximation for the ANF problem described in section 1, where $K = |\mathcal{T}|$.

8.2. A modified flow-linked decomposition theorem. Let $G_1, G_2, \ldots, G_\ell$ be the node disjoint subgraphs of $\mathcal{G}$ produced by the CKS decomposition procedure. We treat the induced subproblems $(G_i, \mathcal{T}_i)$, $\forall i$ independently. Given (G_i, Y_i) such that Y_i is ρ_i-flow-linked in G_i, there are two postprocessing stages needed in order to repair the min-cut conditions while maintaining the flow-linked conditions across G_i, $\forall i$.

1. **Min-cut processing stage.** Formally, let $V(G_i)$ be the current set of vertices of G_i. We keep cutting off the smaller side S of a minimum cut, in terms of weight ρ_i, from G_i when $\mathsf{cap}(S, V(G_i) \setminus S)$ is less than $\hat{c}$, until every cut in G_i is at least $\hat{c}$, where we set $\hat{c} = \Omega(\log^3 n)$.
   By cutting off, we remove both nodes in S and edges that are adjacent to S in current G_i; this includes the cases when we get rid of any single node whose degree falls below $\hat{c}$ from its original degree of $\Omega(\log^5 n)$. We call such a stage a min-cut processing stage. Lemmas 10.3 and 10.4 bound the total number of edges that we lose from $G_1, \ldots, G_\ell$ and the flow that we further lose from $\mathsf{remaining\text{-}flow}$ (and hence $\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$) as in (8.2).
2. **Sparsest-cut processing stage.** In order to guarantee that we have an instance Y_i' that is ϖ_i-flow-linked in G_i for a new weight function ϖ_i, we need to further "mute" some terminals with a positive weight under ρ by setting their weight to zero under ϖ_i. This way, we can guarantee that every cut in G_i is good with respect to a product multicommodity flow demand that is defined based on the new weight function ϖ_i. We emphasize that we do not remove any nodes or edges in this stage; hence the min-cuts are guaranteed to be $\hat{c} = \Omega(\log^3 n)$. Lemma 10.4 bounds the flow that we further lose from $\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$, and Lemma 10.6 shows the final bound on $\mathsf{remaining\text{-}flow}$.

We have the following theorem about the instances that we have by the end of this postprocessing stage. The proof of this theorem is in section 10.

THEOREM 8.3. *Suppose we are given a graph $\mathcal{G}$ with min-cut value $C^0 \geq (4a_0\lambda(n) + a_0 + 2)\hat{c}$ for some $a_0 \geq 2$. By the end of the sparsest-cut processing, we obtain a set of node-disjoint induced subgraphs $\check{G}_1, \ldots, \check{G}_\ell$, all with min-cut at* *least $\hat{c}$, and the corresponding disjoint subsets $\mathcal{T}'_1, \dots, \mathcal{T}'_\ell$ of $\mathcal{T}$, such that terminal pairs in $\mathcal{T}'_i$ belong to $\check{G}_i$ and there exists a set of weight functions $\varpi_i : V(\check{G}_i) \to \mathbb{R}^+$ with the following properties:*

1. *there are $\eta_1, \dots, \eta_k$ such that $\forall u$ in a subgraph $\check{G}_i$, $\varpi_i(u) = \sum_{i:s_i=u,t_i\in\check{G}_i} 2\eta_i x_i$; note that this implies that $\forall (s_i,t_i) \in \mathcal{T}'_i$, x_i contributes the same amount of weight to $\varpi_i(s_i)$ and $\varpi_i(t_i)$;*
2. *Y'_i is ϖ_i-flow-linked in G'_i, where Y'_i is the set of terminals of T'_i;*
3. *$\forall u$ in a subgraph $\check{G}_i$ s.t. $\varpi_i(Y'_i) \geq \Omega(\log^3 n)$, $\varpi_i(u) \leq \sum_{i:s_i=u,t_i\in\check{G}_i} \frac{x_i}{\beta(\mathcal{G})\lambda(n)}$;*
4. *$\sum_{i=1}^{\ell} \varpi_i(Y'_i) = \Omega(\frac{\mathsf{OPT}^*(\mathcal{G},\mathcal{T})}{\lambda(n)\beta(\mathcal{G})})$, where $\lambda(n) = \beta(\mathcal{G}) \log \mathsf{OPT}^*(\mathcal{G}, T)$ and $\beta(G)$ is the worst-case min-cut–max-flow gap on product multicommodity flow instances on $\mathcal{G}$.*

8.3. Defining weights for Theorem 3.1. Finally, we define a weight function π on $V(\mathcal{G})$ as follows: (a) $\forall i, \forall u \in \check{G}_i$, where $\check{G}_i$ is a subgraph of $\mathcal{G}$, we assign $\pi(u) = \varpi_i(u)/2$; and (b) we assign $\pi(u) = 0$ for nodes of $V(G)$ not in any $\check{G}_i$. We thus have defined the weight function $\pi : V(\mathcal{G}) \to \mathbb{R}^+$ on the entire set of nodes of $\mathcal{G}$ as required by Theorem 3.1 with the same decomposition as we obtain for Theorem 8.3.

9. Details regarding CKS flow-linked decompositions. Recall that all subgraphs $G_1, G_2, \dots, G_\ell$ produced by the recursive decomposition procedure in Theorem 8.1 satisfy one of the following conditions:

1. The flow is sufficiently small, in that $\gamma(G_i) \leq \lambda(n)/10$.
2. Else, $f_0\mathsf{dem}(u,v)$ units of flow can be simultaneously routed $\forall uv$ in G_i with congestion 1 in G_i, where

$$f_0 = \frac{1}{2\lambda(n)}, \tag{9.1}$$

and the product demands are specified based on the original induced flow values $\gamma(u, G_i)$ at each node $u \in V(G_i)$ of $\bar{f}$ in G_i, $\forall i$ as follows:

$$\forall u, v \in V(G_i), \quad \mathsf{dem}(u,v) = \frac{\gamma(u,G_i)\gamma(v,G_i)}{\gamma(G_i)}. \tag{9.2}$$

It is clear from (9.2) and (9.1) that for a scaled-down product flow problem $\mathsf{dem}^{\rho_i}(u,v)$, such that each demand is f_0 of the original, $\forall uv \in V(G_i)$,

$$\mathsf{dem}^{\rho_i}(u,v) = \frac{\rho_i(u)\rho_i(v)}{\rho_i(Y_i)} = \frac{\gamma(u,G_i)\gamma(v,G_i)}{2\lambda(n)\gamma(G_i)} = \frac{\mathsf{dem}(u,v)}{2\lambda(n)} = f_0\mathsf{dem}(u,v),$$

there is a feasible flow in G_i since the concurrent max-flow value is at least 1. This actually applies to the case when $\gamma(G_i) \leq \lambda(n)/10$. Depending on the context, we may prefer to use the original product flow $\mathsf{dem}(u,v)$ instead of the feasible product flow $\mathsf{dem}^{\rho_i}(u,v)$, or the other way around.

10. Proof of Theorem 8.3. The analysis of this section will lead to the proof of Theorem 8.3 eventually. Throughout this section, we keep reducing the set of terminal pairs of $\mathcal{T}_i$ that are relevant, in the sense that these pairs will remain to be candidate pairs that we eventually route edge disjointly in $\mathcal{G}$. Therefore, we keep track of the following set of parameters in each subgraph G_i that we obtain through flow decomposition:

- $\mathcal{T}_i$: the induced pairs of $\mathcal{T}$ in G_i that we still consider to route edge disjointly;

- a weight function ρ_i defined on the $V(G_i)$, with positive values only on terminals X_i of $\mathcal{T}_i$; it evolves from ρ_i to ϖ_i, upon which π is defined;
- finally, we use $\mathsf{remaining\text{-}flow}$ as defined and initially bounded in (8.2) to keep track of the total remaining flow of $\bar{f}$ between terminal pairs in G_i, across all i; note that $\mathsf{remaining\text{-}flow}$ is the lower bound on $\sum_i |\mathcal{T}_i|$.

