CS 598CSC: Topics in Graph Algorithms	Lecture date: 09/10/2024
Instructor: Chandra Chekuri	Scribe: CC

1 Recent Shortest Path Algorithms

Shortest path problems in graphs are fundamental algorithmic primitives. For non-negative edge lengths, Dijkstra's algorithm for single-source shortest path (SSPP) can be implemented in $O(m + n \log n)$ time and there is also a complicated $O(m + n)$-time algorithm in the RAM model for integer length [Tho99]. The algorithmic complexity of the all-pairs shortest path problem (APSP) is quite fundamental and is an active area of study in fine-grained complexity — see `https://algorithm-wiki.csail.mit.edu/wiki/APSP`. The situation for negative lengths saw breakthrough progress recently. The well-known Shimbel-Bellman-Ford-Moore algorithm for SSSP in directed graphs with potential negative length edges runs in $O(mn)$ time on a directed graph and either finds a negative length cycle or correctly outputs the shortest path distances from the source. In the past there have been scaling algorithms that improved the running time. Recently there have been two important results. First, Bernstein, Nanongkai and Wulff-Nielsen describe a randomized Las Vegas algorithm that runs in $O(m \log^8 n \log W)$-time where weights are assumed to be integer and W is the absolute value of least negative length edge [BNWN22]; subsequent work reduced the number of log factors [BCF23]. Second, a strongly polynomial time algorithm that runs in randomized $\tilde{O}(mn^{8/9})$ due to Fineman [Fin24]; see recent improvement [HJQ24].

The scaling algorithm of [BNWN22], surprisingly, uses the notion of low-diameter graph decomposition which was used mainly in distributed algorithms and approximation algorithms in the past (it is a concept from the 80's). The recent breakthrough work on min-cost flow that achieved almost-linear running time is based on dynamic algorithms for shortest paths with negative lengths. There is a synergy and close connection between algorithms for flows and cuts in graphs and shortest paths. In the past the main use of these ideas was in approximation algorithms and distributed algorithms but recent work has shown the utility of these notions for developing fast algorithms. In order to develop some intuition and background we will first go over some results on (multicommodity) flows and cuts via metric embeddings and develop necessary tools along the way.

The rest of the notes below are borrowed from the instructor's previous notes on approximation algorithms which were scribed by students — see the course webpage for proper attribution.

2 s-t mincut via LP Rounding and Maxflow-Mincut

Let $G = (V, E)$ be a directed graph with edge costs $c : E \to \mathbb{R}_+$. Let $s, t \in V$ be distinct vertices. The s-t mincut problem is to find the cheapest set of edges $E' \subseteq E$ such that there is no s-t path in $G - E'$. An s-t cut is often also defined as $\delta^+(S)$ for some $S \subset V$ where $s \in S, t \in V - S$. Suppose E' is an s-t cut. Let S be the set of nodes reachable from s in $G - E'$, then $\delta(S) \subseteq E'$ and moreover $\delta(S)$ is an s-t cut. Thus, it suffices to focus on such limited type of cuts, however in some more general settings it is useful to keep these notions separate.

It is well-known that s-t mincut can be computed efficiently via s-t maximumflow which also establishes the maxflow-mincut theorem. This is a fundamental theorem in combinatorial optimization with many direct and indirect applications.

Theorem 1. *Let $G = (V, E)$ be a directed graph with rational edge capacities $c : E \to \mathbb{Q}_+$ and let $s, t \in V$ be distinct vertices. Then the s-t maximum flow value in G is equal to the s-t minimum cut value and both can be computed in strongly polynomial time. Further, if c is integer valued then there exists an integer-valued maximum flow.*

The proof of the preceding theorem is typically established via the augmenting path algorithm for computing a maximum flow. Here we take a different approach to finding an s-t cut via an LP relaxation whose dual can be seen as the the maxflow LP.

Suppose we want to find an s-t mincut. We can write it as an integer program as follows. For each edge $e \in E$ we have a boolean variable $x_e \in \{0, 1\}$ to indicate whether we cut e. The constraint is that for any path $P \in \mathcal{P}_{s,t}$ (here $\mathcal{P}_{s,t}$ is the set of all s-t paths) we must choose at least on edge from P. This leads to the following IP.

$$
\begin{aligned}
\min \sum_{e \in E} c(e) x_e & & & \\
\sum_{e \in P} x_e & \geq & 1 & \quad P \in \mathcal{P}_{s,t} \\
x_e & \in & \{0, 1\} & \quad e \in E.
\end{aligned}
$$

The LP relaxation is obtained by changing $x_e \in \{0, 1\}$ to $x_e \geq 0$ since we can omit the constraints $x_e \leq 1$. We note that the LP has an exponential number of constraints, however, we have an efficient separation oracle since it corresponds to computing the shortest s-t path. The LP can be viewed as assigning lengths to the edges such that the shortest path between s and t according to the lengths is at least 1. This is a fractional relaxation of the cut.

Rounding the LP: We will prove that the LP relaxation can be rounded without any loss! The rounding algorithm is described below.

