Faster Sparse Minimum Cost Flow by Electrical Flow Localization

Kyriakos Axiotis* Aleksander Mądry† Adrian Vladu‡

Abstract

We give an $\widetilde{O}(m^{3/2-1/762}\log(U+W))$ time algorithm for minimum cost flow with capacities bounded by U and costs bounded by W. For sparse graphs with general capacities, this is the first algorithm to improve over the $\widetilde{O}(m^{3/2}\log^{O(1)}(U+W))$ running time obtained by an appropriate instantiation of an interior point method [Daitch-Spielman, 2008].

Our approach is extending the framework put forth in [Gao-Liu-Peng, 2021] for computing the maximum flow in graphs with large capacities and, in particular, demonstrates how to reduce the problem of computing an electrical flow with general demands to the same problem on a sublinear-sized set of vertices—even if the demand is supported on the entire graph. Along the way, we develop new machinery to assess the importance of the graph's edges at each phase of the interior point method optimization process. This capability relies on establishing a new connections between the electrical flows arising inside that optimization process and vertex distances in the corresponding effective resistance metric.

*MIT, kaxiotis@mit.edu
†MIT, madry@mit.edu
‡CNRS & IRIF, Université de Paris, vladu@irif.fr

Contents

1 Introduction

In the last decade, continuous optimization has proved to be an invaluable tool for designing graph algorithms, often leading to significant improvements over the best known combinatorial algorithms. This has been particularly true in the context of flow problems—arguably, some of the most prominent graph problems [DS08, CKM+11, LRS13, Mąd13, She13, KLOS14, LS14, Pen16, Mąd16, CMSV17, She17b, She17a, ST18, LS20, KLS20, AMV20, vdBLN+20, vdBLL+21]. Indeed, these developments have brought a host of remarkable improvements in a variety of regimes, such as when seeking only approximate solutions, or when the underlying graph is dense. However, most of these improvements did not fully address the challenge of seeking *exact* solutions in *sparse* graphs. Fortunately, the improvements for that regime eventually emerged [Mąd13, Mąd16, CMSV17, LS20, KLS20, AMV20]. They still suffered though from an important shortcoming: they all had a polynomial running time dependency in the graph's capacities, and hence—in contrast to the classical combinatorial algorithms—they did not yield efficient algorithms in the presence of arbitrary capacities. Recently, Gao, Liu and Peng [GLP21] have finally changed this state of affairs by providing the first *exact* maximum flow algorithm to break the $\widetilde{O}\left(m^{3/2}\log^{O(1)} U\right)$ barrier for sparse graphs with *general* capacities (bounded by U). Their approach, however, crucially relies on a preconditioning technique that is specific to the maximum flow problem and, in particular, having an s-t demand—rather than a general one. As a result, the corresponding improvement held only for that particular problem.

In this paper, we demonstrate how to circumvent these limitations and provide the first algorithm that breaks the $\widetilde{O}\left(m^{3/2}\log^{O(1)}(U+W)\right)$ barrier for the *minimum cost flow* problem in *sparse* graphs with general demands, capacities (bounded by U), and costs (bounded by W). This algorithm runs in time $\widetilde{O}\left(m^{3/2-1/762}\log(U+W)\right)$.

1.1 Previous work

In 2013, Mądry [Mąd13] presented the first running time improvement to the maximum flow problem since the $\widetilde{O}\left(m\sqrt{n}\log U\right)$ algorithm of [GR98] in the regime of sparse graphs with small capacities. To this end, he presented an algorithm that runs in time $\widetilde{O}\left(m^{10/7}\mathrm{poly}(U)\right)$, where U is a bound on edge capacities, breaking past the $\widetilde{O}\left(m^{3/2}\right)$ running time barrier that has for decades resisted improvement attempts. The main idea in that work was to use an interior point method with an improved number of iterations guarantee that was delivered via use of an adaptive re-weighting of the central path and careful perturbations of the problem instance. Building on this framework, a series of subsequent works [Mąd16, LS20, KLS20] has brought the runtime of sparse max flow down to $\widetilde{O}\left(m^{4/3}\text{poly}(U)\right)$. (With the most recent of these works crucially relying on nearly-linear time ℓ_p flows [KPSW19].) In parallel [CMSV17, AMV20], the running time of the more general minimum cost flow problem was reduced to $\widetilde{O}\left(m^{4/3}\text{poly}(U)\log W\right)$, where W is a bound on edge costs.

However, even though these algorithms offer a significant improvement when U is relatively small, the question of whether there exists an algorithm faster than $\widetilde{O}\left(m^{3/2}\log^{O(1)} U\right)$ for sparse graphs with general capacities remained open. In fact, a polynomial dependence on capacities or costs seems inherent in the central path re-weighting technique used in all the aforementioned works. Recently, [GLP21] finally made progress on this question by developing an algorithm for the maximum flow problem that runs in time $\widetilde{O}\left(m^{3/2-1/328}\log U\right)$. The source of improvement here was different from previous works, in the sense that it was not based on decreasing the number of iterations of the interior point method. Instead, it was based on devising a data structure to solve the dynamically changing Laplacian system required by the interior point method in sublinear time per iteration.

The new approach put forth by [GLP21], despite being quite different to the prior ones, still leaned on the preconditioning approach of [Mądry16], as well as on other properties that are specific to the maximum flow problem. For this reason, this improvement did not extend to the minimum cost flow problem with general capacities, for which the fastest known runtime was still $\widetilde{O}\left(m\log(U+W) + n^{1.5}\log^2(U+W)\right)$ [vdBLL+21] and $\widetilde{O}(m^{3/2}\log^{O(1)}(U+W))$ [DS08] in the sparse regime.

1.2 Our result

In this work, we give an algorithm for the minimum cost flow problem with a running time of $\widetilde{O}\left(m^{3/2-1/762}\log(U+W)\right)$. This is the first improvement for sparse graphs with general capacities over [DS08], which runs in time $\widetilde{O}\left(m^{3/2}\log^{O(1)}(U+W)\right)$. Specifically, we prove that:

Theorem 1.1. *Given a graph $G(V,E)$ with edge costs $\boldsymbol{c} \in \mathbb{Z}^m_{[-W,W]}$, a demand $\boldsymbol{d} \in \mathbb{R}^n$, and capacities $\boldsymbol{u} \in \mathbb{Z}^m_{(0,U]}$, there exists an algorithm that with high probability runs in time $\widetilde{O}\left(m^{3/2-1/762}\log(U+W)\right)$ and returns a flow $\boldsymbol{f} \in [\boldsymbol{0}, \boldsymbol{u}]$ in G such that $\boldsymbol{f}$ routes the demand $\boldsymbol{d}$ and the cost $\langle \boldsymbol{c}, \boldsymbol{f}\rangle$ is minimized.*

