Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2003-002
MIT-LCS-TR-910

December 12, 2005

On the Max-Flow Min-Cut Ratio for Directed Multicommodity Flows

MohammadTaghi Hajiaghayi and F. Thomson Leighton

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

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

MohammadTaghi Hajiaghayi[*] *F. Thomson Leighton*[*]

Abstract

We give 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 Greedy algorithm for maximum edge disjoint path problem. More precisely, our upper bound improves the approximation factor for this problem to $O(n^{3/4})$. Finally, we demonstrate how even for very simple graphs the aforementioned ratio might be very large.

Leighton and Rao [5] first introduced a relation between maximum multicommodity flow and minimum cut 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 efficiently for 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 minimum cut 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 k-commodity flow ratio is no less than $k - \epsilon$, in contrast with the $\Omega(\log k)$ upper bound in the undirected case (an upper bound k is trivial.) However, in this instance, the number of vertices is exponential in k. Also Cheriyan et al. [2] obtain the upper bound $O(\sqrt{n \ln(k)})$ which is further improved to $O(\sqrt{n})$ by Gupta [3]. However, still there is a big gap between the lower bound $O(\log(n))$ and the upper bound $\min\{O(\sqrt{n}), k\}$.

So far, many researchers have considered the directed multicut problem described above. Another variant called the *sparsest cut problem* considered by Leighton and Rao [5] is as follows: given a graph G with k commodities each has its own demand, we search for a cut whose ratio of the capacity of the cut to the separated demands is minimized. One can observe that the sparsest cut problem for directed graphs can be approximated within factor $\alpha \log D$, where D is the sum of the demands and α is the best approximation ratio for directed multicut (the same reduction from sparsest cut to multicut in undirected graphs works for directed graphs, e.g., see Section 5.3.2 of [7].) Thus using the result of Gupta [3], we have $O(\log k \sqrt{n})$-approximation for directed sparsest cut problem in which all demands are unit. We note that in fact Leighton and Rao [5] consider this problem which has many algorithmic applications. Therefore any progress in improving the max-flow min-cut ratio and hence the approximation factor of multicut has direct consequences in efficient decompositions of directed graphs and obtaining approximation algorithms for many NP-hard problems on these graphs.

In this paper, we mainly focus on distinguishing between *polylog*(n) and $O(n^{\epsilon})$ for the max-flow min-cut ratio in directed graphs which is in fact the integrality gap of an LP. To this end, we introduce a pure combinatorial problem whose solution determines this ratio. In addition, as pointed out by Chekuri and

[*]Department of Mathematics and Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, U.S.A. {hajiagha, ftl}@math.mit.edu

Khanna [1] this combinatorial problem has applications in improving the approximation factor of the greedy algorithm for the *edge disjoint path* problem (EDP) on directed graphs in which given a graph G and 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 $O(n^{3/4})$. Finally, we demonstrate how, even for very simple graphs, the integrality gap might be very large.

1 Cutting far pairs

In this paper we consider the 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 is no path from a s_i to t_i, $1 \leqslant i \leqslant k$. We note that here $s_1, t_1, \ldots, s_k, t_k$ can belong to C. The LP relaxation for this problem is as follows:

$$F = Min \sum_{v \in V} x_v; \text{such that} : \text{dist}_x(s_i, t_i) \geqslant 1, \forall i \in \{1, \cdots, k\}; x_v \geqslant 0, \forall v \in V \quad (1)$$

The reason that we denote the optimal solution by F is that in fact this LP is a dual of max-sum multi-commodity flow problem in which we want to maximize the sum of flows from sources to sinks such that the total flow passing through each vertex is at most one. We now introduce our combinatorial problem.

The *Cutting Far Pairs* problem (CFP) can be 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 such that the shortest distance in G between source and sink is at least l. What is the size of the smallest cut in terms of n and l that separates all pairs in $\mathcal{T}$?

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

Theorem 1.1. *If there exists a graph for which there are solutions to CFP of size at least $\Omega((\frac{n}{l})^{1+\alpha})$ for some $l = O(n^{1-\epsilon})$ where $\alpha, \epsilon > 0$ then the integrality gap of LP (1) is at least $n^{\epsilon\alpha}$. In the other hand, if the solution of CFP for every graph G and for all l is in $O(\frac{n}{l}polylog(n))$ then the integrality gap of the LP (1) is in $O(polylog(n))$.*

Proof. The first part is easy. Consider the instance mentioned in the Theorem. Suppose we set $x_v = 1/l$ for each $v \in V(G)$. Since this is a feasible solution of 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 at least $n^{\epsilon\alpha}$, as desired.

