TTIC 31080 - Approximation Algorithms **Note 2**

The Sparsest Cut Problem

Scribe: Antares Chen 11/09/2023

In which we discuss the sparsest cut problem, and an $O(\log n)$-approximation algorithm due to Linial, London & Rabinovich.

1 Introduction

Previously, we studied the multicut problem which given a weighted graph, and a set of demand pairs, asked to compute a minimum cost subset of edges such that its removal from the graph separates all demand pairs. Today, we study a close relative of the multicut problem, called the sparsest cut problem.

In the sparsest cut problem, our goal is to delete a minimum cost subset of edges such that as many (but not necessarily all) demand pairs are separated. There are two ways of defining this problem.

Definition 1 (Edge Set / Non-bipartite Version). *Given a graph $G = (V, E)$ with edge costs c_e, and a set of k demand pairs $\mathcal{D} = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$, the sparsest cut problem asks to output $E' \subseteq E$ such that the ratio between $c(E')$ and the total number of demand pairs disconnected in $G \setminus E'$ is minimized. (We require that at least on demand pair is disconnected)*

Definition 2 (Vertex Set / Bipartite Version). *Let $G = (V, E)$ be a graph with edge costs c_e, and $\mathcal{D} = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$ be a set of k demand pairs. The sparsity of a cut $S \subseteq V$ is defined as*

$$\varphi_G(S) \stackrel{\text{def}}{=} \frac{c(E(S, \bar{S}))}{|D(S, \bar{S})|}$$

where $D(S, \bar{S})$ is the number of source-sink pairs that are "disconnected" by the cut.

- *If the graph is* undirected*, then $D(S, \bar{S})$ contains every demand pair (s_i, t_i) such that exactly one of s_i, t_i lies in S. That is, we can write $|D(S, \bar{S})|$ as*

$$|D(S, \bar{S})| = \sum_{i=1}^{k} \mathbf{1}\{(s_i, t_i) \text{ is separated by } S\}, \; \mathbf{1}\{(s_i, t_i) \text{ is separated by } S\} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } |S \cap \{s_i, t_i\}| = 1 \\ 0 & \text{otherwise} \end{cases}$$

- *If the graph is* directed *then $D(S, \bar{S})$ contains all demand pairs (s_i, t_i) such that $s_i \in S$ and $t_i \notin S$. In this case, $E(S, \bar{S})$ also only contains edges of G that are directed from S to $\bar{S}$.*

The goal of the sparsest cut problem *is then to output a cut $S \subseteq V$ with minimum sparsity. We will denote $\varphi_G \stackrel{\text{def}}{=} \min_{S \subseteq V} \varphi(S)$*

These two definitions of the sparsest cut problem are equal when G is an undirected graph. When G is directed, the edge-set, and vertex-set definitions of cost could differ significantly! In this note, we will demonstrate an $O(\log n)$-approximation to the *vertex-set based definition* of sparsest cut for *undirected graphs*.

Theorem 3. *There exists an efficient, randomized algorithm $\mathcal{A}$ such that, when given an undirected graph $G = (V, E)$ with edge costs $c_e > 0$ for all $e \in E$, along with demands $\mathcal{D} = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$, $\mathcal{A}$ outputs an $O(\log n)$-approximation to the sparsest cut problem, i.e. $S \subseteq V$ satisfying*

$$\varphi_G \leq \varphi_G(S) \leq O(\log n) \cdot \varphi_G(S)$$

We present the analysis first given by Linial, London & Rabinovich [1]. This algorithm is also based off of LP-rounding and so these notes will be organized like so

(1) First we derive the metric LP relaxation from the ℓ_1 formulation for sparsest cut.
(2) We then write down the blueprint of the algorithm and recall some tools from metric geometry.
(3) Finally, we will work through the analysis of the algorithm, and finally describe the algorithm in full detail.

2 A Metric LP Relaxation for Sparsest Cut

Unlike typical applications of the LP relaxation framework, the derivation of the metric LP for sparsest cut we discuss doesn't start with writing an ILP. Instead, we derive a certain minimization problem concerning the ratio of ℓ_1-metric quantities.

2.1 A Primer on Metrics

Before proceeding, it is helpful to define what a metric space is.

Definition 4 (Metric Spaces)**.** *A metric space is a tuple $(\mathcal{X}, d)$ where $\mathcal{X}$ is a set of points, and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a distance function satisfying the following conditions.*

1. *(Positivity) for all $x, y \in \mathcal{X}$ we have $d(x, y) \geq 0$ with $d(x, y) = 0$ if and only if $x = y$.*
2. *(Symmetric) for all $x, y \in \mathcal{X}$, we have $d(x, y) = d(y, x)$.*
3. *(Triangle Inequality) for all x, y, z we have $d(x, y) + d(y, z) \geq d(x, z)$.*

We call $d(\cdot, \cdot)$ a metric. If $d(x, y)$ satisfies all the conditions except for the fact that $d(x, y) = 0$ if and only if $x = y$, then we call d a semi-metric.