1.3 High level overview of our approach

As we build on the approach presented in [GLP21], we first briefly overview some of the key ideas introduced there that will also be relevant for our discussion. The maximum flow interior point method by [Mąd16] works by, repeatedly over $\widetilde{O}(\sqrt{m})$ steps, taking an electrical flow step that is a multiple of

$$\widetilde{\boldsymbol{f}} = \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{B}^{\top}\boldsymbol{1}_{st}\,,$$

where $\boldsymbol{L} = \boldsymbol{B}^{\top}\boldsymbol{R}^{-1}\boldsymbol{B}$ is a Laplacian matrix and $\boldsymbol{r}$ are resistances that change per step. However, $\widetilde{\boldsymbol{f}}$ has m entries and takes $\widetilde{O}(m)$ to compute, which gives the standard $\widetilde{O}\left(m^{3/2}\right)$ bound. To go beyond this, [GLP21] show that it suffices to compute $\widetilde{\boldsymbol{f}}$ for only a *sublinear* number of high-congestion entries of $\widetilde{\boldsymbol{f}}$, where congestion is defined as $\boldsymbol{\rho} = \sqrt{\boldsymbol{r}}\widetilde{\boldsymbol{f}}$. By known linear sketching results, these edges can be detected by computing the inner product $\langle \boldsymbol{q}, \boldsymbol{\rho}\rangle$ for a small number of randomly chosen vectors $\boldsymbol{q} \in \mathbb{R}^m$. Crucially, given a vertex subset $C \subseteq V$ of sublinear size that contains s

and t, this inner product can be equivalently written as the following sublinear-sized inner product

$$\langle \boldsymbol{q}, \boldsymbol{\rho} \rangle = \left\langle \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right), SC^+ \boldsymbol{d} \right\rangle, \tag{1}$$

where $SC := SC(G, C)$ is the *Schur complement* of G onto C, $\boldsymbol{d}$ is equal to $\boldsymbol{B}^\top \boldsymbol{1}_{st}$, and $\boldsymbol{\pi}^C(\cdot)$ is a *demand projection* onto C. Therefore, the problem is reduced to maintaining two quantities: $\boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right)$ and $(SC(G, C))^+ \boldsymbol{d}$ in sublinear time per operation. The latter is computed by using the dynamic Schur complement data structure of [DGGP19], and the former can be maintained by a careful use of random walks.

We now describe our approach. Instead of using the interior point method formulation of [Mąd16] which only applies to the maximum flow problem, we use the one by [AMV20] for the, more general, minimum cost flow problem.

There are now several obstacles to making this approach work by maintaining the quantity $\langle \boldsymbol{q}, \boldsymbol{\rho} \rangle$:

Preconditioning A significant difference between [Mąd16] and [AMV20] is that while the former is able to guarantee that the magnitude of the electrical potentials computed in each step is inversely proportional to the duality gap, meaning that a large duality gap implies potential embeddings of low stretch, no such preconditioning method is known for minimum cost flow. In fact, [AMV20] used demand perturbations to show that a *weaker* bound on the potentials can be achieved, which was still sufficient for their purposes. Unfortunately, this bound is not strong enough to be used in the analysis of [GLP21].

In order to alleviate this issue, we completely remove preconditioning from the picture by only requiring a bound on the *energy* of the electrical potentials (instead of their magnitude). In particular, given an approximate demand projection $\widetilde{\boldsymbol{\pi}}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right)$, identity (1) is used to detect congested edges. In [GLP21], there is a uniform upper bound on the entries of the potential embedding $\boldsymbol{\phi} = SC^+ \boldsymbol{d}$ because of preconditioning, thus the error in (1) can be bounded by

$$\left\| \widetilde{\boldsymbol{\pi}}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right) - \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right) \right\|_1 \left\| \boldsymbol{\phi} \right\|_\infty .$$

As we do not have a good bound on $\left\| \boldsymbol{\phi} \right\|_\infty$, we instead use an alternative upper bound on the error:

$$\sqrt{\mathcal{E}_{\boldsymbol{r}} \left(\widetilde{\boldsymbol{\pi}}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right) - \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right) \right)} \sqrt{E_{\boldsymbol{r}}(\boldsymbol{\phi})},$$

where $\mathcal{E}_{\boldsymbol{r}}(\cdot)$ gives the energy to route a demand with resistances $\boldsymbol{r}$, and $E_{\boldsymbol{r}}(\cdot)$ gives the energy of a potential embedding with resistances $\boldsymbol{r}$. As the standard interior point method step satisfies $E_{\boldsymbol{r}}(\boldsymbol{\phi}) \leq 1$, all our efforts focus on ensuring that

$$\sqrt{\mathcal{E}_{\boldsymbol{r}} \left(\widetilde{\boldsymbol{\pi}}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right) - \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right) \right)} \leq \varepsilon \tag{2}$$

for some error parameter ε. One issue is the fact that the energy depends on the current resistances, therefore even if at some point the error of the demand projection is low, after a few iterations it might increase because of resistance changes. We deal with this issue by taking the stability of resistances along the central path into account. This allows us to upper bound how much this error increases after a number of iterations. The resistance stability lemma is a generalization of the one used in [GLP21].

Unfortunately, even though (2) seems like the right type of guarantee, it is unclear how to ensure that it is always true. Specifically, it involves efficiently computing the hitting probabilities from some vertex v to C in an appropriate norm, which ends up being non-trivial. Instead, we show that the following *weaker* error bound can be ensured with high probability:

$$\left|\left\langle \widetilde{\boldsymbol{\pi}}^{C}\left(\boldsymbol{B}^{\top}\frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right)-\boldsymbol{\pi}^{C}\left(\boldsymbol{B}^{\top}\frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right),\boldsymbol{\phi}\right\rangle\right|\leq\varepsilon\,, \tag{3}$$

where $\boldsymbol{\phi}$ is a *fixed* potential vector with $E_{\boldsymbol{r}}(\boldsymbol{\phi})\leq 1$. Interestingly, this guarantee is still sufficient for our purposes.

Costs and general demand There is a fundamental obstacle to using the approach of [GLP21] once edge costs are introduced. In particular, for the maximum flow problem, the demand pushed by the electrical flow in each iteration is an s-t demand, so—up to scaling—it is always constant. In minimum cost flow on the other hand, the augmenting flow is a multiple of $\boldsymbol{c}-\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{B}^{\top}\boldsymbol{c}$. Here it is not possible to locate a sublinear number of congested edges just by looking at the electrical flow term $\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{B}^{\top}\boldsymbol{c}$, as there might be significant cancellations with $\boldsymbol{c}$. We instead use the following equivalent form: $\frac{\frac{1}{\boldsymbol{s}^{+}}-\frac{1}{\boldsymbol{s}^{-}}}{\boldsymbol{r}}-\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{B}^{\top}\frac{\frac{1}{\boldsymbol{s}^{+}}-\frac{1}{\boldsymbol{s}^{-}}}{\boldsymbol{r}}$, which allows us to ignore the first term because it is small and concentrate on the electrical flow term. One issue that arises is the fact that the demand vector $\boldsymbol{B}^{\top}\frac{\frac{1}{\boldsymbol{s}^{+}}-\frac{1}{\boldsymbol{s}^{-}}}{\boldsymbol{r}}$ now depends on slacks, and as a result changes throughout the interior point method. This issue can be handled relatively easily.