We are going to keep computing the original flow of $\bar{f}$ that we lose during the post-processing stages. We specify the following parameters that are related to min-cuts:

1. $\hat{c}$: the smallest minimum cut value that we allow in G_i, $\forall i$, which is $\Theta(\log^3 n)$.
2. $\mathcal{C}^0$: the minimum cut value in the original graph $\mathcal{G}$, which is $\Omega(\log^5 n)$.
3. $\mathsf{LOSS} \le \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})/2$: the number of edges that are cut during the CKS flow-decomposition process.

We analyze the min-cut processing in the next two sections. Formally, let $V(G_i)$ be the current set of vertices of G_i, which keeps shrinking as follows. We keep cutting off the smaller side S of a minimum cut, in terms of weight ρ_i, from G_i when $\mathsf{cap}(S, V(G_i) \setminus S)$ is less than $\hat{c}$, until every cut in G_i is at least $\hat{c}$. By cutting off, we remove both nodes in S and edges that are adjacent to S in current G_i.

Let $S_i^1, S_i^2, \dots, S_i^{x_i}$ be the sets of vertices that we take away from G_i and in that order. We define the following notation to track this process of updating G_i.

- $G_i^0 = (V_i^0, E_i^0)$: the subgraph G_i before any of $S_i^t, t = 1, \dots, x_i$, has been taken out.
- Y_i^0: the set of terminals of G_i^0 right after flow decomposition, such that Y_i^0 is ρ_i-flow-linked in G_i^0, as guaranteed by the CKS decomposition.
- $G_i^t = (V_i^t, E_i^t)$, $\forall t = 1, \dots, x_i$: the remaining subgraph of G_i^0 after removing $S_i^1, \dots, S_i^t$ and their adjacent edges; hence $V_i^t = V_i^0 \setminus \cup_{j=1,\dots,t} S_i^j$.
- Let the remaining subgraph of G_i^0 by the end of the min-cut processing stage be

$$\check{G}_i = (\check{V}_i, \check{E}_i) := G_i^{x_i} = (V_i^{x_i}, E_i^{x_i}). \tag{10.1}$$

10.1. Bound edges lost due to the min-cut processing. Denote the number of edges that we take away from G_i^0 due to the min-cut processing by $\mathsf{edge\text{-}loss}_i$, $\forall i$.

DEFINITION 10.1. $\mathsf{edge\text{-}loss}_i$ *is the sum of capacities of the minimum cuts that have caused* $S_i^1, \dots, S_i^{x_i}$ *to be cut off from* $G_i, \forall i$. *Denote the sum of* $\mathsf{edge\text{-}loss}_i$ *across all* i *by* $\mathsf{edge\text{-}loss}$*:*

$$\mathsf{edge\text{-}loss} = \sum_{i=1,2,\dots} \mathsf{edge\text{-}loss}_i = \sum_{i=1,2,\dots} \sum_{t=1,\dots,x_i} \mathsf{cap}(S_i^t, V_i^t).$$

In addition, we define the total flow of $\bar{f}$ *that we lose during this process by* $\mathsf{flow\text{-}loss}_1$.

Remark 10.2. Note that the number of edges that we take away from the final set of nodes $V(G_i) = V_i^{x_i} = V_i^0 \setminus \cup_{j=1,\dots,x_i} S_i^j$ due to the min-cut processing is upper bounded and in fact may be smaller than $\mathsf{edge\text{-}loss}_i$, $\forall i$.

We prove the following lemma in this section.

LEMMA 10.3. *The total number of edges that we take away from decomposed subgraphs* $G_i^0, G_i^1, \dots$ *for* $C^0 > 2\hat{c}$ *is at most*

$$\mathsf{edge\text{-}loss} = \sum_{i=1,2,\dots} \mathsf{edge\text{-}loss}_i \le \frac{2 \cdot \mathsf{LOSS} \cdot \hat{c}}{\mathcal{C}^0 - 2\hat{c}}. \tag{10.2}$$

Proof. We use a potential function $\psi(G_i)$ to count the number of edges we lose from nodes currently in G_i, as compared to the original graph $\mathcal{G} = (V, E)$, while G_i

keeps shrinking due to its min-cut processing. The counting process is as follows. We start with a component G_i such that $\psi_i^0 = \mathsf{LOSS}_i$ denotes the number of edges that we initially lose from nodes in G_i^0 right after the CKS flow decomposition procedure. Hence

$$\psi_i^0 = \psi(G_i^0) = \mathsf{LOSS}_i \geq 0 \text{ and } \sum_{i=1,2,\ldots} \mathsf{LOSS}_i = 2 \cdot \mathsf{LOSS}. \tag{10.3}$$

When a subset S is cut off, it claims away some credit from the current $\psi(G_i)$, since S is cut off because $\mathsf{cap}(S, V(\mathcal{G}) \setminus S)$ has decreased from above $\mathcal{C}^0$ to its current size in G_i, $\mathsf{cap}(S, V(G_i) \setminus S)$, which is $\leq \hat{c}$, due to edges lost from nodes in S during the CKS flow decomposition. That is, edges lost from nodes in S have contributed to the current value of $\psi(G_i)$.

Let ψ_i^t be the value of $\psi(G_i)$ after taking t sets of vertices $S_i^1, \ldots, S_i^t$ and their adjacent edges away from G_i. Let $(S_i^{t+1}, V_i^t \setminus S_i^{t+1})$ be the minimum cut in G_i^t and S_i^{t+1} be the $(t+1)$st set of vertices that we cut off from G_i because $\mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1})$ is less than $\hat{c}$. Let us denote the size of the original cut $(S_i^{t+1}, V \setminus S_i^{t+1})$ in $\mathcal{G}$ with

$$\ell_i^{t+1} := \mathsf{cap}(S_i^{t+1}, V \setminus S_i^{t+1}) \geq \mathcal{C}^0. \tag{10.4}$$

The amount of credit S_i^{t+1} takes away from $\psi(G_i)$ is $(\mathsf{cap}(S_i^{t+1}, V \setminus S_i^{t+1}) - \mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1}))$, and the credit it puts back is $\mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1})$, since we remove edges in $(S_i^{t+1}, V_i^t \setminus S_i^{t+1})$ from G_i^t, in addition to the subgraph induced by S_i^{t+1} in G_i^t. Hence, we update $\psi(G_i)$ as follows:

$$\begin{aligned}\psi_i^{t+1} &= \psi_i^t - (\mathsf{cap}(S_i^{t+1}, V \setminus S_i^{t+1}) - \mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1})) + \mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1}) \\ &= \psi_i^t - (\ell_i^{t+1} - \mathsf{cap}(S_i^{t+1}, V_i^{t+1})) + \mathsf{cap}(S_i^{t+1}, V_i^{t+1}).\end{aligned}$$

Since $\mathsf{cap}(S_i^{t+1}, V_i^{t+1}) \leq \hat{c}$, we have $\psi_i^{t+1} \leq \psi_i^t - (\ell_i^{t+1} - \hat{c}) + \hat{c}$.

Since the credit that a cut puts back is much less than the credit that it spent, there is only a finite number x_i of such small cuts in G_i, $\forall i$. By the end of x_i rounds, there must be a nonnegative credit in $\psi(G_i)$, since nodes in current G_i can never gain any edges. Hence

$$0 \leq \psi(G_i) = \psi_i^x \leq \mathsf{LOSS}_i - (\ell_i^1 - \hat{c}) + \hat{c} - (\ell_i^2 - \hat{c}) + \hat{c} - \cdots - (\ell_i^{x_i} - \hat{c}) + \hat{c}.$$

Summing the above inequalities over all i, we have by (10.4) and (10.3)

$$\sum_{i=1,2,\ldots} x_i \cdot C^0 \leq \sum_{i=1,2,\ldots} \sum_{j=1,2,\ldots,x_i} \ell_i^j \leq 2 \cdot \mathsf{LOSS} + 2 \sum_{i=1,2,\ldots} x_i \cdot \hat{c}.$$

Hence the total number of minimum cuts across all G_i that we process is

$$\sum_{i=1,2,\ldots} x_i \leq \frac{2 \cdot \mathsf{LOSS}}{\mathcal{C}^0 - 2\hat{c}}. \tag{10.5}$$

Now summing $\mathsf{edge\text{-}loss}_i$ across all i as in Definition 10.1, we have

$$\mathsf{edge\text{-}loss} = \sum_{i=1,2,\ldots} \mathsf{edge\text{-}loss}_i = \sum_{i=1,2,\ldots} \sum_{t=1,\ldots,x_i} \mathsf{cap}(S_i^t, V_i^t) \leq \sum_{i=1,2,\ldots} x_i \cdot \hat{c}. \tag{10.6}$$

Plugging (10.5) into (10.6) shows that (10.2) holds. $\square$

10.2. Bound the flow loss due to the min-cut processing. In this section, we compute the total amount of flow that we lose from $\bar{f}$ due to the min-cut processing in the previous section.

