1 Introduction to Sparsest Cut

Sparsest Cut is a fundamental problem in graph algorithms with many applications and connections. There are several variants that are considered in the literature and they are closely related but it is useful to have proper terminology and understand the similarities and differences.

Non-Uniform Sparsest Cut: We consider the general one first. The input is a graph $G = (V, E)$ with non-negative edge capacities $c : E \to \mathbb{R}_+$ and a set of pairs $(s_1, t_1), ..., (s_k, t_k)$ along with non-negative demand values $D_1, D_2, \ldots, D_k$. When considering undirected graphs the demand pairs are unordered — by this we mean that we do not distinguish (s_1, t_1) from (t_1, s_1). One can also think of the demand values as "weights" but the demand terminology makes more sense when considering the dual flow problem. Given a set/cut $S \subseteq V$ the *sparsity* of the cut S is defined as the ratio $\frac{c(\delta(S))}{(\sum_{i:S\cap\{s_i,t_i\}=1} D_i)}$. The numerator is the capacity of the cut and the denominator is the total demand of the pairs separated by S. The goal is to find the cut S with minimum sparsity. In other words we are trying to find the best "bang per buck" cut: how much capacity do we need to remove per amount of demand separated? It is sometime convenient to consider G as the supply graph and the demands as forming a demand graph $H = (V, F)$ where F represents the pairs and we associate $D : F \to \mathbb{R}_+$ to represent the demand value (alternatively we can also consider multigraphs). With this representation of the demand pairs, the sparsity of cut S is simply $\frac{c(\delta_G(S))}{D(\delta_H(S))}$ note that $\delta_G(S)$ represents the supply edges crossing S and $\delta_H(S)$ represents the demand edges crossing the cut S.

Remark 1. *One can define a cut as removing a set of edges. This may lead to more than two components. In the case of sparsest cut in undirected graphs it suffices to restrict attention to cuts of the form $\delta(S)$ for some $S \subseteq V$. It is a useful exercise to see why there is always a sparsest cut of that form for any given instance. This is not necessarily true for directed graphs or even in undirected graphs with node-weights or in hypergraphs.*

Uniform Sparsest Cut: Very often when people say Sparsest Cut they mean the uniform version. This is the version in which $D(u, v) = 1$ for each unordered pair of vertices (u, v). For these demands the a cut S is $\frac{c(\delta_G(S))}{|S||V \setminus S|}$. Alternatively the demand graph H is a complete graph with unit demand values on each edge. A slightly generalization of Uniform Sparsest Cut is obtained by considering demands induced by weights on vertices (the dual flow instances are called Prodcut Multicommodity Flow instances). There is a weight function $\pi : V \to \mathbb{R}_+$ on the vertices and and demand $D(u, v)$ for pair (u, v) is set to be $\pi(u)\pi(v)$. Note that if $\pi(u) = 1$ for all u then we obtain Uniform Sparsest Cut. If $\pi(u) \in \{0, 1\}$ for all u then we are focusing our attention on sparsity with respect to the set $V' = \{v \mid \pi(v) = 1\}$ since the the vertices with $\pi(u) = 0$ play no role. This may seem unnatural at first but it is closely connected to expansion and conductance as we will see below.

EXPANSION: The expansion of a multi-graph $G = (V, E)$ is defined as $\min_{S:|S|\leq|V|/2} \frac{|\delta(S)|}{|S|}$. Recall that G is an α-expander if the expansion of G is at least α. A random 3-regular graph is α-expander with $\alpha = \Omega(1)$ with high probability. Thus, to find an α-expander one can obtain an efficient randomized algorithm by picking a random graph and then verifying its expansion. However, checking expansion is coNP-Hard. EXPANSION is closely related to UNIFORM SPARSEST CUT. Note that when $|S| \leq |V|/2$ we have

$$\frac{1}{|V|}\frac{|\delta(S)|}{|S|} \leq \frac{|\delta(S)|}{|S||V \setminus S|} \leq \frac{2}{|V|}\frac{|\delta(S)|}{|S|}.$$

Thus EXPANSION and UNIFORM SPARSEST CUT are within a factor of 2 of each other. Sometimes it is useful to consider expansion with vertex weights $w : V \to \mathbb{R}_+$. Here the expansion is defined as $\min_{S:w(S)\leq w(V)/2} \frac{|\delta(S)|}{w(S)}$. This corresponds to product multicommodity flow instances where $\pi(v) = w(v)$. The term CONDUCTANCE is used to denote the quantity $\frac{|\delta(S)|}{\text{vol}(S)}$ where $\text{vol}(S) = \sum_{v\in S} \deg(v)$ (here vol is short for volume). When a graph is regular the definition of expansion and conductance are the same but not in the general setting. Note that we can capture conductance by setting weights on vertices where $w(v) = \deg(v)$.

Some key applications: UNIFORM SPARSEST CUT is fundamentally interesting because it helps us directly and indirectly solve the BALANCED SEPARATOR problem. In the latter problem we want to partition $G = (V, E)$ into two pieces $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ where $|V_1|$ and $|V_2$ are roughly the same size so that we minimize the number of edges between V_1 and V_2. One can repeatedly use a sparse cut routine to get an approximately balanced separator. The other key application is to certify expansion of a graph. Expander graphs and relatives arise in many many applications and knowing whether a graph is expanding or not is very useful — a well-known survey is by Hoory, Liniar and Wigderson [HLW06].

