On the max-flow min-cut ratio for directed multicommodity flows

MohammadTaghi Hajiaghayi[*] *Tom Leighton*[*]

Abstract

We present a pure combinatorial problem whose solution determines max-flow min-cut ratio for directed multicommodity flows. In addition, this combinatorial problem has applications in improving the approximation factor of the greedy algorithm for the maximum edge disjoint path problem. More precisely, our upper bound improves the approximation factor for this problem to $O(n^{3/4})$.

1 Introduction

Leighton and Rao [5] first introduced a relation between minimum sparsest cut and maximum concurrent multicommodity flow in undirected graphs. This relation has been used to develop novel tools for designing divide-and-conquer approximation algorithms for NP-complete problems on undirected graphs (See Shmoys [7] for a survey). The directed variant of the problem appears much harder (e.g., it is NP-hard for $k = 2$, the case which can be solved in polynomial time in undirected graphs). Despite persistent research efforts, the current bounds for directed graphs are weak. To state the bound, we need to define the problem more precisely. In a given directed graph G, with capacities on edges together with a list of k source-sink pairs of vertices called *commodities*, we want to find a cut of minimum capacity called a minimum *multicut* whose removal disconnects all source-sink pairs. In a recent work, Saks et al. [6] construct a family of k-commodity networks, for all k and $\epsilon > 0$, where the minimum multicut to maximum sum multicommodity flow ratio is no less than $k - \epsilon$, in contrast with the $O(\log k)$ upper bound in the undirected case [5] (an upper bound k is trivial.) However, in this instance, the number of vertices is exponential in k. Cheriyan et al. [2] obtain the upper bound $O(\sqrt{n \log k})$ for this ratio which is further improved to $O(\sqrt{n})$ by Gupta [3]. However, still there is a big gap between the lower bound $\Omega(\log n)$ and the upper bound $\min\{O(\sqrt{n}), k\}$.

In this paper, we mainly focus on distinguishing between polylog(n) and n^ϵ for the max-flow min-cut ratio in directed graphs which is indeed the integrality gap of an LP. To this end, we introduce a pure combinatorial problem whose solution determines this ratio. In addition, we demonstrate how this combinatorial problem has applications in improving the approximation factor of the greedy algorithm for the maximum *edge disjoint path* problem (EDP) on directed graphs in which given a graph G and a set $\mathcal{T}$ of k source-sink pairs, our objective is to connect a maximum number of these pairs via edge-disjoint paths. Our upper bound improves the approximation factor for this problem from $O(n^{4/5})$ [1] to $O(n^{3/4})$. Guruswami et al [4] showed that it is NP-hard to approximate EDP in directed graphs within an $\Omega(n^{1/2-\epsilon})$ factor for every fixed $\epsilon > 0$.

2 Cutting Far Pairs

In this paper, we consider a more convenient formulation of the multicut problem as follows. Given a set of pairs $\mathcal{T} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$, we want to pick a set C of vertices such that in the remaining graph there

[*]Department of Mathematics and Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA 02139, U.S.A. Emails: {hajiagha, ftl}@theory.csail.mit.edu.

is no path from s_i to t_i, $1 \leqslant i \leqslant k$. We note that $s_1, t_1, \ldots, s_k, t_k$ can belong to C. The LP relaxation for this problem is as follows (below by $\text{dist}_x(s_i, t_i)$, we mean the length of the shortest path from s_i to t_i when vertex-weights are x_v's):

$$
\begin{aligned}
F = \text{minimize} & & \sum_{v \in V} x_v & \qquad (1)\\
\text{subject to} & \quad \forall i \in \{1, \cdots, k\} & \text{dist}_x(s_i, t_i) \geqslant 1 \\
& \quad \forall v \in V & x_v \geqslant 0
\end{aligned}
$$

The above LP is the dual of maximum sum multicommodity flow in which the goal is to maximize the sum of flows from sources to the corresponding sinks such that the total flow passing through each vertex is at most one. Indeed, the integrality ratio of this LP is the max-flow min-cut ratio, which we discussed in Section 1.

We are now ready to introduce our combinatorial problem. The *Cutting Far Pairs (CFP)* problem is defined as follows. Let $G(V, E)$ be a simple unit-capacity directed graph and let $\mathcal{T}$ be the set of all source-sink pairs for which the distance in G from each source to the corresponding sink is at least l. What is the size of the smallest vertex-cut in terms of n and l that separates all pairs in $\mathcal{T}$?[1]

The next theorem shows how the CFP problem captures the hardness of the integrality gap of LP (1).

Theorem 2.1. *If there exists a graph for which any solution to CFP is in $\Omega((\frac{n}{l})^{1+\alpha})$ for some $l = O(n^{1-\epsilon})$ where $\alpha, \epsilon > 0$, then the integrality gap of LP (1) is in $\Omega(n^{\epsilon\alpha})$. On the other hand, if the solution to CFP for every graph G and for all l is in $O(\frac{n}{l}\text{polylog}(n))$, then the integrality gap of LP (1) is in $O(\text{polylog}(n))$.*

Proof. The first part is easy. Consider the graph instance G mentioned in the statement of the theorem. Suppose we set $x_v = 1/l$ for each $v \in V(G)$. Since this is a feasible solution to LP (1), we have $F \leqslant \frac{n}{l}$. Since the integer solution is in $\Omega((\frac{n}{l})^{1+\alpha})$, by setting $l = O(n^{1-\epsilon})$, the integrality gap is in $\Omega(n^{\epsilon\alpha})$, as desired.

