Princeton Univ. F'24 COS 597B: Recent Advances in Graph Algorithms

Lecture 7: A "combinatorial" max flow algorithms in $n^{2+o(1)}$ time using directed expander hierarchy

Lecturer: *Huacheng Yu* Scribe: *Ijay Narang*

1 Introduction

In these notes, we explore the weighted push-relabel algorithm combined with expander hierarchies to efficiently compute maximum flows in directed graphs. The classical push-relabel algorithm faces challenges in dense graphs due to excessive relabeling. By decomposing the graph into (DAGs and expander levels), the improved algorithm achieves a time complexity of $O(n^{2+o(1)})$. Our core problem is:

Find good w such that

- There exists an approximate max-flow such that every flow path P has weight $w(P) = \sum_{e \in P} w(e) \leq h \leq n^{1+o(1)}$
- $\sum_e 1/w(e) \leq n^{1+o(1)}$

In terms of notation, we have that if $F \subseteq E$, then $deg_F(u)$ is the number of edges in F incident to u in both directions. We define $vol_F(S) = \sum_{u \in S} deg_F(u)$. We say that a graph is a A **φ-expander** where, for every non-empty subset $S \subset V$, the edge boundary satisfies:

$$\frac{|E(S, S^c)|}{\min(\text{vol}_F(S), \text{vol}_F(S^c))} \geq \varphi,$$

We can use this notion to extend our typical definition of expanders to directed graphs as follows: An **expander hierarchy** for a directed graph $G = (V, E)$ is a sequence of edge sets $H = (D, X_1, \ldots, X_\eta)$ such that:

- D contains the **DAG edges** connecting strongly connected components (SCCs).
- Each X_i is a **φ-expander** with respect to the remaining graph after removing edges from higher levels:
$$G_i = G \setminus \bigcup_{j<i} X_j.$$

For the purpose of this problem, we want (in our expander decomposition) $|X_i| \leq \tilde{O}(\phi^{i-1} m) \Rightarrow \eta \leq O(\log_{\frac{1}{\phi}} m)$.

2 Expanders + DAG + ϕ-expanding X_2

Our goal is to show that there exists an approximate max flow f such that the flow path uses $\tilde{O}(\frac{1}{\phi})$ edges in X_2 and less than $\tilde{O}(\frac{1}{\phi})$ in each expander.

A lemma of importance to us (that we will not prove here) is the following:

Lemma 1 *Let $F \subseteq E$, and G be $\phi-$expanding, then any flow instance (Δ, ∇), such that for all u, we have that $\Delta(u) \leq deg_F(u)$ and $\nabla(u) \leq deg_F(u)$, then there exists a flow f where:*

- *$cong(f) \leq \tilde{O}(\frac{1}{\phi})$*
- *every flow path has $\tilde{O}(\frac{1}{\phi})$ edges in F*

Now, consider the flow problem on f and X_1 and X_2, note that by an application of the lemma, we can obtain a flow path f_1 such that every path of f uses $\tilde{O}(\frac{1}{\phi})$ edges in X_2 with congestion $\tilde{O}(\frac{1}{\phi})$. Now, consider a flow path in f_1 and an expander U_i. If length of P in $U_i \leq \tilde{O}(\frac{1}{\phi})$, then we do not need to reroute and we are done. Otherwise, let $l = \tilde{O}(\frac{1}{\phi})$, consider the first l vertices on P in U_i $(u_1, u_2, \cdots, u_l)$ and the last l vertices on P in U_i, $v_1, v_2, \cdots v_l$ in U_i. Observe that when we route $1/l$ units of flow from each of the u_j to the v_j on every long path P implies that for every vertex u, we have that:

- $\Delta(u) \leq 1/l \cdot vol(u) \cdot cong(f_1)$
- $\nabla(u) \leq 1/l \cdot vol(u) \cdot cong(f_1)$

It thus follows that Lemma 1 implies that (Δ, ∇) can be routed by f_2 such that

- $cong(f_2) \leq \tilde{O}(\frac{1}{\phi}) \cdot 1/l \cdot cong(f_1)$
- Every path uses $\tilde{O}(\frac{1}{\phi})$ edges

Therefore, the congestion after rerouting is upper bounded by $cong(f_1) + cong(f_2) \leq (1 + \tilde{O}(\frac{1}{\phi} \cdot 1/l)) \cdot cong(f_1) \leq (1 + 1/polylog(n)) \cdot cong(f_1)$ Keeping this in mind (and assuming the existence of the hierarchy), we can first find a topological order with respect to the DAG and expanders from the hierarchy. Then, by setting the weight of each edge to $w(u, v) = |\tau(u) - \tau(v)|$ we get that the total weight of a flow path meets our desired result, as there are $\tilde{O}(|U_i|/\phi)$ expander edges in U_i, $\tilde{O}(\frac{n}{\phi})$ backward edges and $\tilde{O}(\frac{n}{\phi})$ forward edges. Thus, the only remaining question is why must a hierarchy $(D, X_1, X_2, \cdots X_\eta)$ exist? This is the focus of the next section.

