Polynomial Flow-Cut Gaps and Hardness of Directed Cut Problems

[Extended Abstract]

Julia Chuzhoy*
School of Mathematics
Institute for Advanced Study
Princeton, NJ 08540
cjulia@math.ias.edu

Sanjeev Khanna†
Dept. of Computer & Information Science
University of Pennsylvania
Philadelphia, PA 19104
sanjeev@cis.upenn.edu

ABSTRACT

We study the multicut and the sparsest cut problems in directed graphs. In the multicut problem, we are a given an n-vertex graph G along with k source-sink pairs, and the goal is to find the minimum cardinality subset of edges whose removal separates all source-sink pairs. The sparsest cut problem has the same input, but the goal is to find a subset of edges to delete so as to minimize the ratio of deleted edges to the number of source-sink pairs that are separated by this deletion. The natural linear programming relaxation for multicut corresponds, by LP-duality, to the well-studied maximum (fractional) multicommodity flow problem, while the natural LP-relaxation for sparsest cut corresponds to maximum concurrent flow. Therefore, the integrality gap of the linear programming relaxation for multicut/sparsest cut is also the *flow-cut gap*: the maximum ratio, achievable for any graph, between the maximum flow value and the minimum cost solution for the corresponding cut problem. Starting with the celebrated max flow-min cut theorem of Ford and Fulkerson, flow-cut gaps have played a central role in combinatorial optimization. For many NP-hard network optimization problems, the best known approximation guarantee corresponds to our understanding of the appropriate flow-cut gap.

Our first result is that the flow-cut gap between maximum multicommodity flow and minimum multicut is $\tilde{\Omega}(n^{1/7})$ in directed graphs. We show a similar result for the gap between maximum concurrent flow and sparsest cut in directed graphs. These results improve upon a long-standing lower bound of $\Omega(\log n)$ for both types of flow-cut gaps. We notice that these polynomially large flow-cut gaps are in a sharp contrast to the undirected setting where both these flow-cut gaps are known to be $\Theta(\log n)$. Our second result is that both directed multicut and sparsest cut are hard to approximate to within a factor of $2^{\Omega(\log^{1-\epsilon} n)}$ for any constant $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{ZPP}$. This improves upon the recent $\Omega(\log n / \log\log n)$-hardness result for these problems. We also show that existence of PCP's for NP with perfect completeness, polynomially small soundness, and constant number of queries would imply a polynomial factor hardness of approximation for both these problems. All our results hold for directed acyclic graphs.

*Supported by a grant of the state of New Jersey to the Institute for Advanced Study.

†Supported in part by an NSF Career Award CCR-0093117 and by NSF Award CCF-0635084.

STOC'07, June 11–13, 2007, San Diego, California, USA.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Algorithms, Theory.

Keywords

Concurrent Flow, Directed Multicut, Directed Sparsest cut, Flow-Cut Gaps, Hardness of Approximation, Multicommodity Flow.

1. INTRODUCTION

Cuts are fundamental combinatorial objects that play an important role in the study of embeddings, graph theory, Markov chains, parallel computation and optimization. Some representative applications of cuts include the use of bisection width and flux in establishing bounds for parallel algorithms, use of conductance in establishing rapid-mixing property of Markov chains, balanced separators for divide-and-conquer algorithms, and sparsest cuts in network design and routing. Essentially all cut problems that arise in above application scenarios are NP-hard. One of the most successful approaches to designing approximation algorithms for cut problems relies on the rounding of natural linear-programming relaxations for these problems. The dual linear programs of these relaxations formulate flow problems in networks. By strong duality, the largest gap between the maximum flow and the minimum cut achievable for any problem instance (also called the flow-cut gap) is exactly the integrality gap of these cut relaxations. If one views a cut computation as revealing some inherent bottleneck to communication in a network, then the flow-cut gap may be viewed as revealing how closely this bottleneck can be approached. Starting from the celebrated max flow-min cut theorem of Ford and Fulkerson, flow-cut gaps have played a central role in combinatorial optimization. For many optimization problems, the best known approximation guarantees correspond to upper bounds on flow-cut gaps. Some examples include approximation guarantees for undirected and directed multicut problems, sparsest cut in planar graphs, well-linked decompositions, and the performance ratio for oblivious routing.

In this paper, we make progress on some long-standing questions concerning flow-cut gaps in directed graphs and approximability of directed cut problems. We start by describing the cut problems that we study and with a brief review of prior work for them.

Directed Multicut: An instance of the *directed multicut* problem consists of a directed n-vertex graph $G(V, E)$ and a collection of k source-sink pairs $\{(s_1, t_1), \ldots, (s_k, t_k)\}$. The goal is to remove the smallest possible number of edges so as to separate all source-sink pairs; a pair (s_i, t_i) is considered separated iff in the resulting graph there is no path connecting s_i to t_i. The parameter k is also referred to as the *number of commodities* in the instance. Vertices in the set $T = \{s_1, t_1, s_2, t_2, ..., s_k, t_k\}$ are called *terminals*, and all the other vertices are non-terminals.

For the single-commodity case, the celebrated max-flow min-cut theorem [20] shows that the size of minimum (s_1, t_1) cut equals the maximum flow from s_1 to t_1. The tight duality between cuts and flows breaks down even in undirected graphs for $k \geq 3$. However, the (worst-case) gap between maximum multicommodity flow and the minimum multicut is well-understood for undirected graphs and is known to be $\Theta(\log k)$ [28, 22]. In a sharp contrast, Saks *et al.* [30] have shown that the flow-cut gap in directed graphs can be as large as $k - \epsilon$ for any $\epsilon > 0$ [30]. Since it is easy to see that the flow-cut gap cannot exceed k, it may seem that the flow-cut gaps are well-understood in the directed case as well. However, the size of the Saks *et al.* construction grows super-exponentially in k, and the gap realized by these instances is only $O(\log n/\log\log n)$, where n is the number of vertices in G. As a function of n, the strongest gap known is $\Omega(\log n)$, and it is achieved by an expander-based construction [28]. This lack of understanding of the directed flow-cut gaps is reflected as well in a large separation between the upper and the lower bounds on the approximability threshold of directed multicut. Cheriyan, Karloff and Rabani [11] gave an $O(\sqrt{n \log n})$-approximation algorithm for directed multicut, and Gupta [24] subsequently improved it to an $O(\sqrt{n})$-approximation. In a very recent work, Agarwal, Alon, and Charikar [1], have further improved the approximation ratio to $\tilde{O}(n^{11/23})$. Since all these algorithms are based on rounding of the natural LP relaxation for multicut, they also give matching upper bounds on the gap between maximum multicommodity flow and the minimum multicut in directed graphs. On the hardness front, recently, Chuzhoy and Khanna [12] established an $\Omega(\log n/\log\log n)$-hardness for directed multicut, assuming that NP is not contained in $\mathsf{DTIME}\left(n^{\mathrm{polylog(n)}}\right)$. While these algorithmic and hardness results represent important steps in closing the gaps in our understanding of the directed multicut problem, they leave open the possibility that the approximability threshold may range anywhere from logarithmic to polynomial function of the input size. We note that the approximability threshold remains unresolved for undirected multicut problem as well, and it currently lies somewhere between APX-hardness [15] and $\Theta(\log k)$ [28, 22]. However, if one assumes the Unique Games Conjecture of Khot [26], undirected multicut problem can be shown to be hard to approximate to within any constant factor [10, 27].