2 LP Relaxation and Maximum Concurrent Flow

How do we write an LP relaxtion for SPARSEST CUT? This is less obvious than it is for MULTICUT and other cut problems where we have explicit terminal pairs that we wish to separate. We consider NON-UNIFORM SPARSEST CUT. First we develop an integer program. We have two sets of variables. For each pair (s_i, t_i) we have a variable y_i to indicate whether we want to separate the pair i. For each edge we have a variable x_e to indicate whether e is cut. If we decide to separate pair i then for every path between s_i and t_i we should cut at least one edge on the path — this is similar to relaxations we have seen before. We let $\mathcal{P}_{s_i,t}$ be the set of all s_i-t_i paths. The following captures the problem:

$$\min \frac{\sum_{e\in E} c_e x_e}{\sum_{i=1}^{k} D_i y_i}$$
$$\begin{aligned} \sum_{e\in p} x_e &\geq y_i \quad p \in \mathcal{P}_{s_i,t_i}, i \in [k] \\ x_e &\in \{0,1\} \quad e \in E \\ y_i &\in \{0,1\} \quad i \in [k] \end{aligned}$$

Note, however, that the objective is a ratio and not linear. It is a standard trick to obtain an LP relaxation wherein we normalize the denominator in the ratio to 1 and relax the variables to be real-valued. Thus we obtain the following LP relaxation.

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & & \\ \sum_{i=1}^{k} D_i y_i &= 1 & \\ \sum_{e \in p} x_e &\geq y_i & p \in \mathcal{P}_{s_i,t_i}, i \in [k] \\ x_e &\geq 0 & e \in E \\ y_i &\geq 0 & i \in [k] \end{aligned}$$

Exercise 1. *Show that the LP is indeed a relaxation for the* Sparsest Cut *problem. Formally, given an integer feasible solution with sparsity λ find a feasible solution to the relaxation such that its value is no more than λ.*

Now we consider the dual LP. For each path $p \in \cup_i \mathcal{P}_{s_i,t_i}$ there is a non-negative variable y_p which is the amount of "flow" sent on path p. There is a variable λ that we will interpret later.

$$\begin{aligned} \max \lambda & & \\ \sum_{p \in \mathcal{P}_{s_i,t_i}} y_p &\geq \lambda D_i & i \in [k] \\ \sum_{i=1}^{k} \sum_{p \in \mathcal{P}_{s_i,t_i}, e \in p} y_p &\leq c_e & e \in E \\ y_p &\geq 0 & p \in \mathcal{P}_{s_i,t_i}, i \in [k] \end{aligned}$$

The dual LP is a multicommodity flow. It solves the Maximum Concurrent Multicommodity Flow problem for the given instance. It finds the largest value of λ such that there is a feasible multicommodity flow for the given pairs in which the flow routed for pair (s_i, t_i) is at least λD_i. It is called *concurrent flow* since we need to route all demand pairs to the same factor which is in constrast to the dual of Multicut which corresponds to the maximum throughput multicommodity flow (in which some pairs may have zero flow while others have a lot of flow).

Exercise 2. *Suppose we have a cut S with sparsity $c(\delta(S))/(\sum_{i:S \cap \{s_i,t_i\}=1} D_i)$. Why is the maximum concurrent flow at most the sparsity of S?*

Note that the LP can be solved via the Ellipsoid method. One can also write a compact LP via distance variables which will help us later to focus on constraining the metric in other ways.

$$\begin{aligned} \min \sum_{uv \in E} c(uv) d(uv) & \\ \sum_{i=1}^{k} D_i d(s_i t_i) &= 1 \\ d \text{ is a metric on } V & \end{aligned}$$

Flow-cut gap: The flow-cut gap in this context is the following equivalent way of thinking about the problem. Consider a multicommodity flow instance on G with demand pairs $(s_1, t_1), \ldots, (s_k, t_k)$ and demand values $D_1, \ldots, D_k$. Suppose G satisfies the *cut-condition*, that is, for every $S \subseteq V$ the capacity $c(\delta(S))$ is at least the demand separated by S. Can we route all the demand pairs? This is true when $k = 1$ but is not true in general even for $k = 3$ in undirected graphs. The question is the maximum value of λ such that we can route λD_i for every pair i? The worst-case integrality gap of the preceding LP relaxation for SPARSEST CUT is precisely the flow-cut gap. One can ask about the flow-cut gap for all graphs, a specific class of graphs, for a specific class of demand graphs, a specific class of supply and demand graphs, and so on.

In these notes we will establish that the flow-cut gap in general undirected graphs is at most $O(\log k)$. And there are instances where the gap is $\Omega(\log k)$ which are uniform instances — in fact the same expander based example we saw for MULTICUT shows that the gap is $\Omega(\log n)$ even for UNIFORM SPARSEST CUT. It is conjectured that the gap is $O(1)$ for planar graphs but the best upper bound we have is $O(\sqrt{\log n})$. Resolving the flow-cut gap in planar graphs is a major open problem.

Exercise 3. *Use the expander construction that we saw for* MULTICUT *to show that the flow-cut gap for* UNIFORM SPARSEST CUT *can be* $\Omega(\log n)$*.*

Remark 2. *Approximating the* SPARSEST CUT *problem is not the same as establishing flow-cut gaps. One can obtain improved approximations for* SPARSEST CUT *via stronger relaxations than the natural LP. Indeed the best approximation ratio for* SPARSEST CUT *is* $O(\sqrt{\log n})$ *via an SDP relaxation.*

3 Rounding LP via Connection to Multicut

There are close connections between SPARSEST CUT and MULTICUT. By repeatedly using SPARSEST CUT routine and SET COVER style analysis prove the following.