We remark that for many graph partitioning problems, it is sufficient to work with semi-metrics. Two examples of metric spaces that we'll see in this note are the following.

- Given a graph $G = (V, E)$ labeled with edge lengths $x_e > 0$, define the distance function $d_G : V \times V \to \mathbb{R}$ to be

$$d_G(i, j) \stackrel{\text{def}}{=} \text{"The shortest path distance between } i, j \text{ in } G \text{ w.r.t. lengths } x_e\text{"}$$

  The metric space (V, d_G) is called the *shortest path metric* associated with G (with respect to lengths x_e).
- Given an n-point metric space $(\mathcal{X}, d)$, suppose there exists $h \in \mathbb{Z}_{\geq 1}$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^h$ such that

$$d(i, j) = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_1 \qquad \forall\, i, j \in \mathcal{X}\,.$$

  Then, we call $(\mathcal{X}, d)$ an ℓ_1-*metric.*

2.2 Deriving the LP Relaxation

Let us now derive the LP relaxation for sparsest cut. We begin by choosing our decision variables. As sparsest cut optimizes over cuts $S \subseteq V$, a natural choice of decision variables is to quantify over 0-1 indicator of cuts S. For a fixed $S \subseteq V$, we associate a hypothetical integral solution $x_i \in \{0, 1\}$ for each $i \in V$ where

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

It is helpful to interpret this assignment to a value x_i for all $i \in V$ as performing a *1-dimensional embedding* of vertices into the real line in the sense that we assign vertices i to position x_i.

With this choice of decision variables, we can now write down the objective. The objective for sparsest cut is a minimization of ratios. Let's try writing down the numerator, and denominator separately in terms of x. For the numerator, consider an edge $(i, j) \in E$. Notice that $|x_i - x_j| = 1$ if and only if $(i, j) \in E(S, \bar{S})$, otherwise it will be zero. Consequently,

$$c(E(S, \bar{S})) = \sum_{ij \in E} c_{ij} \cdot |x_i - x_j|$$

Since the denominator is the number of demand pairs separated by S, we can similarly write

$$|D(S, \bar{S})| = \sum_{i=1}^{k} \mathbf{1}\{(s_i, t_i) \text{ is separated by } S\} = \sum_{i=1}^{k} |x_{s_i} - x_{t_i}|$$

Putting the above together, we have, in effect, derived the following statement. Given any collection of $x_i \in \{0, 1\}$ where each $i \in V$, let us associate with it a more compact representation as the vector $\boldsymbol{x} \in \{0, 1\}^n$ where the i-th coordinate of $\boldsymbol{x}$ is given by x_i. Let $\psi_G(\boldsymbol{x})$ be defined as

$$\psi_G(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{\sum_{ij \in E} c_{ij} \cdot |x_i - x_j|}{\sum_{i=1}^{k} |x_{s_i} - x_{t_i}|}$$

We will colloquially refer to this as the ℓ_1-*cost* of the 1-dimensional embedding given by the x_i's. The above computation subsequently shows that, if $\boldsymbol{x} \in \{0, 1\}^n$ is the 0-1 indicator of S, then the ℓ_1-cost of the assigned values x_i is equivalent to the sparsity of S: $\psi_G(\boldsymbol{x}) = \varphi_G(S)$. Consequently,

$$\psi_G \stackrel{\text{def}}{=} \min_{S \subseteq V} \psi_G(S) = \min_{S \subseteq V} \varphi_G(S) = \varphi_G$$

At this point, we could try to relax $x_i \in \{0, 1\}$ to the continuous constraint $x_i \geq 0$, and hope that the subsequent problem is efficiently solvable. This does not immediately work due to the following result.

Theorem 5. *Given graph $G = (V, E)$ with edge weights $c_e > 0$, and demands $\mathcal{D} = \{(s_i, t_i)\}_{i=1,\ldots,k}$, we have*

$$\min_{\boldsymbol{x} \in \{0,1\}^n} \psi_G(\boldsymbol{x}) = \min_{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}} \psi_G(\boldsymbol{x}) \,.$$

Furthermore, there exists an efficient algorithm $\mathcal{A}$ such that given $\boldsymbol{z} \in \mathbb{R}^n_{\geq 0}$, $\mathcal{A}$ outputs a cut $S \subseteq V$ satisfying

$$\varphi_G(S) \leq \psi_G(\boldsymbol{z}) \,.$$

In the parlance of metric geometry, Theorem 5 says that "every line metric (the side $\min_{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}} \psi_G(\boldsymbol{x})$) embeds isometrically into a distribution of cut metrics (the side $\min_{\boldsymbol{x} \in \{0,1\}^n} \psi_G(\boldsymbol{x})$)". Additionally, it implies that solving $\min_{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}} \psi_G(\boldsymbol{x})$ is NP-hard. To see why, suppose that we could solve the above optimization problem optimally. Let $\boldsymbol{z}^* = \operatorname{argmin}_{x_i \geq 0} \psi_G(\boldsymbol{x})$. Then, using the algorithm $\mathcal{A}$ present in Theorem 5, we could round a cut $S \subseteq V$ whose sparsity satisfies the following:

$$\varphi_G(S) \leq \psi_G(\boldsymbol{z}) = \min_{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}} \psi_G(\boldsymbol{x}) = \min_{\boldsymbol{x} \in \{0,1\}^n} \psi_G(\boldsymbol{x}) = \psi_G = \varphi_G \,.$$