We now consider the second part. Consider a solution x to LP (1). Suppose that for each $v \in V(G)$, we round up x_v to the nearest multiple of n^{-1}. After this rounding process, we have a feasible solution x' such that $\frac{\sum_{v \in V(G)} x'_v}{F}$ is at most $1 + n\frac{1}{n} = 2$ (since $F \geqslant 1$). Now, in order, for each vertex v with $x'_v = 0$ first we delete v from the graph and then for each pair of edges $(u, v), (v, w) \in E(G)$, we add an edge (u, w) to the new graph and call the new graph G'. Also, if v was a source in a pair (v, t) (a sink in a pair (s, v)), we omit this pair and instead we add all pairs (w, t) ((s, w)) for which $(v, w) \in E(G)$ ($(w, v) \in E(G)$) and there is a path from w to t (s to w) in G. Finally, for each vertex v which has $x'_v > 0$, we replace v by a path of length x'_v/n^{-1} (recall that x'_v is a multiple of n^{-1}). Call the new graph G''. In the new graph we consider all pairs for which their distance with respect to x is at least 1 and thus their distance in G'' is at least n. Every cut of our new instance corresponds to a cut in the original graph G. Also, $\sum_{v \in V(G)} x'_v = |V(G'')| n^{-1}$. By our assumption, we can cut all the far pairs by at most $O(\frac{|V(G'')|}{n}\text{polylog}(|V(G'')|)) = O(\frac{|V(G'')|}{n}\text{polylog}(n))$ vertices ($|V(G'')|$ is at most n^2). Since this cut corresponds to a cut in the original graph G, the integrality gap of LP (1) is at most $O(\frac{\frac{|V(G'')|}{n}\text{polylog}(n)}{F}) = O(\frac{\frac{|V(G'')|}{n}\text{polylog}(|V(G'')|)}{\sum_{v \in V(G)} x'_v}) = O(\text{polylog}(n))$ as desired. □

One can easily observe that if we have an instance for the CFP problem with solution in $\Omega((\frac{n}{l'})^{1+\alpha})$ for any length $l' = O(n^{1-\epsilon})$, then by subdividing nodes as we did in the proof of Theorem 2.1, we can obtain

[1] The edge-cut version of CFP has been studied by Chekuri and Khanna [1].

an instance for $l = \Theta(n^{1-\epsilon})$ with solution in $\Omega((\frac{n}{l})^{1+\alpha})$. Thus this observation and Theorem 2.1 implicitly say that the hardest case of CFP is the case in which $l = \Theta(n^{1-\epsilon})$.

The next theorem shows how we can obtain an upper bound $O(\frac{n^2}{l^2})$ for the CFP problem.

Theorem 2.2. *For any length l, the CFP problem has a solution of size in $O(\frac{n^2}{l^2})$.*

Proof. Here, we show that there exists a vertex-cut C of the desired size. First we initiate C with an empty set. Then we add vertices to C during a number of iterations. In the beginning of the jth iteration, if there exists no $s_i - t_i$ path in the residual graph G (G will be updated after each iteration) where s_i and t_i consist of a far pair in the original graph, we are done. Otherwise choose a far pair (s, t) for which there exists a path from s to t in G. Remove all vertices v of G for which there exists no (simple) directed $s-t$-path which goes through v. We call the remaining graph G'. We now do a breadth-first search from s in graph G' and call the vertices at distance i from s *layer* L_i. Also we let $X = L_1 \cup L_2 \cup \cdots \cup L_{\frac{l}{3}}$, $Y = L_{\frac{l}{3}+1} \cup L_2 \cup \cdots \cup L_{\frac{2l}{3}}$ and $Z = V(G') - X - Z$. Assume $|Y| = c_j$. Since the layers are disjoint, there exists a layer of size at most $\frac{3c_j}{l}$ in Y. We add vertices of such a layer to C and remove them from G. Clearly after the termination of the algorithm, set C is a directed multicut. We show that $|C|$ is in $O(n^2/l^2)$. Let k' be the total number of iterations of our algorithm. We double-count the number of (a, b)-pairs in G for which there exists a path from a to b in G. This number is at most n^2. On the other hand, consider the jth iteration and a vertex $v \in Y$. After cutting the layer within Y, either there is no path from v to t, or there exists no path from s to v. We consider the former case and the latter case has a very similar analysis. We know that before cutting, there is a (simple) path P from s_i to t_i which goes through v. Thus there are at least $l/3$ vertices of path P in Z to which there were paths from v, but now there is no path. Similarly, in the latter case, if there exists no path from s to v, there are $l/3$ vertices in X from which there exists no path to v now, but it was before. Thus for c_j vertices in Y, we separate at least $\frac{c_j l}{3}$ pairs of vertices that were connected before the jth iteration. We now observe that the total number of vertices in C is at most $\sum_{j=1}^{k'} \frac{3c_j}{l}$ subject to $\sum_{j=1}^{k'} \frac{c_j l}{3} \leqslant n^2$. In this case the maximum size of $|C|$ is in $O(n^2/l^2)$ as desired. □