Exercise 4. *Suppose there is an* $\alpha(k, n)$*-approximation for* NON-UNIFORM SPARSEST CUT*. Prove that this implies an* $O(\alpha(k, n) \ln k)$*-approximation for* MULTICUT*.*

Can we prove some form a converse? That is, can we use an approximation algorithm for MULTICUT to obtain an approximation algorithm for SPARSEST CUT? Note that if someone told us the pairs to separate in an optimum solution to the SPARSEST CUT instance then we can use an (approximation) algorithm for MULTICUT to separate those pairs. Here we show that one can use information from the LP solution to figure out which pairs to separate. We sketch the argument and focus our attention on the simpler case when $D_i = 1$ for all $i \in [k]$. We give this argument even though it does not lead to the optimum ratio, for historical interest, as well as to illustrate a useful high-level technique that has found applications in other settings.

Identifying the pairs to separate from LP solution: Suppose we solve the LP and obtain a feasible solution (x, y). y_i indicates the extent to which pair i is separated. Suppose we have an ideal situation where $y_i \in \{0, p\}$ for every i. Let $A = \{i \mid y_i = p\}$. We have $|A| = 1/p$ since $\sum_i y_i = 1$. Then it is intutively clear that the LP is separating the pairs in A. We can then solve the Multicut problem for the pairs in A and consider the ratio of the cost of the cut to $|A|$. How do we argue about this algorithm? We do the following. Consider a fractional assignment $x' : E \to \mathbb{R}_+$ where $x'_e = \min\{1, x_e/p\}$; in other words we scale each x_e by $1/p$. Note that $y_i = d_x(s_i, t_i)$. Since we scaled up each x_e by $1/p$ it is not hard to see that $d_{x'}(s_i, t_i) \geq 1$; in other words x' is a feasible solution to the Multicut instance on G for the pairs in A. The fractional cost of x' is $\sum_e c_e x'_e \leq \sum_e c_e x_e/p$. Thus, by the algorithm for Multicut in the previous chapter, we can find a feasible Multicut $E' \subseteq E$ that separates all pairs in A and $c(E') = O(\log k) \sum_e c_e x_e/p$. What is the sparsity of this cut? It is $c(E')/|A|$ which is $O(\log k) \sum_e x_e$. Thus the sparsity of the cut is $O(\log k)\lambda$ where λ is the value of the LP relaxation.

Now we consider the general setting. Recall that $\sum_i y_i = 1$. We partition the pairs into groups that have similar y_i values. For $j \geq 0$, let $A_j = \{i \mid y_i \in (1/2^{j+1}, 1/2^j]$. Thus all pairs in A_j have a y_i value that are within a factor of 2 of each other.

Claim 3. *There exists a $j \leq \log_2 k$ such that $\sum_{i \in A_j} y_i \geq \frac{1}{2(1+\log_2 k)} \geq \frac{1}{4 \log k}$.*

Proof. Consider any i such that $i \in A_j$ where $j > \log_2 k$. By definition we have $y_i \leq 1/2k$. Since there are only k pairs, $\sum_{j > \log_2 k} \sum_{i \in A_j} y_i \leq k/2k \leq 1/2$. Thus $\sum_{j \leq \log_2 k} \sum_{i \in A_j} y_i \geq 1/2$ and therefore, there must be a $j \leq \log_2 k$ such that $\sum_{i \in A_j} y_i \geq \frac{1}{2(1+\log_2 k)}$ (there are only so many groups). ■

Consider the A_j with $\sum_{i \in A_j} y_i \geq \frac{1}{4 \log_2 k}$. For each $i \in A_j$ we have $1/2^{j+1} \leq y_i \leq 1/2^j$. Therefore $|A|_j \geq \min\{1, 2^j/4 \log_2 k\}$. The algorithm now separates the pairs in A_j via an algorithm for Multicut.

Claim 4. *Consider the fractional solution $x' : E \to [0, 1]$ where $x'_e = \min\{1, 2^{j+1} x_e\}$. Then $d_{x'}(s_i, t_i) \geq 1$ for all $i \in A_j$. Thus x' is a feasible fractional solution to the Multicut LP for separating the pairs in A_j.*

Via the rounding algorithm in the preceding chapter we have there is a set $E' \subseteq E$ such that E' is a feasible multicut for the pairs in A_j and $c(E') = O(\log k) 2^{j+1} \sum_e c_e x_e$. The sparsity of this cut is $c(E')/|A_j| = O(\log^2 k) \sum_e c_e x_e$. Thus we obtained an $O(\log^2 k)$-approximation for Sparsest Cut when $D_i = 1$ for each pair.

