An $O(\sqrt{n})$-Approximation Algorithm For Directed Sparsest Cut

Mohammad Taghi Hajiaghayi[*] *Harald Räcke*[†]

Abstract

We give an $O(\sqrt{n})$-approximation algorithm for the Sparsest Cut Problem on directed graphs. A naïve reduction from Sparsest Cut to Minimum Multicut would only give an approximation ratio of $O(\sqrt{n}\log D)$, where D is the sum of the demands. We obtain the improvement using a novel LP-rounding method for fractional Sparsest Cut, the dual of Maximum Concurrent Flow.

Keywords: approximation algorithms, directed graphs, sparsest cut, multicommodity flow.

1 Introduction

Suppose we are given a directed graph $G = (V, E)$ with an edge capacity function $c : E \to \mathbb{R}_{>0}$, and a collection of k source-target pairs $\{(s_i, t_i)\}_{i=1}^{k}$, where a positive demand $\mathrm{dem}(i)$ is associated with each of k pairs $(s_i, t_i) \in V \times V$. A *cut* $C \subseteq E$ is said to *separate* a pair (s_i, t_i), if there is a path from s_i to t_i in G and no path in $G' := (V, E-C)$. The *Sparsest Cut Problem* asks for a non-empty cut C that minimizes the ratio $c(C)/\mathrm{dem}(C)$, where $c(C) := \sum_{e\in C} c(e)$ denotes the capacity of all edges in the cut, and $\mathrm{dem}(C) := \sum\{\mathrm{dem}(i) : (s_i, t_i)$ is separated by $C\}$ denotes the total demand that is separated by the cut.

It is known that this problem has numerous applications in the design of approximation algorithms for NP-hard optimization problems in the areas of VLSI-design, routing, embedding, etc. (see [LR99, Shm97] for a survey). However, the situations of this problem for directed graphs and undirected graphs are quite different. While an algorithm with a polylogarithmic approximation ratio for the undirected case was obtained by Leighton and Rao [LR99] in 1988, and has been subsequently improved in a long series of papers (e.g. [ARV04, CGR05]) to an $O(\log^{3/4} k)$-approximation, the first non-trivial algorithms for the directed case are due to Cheriyan et al. [CKR01] in 2001. They consider the closely related *Minimum Multicut Problem*, where the goal is to find the minimum capacity cut that separates all source-target pairs, and they obtain an $O(\sqrt{n\log n})$-approximation algorithm. This result is further sharpened by Gupta [Gup03] who gives an $O(\sqrt{n})$-approximation. Recently, Agarwal, Charikar, Makarychev, and Makarychev [ACMM05] gave an $O(\sqrt{\log n})$ approximation algorithm for the directed

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

[†]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A., harry@cs.cmu.edu

sparsest cut problem for the special case of uniform demands (in the uniform demand case, there is the same demand between every pair of nodes). In undirected graphs such an approximation-ratio has been previously obtained in the breakthrough paper by Arora, Rao, and Vazirani [ARV04]. The reader is referred to [CLR05, HL03, KKN05, VV04] to see more recent results regarding (directed) multicut.

In this paper, using a novel LP-rounding method for fractional Sparsest Cut, we present an $O(\sqrt{n})$-approximation algorithm for the Sparsest Cut problem. In our LP rounding algorithm, we use some techniques of Cheryan et al. [CKR01] and Gupta [Gup03]. It is worth mentioning that all these results (Cheryan et al. [CKR01], Gupta [Gup03], this paper) not only give an approximation algorithm, but also bound the ratio between a solution to the cut problems (i.e., Sparsest Cut and Minimum Multicut) and the corresponding flow problems (Maximum Concurrent Multiflow and Maximum Multiflow, respectively). It has been shown by Saks et al. [SSZ04] that for directed graphs this ratio can be arbitrarily close to k, as opposed to undirected graphs where it is bounded by $O(\log k)$ (see [LR99]). However, in the construction of Saks et al. [SSZ04], $k = \Theta(\log n)$ and thus ratio $O(\log n)$ is not ruled out.