Theta-Rounding(G, s, t)

1. Solve LP to obtain fractional solution y
2. For each $v \in V$ let $d_y(s, v)$ be the shortest path distance from s to v according to edge lengths y_e.
3. Pick θ uniformly at random from $(0, 1)$
4. Output $E' = \delta^+(B(s, \theta))$ where $B(s, \theta) = \{v \mid d_y(s, v) \leq \theta\}$ is the ball of radius θ around s

It is easy to see that the algorithm outputs a valid s-t cut since $d_y(s, t) \geq 1$ by feasibility of the LP solution y and hence $t \notin B(s, \theta)$ for any $\theta < 1$.

Lemma 2. *Let $e = (u, v)$ be an edge.* $\mathbf{P}[e$ *is cut by algorithm*$] \leq y(u, v)$.

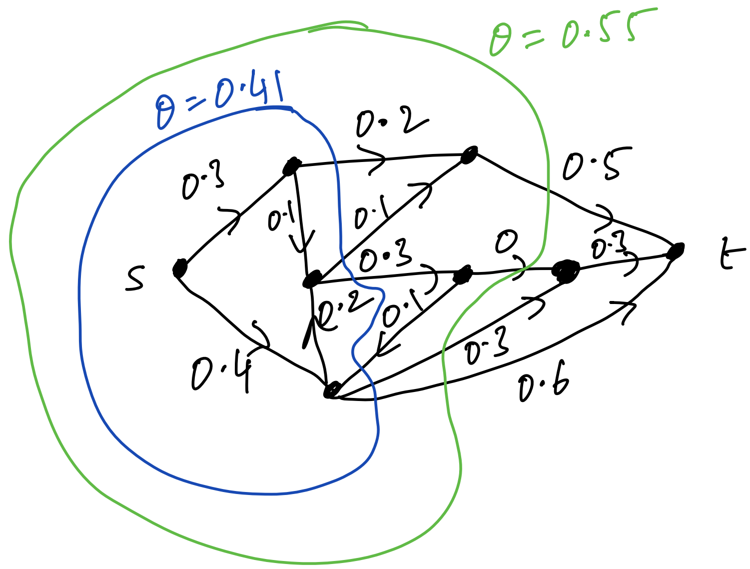


Figure 1: Example of fractional solution to s-t LP and balls of different radii.

Proof. An edge $e = (u, v)$ is cut iff $d_y(s, u) \leq \theta < d_y(s, v)$. Hence the edge is not cut if $d_y(s, v) \leq d_y(s, u)$. If $d_y(s, v) > d_y(s, u)$ we have $d_y(s, v) - d_y(s, u) \leq y(u, v)$. Since θ is chosen uniformly at random from $(0, 1)$ the probabilty that θ lies in the interval $[d_y(s, u), d_y(s, v)]$ is at most $y(u, v)$. ∎

Corollary 3. *The expected cost of the cut output by the algorithm is at most* $\sum_e c(e) y_e$.

The preceding corollary shows that there is an integral cut whose cost is at most that of the LP relaxation which implies that the LP relaxation yields an optimum solution. The algorithm can be easily derandomized by trying "all possible value of θ". What does this mean? Once we have y we compute the shortest path distances from s to each vertex v. We can think of these distances as producing a *line embedding* where we place s at 0 and each vertex v at $d_y(s, v)$. The only interesting choices for θ are given by the n values of $d_y(s, v)$ and one can try each of them and the corresponding cut and find the cheapest one. It is guaranteed to be at most $\sum_e c(e) y_e$.

What is the dual LP? We write it down below and you can verify that it is the path version of the maxflow!

$$
\begin{aligned}
\max_{P \in \mathcal{P}_{s,t}} \; & z_P \\
\sum_{P : e \in P} z_P & \leq c(e) \quad e \in E \\
z_P & \geq 0 \quad P \in \mathcal{P}_{s,t}.
\end{aligned}
$$

Thus, we have seen a proof of the maxflow-mincut theorem via LP rounding of a relaxation for the s-t cut problem.

A compact LP via distance variables: The path based LP relaxation for the s-t mincut problem is natural and easy to formulate. We can also express shortest path constraints via distance variables. We first write a bigger LP than necessary via variables $d(u,v)$ for all ordered pairs of vertices (hence there are n^2 variables). We need triangle inequality constraints to enforce that $d(u,v)$ values respect shortest path distances.

$$
\begin{aligned}
\min_{(u,v)\in E} c(u,v)d(u,v) & & & \\
d(u,v) + d(v,w) - d(u,w) &\geq 0 & & u,v,w \in V \\
d(s,t) &\geq 1 & & \\
d(u,v) &\geq 0 & & (u,v) \in V \times V
\end{aligned}
$$

Although the preceding LP is wasteful in some ways it is quite generic and can be used for many cut problems where we are interested in distances between multiple pairs of vertices.