Remark 5. *When demands are not* 1 *(or identical) the preceding argument yields an $O(\log k \log D)$ approximation where $D = \sum_i D_i$ with the normalization that $D_i \geq 1$ for all i.*

4 Rounding via ℓ_1 embeddings

Leighton and Rao [LR99], in their seminal work, obtained an $O(\log n)$ approximation and flow-cut gap for Uniform Sparsest Cut and showed many applications. They also showed a lower bound of $\Omega(\log n)$ on the flow-cut gap for Uniform Sparsest Cut via expanders which was also an important connection. This led to $O(\log^2 n)$-approximation for Sparsest Cut and it was an open problem to obtain a tight conjectured bound of $O(\log n)$. The optimal rounding of the LP relaxation turns out to go via metric embedding theory and this connection was discovered by Linial, London and Rabinovich [LLR95] and Aumann and Rabani [AR98]. We need some basics in metric embeddings to point out the connection and rounding. Even though the metric embedding machinery is powerful, it can seem like magic. The more basic ideas for Uniform Sparsest Cut based on region growing is useful to know. One can find the details in [LR99] or in a more accessible form in the book of Williamson and Shmoys [WS11].

4.1 A digression through trees

It is instructive to consider the simple setting when G is a tree $T = (V, E)$. In this case it is easy to find the sparsest cut. For each edge $e \in T$ we can associate a cut S_e which is one side of the two components in $T - e$. The capacity of the cut $\delta(S_e)$, by defintion, is c_e. Let $D(e) = \sum_{i:|S_e \cup \{s_i,t_i\}|=1} D_i$ be the demand separated by e. The sparsity of the cut S_e is simply c_e/D_e. Finding sparsest cut in a tree is easy from the following exercise.

Exercise 5. *The sparsest cut in a tree is given by* $\arg\min_e c_e/D_e$.

A more interesting exercise is to prove that the LP relaxation gives an optimum solution on a tree.

Lemma 6. *Let (x, y) be a feasible solution to the LP with objectie value λ. If G is a tree T then there is an edge $e \in T$ such that $c_e/D_e \leq \lambda$.*

Proof. We have $\lambda = \frac{\sum_e c_e x_e}{\sum_i D_i d_x(s_i,t_i)}$ where $d_x(s_i, t_i)$ is the shortest path distance between s_i and t_i. There is a unique path P_{s_i,t_i} from s_i to t_i in a tree so $d_x(s_i, t_i) = \sum_{e \in P_{s_i,t_i}} x_e$. Thus,

$$\begin{aligned}
\lambda &= \frac{\sum_e c_e x_e}{\sum_i D_i d_x(s_i, t_i)} \\
&= \frac{\sum_e c_e x_e}{\sum_i D_i \sum_{e \in P_{s_i,t_i}} x_e} \\
&= \frac{\sum_e c_e x_e}{\sum_e D_e x_e} \\
&\geq \min_e \frac{c_e}{D_e}.
\end{aligned}$$

In the last inequality we are using the simple fact that $\frac{a_1+a_2+\ldots+a_n}{b_1+b_2+\ldots+b_n} \geq \min_i \frac{a_i}{b_i}$ for positive a's and b's. ∎

What made the proof work for trees? Is there a more general phenomenon than the fact that trees are pretty simple structures? It turns out that the key fact is that shortest path distances induced by a tree are ℓ_1 metrics or equivalently cut metrics.

4.2 Cut metrics, line metrics, and ℓ_1 metrics

Let (V, d) be a finite metric space. We will be interested in two special types of metrics.

Definition 1. *Let V be a finite set and let $S \subseteq V$. The metric d_S associated with the cut S is the following: $d_S(u, v) = 1$ if $|S \cap \{u, v\}| = 1$ and $d_S(u, v) = 0$ otherwise.*

Definition 2. *Let (V, d) be a finite metric space. The metric d is a* cut metric *if there is a set $S \subset V$ such that $d = d_S$. d is in the cut cone (or in the cone of cut metrics) if there exist non-negative scalars $y_S, S \subset V$ such that $d(u, v) = \sum_{S \subset V} y_S d_S(u, v)$ for all $u, v \in V$.*

Definition 3. *Let (V, d) be a finite metric space. The metrid d is a* line metric *if there is a mappting $f : V \to \mathbb{R}$ (the real line) such that $d(u, v) = |f(u) - f(v)|$ for all $u, v \in V$.*

Definition 4. *Let (V, d) be a finite metric space. The metric d is an ℓ_1 metric*[1] *if there is some integer d and a mapping $f : V \to \mathbb{R}^d$ (Euclidean space in d dimensions) such that $d(u, v) = |f(u) - f(v)|_1$ (the ℓ_1 distance) for all $u, v \in V$.*

Claim 7. *A metric (V, d) is an ℓ_1 metric iff it is a non-negative combination of line metrics (in the cone of line metrics).*

Proof Sketch. If d is an ℓ_1 metric then each dimension corresponds to a line metric and since the ℓ_1 metric is separable over the dimensions it is a non-negative combination of line metric. Conversely, any non-negative combination of line metrics can be made into an ℓ_1 metric where each line metric becomes a separate dimension (scalar multiplication of a line metric is also a line metric). ■

Lemma 8. *d is an ℓ_1 metric iff d is in the cut cone.*

Proof. Consider the metric d_S. It is easy to that it is a simple line metric. Map all vertices in S to 0 and all vertices in $V - S$ to 1. If d is in the cut cone then it is a non-negative combination of the cut metrics, and hence it is a non-negative combination of line metrics, and hence an ℓ_1 metric.