3 Building an Expander Hierarchy

In this section, we show how to construct the expander hierarchy of the input graph that was critical towards deriving the weight function fof the push-relabel algorithm. The original paper formally defines this as follows:

Theorem 2 *([1]) There is a randomized algorithm that, given an n-vertex graph with capacities (G, c), with high probability constructs a $1/n^{o(1)}$-expander hierarchy $H = (D, X_1, \ldots, X_\eta)$ of (G, c) with $\eta = O(\log n)$ in $n^{2+o(1)}$ time.*

The capacities c are all unit capacities, as it is not too difficult to generalize from there. A more general variant of the above theorem is below. Note that the first theorem follows immediately when we choose $\phi = \exp\left(-\frac{\log n}{(\log\log n)^{1/3}}\right)$.

Theorem 3 *([1]) Given an n-vertex graph (G, c) and a parameter $0 < \phi < 2^{-\omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)}$ sufficiently small, there is a randomized $n^{2+o(1)}\phi^3$ time algorithm that with high probability constructs a $\phi/n^{o(1)}$-expander hierarchy $H = (D, X_1, \ldots, X_\eta)$ of (G, c) with $\eta = O(\log n)$.*

The high-level methodology towards constructing each X_i is to repeatedly find cuts in G and removing edges from one of the directions (i.e., $E_G(S, S)$ or $E_G(S, S)$ for some S) which disconnects the two sides of the cuts. We provide more detail in the next section.

The naive approach for computing $(D, X_1, X_2, \cdots, X_\eta)$ is to greedily find sparse cuts in G, and to move edges in the sparse direction to X_2 and recurse on both sides. That is, via standard expander decomposition, we can get three edge sets D, X_1, X_2 such that D is a DAG, X_1 is ϕ-expanding in G_1, and $|X_2|$ is small (on the order of ϕm). To construct the second level and onward, one immediate idea is to simply do expander decomposition with respect to the volume induced by X_2. If the returned edge set X_3 happens to be a subset of X_2, then we can set $X_2 \leftarrow X_2 \setminus X_3$ and continue to run expander decomposition on X_3. As the number of edges in the terminal set decreases roughly by a factor of ϕ each time, after $O(\log_{1/\phi} n)$ iterations we will get the desired expander hierarchy.

The issue is that the edge set X_3 we need to cut when doing expander decomposition with respect to the volume induced by X_2 may not be a subset of X_2. It might necessarily be the case that X_3 includes edges from X_1 or even from D. When this happens, $X_3 \nsubseteq X_2$ the graph $G_1 = G \setminus X_{>1}$ in which X_1 is expanding changes which decreases the well-connectivity of G_1 and make X_1 no longer expanding (breaking the hierarchy).

Intuitively, we can resolve this by immediately removing $X_3 \setminus X_2$ from G_1 (note that it suffices to remove $X_3 \setminus X_2$)and then further refine the strongly connected components of G_1 into smaller pieces so that X_1 is still expanding in this graph. That is, we repeatedly find the set S such that $min\{|E(S, \bar{S})|, |E(\bar{(S)}, S)|\} < \phi min\{vol_{X_2}(S), vol_{X_2}(\bar{S})\}$ and move edges in the more sparse direction to X_3 and once again recurse on the two sides. Because of the charging argument, we have that we moved at most $\tilde{O}(\phi \cdot vol(X_2)) \leq \tilde{O}(\phi \cdot |X_2|)$ to X_3.

The issue with this scheme is that we may move edges in X_1 or D to X_3, breaking the expanders in the first level.Thus, in order to fix this issue, we iteratively fix the expansion in all levels and move a set of edges R to X_3. This procedure works as follows:

1. For every expander in X_1, while there exists a cut less than $\phi/4$-sparse (3/4 of the edges are in R), we move all the edges in the sparse direction.

2. For X_2, we iteratively find sparse cuts $w < \phi/4$ and move the edges to X_3

In the paper, they show that there are less than $O(\phi \cdot |R|)$ edges moved to X_3. Though we will not show the actual proof itself, the idea is that by changing the new edges moved to X_2 to be edges in R, we can show that we only move $O(|R|)$ edges to X_2.

References

[1] Aaron Bernstein, Joakim Blikstad, Thatchaphol Saranurak, and Ta-Wei Tu. Maximum flow by augmenting paths in $n^{2+o(1)}$ time, 2024.