We note that in the proof of Theorem 2.2, the ratio of the number of pairs which are disconnected (i.e., $\frac{lc_j}{3}$) to the deleted vertices (i.e., $3c_j/l$), which can be considered as a notion of *efficiency*, is in $O(l^2)$ which is tight for some graphs (e.g., consider a source s and a sink t among which there are n/l disjoint paths of length l directed from s to t and there are no more edges in the graph).

The main remaining open problem is closing the gap between the trivial lower bound $O(n/l)$ and the upper bound $O(n^2/l^2)$ in the worst case of the CFP problem.

Finally, it is worth mentioning that by the same method of Theorem 2.2, we can improve the approximation factor of the EDP problem (see the definition in the introduction) from $O(n^{4/5})$ [1] to $O(n^{3/4})$. Since the submission date of this note, our $O(n^{3/4})$ approximation factor has been improved to $O(n^{2/3} \log^{2/3} n)$ by Varadarajan and Venkataraman [8] [2].

Theorem 2.3. *There exists an $O(n^{3/4})$-approximation algorithm for the edge disjoint path problem (EDP) in directed graphs.*

Proof. The algorithm is a simple greedy algorithm which considers the shortest path length for each unrouted pair $(s_i, t_i) \in \mathcal{T}$ and connects a pair with minimum shortest path length via its shortest path.

The $O(n^{3/4})$ approximation factor for the above algorithm can be obtained by plugging Lemma 2.4 below into the analysis of Chekuri and Khanna [1] (Lemma 2.4 is an improvement to their Theorem 3.2).

[2] Varadarajan and Venkataraman [8] refer to our paper as a previous work.

Lemma 2.4. *Let $G(V, E)$ be a simple unweighted directed graph and let $\mathcal{T}$ be the collection of all source-sink pairs such that the distance in G from each source to its corresponding sink is at least l. Then there exists an edge-cut C of size $O(n^3/l^3)$ that separates all pairs in $\mathcal{T}$.*

Proof. The proof is essentially the same as that of Theorem 2.2 with this difference that here we want to delete edges instead of vertices to separate far pairs. We partition the layers defined in the proof of Theorem 2.2 into blocks of two adjacent layers each such that the ith *block* B_i consists of vertices in the layers L_{2i+1} and L_{2i+2}. Now in each iteration we add all edges within a block with minimum number of vertices in Y to the edge-cut C [3] (the number of edges within this block is at most $\frac{36c_j^2}{l^2}$ since the number of vertices in this block is at most $\frac{c_j}{l/6}$ [4]). We can observe that the total number of edges in C is at most $\sum_{j=1}^{k'} \frac{36c_j^2}{l^2}$ subject to $c_j \leqslant n$ and $\sum_{j=1}^{k'} \frac{c_j l}{3} \leqslant n^2$. We see that $|C|$ is maximized when $\min\{3n/l, k'\}$ of c_j's are n and the rest are zero. In this case $|C|$ is in $O(n^3/l^3)$ as desired. □

□

Acknowledgment: We thank Anupam Gupta and Baruch Schieber for several fruitful discussions.

References

[1] C. Chekuri and S. Khanna, *Edge disjoint paths revisited*, in Proceedings of the fourteenth annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 628–637.

[2] J. Cheriyan, H. J. Karloff, and Y. Rabani, *Approximating directed multicuts*, in Proceedings of the 42th Annual Symposium on Foundations of Computer Science, 2001, pp. 348–356.

[3] A. Gupta, *Improved results for directed multicut*, in Proceedings of the fourteenth annual ACM-SIAM Symposium on Discrete Algorithms, 2003. 454-455.

[4] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis, *Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems*, J. Comput. System Sci., 67 (2003), pp. 473–496.

[5] T. Leighton and S. Rao, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, J. ACM, 46 (1999), pp. 787–832.

[6] M. Saks, A. Samorodnitsky, and L. Zosin, *A lower bound on the integrality gap for minimum multicut in directed networks*. To appear in Combinatorica.

[7] D. B. Shmoys, *Cut problems and their application to divide-and-conquer*, in Approximation Algorithms for NP-hard Problem, PWS Publishing Company, 1997, pp. 192–235.

[8] K. Varadarajan and G. Venkataraman, *Graph decomposition and a greedy algorithm for edge-disjoint paths*, in Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms, 2004, pp. 379–380.

[3] Without loss of generality, we assume that l is divisible by 6, since otherwise we can decrease l by at most 5 to obtain such a property.

[4] Note that we could just remove all the edges going from the first layer to the second layer of a block instead of removing all the edges within the block. Here, we do not necessarily obtain the best constants on desired bounds.