To prove the converse, it suffices to argue that any line metric is in the cut cone. Let $V = \{v_1, v_2, \ldots, v_n\}$ and let d be a line metric on V. Without loss of generality assume that the coordinates of the points corresponding to the line metric d are $x_1 \leq x_2 \leq x_n$ on the real line. For $1 \leq i < n$ let $S_i = \{v_1, v_2, \ldots, v_i\}$. It is not hard to verify that $\sum_{i=1}^{n-1} |x_{i+1} - x_i| d_{S_i} = d$. ■

4.3 Brief introducton to metric embeddings

Let (V, d) me a finite metric space. Note that any finite metric space can be viewed as one that is derived from the shortest path metric induced on a graph with some non-negative edge lengths. If $G = (V, E)$ is a simple graph and $\ell : E \to \mathbb{R}_+$ are some edge-lengths, the metric induced on V depends both on the "topology" of G as well as the lengths. Finite metrics can encode graph structure and hence can be diverse. When trying to round we may want to work with simpler metric spaces. One way to do this is to *embed* a given metric space (V, d) into a simpler *host* metric space (V', d'). An embedding is simply a mapping of V to V'. Even though we may be interested in finite metric spaces, the host metric space can be continuous/infinite such as the Euclidean space in some dimenstion d. Embedding typically *distorts* the distances and thus one wants to find embeddings with small distortion. We will focus on relative notion of distortion; additive notions are also explored in the literature although they are very restrictive due to lack of scale invariance.

Definition 5. *An embedding of a finite metric space (V, d) to a host metric space (V', d') is a mapping $f : V \to V'$. The embedding is an* isometric embedding *if $d(u, v) = d'(f(u), f(v))$ for all $u, v \in V$. An embedding is a contraction if $d'(f(u), f(v)) \leq d(u, v)$ for all $u, v \in V$. An embedding is non-contracting if $d'(f(u), f(v)) \geq d(u, v)$ for all $u, v \in V$.*

Definition 6. *Let (V, d) and (V', d') be two metric spaces and let $f : V \to V'$ be an embedding. The* distortion *of f is $\max_{u,v \in V, u \neq v} \max\{\frac{d(u,v)}{d'(f(u),f(v))}, \frac{d'(f(u),f(v))}{d(u,v)}\}$.*

[1] We define ℓ_1 metric with respect to finite dimensional embeddings. Technically we can allow infinite dimensional embeddings but they are not needed for finite metrics. Moreover, it is algorithmically more useful to confine attention to finite dimensional embeddings.

Of particular importance are embeddings of finite metric spaces into Euclidean space $\mathbb{R}^d$ where the distance in the host space is measured under a norm. Examples include $\ell_1, \ell_2, \ell_\infty$. An embedding of a finite metric space (V, d) into $\mathbb{R}^d$ means that we map each v to a point $(x_1, x_2, \ldots, x_d)$ and the distance between say x, y is measured as $\|x - y\|$ for some norm of interest.

The dimension d is also important in various applications but in some settings like with Sparsest Cut the dimension is not important.

Theorem 9 (Bourgain)**.** *Any n-point finite metric space can be embedded into ℓ_2 (and hence also ℓ_1) with distortion $O(\log n)$. Moreover the embedding is a contraction and can be constructed in randomized polynomial time and embeds points into $\mathbb{R}^d$ where $d = O(\log^2 n)$.*

In fact one can obtain a refined theorem that is useful for Sparsest Cut.

Theorem 10 (Bourgain)**.** *Let (V, d) be n-point finite metric and let $S \subseteq V$ with $|S| = k$. Then there is a randomized polynomial time algorithm to compute an embedding $f : V \to \mathbb{R}^{O(\log^2 n)}$ such that (i) the embedding is a contraction (that is, $\|f(u) - f(v)\|_1 \leq d(u, v)$ for all $u, v \in V$ and (ii) for every $u, v \in S$, $\|f(u) - f(v)\|_1 \geq \frac{c}{\log k} d(u, v)$ for some universal constant c.*

4.4 Utilizing the ℓ_1 embedding

We saw that the integrality gap of the LP is 1 on trees since the shortest path metric on trees is in the cut cone (equivalently ℓ_1-embeddable). More generally one can prove that if the shortest path metric on a graph G embeds into ℓ_1 with distortion α then the integrality gap of the LP for Sparsest Cut is at most α. This will imply an $O(\log n)$-integrality gap via Bourgain's theorem since any n point finite metric embeds in to ℓ_1 with distortion $O(\log n)$.

Theorem 11. *Let $G = (V, E)$ be a graph. Suppose any finite metric induced by edge lengths on E can be embedded into ℓ_1 with distortion α. Then the integrality gap of the LP for* Sparsest Cut *is at most α for any instance on G.*

Proof. Let (x, y) be a feasible fractional solution and let d be the metric induced by edge lengths given by x. Let λ be the value of the solution and recall that $\lambda = \frac{\sum_{uv \in E} c(uv) d(uv)}{\sum_{i=1}^k D_i d(s_i, t_i)}$.