Now we consider a more compact LP formulation. We have two types of variables, $x(u,v)$ for each edge $(u,v) \in E$ and d_v variables for each $v \in V$ to indicate distances from s.

$$
\begin{aligned}
\min_{(u,v)\in E} c(u,v)x(u,v) & & & \\
d_v &\leq d_u + x(u,v) & & (u,v) \in E \\
d_t &\geq 1 & & \\
d_v &\geq 0 & & v \in V \\
x(u,v) &\geq 0 & & (u,v) \in E
\end{aligned}
$$

Exercise 1. *Write the dual of the above LP and see it as the standard edge-based flow formulation for s-t maximum flow.*

3 Multicut and Approximation via Randomized Decomposition

In the MULTICUT problem, we are given a graph $G = (V, E)$, a capacity function that assigns a capacity c_e to each edge e, and a set of pairs $(s_1,t_1),...,(s_k,t_k)$. The MULTICUT problem asks for a minimum capacity set of edges $F \subseteq E$ such that removing the edges in F disconnects s_i and t_i, for all i. MULTICUT is NP-Hard even on trees. We describe an $O(\log k)$ approximation algorithm for MULTICUT which, as a corollary, also proves a multicommodity flow-cut gap result. It turns out that the bound of $O(\log k)$ is tight in general graphs. For planar graphs one can get an $O(1)$-approximation and flow-cut gap. These results are for undirected graphs. The situation is more complicated in directed graphs and we will discuss that later. The rest of the section is about undirected graphs.

We start by describing an LP formulation for the problem. For each edge e, we have a variable d_e. We interpret each variable d_e as a distance label for the edge. Let $\mathcal{P}_{s_i,t_i}$ denote the set of all paths between s_i and t_i. We have the following LP for the problem:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e d_e \\
\text{s.t.} \quad & \\
& \sum_{e \in p} d_e \geq 1 \quad p \in \mathcal{P}_{s_i,t_i}, 1 \leq i \leq k \\
& \quad d_e \geq 0 \qquad\qquad\quad e \in E
\end{aligned}
$$

The LP assigns distance labels to edges so that, on each path p between s_i and t_i, the distance labels of the edges on p sum up to at least one. Note that, even though the LP can have exponentially many constraints, we can solve the LP in polynomial time using the ellipsoid method and the following separation oracle. Given distance labels d_e, we set the length of each edge to d_e and, for each pair (s_i, t_i), we compute the length of the shortest path between s_i and t_i and check whether it is at least one. If the shortest path between s_i and t_i has length smaller than one, we have a violated constraint. Conversely, if all shortest paths have length at least one, the distance labels define a feasible solution.

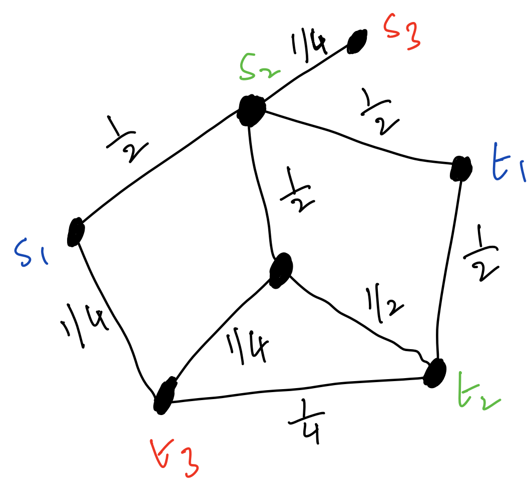


Figure 2: Example of fractional solution to a Multicut instance. Note that even when (s_3, t_2) is not an input pair, the LP solution separates them fractionally.

We also consider the dual of the previous LP. For each path p between any pair (s_i, t_i) we have a dual variable f_p. We interpret each variable f_p as the amount of flow between s_i and t_i that is routed along the path p. We have the following dual LP:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{k} \sum_{p \in \mathcal{P}_{s_i,t_i}} f_p \\
\text{s.t.} \quad & \\
& \sum_{p:\, e \in p} f_p \leq c_e \qquad\qquad e \in E(G) \\
& \qquad f_p \geq 0 \quad p \in \mathcal{P}_{s_1,t_1} \cup \ldots \cup \mathcal{P}_{s_k,t_k}
\end{aligned}
$$

The dual is an LP formulation for the Maximum Throughput Multicommodity Flow problem. In the Maximum Throughput Multicommodity Flow problem, we have k different commodities. For each i, we want to route commodity i from the source s_i to the destination t_i. Each commodity must satisfy flow conservation at each vertex other than its source and its destination. Additionally, the total flow routed on each edge must not exceed the capacity of the edge. The goal is to maximize the sum of the commodities routed.

The dual LP tries to assign an amount of flow f_p to each path p so that the total flow on each edge is at most the capacity of the edge (the flow conservation constraints are automatically satisfied). Note that the endpoints of the path p determine which kind of commodity is routed along the path.