For a set of nodes $S_i^t \in V_i^0$ for $t = 1, \ldots, x_i$, in $G_i^0 = (V_i^0, E_i^0)$, we denote the size of cut $(S_i^t, V_i^0 \setminus S_i^t)$ with $\Delta(S_i^t) := \mathsf{cap}(S_i^t, V_i^0 \setminus S_i^t)$. As we shall see, $\Delta(S_i^t)$ determines the amount of flow of $\bar{f}$ that we take away from $\gamma(G_i)$ when we remove S_i^t from G_i as the smaller side (in terms of weight ρ_i) of a min-cut (S_i^t, V_i^t) in G^{t-1}.

Thus we first derive an upper bound on $\Delta(S_i^t)$ by the following observation: Edges in $\Delta(S_i^t)$ come either from previous min-cuts, $\{(S_i^j, V_i^j) \text{ for } j < t\}$, or from a set of new edges that contribute to $\mathsf{cap}(S_i^t, V_i^t)$; however, each edge e counted in $\mathsf{edge\text{-}loss}_i$ can be used at most twice toward $\sum_{t=1}^{x_i} \Delta(S_i^t)$—once for each of the two neighboring sets in $\{S_i^t, t = 1, \ldots, x_i\}$ that share $e \in G_i^0$. Thus we have

$$\sum_{t=1}^{x_i} \Delta(S_i^t) \le 2 \sum_{t=1}^{x_i} \mathsf{cap}(S_i^t, V_i^t) = 2 \cdot \mathsf{edge\text{-}loss}_i$$

$$\text{and} \quad \sum_{i=1,2,\ldots} \sum_{t=1}^{x_i} \Delta(S_i^t) \le 2 \cdot \mathsf{edge\text{-}loss}. \tag{10.7}$$

LEMMA 10.4. *The total flow of $\bar{f}$ that we lose from min-cut processing is*

$$\mathsf{flow\text{-}loss}_1 \le \mathsf{edge\text{-}loss} \cdot (2\lambda(n) + 1/2).$$

Proof. We first derive a lower bound on $\Delta(S_i^t)$ based on the fact that Y_i^0 is ρ_i-flow-linked and hence Y_i^0 is also $\rho_i/2$-cut-linked in G_i^0 by Proposition 2.5. Hence

$$\Delta(S_i^t) := \mathsf{cap}(S_i^t, V_i^0 \setminus S_i^t) \ge \frac{\rho_i(S_i^t \cap Y_i^0)}{2} = \frac{\sum_{u \in S_i^t} \gamma(u, G_i)}{2\lambda(n)}, \tag{10.8}$$

where the equality is due to (8.1a) and (8.1b).

Now fix $\Delta(S_i^t)$ for some t. We now calculate the amount of flow of $\bar{f}$ that we lose by cutting off S_i^t. The flow that we lose falls into one of four types:

1. flow whose paths are entirely contained in the subgraph of G_i induced by S_i^t;
2. flow that has to go through edges that are counted in $\Delta(S_i^t)$ but not counted in (S_i^t, V_i^t);
3. flow that has to cross (S_i^t, V_i^t) with at least one endpoint in S_i^t;
4. flow with both endpoints $u'v'$ belonging to V_i^t such that the flow path intersects the min-cut (S_i^t, V_i^t) at least twice.

Flow of type 1 is counted in $\sum_{u \in S_i^t} \gamma(u, G_i)$ twice. Flow of type 2 has been counted before when S_i^j were cut off for some $j < t$. Flow of type 3 contributes its flow amount once to $\sum_{u \in S_i^t} \gamma(u, G_i)$ and once to the usage of $\mathsf{cap}(S_i^t, V_i^t)$. Flow of type 4 is counted twice in the usage of $\mathsf{cap}(S_i^t, V_i^t)$. Note that flow that crosses cut (S_i^t, V_i^t) either has been counted in $\sum_{u \in S_i^t} \gamma(u, G_i)$ at least once or crosses (S_i^t, V_i^t) at least twice. Hence we obtain an upper bound on the amount of flow that we lose from $\bar{f}$ which has not been counted earlier (i.e., of types 1, 3, and 4) due to cutting off the induced subgraph of S_i^t from G_i^{t-1} by

$$\begin{aligned} \frac{1}{2} \sum_{u \in S_i^t} \gamma(u, G_i) + \frac{1}{2}\mathsf{cap}(S_i^t, V_i^t) &\le \frac{1}{2}\rho_i(S_i^t \cap Y_i^0)\lambda(n) + \frac{1}{2}\mathsf{cap}(S_i^t, V_i^t) \\ &\le \Delta(S_i^t)\lambda(n) + \frac{1}{2}\mathsf{cap}(S_i^t, V_i^t), \end{aligned}$$

where the last inequality is due to (10.8). Summing over all S_i^t, $\forall t$, we obtain

$$\begin{aligned}\textsf{flow-loss}_1 &\le \sum_{i=1,2,\ldots}\sum_{t=1}^{x_i} 1/2\left(\sum_{u\in S_i^t}\gamma(u,G_i)+\textsf{cap}(S_i^t,V_i^t)\right)\\ &\le \sum_{i=1,2,\ldots}\sum_{t=1}^{x_i}\left(\Delta(S_i^t)\lambda(n)+\frac{1}{2}\textsf{cap}(S_i^t,V_i^t)\right)\\ &= \sum_{i=1,2,\ldots}\sum_{t=1}^{x_i}\Delta(S_i^t)\lambda(n)+\frac{1}{2}\textsf{edge-loss}.\end{aligned}$$

Thus the lemma holds given (10.7). $\square$

Let $1/a_0$ denote an upper bound on the ratio between the flow of $\bar{f}$ that we lose during min-cut processing and $\textsf{LOSS}$ in the CKS flow decomposition, $\frac{\textsf{flow-loss}_1}{\textsf{LOSS}} \le \frac{1}{a_0}$, which indeed holds for $\mathcal{C}^0 \ge (4a_0\lambda(n)+a_0+2)\cdot\hat{c}$, given Lemma 10.4 and (10.2):

$$\begin{aligned}\textsf{flow-loss}_1 &\le 2\textsf{edge-loss}\cdot\lambda(n)+\frac{1}{2}\textsf{edge-loss}\\ &\le \frac{2\textsf{LOSS}\cdot\hat{c}}{\mathcal{C}^0-2\hat{c}}(2\lambda(n)+1/2)\le\frac{\textsf{LOSS}}{a_0}.\end{aligned}\tag{10.9}$$

Finally, plugging $\mathcal{C}^0 \ge (4a_0\lambda(n)+a_0+2)\cdot\hat{c}$ into (10.2), we obtain the following bound on edge loss due to the min-cut processing in G_i:

$$\textsf{edge-loss}\le\frac{2\textsf{LOSS}\cdot\hat{c}}{\mathcal{C}^0-2\hat{c}}\le\frac{2\textsf{LOSS}\cdot\hat{c}}{a_0(4\lambda(n)+1)\cdot\hat{c}}=\frac{\textsf{LOSS}}{a_0(2\lambda(n)+\frac{1}{2})}.\tag{10.10}$$

10.3. Obtaining the final set of terminals. Recall that $G_i^0=(V_i^0,E_i^0)$ denotes the subgraph G_i we obtain through the CKS flow decomposition before any subset of nodes has been removed. Recall that the set of terminals Y_i of G_i is ρ_i-flow-linked in G_i. Now $\check{G}_i=(\check{V}_i,\check{E}_i)$, $\forall i$ are the remaining subgraphs of G_i, $\forall i$ at the end of the min-cut processing stage as in (10.1). In the sparsest-cut processing, we remove regions $Q_i^1,\ldots,Q_i^{y_i}$ from the graph $\check{G}_i$ (and in that order) that do not meet a certain sparsest-cut condition using the algorithm shown in Figure 10.1. In the end, we have a subgraph

$$G_i' := \check{G}_i\left[\check{V}_i\setminus\bigcup_{j=1}^{y_i}Q_i^j\right]$$

that does meet the sparsest-cut condition on the demands in the remaining subgraph (cf. (10.12) and (10.11)).