Directed Sparsest Cut: The input to the *directed sparsest cut* problem is the same as for multicut, but the objective now is to find a subset E' of edges so as to minimize the ratio $|E'|/|S_{E'}|$ where $S_{E'}$ is the set of source-sink pairs, which are disconnected in the graph $G(V, E \setminus E')$. In general, the notion of a sparsest cut in a graph can be defined in two distinct ways. In one version of the problem, which we refer to as the *bipartite sparsest cut*, the sparsest cut in a graph is a bipartition of vertices into two sets S and $\bar{S}$ that minimizes the ratio of $|\delta(S, \bar{S})|$[1] to $|\{(s_i, t_i) \mid s_i \in S,\ t_i \in \bar{S}\}|$. In the second version, which we refer to as the *non-bipartite sparsest cut* or simply as the *sparsest cut*, we seek to minimize the ratio of the number of edges deleted to the resulting number of pairs separated. We note here that the dual of concurrent flow problem corresponds to a relaxation for the non-bipartite sparsest cut problem. In undirected graphs, it is easy to see that the two notions are equivalent. However, in directed graphs, as highlighted in the very recent work of Charikar *et al.* [9], these versions seem to behave quite differently. In particular, using a result of Feige and Kogan [19], it is shown in [9] that bipartite sparsest cut is hard to approximate to within $2^{\Omega((\log n)^{\delta})}$ for some $\delta > 0$ unless 3SAT has subexponential-time algorithms. Furthermore, this hardness can be strengthened to an n^{δ}-hardness for some $\delta > 0$ assuming a hypothesis concerning hardness of random 3SAT, as described by Feige [18]. In contrast, for the directed non-bipartite sparsest cut, so far only an $\Omega(\log n/\log\log n)$-hardness is known, due to Chuzhoy and Khanna [12]. On the positive side, Hajiaghayi and Räcke [25] gave an $O(\sqrt{n})$-approximation for directed non-bipartite sparsest cut. The recent work of Agarwal *et al.* [1] improves this to an $\tilde{O}(n^{11/23})$-approximation, and as before, this is also the best known upper bound on the gap between concurrent flow and sparsest cut. Thus, at present, the known upper bounds on the approximability as well as flow-cut gaps of both directed multicut and directed non-bipartite sparsest cut are similar.

Our Results and Techniques: Our first main result establishes a polynomial lower bound on directed flow-cut gaps.

THEOREM 1.1. *The flow-cut gap between maximum multicommodity flow and directed multicut is $\tilde{\Omega}(n^{1/7})$. The flow-cut gap between maximum concurrent flow and directed (non-bipartite) sparsest cut is also $\tilde{\Omega}(n^{1/7})$. Both results hold even on directed acyclic graphs.*

We now give an overview of our techniques for the flow-

[1] $\delta(S, \bar{S})$ refers to all edges (x, y) in G where $x \in S$ and $y \in \bar{S}$.

cut gap results above. We will focus on the gap between maximum multicommodity flow and directed multicut. For clarity of exposition, we work with the vertex version of the problem, where the goal is to remove the minimum-cardinality subset of non-terminal vertices that disconnects all the source-sink pairs. A standard transformation allows us to translate a flow-cut gap result for the vertex version to the edge version.

Our starting point is an instance H of the multicut problem that is formed by a union of k graphs $H_1, ..., H_k$ defined over the same set of vertices. Instance H_i corresponds to the source-sink pair s_i-t_i and it is a layered graph with L layers for some parameter L. We say that path P connecting a source s_i to its sink t_i is *canonical* if it is entirely contained in H_i. The main property of instance H is that if we are required to eliminate only the canonical paths, then fractional solution that assigns $1/L$ to each vertex is a feasible solution of cost n/L, while the cost of any integral solution must be at least $\Omega(n)$. The fractional solution above is feasible for the restricted problem because the canonical paths in instance H are long (length at least L). In order to convert this to a true integrality gap result, we need to rule out "short" non-canonical paths. Towards this end, we define another instance $\mathcal{L}$ of directed multicut, which can also be viewed as a union of k graphs $\mathcal{L}_1, \dots, \mathcal{L}_k$. Each $\mathcal{L}_i$ is a graph with $O(L)$ layers that contains many source-sink pairs. A path connecting a source-sink pair in instance $\mathcal{L}$ is called canonical if it uses only edges from $\mathcal{L}_i$. The main property of graph $\mathcal{L}$ is that while the canonical paths share many vertices, no non-canonical paths exist in the graph. We will refer to graph $\mathcal{L}$ as the *labeling scheme*. The basic idea of using a labeling scheme to ensure that only canonical paths exist between source-sink pairs was first used by Andrews and Zhang [3] to show hardness of directed congestion minimization. The dependence of the size of the labeling scheme on the parameters k and L is crucial to determining the final gap or hardness result. The scheme in [3] gives a graph $\mathcal{L}$ of size $L^{O(\log k)}$, which is insufficient to obtain polynomial gaps. We present a simple new labeling scheme that results in a graph of size $\text{poly}(k, L)$. We note that we use a similar labeling scheme in our parallel result on the hardness of directed routing with congestion [13]. The same labeling scheme has also been used independently by [21] for establishing the hardness of directed routing problems with congestion. A merged version of the results from [13] and [21] appears in [14].

It is worth highlighting an important point of departure from the usage of labeling schemes in context of disjoint path problems. A gap result of $\Omega(f(n))$ for directed multicut necessarily requires that the total number of paths connecting source-sink pairs be exponential in $f(n)$. Otherwise, it is easy to see that the flow-cut gap cannot exceed $o(f(n))$. This issue does not arise in the setting of routing problems. Consequently, we modify our basic labeling scheme to ensure that it does not permit any small integral solution. This transformation preserves the property that no non-canonical source-sink paths exist but introduces short canonical paths. The final step is to appropriately compose together graphs H and $\mathcal{L}$ to create a new instance where all canonical paths are long, no non-canonical paths exist and integral solutions have high cost. The resulting instance gives us the desired flow-cut gap.

Our second result shows that the flow-cut gap results above can be extended to almost-polynomial hardness of approximation results for directed multicut and sparsest cut.

THEOREM 1.2. *The directed multicut problem and the directed (non-bipartite) sparsest cut problem are $2^{\Omega(\log^{1-\epsilon} n)}$-hard to approximate for any constant $\epsilon > 0$, even on directed acyclic graphs, unless* $\mathsf{NP} \subseteq \mathsf{ZPP}$.

One way to show the hardness result above is by replacing the graph H in the flow-cut gap construction with an appropriate graph encoding the constraints from a Raz verifier. There is a natural way of doing such an encoding by strongly using the projection property of the solutions to the Raz verifier constraints. However, this approach cannot give a polynomial hardness since the size of the constraint satisfaction system associated with Raz verifier grows fast as the soundness decreases, and thus soundness which is polynomially small in the system size cannot be achieved.

In order to go to polynomial-hardness, we need to allow encoding of general constraint systems which might not have any analog of the projection property. Using several additional ideas we show that we can create desired encodings even when the underlying constraints lack the projection property. An important consequence of our construction is the following theorem.

THEOREM 1.3. *If* NP *has probabilistically checkable proof systems with constant number of queries, proof table entries defined over a field F of polynomial size, logarithmic number of random bits, perfect completeness and polynomially small soundness, then directed multicut and directed sparsest cut are $n^{\Omega(1)}$-hard to approximate.*

Existence of PCP's for NP with above properties was first conjectured by Bellare *et al.* [8]. While the conjecture remains unproven yet, a sequence of papers have made progress towards proving this conjecture. In particular, Dinur *et al.* [16] have shown that for any $\epsilon > 0$, NP has a polynomial-size PCP that queries $O(1)$ variables ranging over a field F with $|F| = 2^{\Theta(\log^{1-\epsilon} n)}$, has perfect completeness, and achieves soundness $O(1/|F|)$. Proving the Bellare et. al. conjecture requires pushing this result to $|F| = 2^{\Theta(\log n)}$, maintaining perfect completeness, and achieving soundness $O(1/|F|^{\beta})$ for some $\beta > 0$.