Though Theorem 5 implies that we can't hope to solve the continuous problem $\min_{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}} \psi_G(\boldsymbol{x})$ efficiently, we can use this continuous problem to produce an LP relaxation. This is how we do it. First, we can rewrite

$\min_{x_i \geq 0} \psi_G(\boldsymbol{x})$ so that it does not optimize over a ratio by observing that $\psi_G(\boldsymbol{x})$ is *positive homogeneous*: for any $\alpha \in \mathbb{R}$, we have $\psi_G(\alpha \cdot \boldsymbol{x}) = \psi_G(\boldsymbol{x})$.

$$\psi_G(\alpha \cdot \boldsymbol{x}) = \frac{\sum_{ij \in E} c_{ij} \cdot |\alpha \cdot x_i - \alpha \cdot x_j|}{\sum_{i=1}^{k} |\alpha \cdot x_{s_i} - \alpha \cdot x_{t_i}|} = \frac{|\alpha| \cdot \sum_{ij \in E} c_{ij} \cdot |x_i - x_j|}{|\alpha| \cdot \sum_{i=1}^{k} |x_{s_i} - x_{t_i}|} = \frac{\sum_{ij \in E} c_{ij} \cdot |x_i - x_j|}{\sum_{i=1}^{k} |x_{s_i} - x_{t_i}|} = \psi_G(\boldsymbol{x})$$

Consequently, for any $\boldsymbol{x}$ that minimizes $\psi_G(\boldsymbol{x})$, there is another solution $\hat{\boldsymbol{x}}$ such that the denominator in $\psi_G(\boldsymbol{x})$ is one. Hence,

$$\min_{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}} \psi_G(\boldsymbol{x}) = \frac{\sum_{ij \in E} c_{ij} \cdot |x_i - x_j|}{\sum_{i=1}^{k} |x_{s_i} - x_{t_i}|} = \begin{cases} \min\limits_{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}} & \sum\limits_{ij \in E} c_{ij} \cdot |x_i - x_j| \\ \text{s.t.} & \sum\limits_{i=1}^{k} |x_{s_i} - x_{t_i}| = 1 \end{cases} \tag{1}$$

From here, then derive a relaxation for sparsest cut. Notice that this problem asks for a way to assign vertices $i \in V$ to positions $x_i \geq 0$ such that (1) the sum of lengths $|x_{s_i} - x_{t_i}|$ between demand pairs is one, but (2) the sum of lengths $|x_i - x_j|$ along edges is collectively small. What makes this problem computationally intractable is the fact that the lengths have to come from a *1-dimensional ℓ_1-metric*. To relax the problem, we can optimize over all possible distances, instead of those given by ℓ_1 distances between two positions in $\mathbb{R}$. Performing the variable replacement

$$|x_i - x_j| \mapsto x_e \qquad \forall\, e \in E : e = (i, j) \qquad\qquad |x_{s_i} - x_{t_i}| \mapsto d_i \qquad \forall\, i = 1, \ldots, k$$

yields the following linear program.

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e \cdot x_e \\ \text{s.t.} \quad & \sum_{i=1}^{k} d_i = 1 \\ & x_e \geq 0, d_i \geq 0 \quad \forall\, e \in E\ i = 1, \ldots, k \end{aligned}$$

2.3 Introducing the Metric

Even though the above program is a linear program, and is efficiently solvable, this LP does not have enough structure for us to produce a good rounding algorithm. In particular, this LP loses the metric structure present in the original ℓ_1-minimization problem! Previously, we were optimizing over ℓ_1 distances, and so, implicitly, feasible solutions assigning each $i \in V$ to $x_i \geq 0$ also satisfied the ℓ_1 triangle inequality. When we performed the relaxation, we lost the fact that feasible solutions had to satisfy the triangle inequality in the resulting LP.