Now, we consider the other case. Consider a solution x of 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 consider 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. Also, for each vertex v for which $x'_v > 0$, 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 distances in G'' is at least n. We can correspond every cut of our new instance 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 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 is corresponded 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 1.1, we can obtain an instance for $l = \Theta(n^{1-\epsilon})$ with solution in $\Omega((\frac{n}{l})^{1+\alpha})$. Thus this observation and Theorem 1.1 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. Notice that using the same ideas mentioned in Theorem 1.2, we improve the approximation factor for the EDP problem in Theorem 1.3.

Theorem 1.2. *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 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 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) $s - t$-path for far pairs (s, t) 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 iteration j and a vertex $v \in Y$. After cutting the edges 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 situation. 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 each 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 1.2, the ratio of the number of pairs which are disconnected ($\frac{lc_j}{3}$) to the deleted vertices ($3c_j/l$), which can be considered as a concept of *efficiency*, is in $O(l^2)$ which is tight for some graphs (e.g. consider a source and a sink among which there are n/l disjoint paths of length l and there are no more edges in the graph).

By the same method of Theorem 1.2, we can improve the approximation factor of the EDP problem (see the definition in the introduction) from $O(n^{4/5})$ [1] to $O(n^{3/4})$. This result has also been claimed independently by K. Varadarajan (via private communications).

Theorem 1.3. *There exists an $O(n^{3/4})$-approximation for the edge disjoint path problem (EDP) in directed graphs.*

Proof. To obtain the result, we state an improvement of Theorem 3.2 of [1], a variant of CFP for which we want to delete edges instead of vertices to separate far pairs. The current bound for this problem is $O(n^4/l^4)$ which we improve to $O(n^3/l^3)$. In fact, our proof uses a different variant of BFS and a more careful counting than the ones in Theorem 3.2 of [1]. The rest of the proof is exactly the same as the proof of Chekuri and Khanna, in which they show that the simple greedy algorithm which considers the shortest path length for each unrouted (s_i, t_i) pair and connects a pair with minimum shortest path length via its shortest path has the desired approximation factor. Here the algorithm terminates as soon as the minimum shortest path length among unrouted pairs exceeds a certain length l ($l = \Theta(n^{3/4})$ for our case.)

The proof of the above claim about the edge variant of CFP is essentially the same as the proof of Theorem 1.2. The only difference is that we consider *blocks* B_i which consist of vertices of layers L_{2i+1} and L_{2i+2} instead of *layers* and we add all edges among a block of minimum size in Y to cut C (the number of edges among a block is at most $\frac{36c_j^2}{l^2}$). We want to minimize $\sum_{j=1}^{k'} \frac{36c_j^2}{l^2}$ 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. □

2 Further observations

In this section, we present some other approaches which are simple, however we think they give good insight to the problem. The proof of the following theorem follows from Kamal Jain's method of LP rounding ([4], Theorem 3.2).

Theorem 2.1. *Consider an optimal solution to LP (1) for a given instance. Suppose we remove all vertices whose x_v is at least $\frac{1}{\alpha}$ from the graph and all source-sink pairs which are cut. Now, if the integrality gap for the new instance is at most β, the integrality gap for the original instance is at most* $\max\{\alpha, \beta\}$.

Proof. The proof follows very similarly to Kamal Jain's method of LP rounding ([4], Theorem 3.2). Let $V_{1/\alpha} = \{v \in V(G) | x_v \geqslant \frac{1}{\alpha}\}$, $G_{res} = G - V_{1/\alpha}$ and F^* be the optimal solution to the LP of the new instance. One can easily observe that the restriction of F to G_{res} is a feasible solution for the new LP. Thus $F^* \leqslant F believe - \sum_{v \in V_{1/\alpha}} x_v$. By the definition of $V_{1/\alpha}$, we have $\alpha x_v \geqslant 1$ for each $v \in V_{1/\alpha}$. Therefore, we have $F \geqslant F^* + |V_{1/\alpha}|/\alpha$. Since the integrality gap of the new instance is at most β, we conclude that there is an integer solution $|V_{res}| \leqslant \beta \mathcal{F}^*$. Therefore, $V_{1/\alpha} \cup V_{res}$ is a solution to LP (1) whose size $|V_{1/\alpha}| + |V_{res}| \leqslant \max\{\alpha, \beta\}(\frac{|V_{1/\alpha}|}{\alpha} + \frac{|V_{res}|}{\beta}) \leqslant \max\{\alpha, \beta\} F$. □

We note that if we continue the iterative rounding process, i.e., iteratively remove vertices whose x_v is at least $V_{1/\alpha}$ until we solve the problem, we obtain a final solution of size at most αF. In fact, we conjecture that LP (1) has a basic optimal solution (see the exact definition in [4]) in which in each step we have at least one vertex v of $x_v \geqslant O(\frac{1}{\sqrt{n}})$, and thus this iterative rounding process also gives an $O(\sqrt{n})$-approximation.