Exercise 2. *Write the* MULTICUT *LP and its dual in a compact form with polynomially many constraints.*

4 Upper Bound on the Integrality Gap

In this section, we will show that the integrality gap of the LP is $O(\log k)$ using a randomized rounding algorithm due to Calinescu, Karloff, and Rabani [CKR01]. The first algorithm that achieved an $O(\log k)$-approximation for MULTICUT is due to Garg, Vazirani, and Yannakakis [GVY93] (see [Vaz13] and [WS11]), and it is based on the region growing technique introduced by Leighton and Rao [LR99]. The reason that we choose to present the randomized rounding algorithm is due to its future application for metric embeddings.

Let $B_d(v, r)$ denote the ball of radius r centered at the vertex v in the metric induced by the distance labels d_e.

CKR-RandomPartition:

Solve the LP to get the distance labels d_e
Pick θ uniformly at random from $[0, 1/2)$
Pick a random permutation σ on $\{1, 2, ..., k\}$
for $i = 1$ to k
$\quad V_{\sigma(i)} = B_d(s_{\sigma(i)}, \theta) \setminus \bigcup_{j<i} V_{\sigma(j)}$
Output $\bigcup_{i=1}^{k} \delta(V_i)$

Lemma 4. *CKR-RandomPartition correctly outputs a feasible multicut for the given instance.*

Proof. Let F be the set of edges output by the algorithm. Suppose F is not a feasible multicut. Then there exists a pair of vertices (s_i, t_i) such that there is a path between s_i and t_i in $G - F$. Therefore there exists a j such that V_j contains s_i and t_i. Since $V_j \subseteq B_d(s_j, \theta)$, both s_i and t_i are contained in the ball of radius θ centered at s_j. Consequently, the distance between s_j and s_i is at most θ and the distance between s_j and t_i is at most θ. By the triangle inequality, the distance between s_i and t_i is at most 2θ. Since θ is smaller than $1/2$, it follows that the distance between s_i and t_i is smaller than one. This contradicts the fact that the distance labels d_e are a feasible solution for the LP. Therefore F is a multicut, as desired. ∎

Lemma 5. *The probability that an edge e is cut is at most $2H_k d_e$, where H_k is the k-th harmonic number and d_e is the distance label of the edge e.*

Proof. Fix an edge $e = (u, v)$. Let:

$$L_i = \min\{d(s_i, u), d(s_i, v)\}$$

$$R_i = \max\{d(s_i, u), d(s_i, v)\}$$

We may assume without loss of generality that $L_1 \leq L_2 \leq ... \leq L_k$ (be reindexing the pairs as needed). See Fig 3.

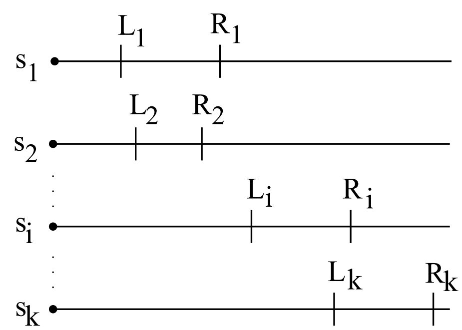


Figure 3: For a fixed edge $e = (u, v)$ we renumber the pairs such that $L_1 \leq L_2 \leq ... \leq L_k$.

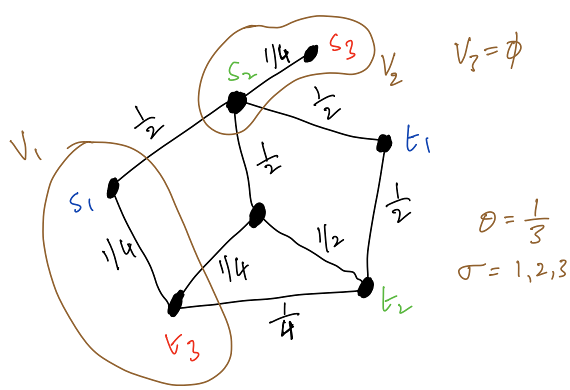


Figure 4: Rounding example.

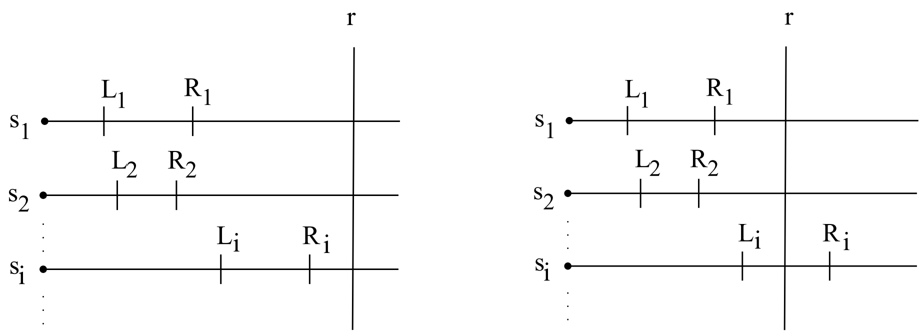
Let A_i be the event that the edge e is cut *first* by s_i. More precisely, A_i is the event that $|V_i \cap \{u, v\}| = 1$ and $|V_j \cap \{u, v\}| = 0$ for all j such that $\sigma(j) < \sigma(i)$. Note that $|V_i \cap \{u, v\}| = 1$ simply says that s_i cuts the edge e. If s_i is the first to cut the edge e, for all j that come before i in the permutation, neither u nor v can be in V_j (if only one of u and v is in V_j, s_j cuts the edge e; if both u and v are in V_j, s_i cannot cut the edge e).