A more significant issue concerns the vertex sparsifier. In fact, the vertex sparsifier framework around which [GLP21] is based only accepts demands that are supported on the vertex set C of the sparsifier. As $|C|$ is sublinear in n, this only captures demands with sublinear support, one such example being max flow with support 2. However, our demand vector $\boldsymbol{B}^{\top}\frac{\frac{1}{\boldsymbol{s}^{+}}-\frac{1}{\boldsymbol{s}^{-}}}{\boldsymbol{r}}$ in general will be supported on n vertices. Even though it might seem impossible to get around this issue, we show that the special structure of C allows us to push the demand to a small number of vertices. More specifically, we show that if one projects all of the demand onto C, the flow induced by this new demand will not differ much from the one with the original demand. Concretely, given a Laplacian system $\boldsymbol{L}\boldsymbol{\phi}=\boldsymbol{d}$, we decompose it into two systems $\boldsymbol{L}\boldsymbol{\phi}^{(1)}=\boldsymbol{\pi}^{C}(\boldsymbol{d})$ and $\boldsymbol{L}\boldsymbol{\phi}^{(2)}=\boldsymbol{d}-\boldsymbol{\pi}^{C}(\boldsymbol{d})$, where $\boldsymbol{\pi}^{C}(\boldsymbol{d})$ is the projection of $\boldsymbol{d}$ onto C. Intuitively, the latter system computes the electrical flow to push all demands to C, and the former to serve this C-supported demand. We show that, as long as C is a *congestion reduction subset* (as it is also the case in [GLP21]), $\boldsymbol{\phi}^{(2)}$ has negligible contribution in the electrical flow, thus it can be ignored. More specifically, in Section 4.1 we prove the following lemma:

Lemma 4.3. *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$ and Laplacian $\boldsymbol{L}$, a β-congestion reduction subset C, and a demand $\boldsymbol{d}=\delta\boldsymbol{B}^{\top}\frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}$ for some $\delta>0$ and $\boldsymbol{q}\in[-1,1]^{m}$. Then, the potential embedding defined as*

$$\boldsymbol{\phi}=\boldsymbol{L}^{+}\left(\boldsymbol{d}-\boldsymbol{\pi}^{C}\left(\boldsymbol{d}\right)\right)$$

has congestion $\delta\cdot\widetilde{O}\left(1/\beta^{2}\right)$, i.e. $\left\|\frac{\boldsymbol{B}\boldsymbol{\phi}}{\sqrt{\boldsymbol{r}}}\right\|_{\infty}\leq\delta\cdot\widetilde{O}\left(1/\beta^{2}\right)$.

Now, for computing $\boldsymbol{\phi}^{(1)}$, we need to get an approximate estimate of $\boldsymbol{\pi}^C(\boldsymbol{d})$. Even though the most natural approach would be to try to maintain $\boldsymbol{\pi}^C(\boldsymbol{d})$ under vertex insertions to C, this approach has issues related to the fact that our error guarantee is based on a *fixed* potential vector. In particular, if we used an estimate of $\boldsymbol{\pi}^C(\boldsymbol{d})$, then the potential vector in (3) would depend on the randomness of this estimate, and as a result the high probability guarantee would not work.

Instead, we show that it is not even neccessary to maintain $\boldsymbol{\pi}^C(\boldsymbol{d})$ very accurately. In fact, it suffices to *exactly* compute it only every few iterations of the algorithm, and use this estimate for the calculation. What allows us to do this is the following lemma, which bounds the change of $\boldsymbol{\pi}^C(\boldsymbol{d})$ measured in energy, after a sequence of vertex insertions and resistance changes.

Lemma 4.12. *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}^0$, $\boldsymbol{q}^0 \in [-1,1]^m$, a β-congestion reduction subset C^0, and a fixed sequence of updates, where the i-th update $i \in \{0, T-1\}$ is of the following form:*

- AddTerminal*(v^i): Set $C^{i+1} = C^i \cup \{v^i\}$ for some $v^i \in V \backslash C^i$, $q_e^{i+1} = q_e^i, r_e^{i+1} = r_e^i$*
- Update*($e^i, \boldsymbol{q}, \boldsymbol{r}$): Set $C^{i+1} = C^i$, $q_e^{i+1} = q_e$ $r_e^{i+1} = r_e$, where $e^i \in E(C^i)$*

Then, with high probability,

$$\sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{\pi}^{C^0,\boldsymbol{r}^0}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^0}{\sqrt{\boldsymbol{r}^0}}\right) - \boldsymbol{\pi}^{C^T,\boldsymbol{r}^T}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^T}{\sqrt{\boldsymbol{r}^T}}\right)\right)} \leq \widetilde{O}\left(\max_{i\in\{0,\dots,T-1\}} \left\|\frac{\boldsymbol{r}^T}{\boldsymbol{r}^i}\right\|_\infty^{1/2} \beta^{-2}\right) \cdot T\,.$$

If we call this demand projection estimate $\boldsymbol{\pi}_{old}$, the quantity that we would like to maintain (1) now becomes

$$\langle \boldsymbol{q}, \boldsymbol{\rho}\rangle \approx \left\langle \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right), SC^+ \boldsymbol{\pi}_{old}\right\rangle\,.$$

Therefore all that's left is to efficiently maintain approximations to demand projections of the form $\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right)$.

Bounding demand projections. An important component for showing that demand projections can be updated efficiently is bounding the magnitude of an entry $\pi_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{1}_e}{\sqrt{r_e}})$ of the projection, for some fixed edge $e = (u,w)$. This is apparent in the following identity which shows how a demand projection changes after inserting a vertex:

$$\boldsymbol{\pi}^{C\cup\{v\}}(\boldsymbol{d}) = \boldsymbol{\pi}^C(\boldsymbol{d}) + \pi_v^{C\cup\{v\}}(\boldsymbol{d}) \cdot \left(\boldsymbol{1}_v - \boldsymbol{\pi}^C(\boldsymbol{1}_v)\right)\,. \tag{4}$$

In [GLP21] this projection entry is upper bounded by $(p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)) \cdot \frac{1}{\sqrt{r_e}}$, where $p_v^{C\cup\{v\}}(u)$ is the probability that a random walk starting at u hits v before C. This bound can be very bad as r_e can be arbitrarily small, although in the particular case of max flow it is possible to show that such low-resistance edges cannot get congested and thus are not of interest.

In order to overcome this issue, we provide a different bound, which in contrast works best when r_e is small.

Lemma 4.6. *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$ and a subset of vertices $C \subseteq V$. For any vertex $v \in V\backslash C$ we have that*

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right| \leq (p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)) \cdot \frac{\sqrt{r_e}}{R_{eff}(v,e)}\,.$$

Here $R_{eff}(v,e)$ is the effective resistance between v and e. In fact, together with the other upper bound mentioned above, this implies that

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right| \leq (p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)) \cdot \frac{1}{\sqrt{R_{eff}(v,e)}},$$

which no longer depends on the value of the resistance r_e.

As we will see, it suffices to approximate $\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right)$ up to additive accuracy roughly $\widehat{\varepsilon}\cdot(p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w))/\sqrt{R_{eff}(C,v)}$ for some error parameter $\widehat{\varepsilon} > 0$. Thus, Lemma 4.6 immediately implies that for any edge e such that $R_{eff}(v,e) \gg R_{eff}(C,v)$, this term is small enough to begin with, and thus can be ignored.

Important edges. In order to ensure that the demand projection can be updated efficiently, we focus only on the demand coming from a special set of edges, which we call *important*. These are the edges that are close (in effective resistance metric) to C relative to their own resistance r_e. In fact, the farther an edge is from C in this sense, the smaller its worst-case congestion, and so non-important edges do not influence the set of congested edges that we are looking for. At a high level, this is because parts of the graph that are very far in the potential embedding have minimal interactions with each other.