Since d can be embedded into ℓ_1 with distortion at most α and any ℓ_1 metric is in the cut-cone, it implies that there are scalaras $z_S, S \subset V$ such that for all u, v

$$\frac{1}{\alpha} \sum_{S \subset V} y_S d_S(u, v) \leq d(u, v) \leq \sum_{S \subset V} y_S d_S(u, v).$$

Here we assumed without loss of generality that the embedding is a contraction. For a set $S \subset V$ we use $\text{Dem}(\delta(S)) = \sum_{i:|S \cap \{s_i, t_i\}|=1} D_i$ to denote the total demand crossing the cut S.

$$\begin{aligned}
\lambda &= \frac{\sum_{uv \in E} c(uv) d(uv)}{\sum_{i=1}^k D_i d(s_i, t_i)} \\
&\geq \frac{1}{\alpha} \frac{\sum_{uv \in E} c(uv) \sum_{S \subset V} z_S d_S(uv)}{\sum_{i=1}^k D_i \sum_{S \subset V} d_S(s_i, t_i)} \quad \text{(using embedding of } d \text{ with distortion } \alpha \\
&= \frac{1}{\alpha} \frac{\sum_{S \subset V} z_S c(\delta(S))}{\sum_{S \subset V} z_S \text{Dem}(\delta(S))} \\
&\geq \frac{1}{\alpha} \min_{S \subset V} \frac{c(\delta(S))}{\text{Dem}(\delta(S))}.
\end{aligned}$$

Thus there is a cut whose sparsity is at most $\alpha \cdot \lambda$. ■

Polynomial-time algorithm: How do we fine a sparse cut? The preceding proof used the embedding of d into the cut-cone. The proof shows that one of the cuts with $z_S > 0$ has sparsity at most $\alpha \cdot \lambda$. Recall the proof that a metric is in the cut-cone iff it is ℓ_1-embeddable. That argument shows the following. Suppose we have an ℓ_1 embedding into d-dimensions. Each dimension corresponds to a line-embedding. Each line embedding is in the cut-cone with only $n-1$ cuts used to express it. Thus, given an ℓ_1 embedding into d dimensions with distortion α we only need to try $d(n-1)$ cuts and one of them will be guaranteed to have sparsity at most $\alpha \cdot \lambda$.

Via Theorem 10 we can obain an $O(\log k)$ randomized approximation and the algorithm is described below.

SparseCutviaEmbedding

1. Solve LP relaxation to obtain (x, y) and metric d_x on V
2. Use Theorem 10 to obtain map $f : V \to \mathbb{R}^d$ where $d = O(\log^2 n)$
3. For $i =$ to d do
   (a) Let $v_{j_1}, v_{j_2}, \ldots, v_{j_n}$ be the sorting of V according to dimension i
   (b) For $h = 1$ to $n-1$ let $S_{i,h} = \{v_{j_1}, v_{j_2}, \ldots, v_{j_h}\}$
4. Among all cuts $S_{i,h}$ with $1 \le i \le d$ and $1 \le h \le n-1$ output the one with the smallest sparsity.

Exercise 6. *Use the refined guarantee in 10 and the proof outline in 11 to show that the described algorithm is a randomized $O(\log k)$-approximation algorithm for* Sparsest Cut.

4.5 Line embeddings, ℓ_1 embeddings and tree embeddings

Bourgain's theorem shows that any finite metric space on n points embeds into ℓ_1 with distortion $O(\log n)$. This can also be derived via probabilistic tree embeddings because every tree metric embeds into ℓ_1 isometrically (that is, without any distortion or with distortion 1).

Exercise 7. *Prove formally that if a finite metric (V, d) can be probabilistically approximated via dominating tree metrics with distortion α then it can be embedded into ℓ_1 with distortion α. Is the resulting map an expansion or a contraction?*

For general metrics, tree embeddings provide a more constrained space while yielding the same worst-case distortion. However, one can ask if ℓ_1 embeddings provide better distortion for concrete graph classes. Indeed this is the case. For instance consider a ring network (a cycle with capacities). One can prove that tree embeddings require a distortion 2. However the ring metric can be isometrically embedded into ℓ_1. Thus, the flow-cut gap on ring networks is 1 which is not obvious.

Exercise 8. *Prove that the metric on a ring network emebeds into ℓ_1 isometrically.*

Planar graphs are an important class of graphs for a number of reasons, both from a theoretical and applications point of view. They also pave the way to the class of proper minor closed families of graphs. There is a famous conjecture that the flow-cut gap in planar graphs is $O(1)$ [GNRS99]. Interestingly, for tree embeddings there is a lower bound of $\Omega(\log n)$ even on the special case of planar graphs called series parallel graphs. Thus, tree embeddings are not powerful enough to prove the conjecture. Rao proved that the flow-cut gap is $O(\sqrt{\log n})$ via ℓ_1 embeddings, thus separating the general graph case from planar graph case. For series-parallel graphs we know that the flow-cut gap is a tight bound of 2 and establishing this tight bound took a fair amount of work. For UNIFORM SPARSEST CUT, [KPR93] showed that the flow-cut gap in planar graphs is $O(1)$. One can show a tight connection between embedebbility into ℓ_1 and flow-cut gap [GNRS99].

We saw that ℓ_1 embeddings are a non-negative combination of line embeddings. A particular type of line-embedding is called a Frechet embedding. Let (V, d) be a metric space and let $S \subseteq V$. Then we can define a mapping $f : V \to \mathbb{R}$ to the real line where $f(v) = d(S, v)$, that is the distance of v from S. Note that all vertices in S get mapped to 0. And the mapping is a contraction. Many results in embeddings into ℓ_p spaces are based on using Frechet embeddings in various clever and often highly non-trivial ways. Bourgain's embedding is, in particular, based on picking many random sets and combining the resulting Frechet embeddings. Line embeddings are simple and have substantial power on their own. Rabinovich [Rab08] defined the following notion of *average* distortion for line embeddings.