In order to reintroduce metric structure, we could add back the triangle inequalities explicitly. However, we take a slightly different route, and instead add a collection of constraints that are *implied* by ℓ_1-distances on $\mathbb{R}$ satisfying the ℓ_1 triangle inequality. For each $i = 1, \ldots, k$, let $\mathcal{P}_i$ denote the set of all paths between s_i and t_i. Now, fix any $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ feasible for (1). Note that it satisfies the triangle inequalities:

$$|x_i - x_j| + |x_j - x_k| \geq |x_i - x_k| \qquad\qquad \forall\, i, j, k \in V$$

These inequalities imply that, for any $i = 1, \ldots, k$ and $p \in \mathcal{P}_i$, the solution $\boldsymbol{x}$ also satisfies

$$\sum_{(u,v) \in p} |x_u - x_v| \geq |x_{s_i} - x_{t_i}| \,.$$

Consequently adding the above inequalities as constraints to (1) does not change the objective value of the problem. The following program has an optimal objective value equivalent to that of (1).

$$\begin{aligned}
\min_{\boldsymbol{x}\in\mathbb{R}^n_{\geq 0}} \quad & \sum_{ij\in E} c_{ij}\cdot|x_i - x_j| \\
\text{s.t.} \quad & \sum_{i=1}^{k}|x_{s_i} - x_{t_i}| = 1 \\
& \sum_{(u,v)\in p} |x_u - x_v| \geq |x_{s_i} - x_{t_i}| \quad \forall\, i = 1,\ldots,k \text{ and } p\in\mathcal{P}_i
\end{aligned}$$

Performing the relaxation as we did above, we derive the *metric LP relaxation* for sparsest cut.

$$\begin{aligned}
\min \quad & \sum_{e\in E} c_e\cdot x_e \\
\text{s.t.} \quad & \sum_{i=1}^{k} d_i = 1 \\
& \sum_{e\in p} x_e \geq d_i \qquad \forall\, i = 1,\ldots,k \text{ and } p\in\mathcal{P}_i \\
& x_e \geq 0, d_i \geq 0 \quad \forall\, e\in E\ i = 1,\ldots,k
\end{aligned} \tag{Metric-LP}$$

Notice that the decision variables in (Metric-LP) include x_e for each edge $e\in E$, and d_i for each demand s_i, t_i pair. This LP also possesses an exponentially many number of constraints. To solve this LP, we can use the ellipsoid method with the following separation oracle: to check one of the constraints $\sum_{e\in p} x_e \geq d_i$ is violated, between every s_i, t_i pair, compute the shortest path in G under edge weights given by x_e. If for any i, the shortest path distance between s_i, t_i is strictly smaller than d_i, then the constraint $\sum_{e\in p} x_e \geq d_i$, for p the shortest path, is violated.

3 The Linial-London-Rabinovich Rounding Algorithm

The process of deriving the LP gives intuition for how one might attempt to construct a rounding algorithm. Previously, we saw how LPs relaxations for graph partitioning problems that assign values x_e to edges can be used to define a natural metric. This is the shortest path metric of G when its edges are weighted by the LP assignment x_e. Using the LP computed shortest path metric, we can then try to transform its pairwise distances into an 1-dimensional embedding whose ℓ_1 cost is not too far away from the optimal LP cost. Why should we aim to produce a 1-dimensional embedding? Because Theorem 5 then tells us that there is an efficient algorithm which can round a cut whose cost is no worse than the ℓ_1-cost of the 1-dimensional embedding. Outputting this cut will then yield us our rounding algorithm.

This is what the rounding algorithm provided by Linial, London & Rabinovich does. Our goal will be to make this sketch formal. In order to do this, we require further tools from metric geometry.

3.1 Metric Embeddings and Bourgain's Theorem

In order to formalize what it means to map one set of distances to another, we use a *metric embedding.*

Definition 6 (Metric Embeddings)**.** *An embedding of a metric space $(\mathcal{X}, d_{\mathcal{X}})$ into another metric space $(\mathcal{Y}, d_{\mathcal{Y}})$ is a function $f:\mathcal{X}\to\mathcal{Y}$. We say that $f:\mathcal{X}\to\mathcal{Y}$ is a* non-expanding *embedding with* distortion $\alpha\geq 1$ *if for every pair $i,j\in\mathcal{X}$ of points:*

$$\frac{1}{\alpha}\cdot d_{\mathcal{X}}(i,j) \leq d_{\mathcal{Y}}\big(f(i), f(j)\big) \leq d_{\mathcal{X}}(i,j)$$

With this, we can introduce a theorem which has become incredibly useful in designing approximation algorithms.

Theorem 7 (Bourgain)**.** *There exists an efficient randomized algorithm that, given any n-point metric space $(\mathcal{X}, d)$, outputs an embedding it into $\left(\{\boldsymbol{y}_i\}_{i\in\mathcal{X}}, \ell_1\right)$. Furthermore, the embedding satisfies the following.*

(1) For each $i \in V$, $\boldsymbol{y}_i \in \mathbb{R}^h$ where $h \leq O(\log^2 n)$.

(2) The embedding is non-expanding, and with high probability, the distortion of the embedding satisfies $\alpha \leq O(\log n)$. That is, for every $i, j \in \mathcal{X}$

$$\frac{1}{O(\log n)} \cdot d(i,j) \leq \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_1 \leq d(i,j)$$

This theorem tells us that given *any* metric space $(\mathcal{X}, d)$, one can produce a mapping from points in $\mathcal{X}$ to vectors in $\mathbb{R}^h$ such that for each pair of points in $\mathcal{X}$, the ℓ_1 distance between their embedding vectors is approximately their distance in $\mathcal{X}$. Furthermore, this theorem is *algorithmic* in that we can compute this mapping in polynomial time.