Definition 1.2 (Important edges)**.** *An edge $e \in E$ is called ε-*important *(or just* important*) if $R_{eff}(C,e) \leq r_e/\varepsilon^2$.*

Based on the above discussion, we seek to find congested edges *only* among important edges.

Lemma 4.9 (Localization lemma)**.** *Let $\boldsymbol{\phi}^*$ be any solution of*

$$\boldsymbol{L}\boldsymbol{\phi}^* = \delta \cdot \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{p}}{\sqrt{\boldsymbol{r}}}\right),$$

where $\boldsymbol{r}$ are any resistances, $\boldsymbol{p} \in [-1,1]^m$, and $C \subseteq V$. Then, for any $e \in E$ that is not ε-important we have $\left|\frac{\boldsymbol{B}\boldsymbol{\phi}^*}{\sqrt{\boldsymbol{r}}}\right|_e \leq 12\varepsilon$.

One issue is that the set of important edges changes whenever C changes. However, we show that, because of the stability of resistances along the central path, the set of important edges only needs to be updated once every few iterations.

2 Preliminaries

2.1 General

For any $k \in \mathbb{Z}_{\geq 0}$, we denote $[k] = \{1, 2, \ldots, k\}$.

For any $x \in \mathbb{R}^n$ and $C \subseteq [n]$, we denote by $x_C \in \mathbb{R}^{|C|}$ the restriction of x to the entries in C. Similarly for a matrix A, subset of rows C, and subset of columns F, we denote by A_{CF} the submatrix that results from keeping the rows in C and the columns in F.

When not ambiguous, we use the corresponding uppercase symbol to a symbol denoting a vector, to denote the diagonal matrix of that vector. In other words $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{r})$.

Given $x, y \in \mathbb{R}$ and $\alpha \in \mathbb{R}_{\geq 1}$, we say that x and y α-approximate each other and write $x \approx_\alpha y$ if $\alpha^{-1} \leq x/y \leq \alpha$.

When a graph $G(V, E)$ is clear from context, we will use $n = |V|$ and $m = |E|$. We use $\boldsymbol{B} \in \mathbb{R}^{m \times n}$ to denote the edge-vertex incidence matrix of G and, given some resistances $\boldsymbol{r}$, we use $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ to denote the Laplacian $\boldsymbol{L} = \boldsymbol{B}^\top \boldsymbol{R}^{-1} \boldsymbol{B}$.

2.2 Minimum cost flow

Given a directed graph $G(V, E)$ with costs $\boldsymbol{c} \in \mathbb{R}^m$, demands $\boldsymbol{d} \in \mathbb{R}^n$, and capacities $\boldsymbol{u} \in \mathbb{R}^m_{>0}$, the *minimum cost flow problem* asks to compute a flow $\boldsymbol{f}$ that

- routes the demand: $\boldsymbol{B}^\top \boldsymbol{f} = \boldsymbol{d}$
- respects the capacities: $\boldsymbol{0} \leq \boldsymbol{f} \leq \boldsymbol{u}$, and
- minimizes the cost: $\langle \boldsymbol{c}, \boldsymbol{f} \rangle$.

We will denote such an instance of the minimum cost flow by the tuple $(G(V, E), \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{u})$.

2.3 Electrical flows

Definition 2.1 (Energy of a potential embedding)**.** *Consider a graph $G(V, E)$ with resistances $\boldsymbol{r}$ and a potential embedding $\boldsymbol{\phi}$. We denote by*

$$E_{\boldsymbol{r}}(\boldsymbol{\phi}) = \sum_{e \in E} \frac{(B\boldsymbol{\phi})_e^2}{r_e}$$

the total energy of the electrical flow induced by $\boldsymbol{\phi}$.

Definition 2.2 (Energy to route a demand)**.** *Consider a graph $G(V, E)$ with resistances $\boldsymbol{r}$, and a vector $\boldsymbol{d} \in \mathbb{R}^n$. If $\boldsymbol{d}$ is a demand ($\langle 1, d \rangle = 0$), we denote by*

$$\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{d}) = \min_{\boldsymbol{\phi} \colon B^\top \frac{B\boldsymbol{\phi}}{\boldsymbol{r}} = \boldsymbol{d}} E_{\boldsymbol{r}}(\boldsymbol{\phi})$$

the total energy that is required to route the demand $\boldsymbol{d}$ with resistances $\boldsymbol{r}$. We extend this definition for a $\boldsymbol{d}$ that is not a demand vector ($\langle \boldsymbol{1}, \boldsymbol{d} \rangle \neq 0$), as $\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{d}) = \mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{d} - \frac{\langle \boldsymbol{1}, \boldsymbol{d} \rangle}{n} \cdot \boldsymbol{1}\right)$.

Fact 2.3 (Energy statements)**.** *Consider a graph $G(V, E)$ with resistances $\boldsymbol{r}$.*

- *For any $x, y \in \mathbb{R}^n$, we have $\sqrt{\mathcal{E}_{\boldsymbol{r}}(x + y)} \leq \sqrt{\mathcal{E}_{\boldsymbol{r}}(x)} + \sqrt{\mathcal{E}_{\boldsymbol{r}}(y)}$.*
- *and a vector $\boldsymbol{d} \in \mathbb{R}^n$. Then,*
$$\max_{\phi : E_{\boldsymbol{r}}(\boldsymbol{\phi}) \leq 1} \langle \boldsymbol{d}, \boldsymbol{\phi} \rangle = \sqrt{\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{d})}\,.$$
- *For any resistances $\boldsymbol{r}' \leq \alpha \boldsymbol{r}$ for some $\alpha \geq 1$ and any $\boldsymbol{d} \in \mathbb{R}^n$, we have $\mathcal{E}_{\boldsymbol{r}'}(\boldsymbol{d}) \leq \alpha \cdot \mathcal{E}_{\boldsymbol{r}}(\boldsymbol{d})$*

Proof. We let $\boldsymbol{L} = \boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B}$ be the Laplacian of G with resistances $\boldsymbol{r}$. For the first one, we have $\sqrt{\mathcal{E}_{\boldsymbol{r}}(x+y)} = \|x+y\|_{\boldsymbol{L}^+} \le \|x\|_{\boldsymbol{L}^+} + \|y\|_{\boldsymbol{L}^+} = \sqrt{\mathcal{E}_{\boldsymbol{r}}(x)} + \sqrt{\mathcal{E}_{\boldsymbol{r}}(y)}$, where we used the triangle inequality.

The second one follows since $E_{\boldsymbol{r}}(\boldsymbol{\phi}) = \|\boldsymbol{\phi}\|_{\boldsymbol{L}}^2$ and $\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{d}) = \|\boldsymbol{d}\|_{\boldsymbol{L}^+}^2$ and the norms $\|\cdot\|_{\boldsymbol{L}}$ and $\|\cdot\|_{\boldsymbol{L}^+}$ are dual.