2 The algorithm

For completeness of exposition, we review some well-known facts about concurrent multicommodity flow and their relationship to the sparsest cut problem. In the concurrent multicommodity flow problem the goal is to establish for each source-target pair (s_i, t_i) a flow from s_i to t_i concurrently. This means that the *total flow* induced on an edge e by all the s_i-t_i flows must not exceed the the capacity of e. The goal is to maximize the *throughput* which is defined as the minimum taken over all commodities of the fraction of the commodity's demand that is met by the solution. Formally this is $\min_i \frac{\text{value(i)}}{\text{dem}(i)}$, where value$(i)$ denotes the value of the flow for the i-th commodity. The following LP gives the optimal throughput for a multicommodity flow instance (P_{uv} denotes the set of all directed paths from a node u to a node v in G).

$$\begin{array}{lll} \text{maximize} & \alpha & \\ \text{subject to} & \sum_{p \in P_{s_i t_i}} f_p \geq \alpha \cdot \text{dem}(i) & \forall \text{ pairs } (s_i, t_i) \\ & \sum_{p: e \in p} f_p \leq c(e) & \forall e \in E \\ & f_p \geq 0, \quad \alpha \geq 0 & \end{array} \tag{LP 1}$$

The dual of LP 1 is

$$\begin{array}{lll} \text{minimize} & \sum_{e \in E} c(e)\text{d}(e) & \\ \text{subject to} & \sum_{e \in p_{uv}} \text{d}(e) \geq x(u,v) & \forall (u,v) \in V \times V, \forall p_{uv} \in P_{uv} \\ & \sum_i x(s_i, t_i) \cdot \text{dem}(i) = 1 & \\ & \text{d}(e) \geq 0, \quad x(u,v) \geq 0 & \end{array} \tag{LP 2}$$

Note that the optimum throughput (the solution to the above LPs) is a lower bound on the minimum sparsity of a cut. Therefore the LPs are a relaxation to the sparsest cut problem in the sense that they give a lower bound on the minimum sparsity. Our main theorem is as follows.

Theorem 1 *There exists a polynomial time algorithm that computes a cut of sparsity $O(\sqrt{n})\cdot\phi$, where ϕ is the optimal value of LP 2.*

The following well-known interpretation of LP 2 forms the basis for our approximation algorithm. Informally speaking, the LP asks for a length-assignment d to the edges of G that creates a large distance between source-target pairs, while keeping the length of edges small. We introduce the following notation that measures how well a length-assignment achieves this goal. We define for a length-assignment d the *weight* $W(\mathrm{d})$ by $W(\mathrm{d}) := \sum c(e)\mathrm{d}(e)$. Further, we define the *demand-separation* $S_\Sigma(\mathrm{d})$ as $\sum_i \mathrm{dem}(i)\mathrm{d}(s_i,t_i)$, where $\mathrm{d}(s_i,t_i) := \min_{p\in P_{s_it_i}}\{\sum_{e\in p}\mathrm{d}(e)\}$ denotes the shortest-path distance between s_i and t_i induced by the length-assignment.[1] Note that if d is the optimum length-assignment for LP 2, the objective function value for the LP is $W(\mathrm{d})/S_\Sigma(\mathrm{d})$. Finally, we define the *max-demand-separation* $S_{\max}(\mathrm{d})$ by $S_{\max}(\mathrm{d}) := \sum_i \mathrm{dem}(i)$ $\min_{p\in P_{s_it_i}}\{\max_{e\in p}\mathrm{d}(e)\}$. The max-demand-separation differs from the demand-separation by the "length" associated with an s_i-t_i path p. In the definition of demand-separation this length is the sum of all edge-lengths in p, while in the definition of max-demand-separation it is the maximum length of an edge. Therefore, $S_{\max}(\mathrm{d}) \le S_\Sigma(\mathrm{d})$ holds for all length-assignments d.

For another way to see that LP 2 is a relaxation to the sparsest cut problem, let for a cut C, d_C denote the length assignment that assigns a length of $1/\mathrm{dem}(C)$ to each edge in the cut, and choose $x(u,v)$ as the shortest path distance $\mathrm{d}_C(u,v)$ between u and v with respect to d_C. Then, $\sum_i \mathrm{dem}(i)x(s_i,t_i) = \sum_i \mathrm{dem}(i)\mathrm{d}_C(s_i,t_i) = \sum\{\mathrm{dem}(i)/\mathrm{dem}(C) : (s_i,t_i) \text{ is separated by } C\} = 1$, and $\sum c(e)\mathrm{d}_C(e) = \mathrm{cap}(C)/\mathrm{dem}(C)$. This shows that there is a feasible solution to LP 2 with objective function value $\mathrm{cap}(C)/\mathrm{dem}(C)$, i.e., the optimum length-assignment d for LP 2 has weight not larger than the sparsity of the cut C.