Bourgain's theorem is incredibly useful. The strategy of producing an optimization problem over ℓ_1-metrics whose value coincides with the combinatorial optimal value, then subsequently relaxing the ℓ_1 problem to an LP which quantifies over all metrics, can very generically be applied to many graph partitioning problems. In such cases, Bourgain's theorem applies as one can use the LP solution to define a metric on vertices in the given graph, then use Bourgain's theorem to produce an embedding back into an ℓ_1-metric. The $O(\log n)$-distortion in the embedding will add an $O(\log n)$ factor to the approximation ratio.

In our blueprint, we can take the shortest path metric, and apply Bourgain's theorem to produce an embedding $i \mapsto \boldsymbol{y}_i$ where $\boldsymbol{y}_i \in \mathbb{R}^h$ for each $i \in V$. The fact that our distortion is low will allow us to demonstrate that the analogous ℓ_1 cost for this embedding is not too far away from the optimal LP cost. However, we still have to deal with the fact that for all $i \in V$, the vectors $\boldsymbol{y}_i$ do not form a 1-dimensional embedding. Luckily, a clever application of the Shopping on Amazon Inequality will allow us to address this.

3.2 An $O(\log n)$-approximation for Sparsest Cut

The rounding algorithm we use for sparsest cut is described in full by algorithm 1. To analyze algorithm 1, we break the argument down into three steps.

(1) First, we will show that the ℓ_1-cost of the embedding associating each $i \in V$ to $\boldsymbol{y}_i \in \mathbb{R}^h$, as computed by Bourgain's theorem in step (3), is at most an $O(\log n)$ factor away from the optimal LP cost.

(2) We then show that the 1-dimensional embedding assigning each $i \in V$ to $z_i \geq 0$, as computed in step (4), has ℓ_1-cost at most that given by the embedding assigning $i \in V$ to $\boldsymbol{y}_i \in \mathbb{R}^h$.

(3) Finally, we will prove that step (5) implements the algorithmic side of Theorem 5. Hence, the cut produced by step (5) will have sparsity at most the ℓ_1-cost given by the values $z_i \geq 0$, for each $i \in V$.

Putting together the pieces listed above will then yield the proof of our main Theorem 3.

3.3 Analysing the Rounding Algorithm

We now proceed with the analysis. Our first step is to show that the ℓ_1-cost of the embedding computed by Bourgain's theorem is at most an $O(\log n)$ factor away from the optimal LP cost. The way we do this uses a very typical argument: using the embedding computed by Bourgain's theorem, produce a feasible solution to (`Metric-LP`) whose objective value is at most α-factor away from the optimal, where α is the distortion of the embedding.

Algorithm 1.

Input: a graph $G = (V, E)$ with edge weights $c_e > 0$, and demand pairs $\mathcal{D} = \{(s_i, t_i)\}_{i=1,\ldots,k}$.

Do: The following.

1. Solve the LP relaxation (Metric-LP) for LP optimal values $\{x_e\}_{e\in E}$ and $\{d_i\}_{i=1,\ldots,k}$.
2. Compute the shortest path metric (V, d_G) where each edge $e \in E$ is weighted by LP values x_e.
3. Apply Bourgain's theorem to embed $(V, d_G) \hookrightarrow \big(\{\boldsymbol{y}_i\}_{i\in V}, \ell_1\big)$.
4. For each $i \in V$, let $\hat{z}_i = y_i(a)$ where $a \in [h]$ is given by

$$a = \operatorname*{argmin}_{a=1,\ldots,h} \frac{\sum_{ij\in E} c_{ij} \cdot |y_i(a) - y_j(a)|}{\sum_{i=1}^{k} |y_{s_i}(a) - y_{t_i}(a)|},$$

   and $y_i(a)$ is the a-th coordinate of $\boldsymbol{y}_i$. Set $z_i = \hat{z}_i - \hat{z}_{\min}$ where $\hat{z}_{\min} = \min_{i\in V} \hat{z}_i$.

5. Sort $i \in V$ in ascending order of $z_1 \le z_2 \le \ldots \le z_n$, then compute the cut $S_\ell \overset{\text{def}}{=} \{1, \ldots, \ell\}$ with minimum sparsity amongst $\ell \in [n]$.

Output: The cut S_ℓ.