Note that the event that the edge e is cut is the union of the disjoint events $A_1, ..., A_k$. Therefore we have:

$$\mathbf{P}[e \text{ is cut}] = \sum_i \mathbf{P}[A_i].$$

Let us fix $r \in [0, 1/2)$ and consider $\mathbf{P}[A_i \mid \theta = r]$. Note that s_i cuts the edge e only if one of u, v is inside the ball of radius r centered at s_i and the other is outside the ball. Differently said, s_i cuts the edge only if $r \in [L_i, R_i)$:

$$\mathbf{P}[A_i \mid \theta = r] = 0 \qquad \text{if } r \notin [L_i, R_i)$$



Now suppose that $r \in [L_i, R_i)$. Let us fix $j < i$ and suppose j comes before i in the permutation (that is, $\sigma(j) < \sigma(i)$). Recall that, since $j < i$, we have $L_j \leq L_i \leq r$. Therefore at least one of u, v is inside the ball of radius r centered at s_j. Consequently, s_i cannot be the first to cut the edge e. Therefore s_i is the first to cut the edge e only if $\sigma(i) < \sigma(j)$ for all $j < i$. See Fig 4. Since σ is a random permutation, i appears before j for all $j < i$ with probability $1/i$. Therefore we have:

$$\mathbf{P}[A_i \mid \theta = r] \leq \frac{1}{i} \qquad \text{if } r \in [L_i, R_i)$$

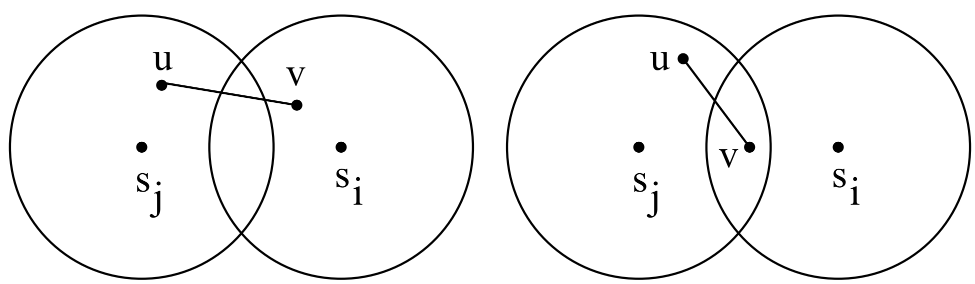


Figure 5: If $\sigma(j) < \sigma(i)$, s_i cannot be the first to cut the edge $e = (u, v)$. On the left s_j also cuts the edge. On the right s_j captures both end points and therefore s_i cannot cut it.

Since θ was selected uniformly at random from the interval $[0, 1/2)$, and independently from σ, we have:

$$\mathbf{P}[A_i] \leq \frac{1}{i} \ \mathbf{P}[\theta \in [L_i, R_i)] = \frac{2}{i} \ (R_i - L_i)$$

By the triangle inequality, $R_i \leq L_i + d_e$. Therefore:

$$\mathbf{P}[A_i] \leq \frac{2d_e}{i}$$

Consequently,

$$\mathbf{P}[e \text{ is cut}] = \sum_i \mathbf{P}[A_i] \leq 2H_k d_e.$$

■

Corollary 6. *The integrality gap of the* MULTICUT *LP is* $O(\log k)$*.*

Proof. Let F be the set of edges outputted by the Randomized Rounding algorithm. For each edge e, let χ_e be an indicator random variable equal to 1 if and only if the edge e is in F. As we have already seen,

$$\mathbf{E}\,\chi_e = \mathbf{P}[\chi_e = 1] \leq 2H_k d_e$$

Let $c(F)$ be a random variable equal to the total capacity of the edges in F. We have:

$$\mathbf{E}\,c(F) = \mathbf{E}\sum_e c_e\chi_e = \sum_e c_e\,\mathbf{P}[\chi_e] \leq 2H_k \sum_e c_e d_e = 2H_k\ \mathrm{OPT}_{\mathrm{LP}}$$

Consequently, there exists a set of edges F such that the total capacity of the edges in F is at most $2H_k$ $\mathrm{OPT}_{\mathrm{LP}}$. Therefore $\mathrm{OPT} \leq 2H_k\ \mathrm{OPT}_{\mathrm{LP}}$, as desired. ■

Corollary 7. *The algorithm achieves an* $O(\log k)$*-approximation (in expectation) for the* MULTICUT *problem.*