Definition 7. *Let (V, d) be a metric space and let $w : V \times V \to \mathbb{R}_+$ be a non-negative symmetric weight/demand function on the vertices. We say that a line embedding $f : V \to \mathbb{R}$ is has* average weighted *distortion α for some $\alpha \geq 1$ if*

$$\sum_{uv} w(uv)|f(u) - f(v)| \geq \frac{1}{\alpha} \sum_{uv} w(uv)d(uv).$$

Note that a contraction implies $\sum_{uv} w(uv)|f(u) - f(v)| \leq \sum_{uv} w(uv)d(uv)$ so we are mainly interested in lower bounding the average distances.

Bourgain's result in fact implies the following.

Theorem 12. *For any n point metric space (V, d) and any weight function $w : V \times V \to \mathbb{R}_+$ there is a line embedding with average weighted distortion $O(\log n)$.*

One can use line embeddings with average distortion to prove flow-cut gap. You should think of the weight $w(uv)$ as the demand $D(uv)$ for pair uv.

Exercise 9. *Prove that the flow-cut gap for* SPARSEST CUT *is $O(\log n)$ using Theorem 12.*

For product multicommodity flow the demand $D(uv)$ is of the form $\pi(u)\pi(v)$. Rabinovich [**?**] showed that existing results such as [KPR93] can be interpreted as providing line embeddings with low average distortion for these demands/weights. Line embeddings turn out to be very relevant when considering node capacities and directed graphs (among other generalizations) where ℓ_1 embedding do not quite work. They are also relevant in spectral methods and SDP methods. We refer the reader to [FHL05] for the relevance of line embeddings in node-capacitated settings and their utility beyond node-capacities [CKRV12].

5 SDP Relaxation

Can we obtain a better approximation than $O(\log k)$ for sparsest cut? Using semi-definite programming based relaxation, Arora, Rao and Vazirani [ARV09] obtained an $O(\sqrt{\log n})$-approximation for Uniform Sparsest Cut (and more generally product instances). Building on this work, Arora, Lee and Naor [ALN08, ALN07] obtained an $O(\sqrt{\log n}\log\log n)$-approximation for Sparsest Cut. Currently these are the best approximation algorithms known for these problems. There was a conjecture that the SDP based relaxation would yield an $O(1)$-approximation but it was shown that the integrality gap is essentially close to $\Omega(\sqrt{\log n})$; this required substantial mathematical work.

6 Spectral Relaxation for Conductance

As we said expander graphs are very important. However, certifying expansion is co-NP hard. In several applications it is important to obtain constant-degree expanders with constant expansion. The approximation algorithms we saw are not useful in this regime. It turns out that there is a very different method based on spectral graph theory that helps in this regime. For an undirected graph on n vertices one can associate a symmetric matrix called the Laplacian $\mathcal{L}_G$. The matrix has the following entries. The diagonal entry for i is $deg(v_i)$. We have $\mathcal{L}_G[i,j] = \mathcal{L}_G[i,j] = -1$. Since this matrix is symmetric all its eigen values are real. More over, this matrix is also positive semidefinite (PSD) and hence all its eigen values are non-negative. Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$ be its eigen values (it is not hard to verify that 0 is an eigen value with eigen vector being the all 1's vector). A well-known and famous result in spectral graph theory is the following Cheeger's inequality on the conductance of G.

$$\frac{\lambda_2}{2} \leq \phi(G) \leq \sqrt{2\lambda_2}$$

Thus λ_2 provides a constant factor approximation for the conductance when the conductance is a constant! Note that expansion and conductance are related by the maximum degree and hence when the degree is small and constant one can use λ_2 to certify expansion. Due to its importance for certifying expansion/conductance, some use λ_2 as the definition of expansion since it is computable and also helps in construction of expanders.

7 Node capacities and Directed Graphs

We typically discuss s-t cuts and other problems such as Sparsest Cut cut with edge capacities. One can define cut problems with node capacities; for instance Menger's theorem on node disjoint paths is an example. In the s-t case it is typical to reduce the node-capacitated problems to edge capacitated case when the graph is *directed*. However this reduction does not apply when the graph is *undirected*. When dealing with more complicated problems involving multicommodity flows and cuts, the directed graph problems often become harder than their undirected graph counterparts. And thus one has to consider node-capacitated undirected graph problems separately.

In the context of Uniform Sparsest Cut, node capacitated cuts are quite important since they are closely related to the notion of *treewidth* of a graph. It is a graph parameter and tree decompositions associated with treewidth play a fundamental role in graph structure theory and many algorithmic applications. Treewidth is NP-hard to compute but an α approximation for Uniform Sparsest Cut implies an $O(\alpha)$-approximation for treewidth. We think of node separator as a set of vertices S such that $G - S$ has (at least) two non-trivial components. Sometimes one uses the terminology (A, S, B) for disjoint sets A, B, S with S as the separation if there is no edge between A and B (all paths between A and B goe via S). The cost/weight of the cut/separation is now the cost/weight of the separator set S. When considering sparse cuts and the like there is a bit of an added complication because S itself is a set of vertices which may have demands associated with them. Handling this requires some care and proper definitions. The existing algorithms for Uniform Sparsest Cut can be generalized with additional technical work (one needs to work with line embeddings rather than ℓ_1 embeddings) to handle node capacities. This leads to an $O(\sqrt{\log n})$-approximation for Uniform Sparsest Cut and a corresponding algorithm for treewidth. See [FHL05].