Organization: Due to space limitations, we will primarily focus on proving Theorem 1.1 in the main body of the paper. We start with some preliminaries in Section 2, and present our flow-cut gap construction in Section 3. Finally, we describe in Section 4 the main difficulty as well the key additional ideas needed in translating the flow-cut gap results to hardness results. We give an overview of the construction used in establishing Theorems 1.2 and 1.3. A full version with complete technical details appears on the webpages of the authors.

2. PRELIMINARIES

Linear programming formulations: We start by defining a natural LP relaxation for directed multicut. For each edge $e \in E$, there is an indicator variable x_e that represents whether or not e is in the solution. For each source-sink pair (s_i, t_i), let $\mathcal{P}_i$ be the set of all the paths connecting s_i to t_i. The multicut LP-relaxation and its dual are as follows:

(LP1-P)

$$\begin{array}{lll} & \min \ \sum_{e \in E} x_e & \\ \text{s.t.} & & \\ & \sum_{e \in p} x_e \geq 1 & \forall i : 1 \leq i \leq k, \forall p \in \mathcal{P}_i \\ & x_e \geq 0 & \forall e \in E \end{array}$$

(LP1-D)

$$\begin{array}{lll} & \max \ \sum_{i=1}^{k} \sum_{p \in \mathcal{P}_i} f_p & \\ \text{s.t.} & & \\ & \sum_{p : e \in p} f_p \leq 1 & \forall e \in E \\ & f_p \geq 0 & \forall i : 1 \leq i \leq k, \forall p \in \mathcal{P}_i \end{array}$$

Notice that (LP1-D) is equivalent to the maximum multicommodity flow problem, where the goal is to maximize the total flow routed between the source-sink pairs, while the flow routed via any edge cannot exceed 1. From LP-duality, the optimal costs of both linear programs are equal, and thus the integrality gap of (LP1-P) is also the flow-cut gap between multicommodity flow and minimum multicut.

We can similarly define an LP-relaxation for sparsest cut. We use the same notation as for the directed multicut LP-formulation. Consider any solution to the sparsest cut problem. For each edge $e \in E$, let x_e denote whether edge e is in the solution, and for each $i : 1 \leq i \leq k$, let h_i denote whether the source-sink pair (s_i, t_i) is disconnected. Let $D = \sum_{i=1^k} h_i$ be the total number of source-sink pairs disconnected by the solution. We now define, for each edge e, $x'_e = x_e / D$, and for each source-sink pair (s_i, t_i), $h'_i = h_i / D$. It is then easy to see that we have defined a feasible solution to the linear program (LP2-P) that appears below, along with its dual (LP2-D). Moreover, the sparsity of the cut equals to the value of the objective function of (LP2-P) on the above solution. Thus, (LP2-P) is a relaxation of the directed sparsest cut problem.

(LP2-P)

$$\begin{array}{lll} & \min \ \sum_{e \in E} x'_e & \\ \text{s.t.} & & \\ & \sum_{e \in p} x'_e \geq h'_i & \forall i : 1 \leq i \leq k, \forall p \in \mathcal{P}_i \\ & \sum_{i=1}^{k} h_i \geq 1 & \\ & x_e, h'_i \geq 0 & \forall e \in E, \forall i : 1 \leq i \leq k \end{array}$$

(LP2-D)

$$\begin{array}{lll} & \max \ \lambda & \\ \text{s.t.} & & \\ & \sum_{p \in \mathcal{P}_i} f(p) \geq \lambda & \forall i : 1 \leq i \leq k \\ & \sum_{p : e \in p} f(p) \leq 1 & \forall e \\ & f(p) \geq 0 & \forall i : 1 \leq i \leq k, \forall p \in \mathcal{P}_i \end{array}$$

Notice that (LP2-D) is equivalent to the maximum concurrent flow problem, in which we need to maximize a value λ, such that λ units of flow can be routed simultaneously for each source-sink pair, while the flow on any edge does not exceed 1. >From LP-duality, the maximum concurrent flow in any graph is equal to the minimum fractional sparsest cut. Therefore, the integrality gap of (LP2-P) is also the flow-cut gap between maximum concurrent flow and minimum sparsest cut.

Vertex version of multicut For the sake of convenience, we consider the vertex version of the directed multicut problem. The input for this problem is denoted by $G = (V, \mathcal{M}, E)$, where V is the set of non-terminal vertices, $\mathcal{M}$ is the set of the source-sink pairs and E is the set of edges. Let $T(\mathcal{M})$ denote the set of all the terminals, $T(\mathcal{M}) = \{s, t \mid (s, t) \in \mathcal{M}\}$. The goal is to remove the minimum cardinality subset of non-terminal vertices so as to disconnect all the source-sink pairs. We prove the lower bounds for flow-cut gap and hardness of approximation for the vertex version of the directed multicut. In order to obtain the same result for the multicut problem itself, we use a standard procedure to convert a vertex version instance G to the instance of directed multicut, as follows. Each non-terminal vertex $v \in V$ is replaced by a *special* directed edge $(v^+ \rightarrow v^-)$. Each edge $e = (u \rightarrow v)$ in the original graph is replaced by an edge $(u^- \rightarrow v^+)$. Let G' denote the new instance. We can assume w.l.o.g., that any integral solution of G' only contains special edges, and thus the integral solution cost for both instances are the same. Moreover, if there is a subset of non-terminal vertices in G whose removal disconnects an α-fraction of source-sink pairs, then there is also a subset of edges in G' of the same size whose removal disconnects the same fraction of source-sink pairs, and vice versa. The linear programming formulation for the vertex version is as follows:

(LP3)

$$\begin{array}{lll} & \min \ \sum_{v \in V} x_v & \\ \text{s.t.} & & \\ & \sum_{v \in p \cap V} x_v \geq 1 & \forall i : 1 \leq i \leq k, \forall p \in \mathcal{P}_i \\ & x_v \geq 0 & \forall v \in V \end{array}$$

Given a fractional solution to (LP3) on graph G, there is a fractional solution of the same value to (LP1-P) on G': each special edge $e = (v^+ \rightarrow v^-)$ is assigned $x_e = x_v$. It follows that the integrality gap and hardness results for the vertex version carry over to our original problems. This is also true for the bi-criteria setting, where only a constant fraction of source-sink pairs needs to be disconnected.

3. INTEGRALITY GAP

Let n be some parameter. We construct a multicut instance G where all the parameters are defined in terms of n, and the instance size (the number of vertices) is $N = O(n^7 / (\log n)^3)$. We show that the integrality gap of G is $\Omega\left(\frac{n}{\log n}\right) = \Omega\left(\frac{N^{1/7}}{\log^{4/7} N}\right)$.

Let $L = n / (4 \log n)$. Our goal is to construct a multicut instance G that has the following properties:

($\mathbf{C}_1$): Any path connecting any source-sink pair contains at least L non-terminal vertices.

($\mathbf{C}_2$): There exists a constant $\epsilon > 0$ such that any integral solution that disconnects a $(1-\epsilon)$-fraction of the pairs contains at least $\Omega(N)$ vertices.

It is easy to see that if G has the above properties, then the integrality gap is at least $\Omega(L) = \Omega\left(\frac{N^{1/7}}{\log^{4/7} N}\right)$, since a feasible fractional solution to (LP3) of cost $O(N/L)$ can be obtained by assigning $1/L$-fraction to each non-terminal vertex.