Our approximation algorithm aims at rounding the length-assignment d that results from solving LP 2 to a length-assignment that corresponds to a cut (i.e., where all edges in the cut have equal length and all other edges have length zero). This is done in several steps. We first show that we can obtain an assignment ℓ that has not only a large demand-separation (like d) but also a large max-demand-separation. Note that a length-assignment is simply a length function on the edge set E. The new length-assignments that we generate are not required to fulfill the constraints of LP 2.

Lemma 2 *For any length assignment* d *with weight $W(\mathrm{d})$ and demand separation $S_\Sigma(\mathrm{d})$, there is a length assignment ℓ with weight $W(\ell) = O(\sqrt{n}\cdot W(\mathrm{d}))$, and max-demand-separation $S_{\max}(\ell) \ge S_\Sigma(\mathrm{d})$.*

Proof. The following proof is similar to the proof of Gupta [Gup03] that gives an $O(\sqrt{n})$-approximation to minimum multicut in directed graphs. There the goal is to round a length-assignment (which is a relaxation to the minimum multicut) into a multicut, i.e., to determine a

[1] We use this slight abuse of notation throughout the paper. For a length-assignment $\mathrm{d} : E \to \mathbb{R}_{\ge 0}$, $\mathrm{d}(u,v)$ is used to denote the induced shortest-path distance from node u to v in G.

set of edges that separates every source target pair. Our goal is to round the length-assignment d into a new length-assignment ℓ in such a way that a source-target pair at distance $\mathrm{d}(s_i, t_i)$ is separated when removing all edges with $\ell(e) \geq \mathrm{d}(s_i, t_i)$ (these edges form a separating cut for the source-target pair). This different objective creates a few technical differences to the proof in [Gup03].

We assume that the source-target pairs (s_i, t_i) are sorted in decreasing order of the distance $\mathrm{d}(s_i, t_i)$ assigned to them by the optimal solution (ties are broken arbitrarily). The following algorithm constructs the new length-assignment. Start with an assignment that assigns length 0 to all edges. In a first phase identify all edges e that lie on some s_i-t_i path and have $\mathrm{d}(e) \geq \frac{1}{\sqrt{n}}\mathrm{d}(s_i, t_i)$. Set the length of these edges to $\mathrm{d}(s_i, t_i)$ (if an edge fulfills the condition for several pairs set its length to the maximum distance among these pairs).

In a second phase the algorithm considers all source-target pairs, ordered by decreasing $d(s_i, t_i)$, in a row (starting with the first pair), and ensures that $\min_{p \in P_{s_i,t_i}}\{\max_{e \in p} \ell(e)\} \geq \mathrm{d}(s_i, t_i)$, by increasing the length of some edges to $\mathrm{d}(s_i, t_i)$. Let E_i denote the set of edges whose length is increased for the *i-th* pair. E_i is determined as follows. Let H_i denote the subgraph of G that contains all s_i-t_i paths for which currently (when considering the *i-th* pair) no edge e on the path has $\ell(e) \geq \mathrm{d}(s_i, t_i)$. Let $W_{H_i}(\mathrm{d}) := \sum_{e \in H_i} c(e)\mathrm{d}(e)$.

The rest of the proof is identical to Gupta's proof, which we restate for completeness of exposition. Define a level cut by deleting, for a parameter r, all edges (u, v) for which $\mathrm{d}(s_i, u) \leq r$ and $\mathrm{d}(s_i, v) \geq r$, i.e., edges that "contain" a point in distance r from s_i. The algorithm chooses E_i as the set of edges in the minimum capacity level cut with $r \in [\frac{1}{3}\mathrm{d}(s_i, t_i), \frac{2}{3}\mathrm{d}(s_i, t_i)]$, i.e., the algorithm searches for a cut in which all separated edges are far away (with respect to the original distance function d) from the terminal nodes s_i and t_i. An averaging argument shows that the capacity of edges in E_i is at most $3W_{H_i}(\mathrm{d})/\mathrm{d}(s_i, t_i)$ (To see this suppose that r is chosen uniformly at random from the interval $[\frac{1}{3}\mathrm{d}(s_i, t_i), \frac{2}{3}\mathrm{d}(s_i, t_i)]$. Then the expected weight of edges cut is at most $3W_{H_i}(\mathrm{d})/\mathrm{d}(s_i, t_i)$ because the probability that an edge e of length $\mathrm{d}(e)$ is cut is less than $\frac{\mathrm{d}(e)}{\frac{1}{3}\mathrm{d}(s_i,t_i)}$). We have that in the *i-th* round the algorithm increases the weight of the length assignment ℓ by at most $3W_{H_i}(\mathrm{d})$ by setting the length of edges in E_i to $\mathrm{d}(s_i, t_i)$.