Sparsest Cut in directed graphs is more difficult. The natural definition, motivated by obstructions to routing, is the following. We are given a directed edge-capcitated graph $G = (V, E)$ and a set of k ordered demand pairs $(s_1, t_1), \ldots, (s_k, t_k)$ each with a non-negative demand D_i. We think of a cut as a set of edges $E' \subseteq E$. Removing E' disconnects some pairs and the sparsity is simply $c(E')/\text{dem}(E')$ where $\text{dem}(E')$ is the demand separated by E'. Unlike undirected graphs minimal E' cuts do no induce bipartititions of the vertex set. It turns out that the flow-cut gap (for Multicut also) can be as large as k [SSZ04], and also $\Omega(n^{11/27}$ [CK09]. More over, [CK09] prove hardness of approximation close to polynomial factors. However, there is an important special case of *symmetric* demands that is quite relevant in several problems. Here the demands are *unordered*, that is, each demand pair is $s_i t_i$ and we say that a pair is separated by an edge set E' if they are not in the same strongly connected component. Note that this means that we are asking that at least one of the pairs (s_i, t_i) or (t_i, s_i) to be separated. It turns out that Multicut and Sparsest Cut and Uniform Sparsest Cut in this formulation have poly-logarithmic approximations and flow-cut gaps. These problems have also connected to the problem of removing edges/nodes to delete all cycles (called Feedback problems). We refer the reader to [ENRS00] on flow-cut gaps and to [ACMM05, CMM06] for SDP based algorithms.

Finally, since it is connected to the work of the instructor, there is a notion of polymatroid networks which connect network flows to submodular functions. This comes from classical work in combinatorial optimziation and one can generalize the notions of multicommodity flows and cuts. Several ideas from regular flows and cuts can be ported over with useful applications and consequences. See [CKRV12] and some recent work on submodular hypergraphs [COT23].

Additional bibliographic information

The highly influential paper of Leighton and Rao [LR99] obtained an $O(\log n)$-approximation and flow-cut gap for Uniform Sparsest Cut and introduced the region growing argument as well as the lower bound via expanders (an important influence is the paper of Sharokhi and Matula [SM90]). [LR99] demonstrated many applications of the divide and conquer approach. There is a large literature on Sparsest Cut and related problems and we only touched upon a small part. An outstanding open problem is whether the flow-cut gap for Non-Uniform Sparsest Cut in planar graphs is $O(1)$ (this called the GNRS conjecture [GNRS99] in the more general context of minor-free graphs); Rao, building on ideas from [KPR93], showed that the gap is $O(\sqrt{\log n})$ [Rao99]. No super-constant lower bound is known for planar graphs. The theory of metric embeddings has been a fruitful bridge between TCS and mathematics and there are several surveys and connections from both perspectives. The argument via MULTICUT is attributed to Nabil Kahale — see the chapter by Shmoys on approximation algorithms for cut problems [Shm97].

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 thirty-seventh annual ACM symposium on Theory of computing*, pages 573–581, 2005.

[ALN07] Sanjeev Arora, James R Lee, and Assaf Naor. Fréchet embeddings of negative type metrics. *Discrete & Computational Geometry*, 38(4):726–739, 2007.

[ALN08] Sanjeev Arora, James R Lee, and Assaf Naor. Euclidean distortion and the sparsest cut. *Journal of the AMS*, 21(1):1–21, 2008.

[AR98] Yonatan Aumann and Yuval Rabani. An o (log k) approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.

[ARV09] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):1–37, 2009.

[CK09] Julia Chuzhoy and Sanjeev Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *Journal of the ACM (JACM)*, 56(2):1–28, 2009.

[CKRV12] Chandra Chekuri, Sreeram Kannan, Adnan Raja, and Pramod Viswanath. Multicommodity flows and cuts in polymatroidal networks. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 399–408, 2012.

[CMM06] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Directed metrics and directed graph partitioning problems. In *SODA*, volume 6, pages 51–60, 2006.

[COT23] Antares Chen, Lorenzo Orecchia, and Erasmo Tani. Submodular hypergraph partitioning: Metric relaxations and fast algorithms via an improved cut-matching game. *arXiv preprint arXiv:2301.08920*, 2023.

[ENRS00] Guy Even, Joseph Seffi Naor, Satish Rao, and Baruch Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *Journal of the ACM (JACM)*, 47(4):585–616, 2000.

[FHL05] Uriel Feige, MohammadTaghi Hajiaghayi, and James R Lee. Improved approximation algorithms for minimum-weight vertex separators. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 563–572, 2005.

[GNRS99] Anupam Gupta, Ilan Newman, Yuri Rabinovich, and Alistair Sinclair. Cuts, trees and l/sub 1/-embeddings of graphs. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 399–408. IEEE, 1999.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[KPR93] Philip Klein, Serge A Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 682–690, 1993.

[LLR95] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[LR99] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999. Conference version is from 1988.

[Rab08] Yuri Rabinovich. On average distortion of embedding metrics into the line. *Discrete & Computational Geometry*, 39(4):720–733, 2008.

[Rao99] Satish Rao. Small distortion and volume preserving embeddings for planar and euclidean metrics. In *Proceedings of the fifteenth annual symposium on Computational geometry*, pages 300–306, 1999.

[Shm97] David B Shmoys. Cut problems and their application to divide-and-conquer. *Approximation algorithms for NP-hard problems*, pages 192–235, 1997.

[SM90] Farhad Shahrokhi and David W Matula. The maximum concurrent flow problem. *Journal of the ACM (JACM)*, 37(2):318–334, 1990.

[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.

[WS11] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.