For the third one, we note that $\left(\boldsymbol{B}^\top \boldsymbol{R}'^{-1}\boldsymbol{B}\right)^+ \preceq \alpha \cdot \left(\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B}\right)^+$, and so $\mathcal{E}_{\boldsymbol{r}'}(\boldsymbol{d}) = \|\boldsymbol{d}\|^2_{(\boldsymbol{B}^\top \boldsymbol{R}'^{-1}\boldsymbol{B})^+} \le \alpha \|\boldsymbol{d}\|^2_{(\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B})^+} = \alpha \cdot \mathcal{E}_{\boldsymbol{r}}(\boldsymbol{d})$. □

Definition 2.4 (Effective resistances)**.** *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$ and any pair of vertices $u,v \in V$. We denote by $R_{eff}(u,v)$ the energy required to route 1 unit of flow from u to v, i.e. $R_{eff}(u,v) = \mathcal{E}_{\boldsymbol{r}}(\mathbf{1}_u - \mathbf{1}_v)$. This is called the* effective resistance between u and v. *We extend this definition to work with vertex subsets $X, Y \subseteq V$, such that $R_{eff}(X,Y)$ is the effective resistance between the vertices x,y that result from contracting X and Y. When used as an argument of R_{eff}, an edge $e = (u,v) \in E$ is treated as the vertex subset $\{u,v\}$.*

Definition 2.5 (Schur complement)**.** *Given a graph $G(V,E)$ with Laplacian $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ and a vertex subset $C \subseteq V$ as well as $F = V \backslash C$, $SC(G,C) := \boldsymbol{L}_{CC} - \boldsymbol{L}_{CF}\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC}$ (or just SC) is called the* Schur complement of G onto C.

Fact 2.6 (Cholesky factorization)**.** *Given a matrix $\boldsymbol{L} \in \mathbb{R}^{n\times n}$, a subset $C \subseteq [n]$, and $F = [n]\backslash C$, we have*

$$\boldsymbol{L}^+ = \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{L}_{FF}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & SC(\boldsymbol{L}, C)^+ \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{L}_{CF}\boldsymbol{L}_{FF}^{-1} & \boldsymbol{I} \end{pmatrix}.$$

2.4 Random walks

Definition 2.7 (Hitting probabilities)**.** *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$. For any $u,v \in V$, $C \subseteq V$, we denote by $p_v^{C,\boldsymbol{r}}(u)$ the probability that for random walk that starts from u and uses edges with probability proportional to $\frac{1}{\boldsymbol{r}}$, the first vertex of C to be visited is v. When not ambiguous, we will use the notation $p_v^C(u)$.*

Definition 2.8 (Demand projection)**.** *Consider a graph $G(V,E)$ and a demand vector $\boldsymbol{d}$. For any $v \in V$, $C \subseteq V$, we define $\pi_v^{C,\boldsymbol{r}}(\boldsymbol{d}) = \sum\limits_{u\in V} d_u p_v^{C,\boldsymbol{r}}(u)$ and call the resulting vector $\boldsymbol{\pi}^{C,\boldsymbol{r}}(\boldsymbol{d}) \in \mathbb{R}^n$ the* demand projection of $\boldsymbol{d}$ onto C. *When not ambiguous, we will use the notation $\boldsymbol{\pi}^C(\boldsymbol{d})$.*

For convenience, when we write $\boldsymbol{\pi}^C(\boldsymbol{d})$ we might also refer to the restriction of this vector to C. This will be clear from the context, and, as $\pi_v^C(\boldsymbol{d}) = 0$ for any $v \notin C$, no ambiguity is introduced.

Fact 2.9 ([GLP21])**.** *Given a graph $G(V,E)$ with Laplacian $\boldsymbol{L}$, a vertex subset $C \subseteq V$, and $\boldsymbol{d} \in \mathbb{R}^n$, we have*

$$\boldsymbol{\pi}^C(\boldsymbol{d}) = \boldsymbol{d}_C - \boldsymbol{L}_{CF}\boldsymbol{L}_{FF}^{-1}\boldsymbol{d}_F\,.$$

Additionally,

$$\left[\boldsymbol{L}^+\boldsymbol{d}\right]_C = SC^+\boldsymbol{\pi}^C(\boldsymbol{d})\,,$$

where SC is the Schur complement of G onto C.

An important property of the demand projection is that the energy required to route it is upper bounded by the energy required to route the original demand. The proof can be found in Section B.

Lemma 2.10. *Let $\boldsymbol{d}$ be a demand vector, let $\boldsymbol{r}$ be resistances, and let $C \subseteq V$ be a subset of vertices. Then*

$$\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^C(\boldsymbol{d})\right) \leq \mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{d}\right) .$$

The following lemma relates the effective resistance between a vertex and a vertex set, to the energy to route a particular demand, based on a demand projection.

Lemma 2.11 (Effective resistance and hitting probabilities)**.** *Given a graph $G(V,E)$ with resistances $\boldsymbol{r}$, any vertex set $A \subseteq V$ and vertex $u \in V \backslash A$, we have $R_{eff}(u,A) = \mathcal{E}_{\boldsymbol{r}}(\mathbf{1}_u - \boldsymbol{\pi}^A(\mathbf{1}_u))$.*

Proof. Let $\boldsymbol{L}$ be the Laplacian of G with resistances $\boldsymbol{r}$ and $F = V\backslash A$. We first prove that

$$\mathcal{E}_{\boldsymbol{r}}(\mathbf{1}_u - \boldsymbol{\pi}^A(\mathbf{1}_u)) = \mathbf{1}_u^\top \boldsymbol{L}_{FF}^{-1}\mathbf{1}_u .$$

This is because

$$\begin{aligned}
&\mathcal{E}_{\boldsymbol{r}}(\mathbf{1}_u - \boldsymbol{\pi}^A(\mathbf{1}_u)) \\
&= \langle \mathbf{1}_u - \boldsymbol{\pi}^A(\mathbf{1}_u), \boldsymbol{L}^+(\mathbf{1}_u - \boldsymbol{\pi}^A(\mathbf{1}_u))\rangle \\
&= \left\langle \mathbf{1}_u - \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ -\boldsymbol{L}_{AF}\boldsymbol{L}_{FF}^{-1} & \boldsymbol{I} \end{pmatrix} \mathbf{1}_u, \boldsymbol{L}^+ \left(\mathbf{1}_u - \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ -\boldsymbol{L}_{AF}\boldsymbol{L}_{FF}^{-1} & \boldsymbol{I} \end{pmatrix} \mathbf{1}_u \right) \right\rangle \\
&= \left\langle \begin{pmatrix} \boldsymbol{I} & \mathbf{0} \\ -\boldsymbol{L}_{AF}\boldsymbol{L}_{FF}^{-1} & \boldsymbol{I} \end{pmatrix} \left(\mathbf{1}_u + \begin{pmatrix} \mathbf{0} \\ \boldsymbol{L}_{AF}\boldsymbol{L}_{FF}^{-1}\mathbf{1}_u \end{pmatrix}\right), \right. \\
&\qquad \left. \begin{pmatrix} \boldsymbol{L}_{FF}^{-1} & \mathbf{0} \\ \mathbf{0} & SC(\boldsymbol{L},A)^+ \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \mathbf{0} \\ -\boldsymbol{L}_{AF}\boldsymbol{L}_{FF}^{-1} & \boldsymbol{I} \end{pmatrix} \left(\mathbf{1}_u + \begin{pmatrix} \mathbf{0} \\ \boldsymbol{L}_{AF}\boldsymbol{L}_{FF}^{-1}\mathbf{1}_u \end{pmatrix}\right) \right\rangle \\
&= \left\langle \mathbf{1}_u, \begin{pmatrix} \boldsymbol{L}_{FF}^{-1} & \mathbf{0} \\ \mathbf{0} & SC(\boldsymbol{L},A)^+ \end{pmatrix} \mathbf{1}_u \right\rangle \\
&= \langle \mathbf{1}_u, \boldsymbol{L}_{FF}^{-1}\mathbf{1}_u \rangle .
\end{aligned}$$