Figure 1: The rounding algorithm for sparsest cut due to Linial, London & Rabinovich

Claim 8. *Let* $\big(\{x_e\}_{e\in E}, \{d_i\}_{i=1,\ldots,k}\big)$ *be an LP optimal solution for* (Metric-LP) *achieving LP cost* opt*, and* $\big(\{\boldsymbol{y}_i\}_{i\in V}, \ell_1\big)$ *be computed as in step (3) of algorithm 1 with distortion* $\alpha \ge 1$*. Then the assignment*

$$\hat{x}_e = \alpha \cdot \|\boldsymbol{y}_u - \boldsymbol{y}_v\|_1 \qquad \forall\, e = (u, v) \in E$$
$$\hat{d}_i = \alpha \cdot \|\boldsymbol{y}_{s_i} - \boldsymbol{y}_{t_i}\|_1 \qquad \forall\, i = 1, \ldots, k$$

satisfies the following.

$$\frac{\sum_{ij\in E} c_{ij} \cdot \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_1}{\sum_{i=1}^{k} \|\boldsymbol{y}_{s_i} - \boldsymbol{y}_{t_i}\|_1} = \frac{\sum_{e\in E} c_e \cdot \hat{x}_e}{\sum_{i=1}^{k} \hat{d}_i} \le \alpha \cdot \mathsf{opt}$$

Proof. We first check that the sum of distances between demands is at least 1. To see this, note that

$$\sum_{i=1}^{k} \|\boldsymbol{y}_{s_i} - \boldsymbol{y}_{t_i}\|_1 \ge \frac{1}{\alpha} \cdot \sum_{i=1}^{k} d_G(s_i, t_i) \ge \frac{1}{\alpha} \cdot \sum_{i=1}^{k} d_i = \frac{1}{\alpha} \tag{2}$$

where the first inequality follows by Bourgain's theorem returning an α-distortion embedding, and the last equality follows by the fact that $\{d_i\}_{i=1,\ldots,k}$ are feasible for (Metric-LP) and hence they sum to 1. The second inequality follows by the constraint $\sum_{e\in p} x_e \ge d_i$. This is important, if p denotes the shortest path between s_i and t_i in G when edges $e \in E$ are weighted by x_e, then the shortest path distance is precisely $\sum_{e\in p} x_e$ for p. Because x_e are feasible for (Metric-LP), the constraint

$$\sum_{e\in p} x_e \ge d_i$$

holds. Continuing from eq. (2), we have that

$$1 \le \alpha \cdot \sum_{i=1}^{k} \|\boldsymbol{y}_{s_i} - \boldsymbol{y}_{t_i}\|_1 = \hat{d}_i \tag{3}$$

Using this, we can compute the cost like so. First note that

$$\frac{\sum_{ij\in E} c_{ij}\cdot\|\boldsymbol{y}_i-\boldsymbol{y}_j\|_1}{\sum_{i=1}^k\|\boldsymbol{y}_{s_i}-\boldsymbol{y}_{t_i}\|_1}=\frac{\sum_{ij\in E} c_{ij}\cdot\alpha\cdot\|\boldsymbol{y}_i-\boldsymbol{y}_j\|_1}{\sum_{i=1}^k\alpha\cdot\|\boldsymbol{y}_{s_i}-\boldsymbol{y}_{t_i}\|_1}=\frac{\sum_{e\in E} c_e\cdot\hat{x}_e}{\sum_{i=1}^k\hat{d}_i}$$

thus deriving the first equality in the claim. We then bound the RHS via the following.

$$\frac{\sum_{e\in E} c_e\cdot\hat{x}_e}{\sum_{i=1}^k\hat{d}_i}\le\sum_{e\in E} c_e\cdot\hat{x}_e=\sum_{ij\in E} c_{ij}\cdot\alpha\cdot\|\boldsymbol{y}_i-\boldsymbol{y}_j\|_1\le\alpha\cdot\sum_{ij\in E} c_{ij}\cdot d_G(i,j)=\alpha\cdot\sum_{e\in E} c_e\cdot x_e=\alpha\cdot\mathsf{opt}$$

where the first inequality holds by the fact that the denominator is at most one, the second equality holds by definition of $\hat{x}_e$, the third inequality holds as the embedding computed by Bourgain's theorem is non-expanding, the fourth equality holds by the fact that the shortest path cost between an adjacent $(i,j)=e\in E$ is the weight of that edge x_e. The final equality follows as $\{x_e\}_{e\in E}$ are LP optimal. □

Our next step is to demonstrate that using $\big(\{\boldsymbol{y}_i\}_{i\in V},\ell_1\big)$, one can compute a 1-dimensional ℓ_1-metric space $\big(\{z_i\}_{i\in V},\ell_1\big)$ whose ℓ_1-cost is no larger than that given by $\boldsymbol{y}_i$ for each $i\in V$. The following claim shows this with a clever use of the Shopping on Amazon Inequality.