Proof. As we have already seen,

$$\mathbf{E}\,c(F) \leq 2H_k\ \mathrm{OPT}_{\mathrm{LP}}$$

where F is the set of edges output by the algorithm and $c(F)$ is the total capacity of the edges in F. Since $\mathrm{OPT}_{LP} \leq \mathrm{OPT}$,

$$\mathbf{E}\,c(F) \leq 2H_k\ \mathrm{OPT} = O(\log k)\mathrm{OPT}$$

■

Remark 8. *The expected cost analysis can be used to obtain an algorithm, via repetition, a randomized algorithm that ouputs an* $O(\log k)$*-approximation with high probability. The algorithm can also be derandomized but it is not straight forward. As we remarked there is an alternative deterministic* $O(\log k)$*-approximation algorithm via region growing.*

Flow-Cut Gap: Recall that when $k = 1$ we have the well-known maxflow-mincut theorem. The integrality gap of the standard LP for MulitCut is the same as the relative gap between flow and cut when k is arbitrary. The upper bound on the integrality gap gives an upper bound on the gap.

Corollary 9. *We have:*

$$\max_{\textit{m.c. flow } f} |f| \leq \min_{\textit{multicut } C} |C| \leq O(\log k) \left(\max_{\textit{m.c. flow } f} |f| \right)$$

where $|f|$ represents the value of the multicommodity flow f, and $|C|$ represents the capacity of the multicut C.

Proof. Let OPT_{LP} denote the total capacity of an optimal (fractional) solution for the MULTICUT LP. Let OPT_{dual} denote the flow value of an optimal solution for the dual LP. Since OPT_{LP} is a lower bound on the capacity of the minimum (integral) multicut, we have:

$$\max_{\text{m.c. flow } f} |f| = \text{OPT}_{dual} = \text{OPT}_{\text{LP}} \leq \min_{\text{multicut } C} |C|$$

As we have already seen, we have:

$$\min_{\text{multicut } C} |C| \leq 2H_k \, \text{OPT}_{\text{LP}} = 2H_k \, \text{OPT}_{dual} = 2H_k \left(\max_{\text{m.c. flow } f} |f| \right)$$

■

5 Lower Bound on the Integrality Gap

In this section, we will show that the integrality gap of the LP is $\Omega(\log k)$. That is, we will give a MULTICUT instance for which the LP gap is $\Omega(\log k)$. This is a non-trivial lower bound and requires the use of expander graphs and their properties. But before that we observe that there is an integrality gap of 2 on trees. This is implicitly obtained by the fact that VERTEX COVER in general graphs reduces to MULTICUT in trees. Take G to be a star graph with center r connected to n leaves $v_1, v_2, \ldots, v_n$; each edge has cost 1. Consider the demand edges to be all pairs (v_i, v_j). It is clear that the optimum integer cut is $n - 1$. On the other hand a feasible fractional solution is to assign $1/2$ to each edge. Thus the integrality gap is $2(1 - 1/n)$.

5.1 Expander Graphs

Definition 1. *A graph $G = (V, E)$ is an α-edge-expander if, for any subset S of at most $|V|/2$ vertices, the number of edges crossing the cut $(S, V \backslash S)$ is at least $\alpha|S|$.*

Note that the complete graph K_n is a $(|V|/2)$-edge-expander. However, the more interesting expander graphs are also sparse. Cycles and grids are examples of graphs that are very poor expanders.

Definition 2. *A graph G is d-regular if every vertex in G has degree d.*

Note that 2-regular graphs consist of a collection of edge disjoint cycles and therefore they have poor expansion. However, for any $d \geq 3$, there exist d-regular graphs that are very good expanders.

Theorem 10. *For every $d \geq 3$ there exists an infinite family of d-regular 1-edge-expanders.*

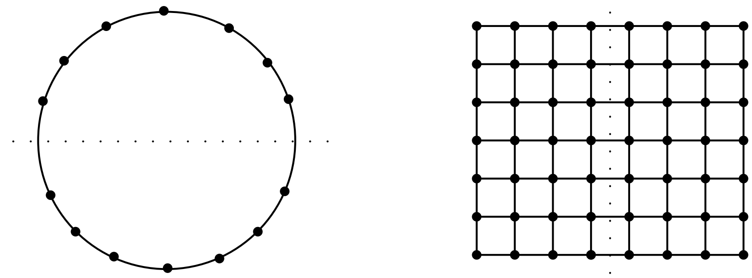


Figure 6: The top half of the cycle has $|V|/2$ vertices and only two edges crossing the cut. The left half of the grid has roughly $|V|/2$ vertices and only $\sqrt{|V|}$ edges crossing the cut.

We will only need the following special case of the previous theorem.

Theorem 11. *There exists a universal constant $\alpha > 0$ and an integer n_0 such that, for all even integers $n \geq n_0$, there exists an n-vertex, 3-regular α-edge-expander.*

Proof Idea. The easiest way to prove this theorem is using the probabilistic method. The proof itself is beyond the scope of this lecture[1]. The proof idea is the following.