On the other hand, note that $R_{eff}(u,A) = \widehat{R}_{eff}(u,\widehat{a})$, where $\widehat{R}$ are the effective resistances in a graph $\widehat{G}$ that results after contracting A to a new vertex $\widehat{a}$. It is easy to see that the Laplacian of this new graph is

$$\widehat{\boldsymbol{L}} = \begin{pmatrix} \boldsymbol{L}_{FF} & \boldsymbol{L}_{FA}\mathbf{1} \\ \mathbf{1}^\top \boldsymbol{L}_{AF} & \mathbf{1}^\top \boldsymbol{L}_{FA}\mathbf{1} \end{pmatrix} .$$

We look at the system $\widehat{\boldsymbol{L}} \begin{pmatrix} x \\ a \end{pmatrix} = \mathbf{1}_u - \mathbf{1}_{\widehat{a}}$, where a is a scalar. The solution is given by

$$\boldsymbol{x} = \boldsymbol{L}_{FF}^{-1}\left(\mathbf{1}_u - a \cdot \boldsymbol{L}_{FA}\mathbf{1}\right) .$$

However, as $\mathbf{1} \in \ker(\widehat{\boldsymbol{L}})$ by the fact that it is a Laplacian, we can assume that $a = 0$ by shifting. Therefore $\boldsymbol{x} = \boldsymbol{L}_{FF}^{-1}\mathbf{1}_u$, and so we can conclude that

$$\begin{aligned}
R_{eff}(u,A) &= \langle \mathbf{1}_u - \mathbf{1}_{\widehat{a}}, \widehat{\boldsymbol{L}}^+(\mathbf{1}_u - \mathbf{1}_{\widehat{a}})\rangle \\
&= \langle \mathbf{1}_u, \boldsymbol{L}_{FF}^{-1}\mathbf{1}_u \rangle .
\end{aligned}$$

So we have proved that $R_{eff}(u,A) = \mathcal{E}_{\boldsymbol{r}}(\mathbf{1}_u - \boldsymbol{\pi}^A(\mathbf{1}_u))$ and we are done. □

Finally, the following lemma relates the effective resistance between a vertex and a vertex set, to the effective resistance between vertices.

Lemma 2.12. *Given a graph $G(V,E)$ with resistances $\boldsymbol{r}$, any vertex set $A \subseteq V$ and vertex $u \in V \backslash A$, we have*

$$\frac{1}{|A|} \cdot \min_{v\in A} R_{eff}(u,v) \leq R_{eff}(u,A) \leq \min_{v\in A} R_{eff}(u,v)\,.$$

Proof. Let $\boldsymbol{L}$ be the Laplacian of G with resistances $\boldsymbol{r}$, and note that $R_{eff}(u,v) = \left\|\boldsymbol{L}^{+/2}(\mathbf{1}_u - \mathbf{1}_v)\right\|_2^2$ and, by Lemma 2.11, $R_{eff}(u,A) = \left\|\boldsymbol{L}^{+/2}(\mathbf{1}_u - \boldsymbol{\pi}^A(\mathbf{1}_u))\right\|_2^2$. Expanding the latter, we have

$$\begin{aligned} R_{eff}(u,A) &= \left\|\sum_{v\in A} \pi_v^A(\mathbf{1}_u)\cdot \boldsymbol{L}^{+/2}(\mathbf{1}_u - \mathbf{1}_v)\right\|_2^2 \\ &= \sum_{v\in A}\left(\pi_v^A(\mathbf{1}_u)\right)^2 \left\|\boldsymbol{L}^{+/2}(\mathbf{1}_u - \mathbf{1}_v)\right\|_2^2 + \sum_{v\in A}\sum_{\substack{v'\in A\\ v'\neq v}} \pi_v^A(\mathbf{1}_u)\pi_{v'}^A(\mathbf{1}_u)\langle \mathbf{1}_u - \mathbf{1}_{v'}, \boldsymbol{L}^{+}(\mathbf{1}_u - \mathbf{1}_v)\rangle\,. \end{aligned}$$

Now, note that $\pi_v^A(\mathbf{1}_u), \pi_{v'}^A(\mathbf{1}_u) \geq 0$. Additionally, let $\boldsymbol{\phi} = \boldsymbol{L}^{+}(\mathbf{1}_u - \mathbf{1}_v)$ be the potential embedding that induces a 1-unit electrical flow from v to u. As the potential embedding stretches between ϕ_v and ϕ_u, we have that $\phi_{v'} \leq \phi_u$, so $\langle \mathbf{1}_u - \mathbf{1}_{v'}, \boldsymbol{L}^{+}(\mathbf{1}_u - \mathbf{1}_v)\rangle = \phi_u - \phi_{v'} \geq 0$. Therefore,

$$\begin{aligned} R_{eff}(u,A) &\geq \sum_{v\in A}\left(\pi_v^A(\mathbf{1}_u)\right)^2 \left\|\boldsymbol{L}^{+/2}(\mathbf{1}_u - \mathbf{1}_v)\right\|_2^2 \\ &\geq \frac{1}{|A|}\sum_{v\in A} \pi_v^A(\mathbf{1}_u)\cdot R_{eff}(u,v) \\ &\geq \frac{1}{|A|}\min_{v\in A} R_{eff}(u,v)\,, \end{aligned}$$

where we used the Cauchy-Schwarz inequality and the fact that $\sum_{v\in A} \pi_v^A(\mathbf{1}_u) = 1$. □

3 Interior Point Method with Dynamic Data Structures

The goal of this section is to show that, given a data structure for approximating electrical flows in sublinear time, we can execute the min cost flow interior point method with total runtime faster than $\widetilde{O}\left(m^{3/2}\right)$.

3.1 LP formulation and background

We present the interior point method setup that we will use, which is from [AMV20]. Our goal is to solve the following minimum cost flow linear program:

$$\begin{aligned} &\min\ \langle \boldsymbol{c}, \boldsymbol{C}\boldsymbol{x}\rangle \\ &\mathbf{0} \leq \boldsymbol{f}^0 + \boldsymbol{C}\boldsymbol{x} \leq \boldsymbol{u}\,, \end{aligned}$$

where $\boldsymbol{f}^0$ is a flow with $\boldsymbol{B}^\top \boldsymbol{f}^0 = \boldsymbol{d}$ and $\boldsymbol{C}$ is an $m \times (m-n+1)$ matrix whose image is the set of circulations in G.

In order to use an interior point method, the following log barrier objective is defined:

$$\min_{\boldsymbol{x}} F_\mu(\boldsymbol{x}) = \left\langle \frac{\boldsymbol{c}}{\mu}, \boldsymbol{C}\boldsymbol{x} \right\rangle - \sum_{e \in E} \left(\log \left(\boldsymbol{f}^0 + \boldsymbol{C}\boldsymbol{x} \right)_e + \log \left(\boldsymbol{u} - (\boldsymbol{f}^0 + \boldsymbol{C}\boldsymbol{x}) \right)_e \right) . \tag{5}$$

For any parameter $\mu > 0$, the optimality condition of (5) is called the *centrality condition* and is given by

$$\boldsymbol{C}^\top \left(\frac{\boldsymbol{c}}{\mu} + \frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-} \right) = \boldsymbol{0} , \tag{6}$$

where $\boldsymbol{f} = \boldsymbol{f}^0 + \boldsymbol{C}\boldsymbol{x}$, and $\boldsymbol{s}^+ = \boldsymbol{u} - \boldsymbol{f}$, $\boldsymbol{s}^- = \boldsymbol{f}$ are called the *positive* and *negative* slacks of $\boldsymbol{f}$ respectively.