Now we assign a zero weight to all vertices in the removed regions so that demands on these regions are zero; we then put $Q_i^1,\ldots,Q_i^{y_i}$ all back in. This graph $\check{G}_i$ is more connected only with regard to the remaining demands induced by $\bar{f}$ inside G_i', $\forall i$. Hence we emphasize that $\check{G}_i$, $\forall i=1,\ldots,\ell$ is the set of subgraphs that we pass on to the next stage. We give an algorithm for computing the final disjoint subsets $\mathcal{T}_1',\ldots,\mathcal{T}_\ell'$ of $\mathcal{T}$ such that terminal pairs in $\mathcal{T}_i'$ belong to G_i', and hence $\check{G}_i$, $\forall i$, and assigning a positive weight ϖ_i to the set of terminals in $\mathcal{T}_i'$, $\forall i$ (cf. (10.19)). In the rest of this section, we prove Theorem 8.3 by first describing our algorithm and setting up the corresponding notation.

0. Given a subgraph $\check{G}_i$.
1. If $\gamma(\check{G}_i) \le (a_1/4)\beta\lambda(n)$, $\varpi_i(u) = \varpi_i(v) = 1$ for some pair $uv \in \mathcal{T}'_i$ with positive flow in $\check{G}_i$; and $\varpi_i(y) = 0$ for $y \ne u, v$.
   Hence we can just route a unit flow between the chosen pair $uv \in \mathcal{T}'_i$ along an integral path; such a path exists since $\check{G}_i$ is a connected component.
2. Suppose that $\gamma(\check{G}_i) > (a_1/4)\beta\lambda(n)$. For $\mathsf{dem}(u,v) = \gamma(u,\check{G}_i)\gamma(v,\check{G}_i)/\gamma(\check{G}_i)$, let f' be the maximum concurrent flow for this instance.
   (a) if $f' \ge f_1$, set $\varpi_i(u) = \frac{\gamma(u,\check{G}_i)}{(a_1/2)\beta\lambda(n)}$ $\forall u \in \check{V}_i$ and stop.
   (b) else $f' < f_1$, find an approximate sparsest cut s.t.
   $\frac{\mathsf{cap}(S, V(\check{G}_i)\setminus S)}{\mathsf{dem}(S, V(\check{G}_i)\setminus S)} \le \beta f'$; set $\varpi_i(u) = 0$, $\forall u \in S$,
   where S is the smaller side in terms of weight ρ_i,
   and shut off edges in $\delta^0(S) = (S, \check{V}_i \setminus S)$
   so that we recurse on $\check{G}_i[V(\check{G}_i) \setminus S]$.
3. End

FIG. 10.1. *Algorithm* FINDING SPARSEST CUTS.

Given a subgraph $\check{G}_i = (\check{V}_i, \check{E}_i) := \check{G}_i^0$, we use the procedure in Figure 10.1 to update $\check{G}_i$ recursively by muting regions that do not satisfy the sparsest-cut condition; by "muting" a region Q, we treat nodes in Q and their adjacent edges as if they were removed from $\check{G}_i$ during the sparsest-cut processing stage, although in the end, we retain these regions entirely in $\check{G}_i$. We define the following parameters given a remaining subgraph $\check{G}_i^t$ of G^i after muting some regions, $Q_i^1, \ldots, Q_i^t$, where $t \ge 1$:

1. $\check{G}_i^t = (\check{V}_i^t, \check{E}_i^t)$: the remaining subgraph of $\check{G}_i$ after muting nodes in $Q_i^1, \ldots, Q_i^t$ and their adjacent edges; $\check{V}_i^t = \check{V}_i \setminus \cup_{j=1,\ldots,t} Q_i^j$ is the remaining set of vertices in $\check{G}_i$ at stage t;
2. $\delta^t(S) = \mathsf{cap}(S, \check{V}_i^t \setminus S)$: the size of a new cut $(S, \check{V}_i^t \setminus S)$ in subgraph $\check{G}_i^t$;
3. $\Delta(S) = \mathsf{cap}(S, V_i^0 \setminus S)$: the size of an original cut $(S, V_i^0 \setminus S)$ in subgraph G_i^0; note that $\Delta(S)$ determines the amount of flow of $\bar{f}$ that we take away from $\gamma(\check{G}_i)$ when we set $\varpi_i(u) = 0$, $\forall u \in S$ in Figure 10.1.

In order for subgraph $\check{G}_i^t$ to satisfy a flow-linked property, we define

$$f_1 = \frac{1}{a_1\beta(G)\lambda(n)}, \quad \text{where} \quad a_1 > 8, \tag{10.11}$$

as the minimum concurrent flow value that one needs to obtain for $\mathsf{dem}^t(u,v)$ in $\check{G}_i^t$, where

$$\forall u, v \in \check{V}_i^t, \ \ \mathsf{dem}^t(u,v) = \frac{\gamma(u,\check{G}_i^t)\gamma(v,\check{G}_i^t)}{\gamma(\check{G}_i^t)} \tag{10.12}$$

specifies demands between any unordered pair of vertices based on their induced flow values $\gamma(u,\check{G}_i^t)$ at each node $u \in \check{V}_i^t$ of $\bar{f}$ in $\check{G}_i^t$. When the actual flow value $f' < f_1$, we can find a set Q_i^{t+1} such that

$$\delta^t(Q_i^{t+1}) := \mathsf{cap}(Q_i^{t+1}, \check{V}_i^t \setminus Q_i^{t+1}) \le \mathsf{dem}^t(Q_i^{t+1}, \check{V}_i^t \setminus Q_i^{t+1})\beta f'. \tag{10.13}$$

We say that Q_i^{t+1} does not meet the sparsest-cut condition for the demands $\mathsf{dem}^t(u,v)$ in subgraph $\check{G}_i^t$, and we set $\varpi_i(u) = 0$, $\forall u \in Q_i^{t+1}$ in $\check{G}_i^t$ and recurse on

$$\check{G}_i^{t+1} := \check{G}_i[\check{V}_i^t \setminus Q_i^{t+1}]. \tag{10.14}$$

When the flow value $f' \geq f_1$, we stop the recursion and assign

$$\varpi_i(u) = \frac{\gamma(u, \check{G}_i^t)}{(a_1/2)\beta\lambda(n)} \ \ \forall u \in \check{V}_i^t. \tag{10.15}$$

We need to tune two parameters, a_0 and a_1, to balance the initial min-cut condition and remaining-flow in Lemma 10.6. We first give a bound on flow-loss due to the sparsest-cut processing.

LEMMA 10.5. *The amount of flow that we lose from $\sum_{i=1}^{\ell} \gamma(\check{G}_i)$ due to the sparsest-cut processing is, for $a_1 > 8$,*

$$\mathsf{flow\text{-}loss}_2 \leq \frac{\mathsf{LOSS}}{2a_0(1 - 8/a_1)}.$$

The proof of Lemma 10.5 is given in section 10.5. Now it is clear that remaining-flow, the total amount of flow of $\bar{f}$ that we retain by the end of min-cut and sparsest-cut processing stages, is the sum of flow of $\bar{f}$ induced in G_i' across all i:

$$\mathsf{remaining\text{-}flow} := \sum_{i=1}^{\ell} \gamma(G_i') = \sum_{i=1}^{\ell} \gamma(\check{G}_i) - \mathsf{flow\text{-}loss}_2. \tag{10.16}$$

We have the following guarantee by the end of the sparsest-cut processing stage for $a_1 > 8$.