Now we show that the final weight of the length-assignment ℓ is at most $O(\sqrt{n} \cdot W(\mathrm{d}))$. Note that edges whose ℓ-length is increased in the first phase of the algorithm do not contribute more than $\sqrt{n} \cdot W(\mathrm{d})$ to the weight of ℓ, since for each such edge $\ell(e) \leq \sqrt{n} \cdot \mathrm{d}(e)$. In order to bound the contribution by edges selected in the second phase we show that $\sum_i W_{H_i}(\mathrm{d}) \leq O(\sqrt{n} \cdot W(\mathrm{d}))$.

Let for a subgraph H_i, $S_i \subset H_i$ denote the subgraph spanned by nodes that can be reached from s_i after the cut-edges E_i are removed. Similarly, let T_i be the subgraph of nodes that can reach t_i after deleting edge set E_i.

We claim that a graph edge e can only belong to $O(\sqrt{n})$ different subgraphs S_i, and $O(\sqrt{n})$ different subgraphs T_i. Furthermore, it can only belong to at most one cut E_i, since for all subsequent pairs the length of the edge (now, $\ell(e) = \mathrm{d}(s_i, t_i)$) is so large that e is not considered anymore (because the source-target pairs are considered in decreasing order of $\mathrm{d}(s_i, t_i)$). This means that the weight of an edge (i.e., $c(e)\mathrm{d}(e)$) only contributes to at most $O(\sqrt{n})$ subgraphs H_i. This yields $W(\ell) \leq \sqrt{n} \cdot W(d) + \sum_i 3W_{H_i}(\mathrm{d}) \leq O(\sqrt{n} \cdot W(\mathrm{d}))$, where in the second expression, the first term is because of the increment in the weight in the first phase and the

second term is because of the increment in the second phase.

We derive a bound on the number of subgraphs S_i containing e, as follows. Suppose, $e \in S_i$, for some i. This means that e must lie on some s_i-t_i path because of the definition of H_i. Denote this path with $p_i(e)$ and let $Q_i(e)$ denote the nodes on the path that lie in T_i. The cut E_i is at distance at most $\frac{2}{3}\mathrm{d}(s_i,t_i)$ from s_i. Therefore, the edges in $p_i(e)$ that have at least one endpoint in T_i must have total length at least $\frac{1}{3}\mathrm{d}(s_i,t_i)$. However, none of these edges has length larger than $\frac{1}{\sqrt{n}}\mathrm{d}(s_i,t_i)$, as it was not considered in the first phase of the algorithm. Hence, there must exist at least $\Omega(\sqrt{n})$ of these edges and therefore $|Q_i(e)| = \Omega(\sqrt{n})$.

Let (s_j,t_j) denote a subsequent terminal pair for which $e \in S_j$. The set E_i forms a cut not only between s_i and t_i but also between edge e and the nodes in T_i. Therefore the path $p_j(e)$ between s_j and t_j does not contain any node from T_i after edge e, because otherwise it would contain an edge from E_i which is impossible because all edges on the path have currently length strictly less than $d(s_j,t_j) \le d(s_i,t_i)$. Therefore, $Q_i(e) \cap Q_j(e) = \emptyset$. Since, $\sum_{i:e\in S_i} |Q_i(e)| \le n$ and $|Q_i(e)| = \Omega(\sqrt{n})$, we get that an edge is at most in $O(\sqrt{n})$ different subgraphs S_i. The argument for subgraphs T_i is analogous. This completes the proof. ∎

Now we show that any length-assignment with a large max-demand-separation can be rounded to a cut with low sparsity. Combining this lemma with Lemma 2 gives an $O(\sqrt{n})$-approximation algorithm for the Sparsest Cut Problem on directed graphs.

Lemma 3 *Given a length assignment ℓ with $S_{\max}(\ell) > 0$, there is a polynomial time algorithm that finds a cut with sparsity at most $W(\ell)/S_{\max}(\ell)$.*

Proof. Consider a length assignment ℓ that has a max-demand-separation $S_{\max}(\ell)$. We group the edges of G into classes $E_1,\dots,E_k$ according to their length in decreasing order. This means that all edges in E_1 have maximum length, and edges in E_k have minimum non-zero length under the length assignment ℓ. Let for $i \in \{1,\dots,k\}$, ℓ_{E_i} denote the length of edges in the respective class E_i. Further, let T_i denote the source-target pairs that are separated by the cut $\cup_{j=1}^{i} E_i$ in G.