This leads us to the following definitions.

Definition 3.1 (μ-central flow). *Given a minimum cost flow instance with costs $\boldsymbol{c}$, demands $\boldsymbol{d}$ and capacities $\boldsymbol{u}$, as well as a parameter $\mu > 0$, we will say that a flow $\boldsymbol{f}$ (and its corresponding slacks $\boldsymbol{s}$ and resistances $\boldsymbol{r}$) is μ-central if $\boldsymbol{B}^\top \boldsymbol{f} = \boldsymbol{d}$, $\boldsymbol{s} > \boldsymbol{0}$, and it satisfies the centrality condition (6), i.e.*

$$\boldsymbol{C}^\top \left(\frac{\boldsymbol{c}}{\mu} + \frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-} \right) = \boldsymbol{0} .$$

Additionally, we will denote such flow by $\boldsymbol{f}(\mu)$ (and its corresponding slacks and resistances by $\boldsymbol{s}(\mu)$ and $\boldsymbol{r}(\mu)$, respectively).

Definition 3.2 ((μ, α)-central flow). *Given parameters $\mu > 0$ and $\alpha \geq 1$, we will say that a flow $\boldsymbol{f}$ with resistances $\boldsymbol{r} > \boldsymbol{0}$ is (μ, α)-central if $\boldsymbol{r} \approx_\alpha \boldsymbol{r}(\mu)$. We will also call its corresponding slacks $\boldsymbol{s}$ and resistances $\boldsymbol{r}$ (μ, α)-central.*

Given a μ-central flow $\boldsymbol{f}$ and some step size $\delta > 0$, the standard (Newton) step to obtain an approximately $\mu/(1+\delta)$-central flow $\boldsymbol{f}' = \boldsymbol{f} + \widetilde{\boldsymbol{f}}$ is given by

$$\begin{aligned} \widetilde{\boldsymbol{f}} &= -\frac{\delta}{\mu} \frac{\boldsymbol{c}}{\boldsymbol{r}} + \frac{\delta}{\mu} \boldsymbol{R}^{-1} \boldsymbol{B} \boldsymbol{L}^+ \boldsymbol{B}^\top \frac{\boldsymbol{c}}{\boldsymbol{r}} \\ &= \delta \frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-}}{\boldsymbol{r}} - \delta \boldsymbol{R}^{-1} \boldsymbol{B} \boldsymbol{L}^+ \boldsymbol{B}^\top \frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-}}{\boldsymbol{r}} \\ &= \delta \cdot g(\boldsymbol{s}) - \delta \boldsymbol{R}^{-1} \boldsymbol{B} \boldsymbol{L}^+ \boldsymbol{B}^\top g(\boldsymbol{s}) \end{aligned}$$

where $\boldsymbol{r} = \frac{1}{(\boldsymbol{s}^+)^2} + \frac{1}{(\boldsymbol{s}^-)^2}$ and we have denoted $g(\boldsymbol{s}) = \frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-}}{\boldsymbol{r}}$.

Fact 3.3. *Using known scaling arguments, can assume that costs and capacities are bounded by* $\mathrm{poly}(m)$*, while only incurring an extra logarithmic dependence in the largest network parameter [Gab83].*

We also use the fact that the resistances in the interior point method are never too large, which is proved in Appendix B.

Fact 3.4. *For any $\mu \in (1/\mathrm{poly}(m), \mathrm{poly}(m))$, we have $\|\boldsymbol{r}(\mu)\|_\infty \leq m^{\widetilde{O}(\log m)}$.*

3.2 Making progress with approximate electrical flows

The following lemma shows that we can make k steps of the interior point method by computing $O(k^4)$ $(1+O(k^{-6}))$-*approximate* electrical flows. The proof is essentially the same as in [GLP21], but we provide it for completeness in Appendix C.2.

Lemma 3.5. *Let $\boldsymbol{f}^1,\ldots,\boldsymbol{f}^{T+1}$ be flows with slacks $\boldsymbol{s}^t$ and resistances $\boldsymbol{r}^t$ for $t\in[T+1]$, where $T=\frac{k}{\varepsilon_{\text{step}}}$ for some $k\le\sqrt{m}/10$ and $\varepsilon_{\text{step}}=10^{-5}k^{-3}$, such that*

- *$\boldsymbol{f}^1$ is $(\mu,1+\varepsilon_{\text{solve}}/8)$-central for $\varepsilon_{\text{solve}}=10^{-5}k^{-3}$*
- *For all $t\in[T]$ and $e\in E$, $f_e^{t+1}=\begin{cases}f_e(\mu)+\varepsilon_{\text{step}}\sum_{i=1}^{t}\widetilde{f}_e^i & \text{if } \exists i\in[t]:\widetilde{f}_e^i\neq 0\\ f_e^1 & \text{otherwise}\end{cases}$, where*

$$\widetilde{\boldsymbol{f}}^{*t}=\delta g(\boldsymbol{s}^t)-\delta(\boldsymbol{R}^t)^{-1}\boldsymbol{B}\left(\boldsymbol{B}^\top(\boldsymbol{R}^t)^{-1}\boldsymbol{B}\right)^+\boldsymbol{B}^\top g(\boldsymbol{s}^t)$$

for $\delta=\frac{1}{\sqrt{m}}$ and

$$\left\|\sqrt{\boldsymbol{r}^t}\left(\widetilde{\boldsymbol{f}}^{*t}-\widetilde{\boldsymbol{f}}^t\right)\right\|_\infty\le\varepsilon$$

for $\varepsilon=10^{-6}k^{-6}$.

Then, setting $\varepsilon_{\text{step}}=\varepsilon_{\text{solve}}=10^{-5}k^{-3}$ and $\varepsilon=10^{-6}k^{-6}$ we get that $\boldsymbol{s}^{T+1}\approx_{1.1}\boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^{k\varepsilon_{\text{step}}^{-1}}\right)$.

From now and for the rest of Section 3 we fix the values of $\varepsilon_{\text{step}},\varepsilon_{\text{solve}},\varepsilon$ based on this lemma. Using this lemma together with the following recentering procedure also used in [GLP21], we can exactly compute a $\left(\mu/(1+\varepsilon_{\text{step}}/\sqrt{m})^{k\varepsilon_{\text{step}}^{-1}}\right)$-central flow.

Lemma 3.6. *Given a flow $\boldsymbol{f}$ with slacks $\boldsymbol{s}$ such that $\boldsymbol{s}\approx_{1.1}\boldsymbol{s}(\mu)$ for some $\mu>0$, we can compute $\boldsymbol{f}(\mu)$ in $\widetilde{O}(m)$.*

3.3 The Locator data structure

From the previous lemma it becomes obvious that the only thing left is to maintain in sublinear time an approximation to

$$\delta g(\boldsymbol{s}^t)-\delta(\boldsymbol{R}^t)^{-1}\boldsymbol{B}(\boldsymbol{B}^\top(\boldsymbol{R}^t)^{-1}\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s}^t)\,.$$

for $\delta=1/\sqrt{m}$. This is the job of the $(\alpha,\beta,\varepsilon)$-Locator data structure, which computes all the entries of this vector that have magnitude $\ge\varepsilon$. We note that the guarantees of this data structure are similar to the ones in [GLP21], but our locator requires an extra parameter α which is a measure of how much resistances can deviate before a full recomputation has to be made.