LEMMA 10.6. *Given a graph $\mathcal{G}$ with min-cut value $C^0 \geq (4a_0\lambda(n) + a_0 + 2)\hat{c}$, where $a_0 > 2$, we have for $a_1 > 8$*

$$\mathsf{remaining\text{-}flow} \geq \frac{\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})}{2} \left(1 - \frac{1}{a_0} - \frac{1}{2a_0(1 - 8/a_1)}\right). \tag{10.17}$$

Proof. The total flow of $\bar{f}$ that remains by the end of the min-cut processing stage is the sum of flow of $\bar{f}$ induced in $\check{G}_i$, across all i; thus we have by definition of LOSS, which is $\leq \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})/2$,

$$\sum_{i=1}^{\ell} \gamma(\check{G}_i) \geq \frac{\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})}{2} - \mathsf{flow\text{-}loss}_1 \geq \frac{1}{2}\mathsf{OPT}^*(\mathcal{G}, \mathcal{T}) \left(1 - \frac{1}{a_0}\right), \tag{10.18}$$

where $\mathsf{flow\text{-}loss}_1 \leq \mathsf{LOSS}/a_0$ by (10.9). Combining this initial condition with Lemma 10.5 and (10.16), we conclude that (10.17) holds. ∎

Let the set of terminal pairs $\mathcal{T}_i'$ be the subset of $\mathcal{T}_i$ that is contained in subgraph G_i'. Let Y_i' be the set of terminals of $\mathcal{T}_i'$. Hence by the end of the sparsest-cut processing stage, we get a new instance Y_i' on $\check{G}_i = (\check{V}_i, \check{E}_i)$ with min-cut at least $\hat{c} = \Omega(\log^3 n)$, such that Y_i' is ϖ_i-flow-linked in $\check{G}_i$, which can be only more connected than G_i'. Properties regarding the set of Y_i' are summarized in the proof of Theorem 8.3.

10.4. Proof of Theorem 8.3. Let $G_i' = \check{G}_i^{y_i} = (\check{V}_i^{y_i}, \check{E}_i^{y_i})$ be the remaining subgraph of $\check{G}_i$ by the end of the sparsest-cut processing stage after muting nodes in $Q_i^1, \ldots, Q_i^{y_i}$, and let their adjacent edges be as described and bounded in (10.11)–(10.17).

Now if $\gamma(G_i') \leq (a_1/4)\beta\lambda(n)$ when the algorithm terminates, we have obtained a terminal pair $\mathcal{T}_i'$ to route in G_i' and a weight assignment that satisfies all three conditions in the theorem following step 1 of Figure 10.1; hence we are done with this subgraph. When $\gamma(G'_i) > (a_1/4)\beta\lambda(n)$, a product flow based on the flow of $\bar{f}$ induced in G'_i is routable with throughput at least f_1 as in (10.11). Hence by assigning a new weight $\varpi_i(\check{V}_i) \to \mathbb{R}^+$,

$$\varpi_i(u) = \frac{\gamma(u, G'_i)}{(a_1/2)\beta\lambda(n)},\ \forall u \in V(G'_i) \text{ and } \varpi_i(u) = 0,\ \forall u \in \check{V}_i \setminus V(G'_i), \tag{10.19}$$

for which we can define a multicommodity flow problem: for any unordered pair of vertices $u, v \in V(G'_i)$, $\mathsf{dem}^{\varpi_i}(u,v) = \varpi_i(u)\varpi_i(v)/\varpi_i(Y'_i)$, there is a feasible flow in both G' and $\check{G}_i$. Hence Y'_i is ϖ_i-flow-linked in $\check{G}_i$, $\forall i$.

Finally, we put Q_i^t, $\forall t$ back in $\check{G}_i$ with zero node weight, while retaining the same weight assignment for nodes in G'_i. Hence the sum of the total weight is

$$\varpi_i(\check{G}_i) = \varpi_i(G'_i) = \sum_{u \in G'_i} \frac{\gamma(u, G'_i)}{(a_1/2)\beta\lambda(n)} = \frac{\gamma(G'_i)}{(a_1/4)\beta\lambda(n)}. \tag{10.20}$$

Hence for both terminating conditions of the algorithm, we have $\varpi_i(G'_i) \geq \frac{\gamma(G'_i)}{(a_1/4)\beta\lambda(n)}$, and thus

$$\sum_{i=1}^{\ell} \varpi_i(Y'_i) \geq \sum_{i=1}^{\ell} \frac{\gamma(G'_i)}{(a_1/4)\beta\lambda(n)} \geq \frac{\mathsf{OPT}^*(\mathcal{G}, \mathcal{T})}{16\beta\lambda(n)},$$

where $\sum_{i=1}^{\ell} \gamma(G'_i) \geq \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})/4$ by taking $a_0 = 4$ and $a_1 = 16$ in Lemma 10.6. ∎

Remark 10.7. We note that only those pairs $(u, v) \in \mathcal{T}'_i$ have their flow of $\bar{f}$ that is entirely contained in G' contributing to ϖ_i in $\check{G}_i$. In particular, for all pairs $v \in G'_i$ of a node $u \in \bigcup_{j=1}^{y_i} Q_i^j$ has zero contribution from $x_{(u,v)}$ toward $\varpi_i(v)$ (cf. condition 1 of Theorem 8.3).

10.5. Proof of Lemma 10.5. The structure of this proof is exactly the same as that of Lemmas 10.3 and 10.4. To analyze the amount of flow that we lose from the sparsest-cut processing stage, we use a potential function $\varphi(\check{G}_i)$ to keep track of the edges of $G_i^0 = (V_i^0, E_i^0)$ that we lose from nodes currently in $\check{G}_i$. Recall that $G_i^0 = (V_i^0, E_i^0)$ denotes the subgraph G_i we obtain through the CKS flow decomposition, where the subset of terminals Y_i is ρ_i-flow-linked in G_i.

Let $\check{G}_i = (\check{V}_i, \check{E}_i)$, $\forall i$ be the remaining subgraphs of G_i, $\forall i$ at the end of the min-cut processing stage as in (10.1). Initially, some nodes in $\check{G}_i$ have lost their edges due to the min-cut processing, and hence we have $\forall i$

$$\varphi_i^0 = \mathsf{edge\text{-}loss}_i \geq 0.$$

We update this function during the sparsest-cut processing stage as follows.

Let φ_i^t be the value of $\varphi(\check{G}_i)$ after removing t sets of vertices $Q_i^1, \ldots, Q_i^t$ and their adjacent edges from $\check{G}_i$, which remains nonnegative since nodes currently in $\check{G}_i^t$, $\forall t \geq 0$, can never gain any internal edges. Note that those lost edges connect to other nodes in V_i^0 from nodes internal to $\check{G}_i^t$ at stage t.

Now let Q_i^{t+1} be the $(t+1)$st set of vertices, where $t \geq 0$, that we shut off from $\check{G}_i$, as the initial boundary capacity of Q_i^{t+1} has decreased from $\Delta(Q_i^{t+1})$ to $\delta^t(Q_i^{t+1}) \leq \mathsf{dem}^t(Q_i^{t+1}, \check{V}_i^t \setminus Q_i^{t+1})\beta f'$. We update $\varphi(G_i)$ as

$$\varphi_i^{t+1} = \varphi_i^t - (\Delta(Q_i^{t+1}) - \delta^t(Q_i^{t+1})) + \delta^t(Q_i^{t+1}),$$

where the two types of cuts are bounded as follows.

Fixing Q_i^{t+1} for some $i, t \in [0, \ldots, y_i - 1]$, we have the following two lemmas, whose proofs are provided at the end of this section.