Corollary 2.2. *If the total number of (simple) paths of length greater than r among the source-sink pairs is M, we have a multicut of size at most* $\max\{r, \ln(M)\}F$.

Proof. The proof follows from Theorem 2.1 (by removing a set of vertices whose deletion removes all path of length at most r), and then considering the multicut problem for the new instance as a set cover instance in which our objective is to remove a set of vertices which cover all $s-t$-paths of length at least r. It is known that the integrality gap of LP (1) which is now the relaxation of set cover is $\ln M$ (M is the number of elements or paths in our case.) □

We note that Corollary 2.2 says to obtain an integrality gap of $\Omega(n^\epsilon)$, the instance must has an exponential number of $s-t$ paths of length $\omega(\text{polylog}(n))$.

3 The problem might be hard even for special graphs

A directed graph G is called a *layered graph* if we can partition its vertices into at most $l \leqslant n$ sets $P_1, \dots, P_l$ such that for each edge $e = (v, u)$ of G where $v \in P_i$ and $u \in P_j$, we have $j = i + 1$.

Theorem 3.1. *For each layered graph G in which each source is in the first layer and each sink is in the last layer, we can obtain a multicut of size $O(n^{1/3})F$.*

Proof. By Gupta [3], we know that the size of a multicut is at most F^2 (his result is in terms of edge multicut to which we can easily reduce the vertex multicut). Now, if $F \leqslant n^{1/3}$, we have the desired multicut. In the other hand, if $F \geqslant n^{1/3}$, by Theorem 2.1, the number of layers must be at least $n^{1/3}$; otherwise we can obtain the result using the iterative rounding method. Now, by cutting a layer of minimum size, we have a cut of size at most $n^{2/3} \leqslant n^{1/3}F$, as desired. □

At the first glance, it seems that layered graphs are very special cases, however the following theorem says that they capture the hardness of the problem for directed acyclic graphs (DAGs).

Theorem 3.2. *If the integrality gap for DAGs is in $\Omega(n^\epsilon)$, then the integrality gap for layered graphs in which each source is in the first layer and each sink is in the last layer is in $\Omega(n^{\epsilon/3})$.*

Proof. The proof is simple. Consider the example of a DAG for which the integrality gap is in $\Omega(n^\epsilon)$. We construct a layered graph with the aforementioned properties as follows. First, we consider a topological ordering for the DAG and consider each vertex as a layer. Then we stretch each edge by adding additional vertices on it such that all new edges goes only between layers. Also, for each source which is not in the first layer, we introduce a node in the first layer with a stretched edge to the previous source (we do similarly for sinks). One can easily observe that a DAG instance of n vertices and m edges will be mapped to a layered graph instance of $O(nm)$ vertices and $O(nm)$ edges with desired properties. Since each cut in the latter graph corresponds to a cut in the former graph, the integrality gap of this new instance can not be in $o(n^{\epsilon/3})$. □

In the special case, any approximation factor of polylog(n) for layered graphs gives a polylog(n) approximation for DAGs that we believe they are as hard as general directed graphs.

The proof of the following Theorem is very similar to the proof of Theorem 4 of Rabani et al. [2].

Theorem 3.3. *For every directed graph G with an underlying H-minor-free graph (H is fixed), we can obtain a directed multicut of size at most $O(n^{1/4})F$.*

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

References

[1] C. Chekuri and S. Khanna, *Edge disjoint paths revisited*, in ACM-SIAM Symposium on Discrete Algorithms (SODA'03), 2003, pp. 628–637.

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

[3] A. Gupta, *Improved results for directed multicut*, in ACM-SIAM Symposium on Discrete Algorithms (SODA'03), 2003. 454-455.

[4] K. Jain, *A factor 2 approximation algorithm for the generalized Steiner network problem*, Combinatorica, 21 (2001), pp. 39–60.

[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*. Manuscript, July 2000.

[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.