Claim 9. *Given $\boldsymbol{y}_i\in\mathbb{R}^h$ for each $i\in V$, there exists a choice of z_i such that $z_i\ge 0$ for each $i\in V$, and*

$$\frac{\sum_{ij\in E} c_{ij}\cdot|z_i-z_j|}{\sum_{i=1}^k|z_{s_i}-z_{t_i}|}\le\frac{\sum_{ij\in E} c_{ij}\cdot\|\boldsymbol{y}_i-\boldsymbol{y}_j\|_1}{\sum_{i=1}^k\|\boldsymbol{y}_{s_i}-\boldsymbol{y}_{t_i}\|_1}\,.$$

Proof. Note that

$$\frac{\sum_{ij\in E} c_{ij}\cdot\|\boldsymbol{y}_i-\boldsymbol{y}_j\|_1}{\sum_{i=1}^k\|\boldsymbol{y}_{s_i}-\boldsymbol{y}_{t_i}\|_1}=\frac{\sum_{ij\in E} c_{ij}\sum_{a=1}^h|y_i(a)-y_j(a)|}{\sum_{i=1}^k\sum_{a=1}^h|y_{s_i}(a)-y_{t_i}(a)|}=\frac{\sum_{a=1}^h\sum_{ij\in E} c_{ij}\cdot|y_i(a)-y_j(a)|}{\sum_{a=1}^h\sum_{i=1}^k|y_{s_i}(a)-y_{t_i}(a)|}$$

and now, by the Shopping on Amazon Inequality, we have

$$\min_{a\in[h]}\frac{\sum_{ij\in E} c_{ij}\cdot|y_i(a)-y_j(a)|}{\sum_{i=1}^k|y_{s_i}(a)-y_{t_i}(a)|}\le\frac{\sum_{a=1}^h\sum_{ij\in E} c_{ij}\cdot|y_i(a)-y_j(a)|}{\sum_{a=1}^h\sum_{i=1}^k|y_{s_i}(a)-y_{t_i}(a)|}$$

Setting $\hat{z}_i=y_i(a)$ for $a=\operatorname{argmin}_{a\in[h]}\frac{\sum_{ij\in E} c_{ij}\cdot|y_i(a)-y_j(a)|}{\sum_{i=1}^k|y_{s_i}(a)-y_{t_i}(a)|}$ thus ensures

$$\frac{\sum_{ij\in E} c_{ij}\cdot|\hat{z}_i-\hat{z}_j|}{\sum_{i=1}^k|\hat{z}_{s_i}-\hat{z}_{t_i}|}\le\frac{\sum_{ij\in E} c_{ij}\cdot\|\boldsymbol{y}_i-\boldsymbol{y}_j\|_1}{\sum_{i=1}^k\|\boldsymbol{y}_{s_i}-\boldsymbol{y}_{t_i}\|_1}\,.$$

Now, to ensure that there is a choice of z_i such that $z_i\ge 0$ for all $i\in V$ and the ℓ_1-cost associated to the z_i's is at most that given by $\boldsymbol{y}_i$, let $z_i\stackrel{\text{def}}{=}\hat{z}_i-\hat{z}_{\min}$ where $\hat{z}_{\min}\stackrel{\text{def}}{=}\min_{i\in V}\hat{z}_i$. By construction $z_i\ge 0$ for each $i\in V$. Further, note that the ℓ_1 costs are equivalent. In particular,

$$\frac{\sum_{ij\in E} c_{ij}\cdot|z_i-z_j|}{\sum_{i=1}^k|z_{s_i}-z_{t_i}|}=\frac{\sum_{ij\in E} c_{ij}\cdot|(\hat{z}_i-\hat{z}_{\min})-(\hat{z}_j-\hat{z}_{\min})|}{\sum_{i=1}^k|(\hat{z}_{s_i}-\hat{z}_{\min})-(\hat{z}_{t_i}-\hat{z}_{\min})|}=\frac{\sum_{ij\in E} c_{ij}\cdot|\hat{z}_i-\hat{z}_j|}{\sum_{i=1}^k|\hat{z}_{s_i}-\hat{z}_{t_i}|}\,,$$

as required. □

Finally we are ready to show that, given a 1-dimensional embedding, there exists an algorithm which outputs a cut whose sparsity is at most the ℓ_1-cost of the embedding. We restate the algorithmic portion of Theorem 5 in the following claim.

Claim 10. *There exists an efficient algorithm $\mathcal{A}$ such that given $\boldsymbol{z} \in \mathbb{R}^n_{\geq 0}$, $\mathcal{A}$ outputs a cut $S \subseteq V$ satisfying*

$$\varphi_G(S) \leq \frac{\sum_{ij \in E} c_{ij} \cdot |z_i - z_j|}{\sum_{i=1}^k |z_{s_i} - z_{t_i}|} .$$

Proof. To produce a cut S, let us first consider the following randomized procedure as a "thought experiment":

(1) Let $z_{\min} = \min_{i \in V} z_i$ and $z_{\max} = \max_{i \in V} z_i$
(2) Sample $r \sim [z_{\min}, z_{\max}]$ uniformly at random
(3) Output the random threshold cut $S_r \stackrel{\text{def}}{=} \{i \in V : z_i \leq r\}$.