Let's fix an even integer n. We will generate a 3-regular random graph G by selecting three random perfect matchings on the vertex set $\{1, 2, ..., n\}$ (recall that a perfect matching is a set of edges such that every vertex is incident to exactly one of these edges). We select a random perfect matching as follows. We maintain a list of vertices that have not been matched so far. While there is at least one vertex that is not matched, we select a pair of distinct vertices u, v uniformly at random from all possible pairs of unmatched vertices. We add the edge (u, v) to our matching and we remove u and v from the list. We repeat this process three times (independently) to get three random matchings. The graph G will consist of the edges in these three matchings. Note that G is actually a 3-regular multigraph since it might have parallel edges (if the same edge is in at least two of the matchings). There are two properties of interest: (1) G is a simple graph and (2) G is an α-edge-expander for some constant $\alpha > 0$. If we can show that G has both properties with positive probability, it follows that there exists a 3-regular α-edge-expander (if no graph is a 3-regular α-edge-expander, the probability that our graph G has both properties is equal to 0).

It is not very hard to show that the probability that G does not have property (1) is small. To show that the probability that G does not have property (2) is small, for each set S with at most $n/2$ vertices, we estimate the expected number of edges that cross the cut $(S, V \backslash S)$ (e.g., we can easily show that $|\delta(S)| \geq |S|/2$). Using tail inequalities (e.g., Chernoff bounds), we can show that the probability that $|\delta(S)|$ differs significantly from its expectation is extremely small (i.e., small enough so that the sum – taken over all sets S – of these probabilities is also small) and we can use the union bound to get the desired result. ∎

Note that explicit constructions of d-regular expanders are also known. Margulis [Mar73] gave an infinite family of 8-regular expanders. There are many explicit construction by now and it is a very important topic of study — we refer the reader to the survey on expanders by Hoory, Linial

[1] A more accurate statement is that the calculations are a bit involved and not terribly interesting for us.

and Wigderson [HLW06]. The vertex set of a graph G_n in Margulis' construction is $\mathbb{Z}_n \times \mathbb{Z}_n$, where $\mathbb{Z}_n$ is the set of all integers mod n. The neighbors of a vertex (x, y) in G_n are $(x+y, y)$, $(x-y, y)$, $(x, y+x)$, $(x, y-x)$, $(x+y+1, y)$, $(x-y+1, y)$, $(x, y+x+1)$, and $(x, y-x+1)$ (all operations are mod n). Another example is the following infinite family of 3-regular expanders. For each prime p, we have a 3-regular graph G_p. The vertex set of G_p is $\mathbb{Z}_p$. The neighbors of a vertex x in G_p are $x+1$, $x-1$, and x^{-1} (as before, all operations are mod p; x^{-1} is the inverse of x mod p, and we define the inverse of 0 to be 0)[2].

We conclude this section with the following observations (they will be very useful in showing the $\Omega(k)$ lower bound on the integrality gap of the LP).

Claim 12. *Let G be an n-vertex d-regular α-edge-expander, for some constants $d \geq 3$ and $\alpha > 0$. Then the diameter of G is $\Theta(\log n)$.*

Proof. For any two vertices u and v, let $dist(u, v)$ denote the length of a shortest path between u and v (the length of a path is the number of edges on the path). Let's fix a vertex s. Let L_i be the set of all vertices v such that $dist(s, v)$ is at most i. Now let's show that $(1+\alpha/d)|L_{i-1}| \leq |L_i| \leq d|L_{i-1}|$. Clearly, $|L_1| = d$ (since s has degree d). Therefore we may assume that $i > 1$. Every vertex in L_i is in L_{i-1} or it has a neighbor in L_{i-1}. Therefore it suffices to bound $|L_i \backslash L_{i-1}|$.

Note that any vertex in L_{i-1} has at least one neighbor in L_{i-1}. Therefore the vertices in L_{i-1} have at most $(d-1)|L_{i-1}|$ neighbors outside of L_{i-1}. Consequently, $|L_i| \leq d|L_{i-1}|$.

Now one of L_{i-1}, $V \backslash L_{i-1}$ has at most $|V|/2$ vertices. Let's assume without loss of generality that L_{i-1} has at most $|V|/2$ vertices (the other case is symmetric). Let $A = L_{i-1}$ and let B be the set of all vertices in $V \backslash L_{i-1}$ that have a neighbor in L_{i-1} (note that $|L_i| = |A| + |B|$). Let F be the set of all edges that cross the cut $(L_{i-1}, V \backslash L_{i-1})$. Now let's look at the bipartite graph $H = (A, B, F)$. Since G is an α-edge-expander, we have $|F| \geq \alpha|A|$. Moreover, $|F| = \sum_{v \in B} d_H(v)$, where $d_H(v)$ is the degree of v in H. Since $d_H(v)$ is at most d, we have $\alpha|A| \leq |F| \leq d|B|$. Therefore we have:

$$L_i = |A| + |B| \geq (1+\alpha/d)|A| = (1+\alpha/d)|L_{i-1}|$$

It follows by induction that $d(1+\alpha/d)^{i-1} \leq |L_i| \leq d^i$. Therefore $dist(s, v)$ is $O(\log n)$ for all v and there exists a vertex v such that $dist(s, v)$ is $\Omega(\log n)$. Since this is true for any s, it follows that the diameter of G is $\Theta(\log n)$. ■