Graph G is constructed in three steps. In the first step, we construct an initial multicut instance H, and we define, for each source-sink pair, a collection of *canonical paths*. Graph H has property (C_2), while property (C_1) holds for canonical paths, i.e., each canonical path contains at least L non-terminal vertices. However, there are also non-canonical paths connecting source-sink pairs in H, which might contain few non-terminal vertices. Thus, property (C_1) does not hold in general. The goal of the next two steps is to eliminate the non-canonical paths, while preserving the other properties of H. To achieve this, in the second step we construct a graph $\mathcal{L}$, called the *labeling scheme*. Graph $\mathcal{L}$ also has a collection of source-sink pairs, and for each source-sink pair a collection of canonical paths connecting the source to the sink is defined. The main property of graph $\mathcal{L}$ is that while the canonical paths share many vertices, no non-canonical paths exist in $\mathcal{L}$. In the third step, we compose the graphs H and $\mathcal{L}$ together to obtain the final graph G, for which both properties (C_1) and (C_2) are true.

3.1 Step 1: Constructing Graph H

In this section we construct our initial graph $H = (V, \mathcal{M}, E)$. Let $V = \{1, \ldots, n\}$ be the set of non-terminal vertices of H, and let $k = n$. For each $i : 1 \le i \le k$, we have a source-sink pair s_i-t_i. Thus, $\mathcal{M} = \{(s_i, t_i) \mid 1 \le i \le k\}$. In order to define set E of edges, we construct, for each source-sink pair s_i-t_i, an auxiliary graph H_i, which is defined over the same set of vertices as H. Graph H can then be viewed as the union of graphs H_i, $1 \le i \le k$. Thus, if we denote by E_i the set of edges of graph H_i, then $E = \cup_{i=1}^{k} E_i$. We call the edges belonging to set E_i *edges of type* i.

The Graph H_i: Fix some $i : 1 \le i \le k$. Graph H_i contains a single source-sink pair (s_i, t_i). The non-terminal vertices of H_i are a subset of V and they are arranged into $L = n/(4\log n)$ layers containing $\delta = \log n$ vertices each. The layers are denoted $S_1(i), S_2(i), \ldots, S_L(i) \subseteq V$ and they are constructed one after another, starting from the first layer. In order to construct the jth layer, for $1 \le j \le L$, we select uniformly at random δ distinct vertices from set $V \setminus (S_1(i) \cup \cdots \cup S_{j-1}(i))$. Notice that since $\delta = \log n$, $L = n/(4\log n)$ and $|V| = n$, it is possible to construct these layers, and in total $|\bigcup_{j=1}^{L} S_j(i)| = n/4$. Edges E_i of graph H_i are defined as follows. There is an edge from s_i to every vertex in the first layer, $S_1(i)$. For every pair of consecutive layers $S_j(i)$, $S_{j+1}(i)$, where $j : 1 \le j < L$, there is an edge from every vertex in $S_j(i)$ to every vertex in $S_{j+1}(i)$. Finally, for every vertex in the last layer $S_L(i)$, there is an edge connecting this vertex to t_i. This concludes the definition of graph H_i. Recall that the final set of edges of graph H is $E = \cup_{i=1}^{k} E_i$, where the edges in set E_i are called edges of type i.

Properties of graph H: A path connecting a source s_i to sink t_i (for $1 \le i \le k$) is called a *canonical path* if it only contains edges of type i.

OBSERVATION 1. *The number of non-terminal vertices on any canonical source-sink path in graph H is at least L.*

The observation follows from the fact that any canonical path connecting s_i to t_i in H also exists in graph H_i, and after leaving source s_i it has to traverse all the layers of graph H_i before reaching sink t_i. Therefore, property (C_1) holds for canonical paths. However, there might also be non-canonical paths connecting s_i to t_i whose length can be short and for which (C_1) is not true.

Next we establish property (C_2) for graph H. We actually establish a stronger version of this property that we need for future analysis. Let $\mathcal{S} \subseteq V$ be any subset of non-terminal vertices, $|\mathcal{S}| \le n/16$. For each $i : 1 \le i \le k$, we say that i is *covered* by $\mathcal{S}$, iff the removal of vertices of $\mathcal{S}$ from graph H_i disconnects s_i from t_i in this graph. Equivalently, $\mathcal{S}$ covers i iff there exists a layer $j : 1 \le j \le L$ with $S_j(i) \subseteq \mathcal{S}$. Let $\mathcal{B}$ be the following bad event: there is a set $\mathcal{S} \subseteq V$ of non-terminal vertices, $|\mathcal{S}| \le n/16$, such that $\mathcal{S}$ covers more than half the indices $i : 1 \le i \le k$. The next lemma bounds the probability of event $\mathcal{B}$.

LEMMA 3.1. *The probability of event $\mathcal{B}$ is at most* 2^{-n}

PROOF. Let $\mathcal{S} \subseteq V$ be any subset of non-terminal vertices, $|\mathcal{S}| \le n/16$. Fix some $i : 1 \le i \le k$ and consider the random choices made when layers $S_1(i), \ldots, S_L(i)$ of graph H_i are constructed. Even though the choices are not independent, when vertices of subset $S_j(i)$ are chosen, the size of the set $V \setminus \bigcup_{\ell=1}^{j-1} S_\ell(i)$ is at least $\frac{3n}{4} + \delta$. Therefore, no matter what vertices have been chosen by $S_1(i), \ldots, S_{j-1}(i)$, the probability that $S_j(i) \subseteq \mathcal{S}$ is at most $\left(\frac{n/16}{3n/4}\right)^{\delta} \le \left(\frac{1}{8}\right)^{\log n} \le \frac{1}{n^3}$ (we use conditional probabilities). Therefore, using the union bound, the probability that i is covered by $\mathcal{S}$ is at most $\frac{L}{n^3} \le \frac{1}{4n^2 \log n}$. Since the random choices made for different graphs H_i, $1 \le i \le k$ are completely independent, the probability that half of these indices are covered is at most:

$$\binom{n}{n/2}\left(\frac{1}{4n^2 \log n}\right)^{n/2} \le 2^{-n\log n/4}$$

The number of possible choices of subset $\mathcal{S}$ is at most 2^n, and applying the union bound for all such subsets finishes the proof. $\square$

Let $\mathcal{S} \subseteq V$ be any subset of non-terminal vertices. Observe that if $\mathcal{S}$ disconnects a source-sink pair s_i-t_i in graph H, then $\mathcal{S}$ also covers i. Therefore, from Lemma 3.1, any solution of size up to $n/16$ disconnects at most half the source-sink pairs in H, with high probability. From now on we assume that $\mathcal{B}$ does not happen.

As we have shown, property (C_2) holds in graph H. As for property (C_1), we are only guaranteed that it holds for the canonical paths. It is therefore possible that for many source-sink pairs short non-canonical paths exist. The goal of the next steps is to resolve this problem while preserving the other properties of graph H.

3.2 Step 2: Handling Non-Canonical Paths

In this section we build a graph $\mathcal{L}$ that represents a labeling scheme. In the final step we combine graphs H and $\mathcal{L}$ together to obtain the final graph G. We notice that we use the labeling scheme and its associated graph in the hardness of approximation construction as well. We start with the definition of a labeling scheme.

A *labeling scheme* with parameters τ and Z, is denoted by $\mathcal{L} = \mathcal{L}_{\tau,Z} = (U, \mathcal{M}', E')$, and it is defined as follows. There is a set $Y = \{1, \ldots, |Y|\}$ of *labels* associated with the labeling scheme. The non-terminal vertices are partitioned into Z layers, where each layer contains one vertex representing each label $y \in Y$. Thus, the set of the non-terminal vertices is: $U = \{u(y, h) \mid y \in Y, 1 \le h \le Z\}$. There are τ different types of source-sink pairs. For each $i : 1 \le i \le \tau$, there are $|Y|$ sources of type i: $\{s_i(y)\}_{y\in Y}$, and $|Y|$ sinks of type i: $\{t_i(y)\}_{y\in Y}$. We describe below how these sources and sinks are paired with each other. We now proceed to describe the edges of $\mathcal{L}$. Each edge either connects a source to a vertex in the first layer, or connects a vertex in the last layer to a sink, or connects a non-terminal vertex in layer j to a non-terminal vertex in layer $j + 1$, for $1 \le j < Z$.