Our goal will be to show that

$$\frac{\mathbb{E}_r[c(E(S_r, \bar{S}_r))]}{\mathbb{E}_r[|D(S_r, \bar{S}_r)|]} \leq \underbrace{\frac{\sum_{ij \in E} c_{ij} \cdot |z_i - z_j|}{\sum_{i=1}^k |z_{s_i} - z_{t_i}|}}_{\psi_G(\boldsymbol{z})} \tag{4}$$

since if this is true, then we are done as

$$\begin{aligned}
\frac{\mathbb{E}_r[c(E(S_r, \bar{S}_r))]}{\mathbb{E}_r[|D(S_r, \bar{S}_r)|]} \leq \psi_G(\boldsymbol{z}) \quad &\iff \quad \mathbb{E}_r[c(E(S_r, \bar{S}_r))] \leq \psi_G(\boldsymbol{z}) \cdot \mathbb{E}_r[|D(S_r, \bar{S}_r)|] \\
&\iff \quad \mathbb{E}_r[c(E(S_r, \bar{S}_r))] - \psi_G(\boldsymbol{z}) \cdot \mathbb{E}_r[|D(S_r, \bar{S}_r)|] \leq 0 \\
&\iff \quad \mathbb{E}_r\left[c(E(S_r, \bar{S}_r)) - \psi_G(\boldsymbol{z}) \cdot |D(S_r, \bar{S}_r)|\right] \leq 0 .
\end{aligned}$$

Consequently, there must exist some $r \in [z_{\min}, z_{\max}]$ such that

$$c(E(S_r, \bar{S}_r)) - \psi_G(\boldsymbol{z}) \cdot |D(S_r, \bar{S}_r)| \leq 0 \quad \iff \quad \frac{c(E(S_r, \bar{S}_r))}{|D(S_r, \bar{S}_r)|} \leq \psi_G(\boldsymbol{z}) .$$

To show eq. (4), let us compute the numerator and the denominator. For the numerator, note that

$$\mathbb{E}_r\left[c(E(S_r, \bar{S}_r))\right] = \mathbb{E}_r\left[\sum_{ij \in E} c_{ij} \cdot \mathbf{1}\{(i,j) \text{ cut by } S_r\}\right] = \sum_{ij \in E} c_{ij} \cdot \mathbb{P}\Big((i,j) \text{ cut by } S_r\Big) .$$

Notice that S_r cuts (i,j) if and only if the sampled r lands between z_i and z_j. Because $r \sim [z_{\min}, z_{\max}]$ uniformly at random, this occurs with probability $\frac{|z_i - z_j|}{z_{\max} - z_{\min}}$ and hence

$$\sum_{ij \in E} c_{ij} \cdot \mathbb{P}\Big((i,j) \text{ cut by } S_r\Big) = \sum_{ij \in E} \frac{c_{ij} \cdot |z_i - z_j|}{z_{\max} - z_{\min}}$$

Computing the denominator is identical, and we get

$$\mathbb{E}_r\left[|D(S_r, \bar{S}_r)|\right] = \sum_{i=1}^k \frac{|z_{s_i} - z_{t_i}|}{z_{\max} - z_{\min}}$$

Putting the numerator and denominator together, we get

$$\frac{\mathbb{E}_r[c(E(S_r, \bar{S}_r))]}{\mathbb{E}_r[|D(S_r, \bar{S}_r)|]} = \frac{\sum_{ij \in E} \frac{c_{ij} \cdot |z_i - z_j|}{z_{\max} - z_{\min}}}{\sum_{i=1}^k \frac{|z_{s_i} - z_{t_i}|}{z_{\max} - z_{\min}}} = \frac{\sum_{ij \in E} c_{ij} \cdot |z_i - z_j|}{\sum_{i=1}^k |z_{s_i} - z_{t_i}|}$$

as required.

Finally, to get the algorithm, notice that there are at most n unique cuts that S_r for r sampled from $[z_{\min}, z_{\max}]$ can produce, and they correspond to sorting the vertices $i \in V$ in ascending order of z_i, and then considering cuts that take the form $\{1, \ldots, \ell\} \subseteq V$. □

3.4 Completing the Analysis

Using the above claims, we can now complete the analysis.

Proof of Theorem 3. Suppose algorithm 1 outputs the cut $S \subseteq V$. The sparsity of S satisfies:

$$\varphi_G(S) \leq \frac{\sum_{ij \in E} c_{ij} \cdot |z_i - z_j|}{\sum_{i=1}^{k} |z_{s_i} - z_{t_i}|} \leq \frac{\sum_{ij \in E} c_{ij} \cdot \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_1}{\sum_{i=1}^{k} \|\boldsymbol{y}_{s_i} - \boldsymbol{y}_{t_i}\|_1} \leq \alpha \cdot \mathsf{opt} \leq O(\log n) \cdot \mathsf{opt}$$

where the first inequality follows by Claim 10, the second inequality follows by Claim 9, and third inequality follows by Claim 8. The last step follows as Bourgain's theorem outputs an $\alpha \leq O(\log n)$ distortion embedding. □

References

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