What happens if we scale the length of all edges in E_1 down to ℓ_{E_2}? The weight of the length-assignment changes by $\Delta W := \sum_{e\in E_1}(\ell_{E_2} - \ell_{E_1})c(e)$, while the max-demand-separation changes by $\Delta S := \sum_{i\in T_1}(\ell_{E_2} - \ell_{E_1})\mathrm{dem}(i)$. If $\Delta W/\Delta S > W(\ell)/S_{\max}(\ell)$ we scale all edges in E_1 down to length ℓ_{E_2} and obtain a new length assignment with smaller $W(\ell)/S_{\max}(\ell)$-ratio.

We can repeat this process until we obtain a length-assignment ℓ for which $\Delta W/\Delta S \le W(\ell)/S_{\max}(\ell)$ (This process terminates since in each round the number of edge-classes decreases by 1. Furthermore if the number of classes (i.e., the number of different edge-lengths) is reduced so far that only one non-zero edge length remains, the condition $\Delta W/\Delta S \le W(\ell)/S_{\max}(\ell)$ is fulfilled). If $\Delta W/\Delta S \le W(\ell)/S_{\max}(\ell)$ the cut E_1 has sparsity at most $\Delta W/\Delta S \le W(\ell)/S_{\max}(\ell)$, as desired. ∎

Combining Lemma 2 and Lemma 3 gives Theorem 1.

Proof of Theorem 1. Let d denote the optimum length assignment for LP 2. Then the objective function value is $W(\mathrm{d})/S_\Sigma(\mathrm{d})$. Since LP 2 is a relaxation to the sparsest cut

problem we have $W(\mathrm{d})/S_\Sigma(\mathrm{d}) \le \phi$. Applying Lemma 2 gives us a length-assignment ℓ with $W(\ell)/S_{\max}(\ell) \le O(\sqrt{n}) \cdot \frac{W(\mathrm{d})}{S_\Sigma(\mathrm{d})} \le O(\sqrt{n}) \cdot \phi$. Now applying Lemma 3 gives us a cut C with sparsity $\mathrm{cap}(C)/\mathrm{dem}(C)$ less than $\sqrt{n} \cdot \phi$. This means we have an $O(\sqrt{n})$-approximation. ■

Acknowledgement: We thank two anonymous referees for their helpful comments.

References

[ACMM05] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $O(\sqrt{\log n})$-approximation algorithms for Min Uncut, Min 2CNF Deletion, and directed cut problems. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC)*, pages 573–581, 2005.

[ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings, and graph partitionings. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, pages 222–231, 2004.

[CGR05] Shuchi Chawla, Anupam Gupta, and Harald Räcke. An improved approximation to sparsest cut. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 102–111, 2005.

[CKR01] Joseph Cheriyan, Howard Karloff, and Yuval Rabani. Approximating directed multicuts. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 320–328, 2001.

[CLR05] Marie-Christine Costa, Lucas Létocart, and Frédéric Roupin. Minimal multicut and maximal integer multiflow: A survey. *European Journal of Operational Research*, 162(1):55–59, 2005.

[Gup03] Anupam Gupta. Improved results for directed multicut. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 454–455, 2003.

[HL03] Mohammad Taghi Hajiaghayi and Frank Thomson Leighton. On the max-flow min-cut ratio for directed multicommodity flows. Technical Report MIT-LCS-TR-910, Massachusetts Institute of Technology, Cambridge, MA, USA, July 2003.

[KKN05] Yana Kortsarts, Guy Kortsarz, and Zeev Nutov. Greedy approximation algorithms for directed multicuts. *Networks*, 45(4):214–217, 2005.

[LR99] Frank Thomson Leighton and Satish B. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[Shm97] David B. Shmoys. Cut problems and their application to divide-and-conquer. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 192–235. PWS Publishing, 1997.

[SSZ04] Michael Saks, Alex Samorodnitsky, and Leonid Zosin. A lower bound on the integrality gap for minimum multicut in directed networks. *Combinatorica*, 24(3):525–530, 2004.

[VV04] Kasturi R. Varadarajan and Ganesh Venkataraman. Graph decomposition and a greedy algorithm for edge-disjoint paths. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 379–380, 2004.