For each $i : 1 \le i \le \tau$, there is a set E'_i of edges of type i, which are defined as follows. We define a permutation $\pi_i : Y \to Y$. For each type-i source $s_i(y)$, we add an edge from $s_i(y)$ to the first-layer vertex $u(\pi_i(y), 1)$. For each non-terminal vertex $u(y, h)$ where $1 \le h < Z$, we add an edge from $u(y, h)$ to $u(\pi_i(y), h+1)$. Finally, for each layer-Z non-terminal vertex $u(y, Z)$, we add an edge from $u(y, Z)$ to sink $t_i(\pi_i(y))$. Therefore, edges of type i form a perfect matching between each pair of consecutive layers, between the type-i sources and the vertices of the first layer, and between the vertices of the last layer and the type-i sinks. Thus, if we start at some type-i source $s_i(y)$ and follow type-i edges, then we will construct a path, denoted by $P_i(y)$, that ends at some type-i sink $t_i(y')$. Moreover, the paths $\{P_i(y)\}_{y\in Y}$ are vertex disjoint. The pairs of endpoints of these paths define the source-sink pairs for the labeling scheme. Thus, we have $|Y|$ source-sink pairs of type i, and each type-i source and sink is involved in exactly one such pair. The paths $P_i(y)$ are called the canonical paths for the corresponding source-sink pairs. Notice that for each source-sink pair there is a unique canonical path connecting it.

A labeling scheme $\mathcal{L}$ is called *valid* iff for every source-sink pair, the canonical path is the only path connecting the source to the sink. In other words, no non-canonical paths connecting source-sink pairs exist. We note that the construction of [3] can be viewed as a labeling scheme, where $|Y| = Z^{O(\log \tau)}$. This construction is insufficient to get a polynomial integrality gap. Below we define a more efficient labeling scheme, where $|Y| = \text{poly}(\tau, Z)$.

In order to define the labeling scheme, we now only need to specify the value of parameter $|Y|$ and to define the permutations π_i, for $1 \le i \le k$. For each $i : 1 \le i \le \tau$, we define an *increment vector* $\mu_i \in \mathbb{Z}^2$, $\mu_i = (i, i^2)$. We view the set of labels Y in the following fashion: each label $y \in Y$ is viewed as a 2-dimensional vector, whose first entry ranges in $[2\tau Z]$ and the second entry ranges over $[2\tau^2 Z]$. Thus, $Y = [2\tau Z] \times [2\tau^2 Z]$, and $|Y| = O(\tau^3 Z^2)$. For each label $y \in Y$, we denote by y_1 and y_2 its first and second coordinate respectively. We define an addition operation between pairs of labels (since all increment vectors $\mu_i \in Y$, this also defines addition of increment vectors and labels). For $y, y' \in Y$, we say that $y \oplus y' = y''$, iff $y''_1 = y_1 + y'_1 \bmod (2\tau Z)$ and $y''_2 = y_2 + y'_2 \bmod (2\tau^2 Z)$. Finally, we define the permutation $\pi_i : Y \to Y$ for each $i : 1 \le i \le k$, as follows: for each $y \in Y$, $\pi_i(y) = y \oplus \mu_i$. This completes the definition of the labeling scheme. It now only remains to show that the above labeling scheme is valid, i.e., no non-canonical source-sink paths exist. Notice that for each type $i : 1 \le i \le \tau$, $s_i(y)$-$t_i(y')$ are a source-sink pair iff $y' = y \oplus ((Z+1)\mu_i)$. Thus, the set of source-sink pairs is: $\mathcal{M}' = \{(s_i(y), t_i(y')) \mid 1 \le i \le \tau, y \in Y, y' = y \oplus ((Z+1)\mu_i)\}$.

LEMMA 3.2. *Let $(s_i(y), t_i(y')) \in \mathcal{M}'$ be any source-sink pair of $\mathcal{L}$. Then the only path connecting $s_i(y)$ to $t_i(y)$ in the graph is the canonical path $P_i(y)$.*

PROOF. Assume otherwise, and let P be a non-canonical path connecting $s_i(y)$ to $t_i(y')$. Recall that $y' = y \oplus ((Z+1) \cdot \mu_i)$. Let $j_1, \ldots, j_{Z+1}$ be the types of edges used along path P. Since path P is non-canonical, at least one of the types $j_q \ne i$, where $1 \le q \le Z+1$. As path P must reach the sink $t_i(y')$, it must be the case that $y' = y \oplus \mu_{j_1} \oplus \cdots \oplus \mu_{j_{Z+1}}$. Recall that the first coordinate in each increment vector lies between 1 and τ, while the second coordinate lies between 1 and τ^2. Since the addition of the first coordinate is performed modulo $2\tau Z$ and the addition of the second coordinate is performed modulo $2\tau^2 Z$, we have that $(Z + 1)\mu_i = \sum_{q=1}^{Z+1} \mu_{j_q}$ (here we use standard addition). Therefore, μ_i is convex combination of $\mu_{j_1}, \ldots, \mu_{j_{Z+1}}$, while for some $q : 1 \le q \le Z + 1$, $\mu_{j_q} \ne \mu_i$. However, the curve (x, x^2) is strictly convex. Therefore, it is impossible that one point on this curve is a convex combination of other points. □

We have completed the construction of the labeling scheme $\mathcal{L}$. The main features of the labeling scheme is that while the canonical paths connecting the source-sink pairs are long and they share many vertices, no non-canonical source-sink paths exist in $\mathcal{L}$. There remains however one major obstacle in combining the labeling scheme with graph H to obtain the final graph G. When viewed as a directed multicut instance, $\mathcal{L}$ has a cheap solution: the removal of all the vertices in one of the layers disconnects all the source-sink pairs. We perform one final transformation to obtain the final labeling scheme $\mathcal{L}' = (U, \mathcal{M}', E'')$. The set of non-terminal vertices and the source-sink pairs in $\mathcal{L}'$ are the same as in $\mathcal{L}$. As for the set E'' of edges, it is defined as follows. For every $i : 1 \le i \le k$, we define a set E''_i of edges of type i. Consider any pair of vertices $x, x' \in U \cup T(\mathcal{M}')$ (they can be either terminal or non-terminal vertices). If there is a path consisting of type-i edges from x to x' in $\mathcal{L}$, and if $(x, x') \notin \mathcal{M}'$ (i.e., they are not a source-sink pair), then we add a type-i edge $(x \to x')$ to E_i''. The final set of edges $E'' = \cup_{i=1}^{k} E_i''$. We refer to $\mathcal{L}'$ as the *modified labeling scheme*. Next we describe its properties.

Let $(s_i(y), t_i(y')) \in \mathcal{M}'$ be any source-sink pair of type i, and let P be any path connecting $s_i(y)$ to $t_i(y')$ in $\mathcal{L}'$. Path P is called *canonical* if it contains type-i edges only. Notice that now for each source-sink pair there are *many* canonical paths.