Claim 13. *Let G be an n-vertex 3-regular α-edge-expander and let $B(v, i)$ be the set of all vertices u such that there is a path between u and v with at most i edges. For any vertex v, $|B(v, \log_3 n/2)| \leq \sqrt{n}$.*

Proof. Note that $B(v, \log_3 n/2)$ is the set of all vertices w such that $dist(v, w)$ is at most $\log_3 n/2$. As we have seen in the proof of the previous claim, we have $|B(v, \log_3 n/2)| \leq 3^{\log_3 n/2} = \sqrt{n}$. ■

5.2 The Multicut Instance

Let n_0, α be as in Theorem 7. Let $n \geq n_0$ and let G be an n-vertex 3-regular α-edge-expander. For each edge e in G, we set the capacity c_e to 1. Now let $X = \{(u, v) | u \notin B(v, \log_3 n/2)\}$. The pairs

[2] Note that, unlike Margulis' construction, this construction is not very explicit since we don't know how to generate large primes deterministically.

in X will be the pairs (s_i, t_i) that we want to disconnect. Let (G, X) be the resulting Multicut instance.

Claim 14. *There exists a feasible fractional solution for (G, X) of capacity $O(n/\log n)$.*

Proof. Let $d_e = 2/\log_3 n$, for all e. Note that, since G is 3-regular, G has $3n/2$ edges. Therefore the total capacity of the fractional solution is

$$\sum_e d_e = \frac{3n}{2} \cdot \frac{2}{\log_3 n} = \frac{3n}{\log_3 n}$$

Therefore we only need to show that the solution is feasible. Let (u, v) be a pair in X. Let's consider a path p between u and v. Since u is not in $B(v, \log_3 n/2)$, the path p has more than $\log_3 n/2$ edges (recall that $B(v, i)$ is the set of all vertices u such that there is a path between u and v with at most i edges). Consequently,

$$\sum_{e \in p} d_e > \frac{\log_3 n}{2} \cdot \frac{2}{\log_3 n} = 1$$

■

Claim 15. *Any integral solution for (G, X) has capacity $\Omega(n)$.*

Proof. Let F be an integral solution for (G, X). Let $V_1, ..., V_h$ be the connected components of $G - F$. Fix an i and let v be an arbitrary vertex in the connected component V_i. Note that, for any u in V_i, there is a path between v and u with at most $\log_3 n/2$ edges (if not, (u, v) is a pair in X which contradicts the fact that removing the edges in F disconnects every pair in X). Therefore V_i is contained in $B(v, \log_3 n/2)$. It follows from Claim 13 that $|V_i| \le \sqrt{n}$. Since G is an α-edge-expander and $|V_i| \le |V|/2$, we have $|\delta(V_i)| \ge \alpha|V_i|$, for all i. Consequently,

$$|F| = \frac{1}{2} \sum_{i=1}^{h} |\delta(V_i)| \ge \frac{\alpha}{2} \sum_{i=1}^{h} |V_i| = \frac{\alpha n}{2}$$

Therefore F has total capacity $\Omega(n)$ (recall that every edge has unit capacity). ■

Theorem 16. *The integrality gap of the* Multicut *LP is $\Omega(\log k)$.*

Proof. Note that $k = |X| = O(n^2)$. It follows from claims 10 and 11 that the LP has integrality gap $\Omega(\log n) = \Omega(\log k)$, as desired. ■

References

[BCF23] Karl Bringmann, Alejandro Cassis, and Nick Fischer. Negative-weight single-source shortest paths in near-linear time: Now faster! In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 515–538. IEEE, 2023.

[BNWN22] Aaron Bernstein, Danupon Nanongkai, and Christian Wulff-Nilsen. Negative-weight single-source shortest paths in near-linear time. In *2022 IEEE 63rd annual symposium on foundations of computer science (FOCS)*, pages 600–611. IEEE, 2022.

[CKR01] G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 8–16. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2001.

[Fin24] Jeremy T Fineman. Single-source shortest paths with negative real weights in $\tilde{O}(mn^{8/9})$ time. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 3–14, 2024.

[GVY93] N. Garg, V.V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 698–707. ACM New York, NY, USA, 1993.

[HJQ24] Yufan Huang, Peter Jin, and Kent Quanrud. Faster single-source shortest paths with negative real weights via proper hop distance. *arXiv preprint arXiv:2407.04872*, 2024.

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

[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.

[Mar73] G.A. Margulis. Explicit constructions of expanders. *Problemy Peredaci Informacii*, 9(4):71–80, 1973.

[Tho99] Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *Journal of the ACM (JACM)*, 46(3):362–394, 1999.

[Vaz13] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

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