LEMMA 10.8. *For all i and all $t \in [0, \ldots, y_i - 1]$,*

$$\Delta(Q_i^{t+1}) = \mathsf{cap}(Q_i^{t+1}, V_i^0 \setminus Q_i^{t+1}) \geq \sum_{u \in Q_i^{t+1}} \frac{\gamma(u, \check{G}_i^t)}{2\lambda(n)}.$$

LEMMA 10.9. *For all i and all $t \in [0, \ldots, y_i - 1]$, and for $a_1 \geq 8$,*

$$\delta^t(Q_i^{t+1}) \leq \frac{4\Delta(Q_i^{t+1})}{a_1} \leq \frac{\Delta(Q_i^{t+1})}{2}. \tag{10.21}$$

Since the credit $\delta^t(Q_i^{t+1})$ that a cut puts back is less than the credit $\Delta(Q_i^{t+1}) - \delta^t(Q_i^{t+1})$ that it spends, and $\varphi_i^t \geq 0$, $\forall t \geq 1$, there is only a finite number y_i of such small cuts. In summary, we have for all i

$$\begin{aligned}
\varphi(\check{G}_i) &= \varphi_i^{y_i} \\
&= \mathsf{edge\text{-}loss}_i - (\Delta(Q_i^1) - \delta^0(Q_i^1)) + \delta^0(Q_i^1) - (\Delta(Q_i^2) - \delta^1(Q_i^2)) + \delta^1(Q_i^2) \\
&\quad - \cdots - (\Delta(Q_i^{y_i}) - \delta^{y_i - 1}(Q_i^{y_i})) + \delta^{y_i - 1}(Q_i^{y_i}) \\
&\geq 0.
\end{aligned}$$

Now summing the above inequalities over all i, we have

$$\sum_{i=1}^{\ell} \sum_{j=1}^{y_i} (\Delta(Q_i^j) - 2\delta^{j-1}(Q_i^j)) \leq \mathsf{edge\text{-}loss} \leq \frac{\mathsf{LOSS}}{a_0(2\lambda(n) + \frac{1}{2})}. \tag{10.22}$$

Plugging (10.21) into (10.22), we get $\sum_{i=1}^{\ell} \sum_{j=1}^{y_i} \Delta(Q_i^j)(1 - 8/a_1) \leq \frac{\mathsf{LOSS}}{a_0(2\lambda(n)+1/2)}$ and

$$\sum_{i=1}^{\ell} \sum_{j=1}^{y_i} \Delta(Q_i^j) = \frac{\mathsf{LOSS}}{a_0(2\lambda(n) + \frac{1}{2})(1 - 8/a_1)}. \tag{10.23}$$

Now fix $\check{G}_i^t = (\check{V}_i^t, \check{E}_i^t)$ for some $t \in [1, \ldots, y_i]$. We now calculate the amount of flow of $\bar{f}$ that we lose from $\sum_{i=1}^{\ell} \gamma(\check{G}_i)$ by shutting off Q_i^{t+1} in $\check{G}_i$. The flow that we lose falls into one of four types:

1. its path is entirely contained in the subgraph of $\check{G}_i$ induced by nodes in Q_i^{t+1};
2. its path contains edges counted in $\Delta(Q_i^{t+1})$ but not those in $\delta^t(Q_i^{t+1})$;
3. its path contains edges counted in $\delta^t(Q_i^{t+1})$ but with at least one endpoint in Q_i^{t+1};
4. flow with both endpoints $u'v' \in \check{V}_i^{t+1}$, such that its path intersects edges counted in $\delta^t(Q_i^{t+1})$ at least twice.

Flow of type 1 contributes to the sum $\sum_{u \in Q_i^{t+1}} \gamma(u, \check{G}_i^t)$ twice. Flow of type 3 contributes its flow value to $\sum_{u \in Q_i^{t+1}} \gamma(u, \check{G}_i^t)$ once and to the usage of $\delta^t(Q_i^{t+1}) = \mathsf{cap}(Q_i^{t+1}, \check{V}_i^{t+1})$ at least once. Flow of type 4 contributes its flow amount at least twice to the usage of $\mathsf{cap}(Q_i^{t+1}, \check{V}_i^{t+1})$. Flow of type 2 has been counted before when Q_i^j were muted for some $j \leq t$ from $\check{G}_i$. Note that those flow that cross (Q_i^{t+1}, V_i^{t+1}) either have been counted in $\sum_{u \in Q_i^{t+1}} \gamma(u, \check{G}_i^t)$ at least once or go through the cut (Q_i^{t+1}, V_i^{t+1}) in $\check{G}_i^t$ at least twice.

Hence the total amount of flow of $\bar{f}$ that we lose from $\gamma(\check{G}_i)$, that have not been counted in stages earlier than t, by muting the induced subgraph of Q_i^{t+1} and its adjacent edges in $\check{G}_i^t$, is bounded by

$$\frac{1}{2}\left(\sum_{u\in Q_i^{t+1}}\gamma(u,\check{G}_i^t)+\mathsf{cap}(Q_i^{t+1},V_i^{t+1})\right)\le\Delta(Q_i^{t+1})\lambda(n)+\frac{1}{2}\delta^t(Q_i^{t+1})$$

by Lemma 10.8. Now summing over all Q_i^t, $\forall t,\forall i$, the total amount of flow lost in the sparsest-cut processing stage is bounded as follows for $a_1\ge 8$:

$$\begin{aligned}\mathsf{flow\text{-}loss}_2 &\le \frac{1}{2}\sum_{i=1}^{\ell}\sum_{t=1}^{y_i}\left(\sum_{u\in Q_i^t}\gamma(u,\check{G}_i^{t-1})+\mathsf{cap}(Q_i^t,V_i^t)\right)\\ &\le \sum_{i=1}^{\ell}\sum_{t=1}^{y_i}\Delta(Q_i^t)(\lambda(n)+2/a_1)\\ &\le \frac{(\lambda(n)+2/a_1)\mathsf{LOSS}}{2a_0(\lambda(n)+1/4)(1-8/a_1)}\\ &\le \frac{\mathsf{LOSS}}{2a_0(1-8/a_1)}\end{aligned}$$

by Lemma 10.9 and (10.23). Thus Lemma 10.5 holds. □

Proof of Lemma 10.8. Note that in $\check{G}_i^t$, Q_i^{t+1} is the smaller side of the cut $(Q_i^{t+1},\check{V}_i^{t+1})$ in terms of weight. Thus we have for $\check{V}_i^t=Q_i^{t+1}\bigcup\check{V}_i^{t+1}$

$$\sum_{u\in Q_i^{t+1}}\gamma(u,\check{G}_i^t)\le\frac{1}{2}\sum_{u\in\check{V}_i^t}\gamma(u,\check{G}_i^t)=\gamma(\check{G}_i^t). \tag{10.24}$$

Next let us define a_2 as the additional flow of $\bar{f}$ for node $u\in Q_i^{t+1}$ induced in G_i^0 as compared to that induced in subgraph $\check{G}_i^t$; hence by definition of a_2 and the fact that $Q_i^{t+1}\subseteq\check{V}_i^t\subseteq V(G_i^0)$, we have

$$a_2+\sum_{u\in Q_i^{t+1}}\gamma(u,\check{G}_i^t):=\sum_{u\in Q_i^{t+1}}\gamma(u,G_i^0),\quad\text{and hence} \tag{10.25}$$

$$\begin{aligned}\sum_{u\in\check{V}_i^t}\gamma(u,\check{G}_i^t)+a_2 &= \sum_{u\in Q_i^{t+1}}\gamma(u,\check{G}_i^t)+a_2+\sum_{u\in\check{V}_i^t\setminus Q_i^{t+1}}\gamma(u,\check{G}_i^t)\\ &\le \sum_{u\in Q_i^{t+1}}\gamma(u,G_i^0)+\sum_{u\in\check{V}_i^t\setminus Q_i^{t+1}}\gamma(u,\check{G}_i^0)\\ &= \sum_{u\in\check{V}_i^t}\gamma(u,G_i^0)\le\sum_{u\in Y_i^0}\gamma(u,G_i^0).\end{aligned} \tag{10.26}$$

Thus we have by (10.25), (10.24), and (10.26)

$$\sum_{u\in Q_i^{t+1}}\gamma(u,G_i^0)=\sum_{u\in Q_i^{t+1}}\gamma(u,\check{G}_i^t)+a_2 \tag{10.27}$$

$$\le\frac{\sum_{u\in\check{V}_i^t}\gamma(u,\check{G}_i^t)}{2}+\frac{a_2}{2}+\frac{a_2}{2}$$

$$\le\sum_{u\in Y_i^0}\frac{\gamma(u,G_i^0)}{2}+\frac{a_2}{2}. \tag{10.28}$$

We have by Theorem 8.1 and Proposition 2.5 that G_i^0 is a $\frac{\rho_i}{2}$-cut-linked instance.