CLAIM 3.1. *There are no non-canonical source-sink paths in $\mathcal{L}'$.*

PROOF. Let P be any non-canonical path connecting some source-sink pair $s_i(y)$-$t_i(y')$. Let $e = (x, x')$ be any edge on path P, and suppose its type is i'. Then there is a path connecting x to x' in graph $\mathcal{L}$ containing type-i' edges only. Therefore, path P corresponds to some non-canonical path connecting $s_i(y)$ to $t_i(y')$ in graph $\mathcal{L}$, which is impossible from Lemma 3.2. □

3.3 Step 3: The Final Graph

We are now ready to describe the construction of the final graph $G = (V^*, \mathcal{M}^*, E^*)$; it is done by combining graph $H = (V, \mathcal{M}, E)$ with the modified labeling scheme $\mathcal{L}' = (U, \mathcal{M}', E'')$. We use the modified labeling scheme with parameters $\tau = k$ and $Z = 32L$. Thus, $|Y| = O(k^3 Z^2) = O(n^3 Z^2)$. The final graph is defined as follows. The set of non-terminal vertices $V^* = U \times V$, that is,

$$V^* = \{v(y, h, p) \mid y \in Y, 1 \le h \le 32L, 1 \le p \le n\}$$

For each non-terminal vertex $v(y, h, p)$, we define its *pre-images* in graphs H and $\mathcal{L}'$ to be $g_H(v(y, h, p)) = p$ and $g_{\mathcal{L}'}(v(y, h, p)) = u(y, h)$, respectively. The set of source-sink pairs is $\mathcal{M}^* = \mathcal{M}'$, that is, it is given by

$$\{(s_i(y), t_i(y')) \mid 1 \le i \le k, y \in Y, y' = y \oplus ((32L + 1)\mu_i)\}.$$

The pre-images of the sources in graphs H and $\mathcal{L}'$ are defined to be $g_H(s_i(y)) = s_i$ and $g_{\mathcal{L}'}(s_i(y)) = s_i(y)$, respectively. The pre-images of the sinks are defined in a similar way. Finally, the edges are defined as follows. For each $i : 1 \le i \le k$, we define a subset E_i^* of edges of type i, and we set $E^* = \bigcup_{i=1}^{n} E_i^*$. We add a type-$i$ edge $(x \to x')$ to set E_i^* iff a type-i edge exists both in graph H and in graph $\mathcal{L}'$ between the corresponding pre-images, that is,

$$\begin{aligned} E_i^* = \{&(x \to x') \mid x, x' \in V^* \cup T(\mathcal{M}^*), \\ &(g_H(x) \to g_H(x')) \in E_i \text{ and } (g_{\mathcal{L}'}(x) \to g_{\mathcal{L}'}(x')) \in E_i''\} \end{aligned}$$

This completes the definition of graph G. The number of non-terminal vertices in graph G is bounded by $N \le |Y|Zn \le O(n^3 L^3 n) = O(n^7/(\log n)^3)$. Since $L = n/(4 \log n)$, we have that $L = \Omega\left(\frac{N^{1/7}}{(\log N)^{4/7}}\right)$.

Fractional solution:. For any source-sink pair $(s_i(y), t_i(y')) \in \mathcal{M}^*$, we say that path P connecting the source to the sink is *canonical* iff it uses edges of type i only. The next lemma establishes the property (C_1) for graph G.

LEMMA 3.3. *For any source-sink pair $s_i(y)$-$t_i(y')$ in G, any path connecting $s_i(y)$ to $t_i(y')$ contains at least L non-terminal vertices.*

The proof relies on the following lemma.

LEMMA 3.4. *No non-canonical source-sink paths exist in graph G.*

PROOF. Assume otherwise. Let P be a non-canonical path connecting some source $s_i(y)$ to its sink $t_i(y')$. Let P' be the sequence of pre-images of vertices of P in graph $\mathcal{L}'$, appearing in the same order as in P. Then P' forms a non-canonical path that connects the source-sink pair $s_i(y)$-$t_i(y')$ in graph $\mathcal{L}'$, which is impossible due to Claim 3.1 □

We can now prove Lemma 3.3.

PROOF. Consider any source-sink pair $(s_i(y), t_i(y')) \in \mathcal{M}^*$, and let P be any path connecting $s_i(y)$ to $t_i(y')$ in G. From Lemma 3.4, P is a canonical path. Let P' be a sequence of vertices containing the pre-images of vertices on path P in H, in the same order as they appear in P. Clearly, P' is a canonical path connecting s_i to t_i in graph H and thus, from Observation 1, it contains at least L non-terminal vertices. It follows that P also contains at least L non-terminal vertices. □

Lemma 3.3 implies that there is a feasible fractional solution of cost N/L, achieved by assigning $1/L$-fraction to each non-terminal vertex of G.

Integral solution:. A key property of our final construction is summarized by the lemma below.

LEMMA 3.5. *Assume that event $\mathcal{B}$ does not happen for H. Then for any subset $\mathcal{S} \subseteq V^*$ of non-terminal vertices, $|\mathcal{S}| \le N/32$, the fraction of source-sink pairs which are disconnected when $\mathcal{S}$ is removed from G is at most $99/100$.*

Before presenting a proof of the above lemma, we consider an immediate consequence of the lemma.

Since there is a fractional solution of cost N/L for graph G, we have that the integrality gap is $\Omega(L) = \Omega\left(\frac{N^{1/7}}{\log^{4/7} N}\right)$. Moreover, this gap holds even when the integral solution needs to disconnect only a $(1-\epsilon)$-fraction of the pairs where $\epsilon \ge 1/100$.

We have thus established the following theorem.

THEOREM 3.1. *The flow-cut gap between the maximum multicommodity flow and minimum multicut in directed graphs is $\tilde{\Omega}(N^{1/7})$. Moreover, this gap holds on directed acyclic graphs and even when the integral solution is required to separate only a $(1-\epsilon)$-fraction of the pairs for some $\epsilon > 0$.*

We now prove Lemma 3.5.

PROOF. Let $\mathcal{S} \subseteq V^*$ be any subset of non-terminal vertices, $|\mathcal{S}| \leq N/32$. For each label $y \in Y$ and for each layer $h : 1 \leq h \leq 32L$, let $\mathcal{S}_{y,h} = \{p : v(y,h,p) \in \mathcal{S}\}$. We say that label-layer pair (y,h) is *good* iff $|\mathcal{S}_{y,h}| \leq n/16$. Clearly, at least half the label-layer pairs are good: otherwise, we have $32L|Y|/2$ non-good label-layer pairs, each of them contributing more than $n/16$ vertices to $\mathcal{S}$, contradicting the fact that $|\mathcal{S}| \leq N/32 = 32L|Y|n/32$.

Fix a good label-layer pair (y,h). We say that index $i : 1 \leq i \leq n$ is *covered at* (y,h) iff there is some $j : 1 \leq j \leq L$, such that $S_j(i) \subseteq \mathcal{S}_{y,h}$ (recall that $S_j(i)$ is the jth layer in graph H_i). Since we assume that $\mathcal{B}$ does not happen, at least half the indices $i : 1 \leq i \leq n$ are not covered at (y,h).

Let $J \subseteq [1,\ldots,k]$ be the set of all indices $i : 1 \leq i \leq k$, such that the number of label-layer pairs (y,h) for which i is **not covered** at (y,h) is at least $4|Y|L$.

CLAIM 3.2. $|J| \geq k/7$.

PROOF. For each $i \in J$, there are at most $32|Y|L$ pairs (y,h) that do not cover i. For each $i \notin J$, there are at most $4|Y|L$ pairs (y,h) that do not cover i. Therefore, in total, the number of triples (y,h,i), where $y \in Y, 1 \leq h \leq 32L$ and $1 \leq i \leq k$ and (y,h) does not cover i is at most:

$$|J| \cdot 32|Y|L + (k - |J|) \cdot 4|Y|L$$

On the other hand, there are at least $16|Y|L$ good label-layer pairs (y,h), and for each of them at least $k/2$ indices $i : 1 \leq i \leq k$ are not covered at (y,h). Therefore, we have that:

$$|J| \cdot 32|Y|L + (k - |J|) \cdot 4|Y|L \geq 8|Y|Lk$$

Hence $|J| \geq k/7$. $\square$

For each $i : 1 \leq i \leq k$, we call the source-sink pairs in set $\{(s_i(y), t_i(y')) \mid y \in Y, y' = y \oplus ((32L+1)\mu_i)\}$ *source-sink pairs of type* i. The next claim will finish the proof.