- Now if $Q_i^{t+1} \subseteq \check{V}_i^t \subseteq V_i^0$ is the smaller side of the cut $(Q_i^{t+1}, V_i^0 \setminus Q_i^{t+1})$ in G_i^0 in terms of weight ρ_i, we have

$$\Delta(Q_i^{t+1}) = \mathsf{cap}(Q_i^{t+1}, V_i^0 \setminus Q_i^{t+1})$$

$$\geq \frac{1}{2}\rho_i(Q_i^{t+1} \cap Y_i^0) = \frac{\sum_{u \in Q_i^{t+1}} \gamma(u, G_i^0)}{2\lambda(n)} \tag{10.29}$$

$$\geq \frac{\sum_{u \in Q_i^{t+1}} \gamma(u, \check{G}_i^t)}{2\lambda(n)}, \tag{10.30}$$

  where (10.29) is by definition of ρ_i in (8.1a) and (8.1b).
- Otherwise, suppose we have $\sum_{u \in Q_i^{t+1}} \gamma(u, G_i^0) \geq \frac{1}{2} \sum_{u \in V_i^0} \gamma(u, G_i^0)$ such that $\rho_i(Q_i^{t+1} \cap Y_i^0) \geq \frac{1}{2}\rho_i(Y_i^0)$. First we have by (10.28)

$$\sum_{u \in V_i^0 \setminus Q_i^{t+1}} \gamma(u, G_i^0) = \sum_{u \in V_i^0} \gamma(u, G_i^0) - \sum_{u \in Q_i^{t+1}} \gamma(u, G_i^0)$$

$$\geq \sum_{u \in Q_i^{t+1}} \gamma(u, G_i^0) - a_2. \tag{10.31}$$

  We then apply the $\rho_i/2$-cut-linked condition to $\rho_i(V_i^0 \setminus Q_i^{t+1})$ to obtain

$$\begin{aligned}
\Delta(Q_i^{t+1}) &= \mathsf{cap}(Q_i^{t+1}, V_i^0 \setminus Q_i^{t+1}) \\
&\geq \frac{\rho_i((V_i^0 \setminus Q_i^{t+1}) \cap Y_i^0)}{2} = \frac{\sum_{u \in V_i^0 \setminus Q_i^{t+1}} \gamma(u, G_i^0)}{2\lambda(n)} \\
&\geq \frac{\sum_{u \in Q_i^{t+1}} \gamma(u, G_i^0) - a_2}{2\lambda(n)} \\
&= \frac{\sum_{u \in Q_i^{t+1}} \gamma(u, \check{G}_i^t)}{2\lambda(n)},
\end{aligned}$$

  where the last two (in)equalities are due to (10.31) and (10.25). ∎

Proof of Lemma 10.9. By the terminating condition 2(b) in Figure 10.1, (10.13), and (10.11), we have for $\mathsf{dem}^t(Q_i^{t+1}, \check{V}_i^t \setminus Q_i^{t+1})$ as defined in (10.12)

$$\begin{aligned}
\delta^t(Q_i^{t+1}) &\leq \mathsf{dem}^t(Q_i^{t+1}, \check{V}_i^t \setminus Q_i^{t+1})\beta f_1 \\
&\leq \frac{\sum_{u \in Q_i^{t+1}} \gamma(u, \check{G}_i^t) \cdot \sum_{v \in \check{V}_i^{t+1}} \gamma(v, \check{G}_i^t)}{a_1 \lambda(n) \cdot \gamma(\check{G}_i^t)} \\
&\leq \frac{2\sum_{u \in Q_i^{t+1}} \gamma(u, \check{G}_i^t)}{a_1 \lambda(n)} = \frac{4}{a_1}\left(\frac{\sum_{u \in Q_i^{t+1}} \gamma(u, \check{G}_i^t)}{2\lambda(n)}\right) \\
&\leq \frac{4}{a_1}\Delta(Q_i^{t+1}),
\end{aligned}$$

where the last two inequalities are due to the fact that $\sum_{v \in \check{V}_i^{t+1}} \gamma(v, \check{G}_i^t) \leq 2\gamma(\check{G}_i^t)$ and to Lemma (10.8). ∎

TABLE 10.1
Parameters related to decomposition of an EDP instance $(\mathcal{G}, \mathcal{T})$.

Parameters	Meaning
a_0	$\sim> 2$; $a_0 = 4$ suffices for Theorem 8.3 to hold
a_1	$\sim> 8$; $a_1 = 16$ in Figure 10.1 suffices for Theorem 8.3 to hold
$\hat{c}$	the smallest min-cut value that we allow in G_i, $\forall i$
$\mathsf{edge\text{-}loss}_i$	sum of min-cuts removed from in G_i: $\sim:= \sum_{t=1}^{x_i} \mathsf{cap}(S_i^t, V_i^t)$
$f_0 := \frac{1}{2\lambda(n)}$	concurrent flow value in the CKS flow-linked decomposition
$f_1 = \frac{1}{a_1\beta(G)\lambda(n)}$	the minimum concurrent flow value for demands in (10.12) in order for subgraph $\check{G}_i^t$ to satisfy the flow-linked property
$\bar{f}$	an optimal fractional solution to (2.2a)–(2.2d)
$\mathsf{flow\text{-}loss}_1$	$\sim\leq \mathsf{LOSS}/a_0$, the flow lost from min-cut processing
$\mathsf{flow\text{-}loss}_2$	$\sim\leq \frac{\mathsf{LOSS}}{2a_0(1-8/a_1)}$, the flow lost from sparsest-cut processing
ℓ	the number of node-disjoint subgraphs G of $\mathcal{G}$
n	the number of nodes in $\mathcal{G}$, the input graph
$\mathsf{remaining\text{-}flow}$	remaining flow of $\bar{f}$ between terminal pairs in $\mathcal{T}_i$, $\forall i$
x_i	amount of flow in OPT^* for (s_i, t_i); see (2.2a)–(2.2d)
$\gamma(\mathcal{G})$	$\sim:= \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$ as in (2.1)
$\gamma(G_i)$	the total flow induced in G_i by the original flow $\bar{f}$ $\sim:= \sum_{p\in\mathcal{P}:p\in G_i} \bar{f}(p) = \frac{1}{2}\sum_{u\in G_i} \gamma(u, G_i)$
$\gamma(u, G_i)$	the flow induced in G_i for u by the original flow $\bar{f}$
$\rho_i(v)$	$\forall i = 1, \ldots, \ell$, weight of $v \in G$ assigned in Theorem 8.1 on G_i
$\varpi_i(v)$	$\forall i = 1, \ldots, \ell$, weight of $v \in \check{G}_i$ in (10.19) for Theorem 8.3
$\pi(v)$	weight in Theorem 3.1: $\sim= \frac{\varpi_i(v)}{2}$, $\forall v \in \check{G}_i, \forall i$; else $\sim= 0$
$\Delta(S_i)$	$\sim:= \mathsf{cap}(S_i, V_i^0 \setminus S_i)$ in G_i
C^0	the min-cut value in $\mathcal{G}$: $\sim\geq (4a_0\lambda(n) + a_0 + 2)\hat{c}$ for some $a_0 \geq 2$
$\mathcal{G}$	the original input graph with min-cut $\Omega(\kappa \log^2 n)$
$(G_i, \mathcal{T}_i)$, $\forall i$	EDP instances in disjoint subgraphs of $\mathcal{G}$ in Theorem 8.1
$G_i^0 := (V_i^0, E_i^0)$	the subgraph G_i obtained in Theorem 8.1
$G_i^t = (V_i^t, E_i^t)$	the remaining subgraph of G_i^0 after removing $S_i^1, \ldots, S_i^t$ and their adjacent edges $\forall t = 1, \ldots, x_i$; hence $V_i^t = V_i^0 \setminus \cup_{j=1,\ldots,t} S_i^j$
$\check{G}_i = (V_i^{x_i}, E_i^{x_i})$	remaining subgraph by the end of min-cut processing $\check{G}_i := G_i^{x_i}$, which we return in Theorem 3.1
$\check{G}_i^t = (\check{V}_i^t, \check{E}_i^t)$	the remaining subgraph of $\check{G}_i$ after muting nodes in $Q_i^1, \ldots, Q_i^t$ and their adjacent edges during sparsest-cut processing
LOSS	$\sim\leq \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})/2$: the number of edges that are cut during the CKS flow-decomposition process in Theorem 8.1
$Q_i^1, \ldots, Q_i^{y_i}$	the sets of regions whose demands are set to 0 in $\check{G}_i$
$S_i^1, \ldots, S_i^{x_i}$	the sets of vertices taken away from G_i^0 and in that order
$\mathcal{T}_i', i = 1, \ldots, \ell$	subsets of disjoint terminals pairs in $\mathcal{T}$ with a positive weight ϖ_i
$\check{V}_i^t$	$\sim:= \check{V}_i \setminus \cup_{j=1,\ldots,t} Q_i^j$; the remaining set of vertices in $\check{G}_i$ at stage t
Y_i^0	the set of terminals of G_i^0 right after CKS flow decomposition s.t. Y_i^0 is ρ_i-flow-linked in G_i^0
Y_i'	terminals in $\mathcal{T}_i'$ s.t. Y_i' is ϖ_i-flow-linked in $\check{G}_i$

REFERENCES

[1] M. ANDREWS, J. CHUZHOY, S. KHANNA, AND L. ZHANG, *Hardness of the undirected edge-disjoint paths problem with congestion*, in Proceedings of the 46th IEEE FOCS, IEEE, Washington, DC, 2005, pp. 226–244.

[2] M. ANDREWS AND L. ZHANG, *Hardness of the undirected congestion minimization problem*, in Proceedings of the 37th ACM STOC, ACM, New York, 2005, pp. 284–293.

[3] M. ANDREWS AND L. ZHANG, *Hardness of the undirected edge-disjoint path problem*, in Proceedings of the 37th ACM STOC, ACM, New York, 2005, pp. 276–283.

[4] M. ANDREWS AND L. ZHANG, *Logarithmic hardness of the directed congestion minimization problem*, in Proceedings of the 38th ACM STOC, ACM, New York, 2006, pp. 517–526.

[5] Y. AUMANN AND Y. RABANI, *Improved bounds for all-optical routing*, in Proceedings of the 6th ACM-SIAM SODA, ACM, New York, SIAM, Philadelphia, 1995, pp. 567–576.

[6] B. Awerbuch, R. Gawlick, F. T. Leighton, and Y. Rabani, *On-line admission control and circuit routing for high performance computing and communication*, in Proceedings of the 35th IEEE FOCS, IEEE, Washington, DC, 1994, pp. 412–423.

[7] A. Z. Broder, A. M. Frieze, and E. Upfal, *Existence and construction of edge-disjoint paths on expander graphs*, SIAM J. Comput., 23 (1994), pp. 976–989.

[8] C. Chekuri and S. Khanna, *Edge disjoint paths revisited*, in Proceedings of the 14th ACM-SIAM SODA, ACM, New York, SIAM, Philadelphia, 2003.

[9] C. Chekuri, S. Khanna, and F. B. Shepherd, *The all-or-nothing multicommodity flow problem*, in Proceedings of the 36th ACM STOC, ACM, New York, 2004, pp. 156–165.

[10] C. Chekuri, S. Khanna, and F. B. Shepherd, *Edge-disjoint paths in planar graphs*, in Proceedings of the 45th IEEE FOCS, IEEE, Washington, DC, 2004, pp. 71–80.

[11] C. Chekuri, S. Khanna, and F. B. Shepherd, *Multicommodity flow, well-linked terminals, and routing problems*, in Proceedings of the 37th ACM STOC, ACM, New York, 2005, pp. 183–192.

[12] C. Chekuri, S. Khanna, and F. B. Shepherd, *Edge-disjoint paths in planar graphs with constant congestion*, in Proceedings of the 38th ACM STOC, ACM, New York, 2006, pp. 757–766.

[13] C. Chekuri, S. Khanna, and F. B. Shepherd, *An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow*, J. Theory Comput., 2 (2006), pp. 137–146.

[14] H. Chernoff, *A measure of asymptotic efficiency of tests of a hypothesis based on the sum of observations*, Ann. Math. Statistics, 23 (1952), pp. 493–507.

[15] J. Chuzhoy, V. Guruswami, S. Khanna, and K. Talwar, *Hardness of routing with congestion in directed graphs*, in Proceedings of the 39th ACM STOC, ACM, New York, 2007, pp. 165–178.

[16] J. Chuzhoy and J. Naor, *New hardness results for congestion minimization and machine scheduling*, in Proceedings of the 36th ACM STOC, ACM, New York, 2004, pp. 28–34.

[17] N. Garg, V. V. Vazirani, and M. Yannakakis, *Primal-dual approximation algorithms for integral flow and multicut in trees*, in Proceedings of the 20th ICALP, Springer-Verlag, London, 1993, pp. 64–75.

[18] N. Garg, V. V. Vazirani, and M. Yannakakis, *Approximate max-flow min-(multi)cut theorems and their applications*, SIAM J. Comput., 25 (1996), pp. 235–251.

[19] V. Guruswami, S. Khanna, R. Rajaraman, F. B. Shepherd, and M. Yannakakis, *Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems*, in Proceedings of the 31th ACM STOC, ACM, New York, 1999, pp. 19–28.

[20] D. R. Karger, *Random sampling in cut, flow, and network design problems*, in Proceedings of the 26th ACM STOC, ACM, New York, 1994, pp. 648–657.

[21] D. R. Karger *Random sampling in cut, flow, and network design problems*, Math. Oper. Res., 24 (1999), pp. 383–413.

[22] R. Khandekar, S. Rao, and U. Vazirani, *Graph partitioning using single commodity flows*, in Proceedings of the 38th ACM STOC, ACM, New York, 2006, pp. 385–390.

[23] J. Kleinberg, *An approximation algorithm for the disjoint paths problem in even-degree planar graphs*, in Proceedings of the 46th IEEE FOCS, IEEE, Washington, DC, 2005, pp. 627–636.

[24] J. Kleinberg and R. Rubinfeld, *Short paths in expander graphs*, in Proceedings of the 37th IEEE FOCS, IEEE, Washington, DC, 1996, p. 86.

[25] J. Kleinberg and E. Tardos, *Approximations for the disjoint paths problem in high-diameter planar networks*, in Proceedings of the 27th ACM STOC, ACM, New York, 1995, pp. 26–35.

[26] J. Kleinberg and E. Tardos, *Disjoint paths in densely embedded graphs*, in Proceedings of the 36th IEEE FOCS, IEEE, Washington, DC, 1995, pp. 52–61.

[27] J. M. Kleinberg, *Approximation Algorithms for Disjoint Paths Problems*, Ph.D. thesis, MIT, Cambridge, MA, 1996.

[28] S. G. Kolliopoulos and C. Stein, *Approximating disjoint-path problems using greedy algorithms and packing integer programs*, in Proceedings of IPCO, Springer-Verlag, London, 1998, pp. 153–168.

[29] P. Kolman and C. Scheideler, *Simple on-line algorithms for the maximum disjoint paths problem*, in Proceedings of the 13th ACM SPAA, ACM, New York, 2001, pp. 38–47.

[30] K. Obata, *Approximate max-integral-flow/min-multicut theorems*, in Proceedings of the 36th ACM STOC, ACM, New York, 2004, pp. 539–545.

[31] L. Orecchia, L. Schulman, U. Vazirani, and N. Vishnoi, *On partitioning graphs via single commodity flows*, in Proceedings of the 40th ACM STOC, ACM, New York, 2008, pp. 461–470.

[32] P. Raghavan and C. D. Thompson, *Randomized roundings: A technique for provably good algorithms and algorithms proofs*, Combinatorica, 7 (1987), pp. 365–374.

[33] N. Robertson and P. D. Seymour, *An outline of a disjoint paths algorithm*, in Paths, Flows and VLSI-layout, Algorithms Combin. 9, Springer-Verlag, Berlin, 1990, pp. 267–292.
[34] A. Srinivasan, *Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems*, in Proceedings of the 38th IEEE FOCS, IEEE, Washington, DC, 1997, p. 416.
[35] K. Varadarajan and G. Venkataraman, *Graph decomposition and a greedy algorithm for edge-disjoint paths*, in Proceedings of the ACM-SIAM SODA, ACM, New York, SIAM, Philadelphia, 2004, pp. 379–380.