CLAIM 3.3. *For each $i \in J$, the fraction of source-sink pairs of type i that are not disconnected by $\mathcal{S}$ is at least 3/31. Therefore, in total, the fraction of source-sink pairs which are not disconnected by $\mathcal{S}$ is at least $\frac{3}{31} \cdot \frac{1}{7} \geq \frac{1}{100}$.*

PROOF. Fix an $i \in J$. We partition the set $Y \times \{1,\ldots,32L\}$ of label-layer pairs into $|Y|$ subsets. For each $y \in Y$, we define a subset T_y, which contains, for each layer $h : 1 \leq h \leq 32L$, the pair (y_h, h), where $y_h = y \oplus (h\mu_i)$. Observe that $\{T_y\}_{y \in Y}$ is indeed a partition of all the label-layer pairs, where each pair appears in exactly one set T_y. Moreover, in graph $\mathcal{L}'$, there is an edge from $s_i(y)$ to every vertex $u(y,h)$ with $(y,h) \in T_y$, and every such vertex connects to $t_i(y')$, which is the sink corresponding to source $s_i(y)$, i.e., $(s_i(y), t_i(y')) \in \mathcal{M}^*$. Additionally, for each pair of vertices $u(y',h')$, $u(y'',h'')$ with $(y',h'), (y'',h'') \in T_y$ and $h' < h''$, there is an edge from $u(y',h')$ to $u(y'',h'')$ in $\mathcal{L}'$.

Let $Y' \subseteq Y$ denote the subset of labels y, for which the number of label-layer pairs $(y,h) \in T_y$ such that i is not covered at (y,h) is at least L. Since the total number of label-layer pairs (y,h) for which i is not covered at (y,h) is at least $4|Y|L$, we have

$$|Y'|(32L) + (|Y| - |Y'|)L \geq 4|Y|L.$$

It follows that $|Y'| \geq (3/31)|Y|$. Now fix *any label* $y \in Y'$. We will show that the source-sink pair $s_r(y)$-$t_r(y')$, where $y' = y \oplus ((32L+1)\mu_i)$ is not disconnected when $\mathcal{S}$ is removed from the graph. Since there are $|Y|$ source-sink pairs of type i, it follows that set $\mathcal{S}$ does not disconnect at least 3/31-fraction of these pairs, which will complete the proof.

For a fixed $y \in Y'$, let $1 \leq h_1 < h_2 < \cdots < h_L < 32L$ be indices of layers, such that for each $j : 1 \leq j \leq L$, i is not covered at the label-layer pair $(y_{h_j}, h_j) \in T_y$, where $y_{h_j} = y \oplus (h_j\mu_i)$. For each such $j : 1 \leq j \leq L$, we know that $S_j(r) \not\subseteq \mathcal{S}_{y_{h_j},h_j}$. In particular, there is some $p_j \in S_j$, such that vertex $v(y_{h_j}, h_j, p_j)$ does not belong to $\mathcal{S}$. We construct a path P connecting source $s_i(y)$ to its sink $t_i(y)$, that contains type-i edges only, as follows:

$$P = \big(s_i(y) \rightarrow v(y_{h_1}, h_1, p_1) \rightarrow \cdots \rightarrow v(y_{h_L}, h_L, p_L) \rightarrow t_i(y')\big)$$

>From the discussion above, the non-terminal vertices appearing on this path do not belong to $\mathcal{S}$. We only need to check that indeed for every pair of consecutive vertices on the path there is a type-i edge connecting them in graph G. This is immediate from the definition of type-i edges in graphs H and $\mathcal{L}'$. $\square$

Concurrent Flow vs. Sparsest Cut: We now build on the preceding result to show that a similar gap result holds for concurrent flow and sparsest cut even in directed acyclic graphs. We apply the transformation outlined in Section 2 to graph G constructed in the previous section to obtain an instance G' of the (edge version of) the directed multicut. Let E^0 denote the set of special edges in G' and let $K = n|Y|$ denote the number of source-sink pairs in G'. Consider the following fractional solution to (LP2-P). For each source-sink pair (s_i, t_i), we set $h'_i = 1/K$. For each special edge $e \in E^0$, we set $x'_e = 1/KL$. This is a feasible solution to (LP2-P) of cost N/KL.

Assume that the integrality gap of (LP2-P) is less than $g(N)$ for some function g. Using an argument similar to the one given in [10] for converting bicriteria hardness of undirected multicut to hardness of sparsest cut, we show that there is a subset E^1 of edges in graph G', $|E^1| = O(N/L)g(N)$, whose removal disconnects 0.99-fraction of source-sink pairs. We perform several iterations, while in each iteration we remove some edges from G' and disconnect some source-sink pairs. The iterations are performed while the number of source-sink pairs disconnected is less than $0.99K$. It is easy to see that at the beginning of each iteration there is a fractional solution to (LP2-P) of cost $100N/KL$: each special edge $e \in E^0$ that belongs to the graph at the beginning of the current iteration is assigned $x'_e = 100/KL$, and each source-sink pair which is still not disconnected (there are at least $K/100$ of them) is assigned $h'_i = 100/K$. This is a feasible solution to (LP2-P) of cost $\varphi = 100N/KL$. Therefore, in each iteration, there is an (integral) cut S of containing at most $\varphi g(N)k'$ edges that separates k' pairs for some integer $k' \geq 1$. We delete all edges in S from G' as well as remove the pairs separated by S. We repeat this until the number of remaining source-sink pairs falls below $K/100$. Summing up over all iterations, we obtain a set of at most $\varphi g(N)K = (100N/L)g(N) = O((N/L)g(N))$ that separates at least $0.99K$ pairs. The following theorem now easily follows from the above discussion and Theorem 3.1.

THEOREM 3.2. *The gap between concurrent multicommodity flow and (non-bipartite) sparsest cut is* $\tilde{\Omega}(N^{1/7})$.

4. HARDNESS OF APPROXIMATION

We give a reduction from a general class of constraint satisfaction problem (CSP). In a constraint satisfaction problem we are given a set X of variables defined over some field F, and a set of constraints Ψ. Each constraint $\psi \in \Psi$ involves D variables, and a list R_ψ of assignment to variables of ψ that satisfy this constraint is given as problem input. The goal is to find assignments to variables so as to satisfy maximum possible number of constraints.

There is a natural way to reduce the constraint satisfaction problem to directed multicut using the techniques developed for the integrality gap construction. The main difference is in the way graph H is constructed. The non-terminal vertices of H will represent the variables and their assignments, i.e., for every variable $x \in X$ and for every assignment $a \in F$ to x, there is a non-terminal vertex $v(x, a)$ in graph H. The source-sink pairs will represent the constraints, namely, for each $\psi \in \Psi$, there is a source-sink pair s_ψ-t_ψ. For each $\psi \in \Psi$, we construct a set E_ψ of edges, by defining subsets $S_j(\psi)$ of vertices (layers), which are connected to each other and to the source-sink pairs as before. The subsets $S_j(\psi)$ will correspond to the satisfying assignments in R_ψ. Thus, if A_j is the jth satisfying assignment in R_ψ, then $S_j(\psi)$ will contain, for each variable x participating in ψ, the vertex $v(x, a)$, where a is the projection of A_j onto x. The rest of the construction, including the labeling scheme $\mathcal{L}$, its transformed version $\mathcal{L}'$ and the composition of $\mathcal{L}'$ with H remain the same.

Unfortunately, this approach does not work with general constraint satisfaction problems. The main difficulty is that the same vertex $v(x, a)$ might belong to several sets $S_j(\psi)$ (for some fixed ψ), and because of this we cannot ensure that in the case of YES-INSTANCEthe "standard" solution will disconnect all the source-sink pairs. Alternatively, if we view the constructed graph as an integrality gap example, we will have some short canonical paths, and hence we will not obtain large gap. As noted in the introduction, this problem does not arise in the CSPs obtained from the Raz verifier (where $D = 2$), due to the projection property. However, the strongest possible hardness achievable via this approach is $2^{\Omega(\log^{1-\epsilon} n)}$. To break this barrier, we need to allow reductions from more general type of CSPs that do not necessarily have an analog of the projection property. We give here an overview of our approach to handle general CSPs. Due to space limitations, we have deferred a formal description of the construction and its analysis to the full version that appears on the authors' webpages.

To overcome the difficulty that the same vertex $v(x, a)$ might belong to several sets $S_j(\psi)$ for some fixed ψ, we create many copies of each vertex $v(x, a)$ representing assignment a to variable x. Now layers $S_j(\psi)$ will use different copies for different indices j, thus avoiding the creation of these bad paths. However, we need to enforce consistency among multiple copies of an assignment to a variable. Specifically, we would like to ensure the following. Fix any solution $\mathcal{S}$, and say that a variable-assignment pair (x, a) is *chosen by* $\mathcal{S}$ iff at least 1/4 of the copies of corresponding vertices belong to $\mathcal{S}$. Let ψ be any constraint for which no satisfying assignment is chosen by $\mathcal{S}$. We want to ensure that in this case there is an s-t pair corresponding to ψ, which is not disconnected by $\mathcal{S}$.

To achieve this goal, instead of choosing the layers $S_1(\psi), \ldots, S_L(\psi)$ just once, we perform $|Y|\Gamma$ such independent choices, each one of them defining a different subset $E_{y,\gamma}$ of edges in graph H, for $y \in Y$, $1 \leq \gamma \leq \Gamma$ (here $\Gamma \leq \text{poly}(n)$ and Y is the set of labels). For a fixed $y \in Y$, $\gamma : 1 \leq \gamma \leq \Gamma$, in order to choose a subset $S_j(\psi)$ of vertices, we randomly choose $\delta = O(\log n)$ copies of every vertex representing the variable-assignment pair (x, a) where x is a variable of ψ and the projection of the jth assignment in R_ψ onto x is a. The random choices are performed with no repetitions across various layers, so each copy may appear in at most one layer for a fixed pair y, γ and a fixed constraint ψ. Let g denote the hardness gap of the constraint satisfaction system. The resulting instance has the property that with high probability, the cost of any No-instance solution is at least $\Omega(g)$ times the cost of an optimal Yes-instance solution even when the No-instance solution is only required to separate a $(1 - \epsilon^*)$-fraction of source-sink pairs for some constant $\epsilon^* > 0$. Using standard arguments, we can then translate the hardness for this bicriteria version to a matching hardness for sparsest cut in directed graphs.

Acknowledgements

We would like to thank Irit Dinur, Piotr Indyk, and Ran Raz for helpful discussions.

5. REFERENCES

[1] A. Agarwal, N. Alon, and M. Charikar. Improved approximation for directed cut problems. In *Proc. of STOC*, 2007.

[2] P. Alimonti and V. Kann. Hardness of approximating

problems on cubic graphs. *Theoretical Computer Science*, 237: 123-134, 2000.

[3] M. Andrews and L. Zhang. Logarithmic hardness of the directed congestion minimization problem. In *Proc. of STOC* '06.

[4] S. Arora, J. R. Lee, A. Naor. Euclidean distortion and the sparsest cut. In *Proc. of STOC*, 2005, pp. 553–562.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[6] S. Arora, S. Rao, and U. V. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proc. of STOC*, 2004, pp. 222–231.

[7] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *JACM*, 45(1):70–122, 1998.

[8] M. Bellare, S. Goldwasser, C. Lund, A. Russeli: Efficient probabilistically checkable proofs and applications to approximations. In *Proc. of STOC*, 1993, pp. 294–304.

[9] M. Charikar, K. Makarychev, Y. Makarychev. Directed Metrics and Directed Graph Partitioning Problems. In *Proc. of SODA*, 2006, pp. 51–60.

[10] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, D. Sivakumar. On the Hardness of Approximating Multicut and Sparsest-Cut. In *Proc. IEEE Conference on Computational Complexity*, 2005, pp. 144–153.

[11] J. Cheriyan, H. J. Karloff, Y. Rabani. Approximating Directed Multicuts. In *Proc. of FOCS*, 2001, pp. 320–328.

[12] J. Chuzhoy and S. Khanna. Hardness of Cut Problems in Directed Graphs. In *Proc. of STOC*, 2006.

[13] J. Chuzhoy and S. Khanna. Hardness of Directed Routing with Congestion. ECCC Technical Report TR06-109, August 2006. *http://eccc.hpi-web.de/eccc-reports/2006/TR06-109/index.html.*

[14] J. Chuzhoy, V. Guruswami, S. Khanna, and K. Talwar. Hardness of Routing with Congestion in Directed Graphs. In *Proc. of STOC*, 2007.

[15] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, M. Yannakakis. The Complexity of Multiterminal Cuts. In *SIAM J. Comput.*, 23(4): 864-894, 1994.

[16] I. Dinur, E. Fischer, G. Kndler, R. Raz and S. Safra. PCP characterizations of NP: towards a polynomially-small error-probability. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC)*, pp. 29–40, 1999.

[17] I. Dinur and S. Safra. On the hardness of approximating label-cover. *Information Processing Letters*, 89(5), pp. 247–254, 2004.

[18] U. Feige. Relations between average case complexity and approximation complexity. In *Proc. of STOC*, 2002, pp. 534–543.

[19] U. Feige and S. Kogan. Hardness of Approximation of the Balanced Complete Bipartite Subgraph Problem. Technical Report MCS04-04, Department of Computer Science and Applied Math., The Weizmann Institute of Science, 2004.

[20] L. R. Ford, D. R. Fulkerson, 1962. Flows in Networks. Princeton University Press, Princeton, NJ.

[21] V. Guruswami and K. Talwar. Hardness of Low Congestion Routing in Directed Graphs. ECCC Technical Report TR06-141, November 2006, *http://eccc.hpi-web.de/eccc-reports/2006/TR06-141/index.html.*

[22] N. Garg, V. Vazirani, M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *Proc. of STOC*, 1993, pp. 698–707.

[23] N. Garg, V. Vazirani, M. Yannakakis. Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees. *Algorithmica*, 18(1):3–20, 1997. Preliminary version in *Proc. of ICALP*, 1993.

[24] A. Gupta. Improved results for directed multicut. In *Proc. of SODA*, 2003, pp. 454-455.

[25] M.T. Hajiaghayi, H. Räcke. An $O(\sqrt{n})$-Approximation Algorithm For Directed Sparsest Cut, *Information Processing Letters*, 97(4): 156-160, 2006.

[26] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. of STOC*, 2002, pp. 767–775.

[27] S. Khot, N. K. Vishnoi. The Unique Games Conjecture, Integrality Gap for Cut Problems and the Embeddability of Negative Type Metrics into ℓ_1. In *Proc. of FOCS*, 2005, pp. 53–62.

[28] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *JACM*, 46(6):787–832, 1999. Preliminary version in *Proc. of FOCS*, 1988.

[29] R. Raz. A parallel repetition theorem. *SIAM J. of Computing*, 27(3):763–803, 1998.

[30] M. E. Saks, A. Samorodnitsky, L. Zosin. A Lower Bound On The Integrality Gap For Minimum Multicut In Directed Networks. *Combinatorica* 24(3): 525–530 (2004).