Definition 3.7 ($(\alpha,\beta,\varepsilon)$-Locator)**.** *An $(\alpha,\beta,\varepsilon)$-Locator is a data structure that maintains valid slacks $\boldsymbol{s}$ and resistances $\boldsymbol{r}$, and can support the following operations against oblivious adversaries with high probability:*

- Initialize($\boldsymbol{f}$)*: Set $\boldsymbol{s}^+=\boldsymbol{u}-\boldsymbol{f}$, $\boldsymbol{s}^-=\boldsymbol{f}$, $\boldsymbol{r}=\frac{1}{(\boldsymbol{s}^+)^2}+\frac{1}{(\boldsymbol{s}^-)^2}$.*

- Update$(e, \boldsymbol{f})$*: Set* $s_e^+ = u_e - f_e$*,* $s_e^- = f_e$*,* $r_e = \frac{1}{(s_e^+)^2} + \frac{1}{(s_e^-)^2}$*. Works under the condition that*

$$r_e^{\max}/\alpha \le r_e \le \alpha \cdot r_e^{\min}\,,$$

where $r_e^{\max}$ *and* $r_e^{\min}$ *are the maximum and minimum resistance values that edge* e *has had since the last call to* BatchUpdate*.*

- BatchUpdate$(Z, \boldsymbol{f})$*: Set* $s_e^+ = u_e - f_e, s_e^- = f_e, r_e = \frac{1}{(s_e^+)^2} + \frac{1}{(s_e^-)^2}$ *for all* $e \in Z$*.*

- Solve()*: Let*

$$\widetilde{\boldsymbol{f}}^* = \delta g(\boldsymbol{s}) - \delta \boldsymbol{R}^{-1}\boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B})^+ \boldsymbol{B}^\top g(\boldsymbol{s})\,, \tag{7}$$

where $\delta = \frac{1}{\sqrt{m}}$*. Returns an edge set* Z *of size* $\widetilde{O}\left(\varepsilon^{-2}\right)$ *that with high probability contains all* e *such that* $\sqrt{r_e}\left|\widetilde{f}_e^*\right| \ge \varepsilon$*.*

The data structure works as long as the total number of calls to Update*, plus the sum of* $|Z|$ *for all calls to* BatchUpdate *is* $O(\beta m)$*.*

In Section 4 we will prove the following lemma, which constructs an $(\alpha, \beta, \varepsilon)$-Locator and outlines its runtime guarantees:

Lemma 3.8 (Efficient $(\alpha, \beta, \varepsilon)$-Locator)**.** *For any graph* $G(V, E)$ *and parameters* $\alpha \ge 1$*,* $\beta \in (0, 1)$*,* $\varepsilon \ge \widetilde{\Omega}\left(\beta^{-2}m^{-1/2}\right)$*, and* $\widehat{\varepsilon} \in \left(\widetilde{\Omega}\left(\beta^{-2}m^{-1/2}\right), \varepsilon\right)$*, there exists an* $(\alpha, \beta, \varepsilon)$-Locator *for* G *with the following runtimes per operation:*

- Initialize$(\boldsymbol{f})$*:* $\widetilde{O}\left(m \cdot \left(\widehat{\varepsilon}^{-4}\beta^{-8} + \widehat{\varepsilon}^{-2}\varepsilon^{-2}\alpha^2\beta^{-4}\right)\right)$*.*

- Update$(e, \boldsymbol{f})$*:* $\widetilde{O}\left(m \cdot \frac{\widehat{\varepsilon}\alpha^{1/2}}{\varepsilon^3} + \widehat{\varepsilon}^{-4}\varepsilon^{-2}\beta^{-8} + \widehat{\varepsilon}^{-2}\varepsilon^{-4}\alpha^2\beta^{-6}\right)$ *amortized.*

- BatchUpdate$(Z, \boldsymbol{f})$*:* $\widetilde{O}\left(m \cdot \frac{1}{\varepsilon^2} + |Z| \cdot \frac{1}{\varepsilon^2\beta^2}\right)$*.*

- Solve()*:* $\widetilde{O}\left(\beta m \cdot \frac{1}{\varepsilon^2}\right)$*.*

Note that even though a Locator computes a set that contains all ε-congested edges, it does not return the actual flow values. The reason for that is that it only works against oblivious adversaries, and allowing (randomized) flow values to affect future updates constitutes an adaptive adversary. As in [GLP21], we resolve this by sanitizing the outputs through a different data structure called Checker, which computes the flow values and works against semi-adaptive adversaries. As the definition and implementation of Checker is orthogonal to our contribution and also does not affect the final runtime, we defer the discussion to Appendix F. To simplify the presentation in this section, we instead define the following idealized version of it, called PerfectChecker.

Definition 3.9 (ε-PerfectChecker)**.** *For any error* $\varepsilon > 0$*, an* ε-PerfectChecker *is an oracle that given a graph* $G(V, E)$*, slacks* $\boldsymbol{s}$*, resistances* $\boldsymbol{r}$*, supports the following operations:*

- Update$(e, \boldsymbol{f})$*: Set* $s_e^+ = u_e - f_e$*,* $s_e^- = f_e$*,* $r_e = \frac{1}{(s_e^+)^2} + \frac{1}{(s_e^-)^2}$*.*

- CHECK(e): *Compute a flow value* $\widetilde{f}_e$ *such that* $\sqrt{r_e}\left|\widetilde{f}_e - \widetilde{f}_e^*\right| \le \varepsilon$, *where*

$$\widetilde{\boldsymbol{f}}^* = \delta g(\boldsymbol{s}) - \delta \boldsymbol{R}^{-1}\boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s})\,,$$

with $\delta = 1/\sqrt{m}$. *If* $\sqrt{r_e}\left|\widetilde{f}_e\right| < \varepsilon/2$ *return* 0, *otherwise return* $\widetilde{f}_e$.

3.4 The minimum cost flow algorithm

Now, we will show how the data structure defined in Section 3.3 can be used to make progress along the central path. The main lemma that analyzes the performance of the minimum cost flow algorithm given access to an $(\alpha, \beta, \varepsilon)$-LOCATOR is Lemma 3.10. Also, the skeleton of the algorithm is described in Algorithm 1.

Algorithm 1 Minimum Cost Flow

procedure MINCOSTFLOW($G, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{u}$)
    $\bar{\boldsymbol{f}}, \mu =$ INITIALIZE($G, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{u}$) ▷ Lemma 3.12. $\bar{\boldsymbol{f}}$ is μ-central at all times.
    $i = 0$
    while $\mu \ge m^{-10}$ **do**
        if i is a multiple of $\lfloor \varepsilon_{\text{solve}}\sqrt{\beta m}/k \rfloor$ **then** ▷ Re-initialize when $|C|$ exceeds $O(\beta m)$.
            $\mathcal{L} =$ LOCATOR.INITIALIZE($\bar{\boldsymbol{f}}$) with error $\varepsilon/2$
        if i is a multiple of $\lfloor \varepsilon_{\text{solve}}\sqrt{\beta_{\text{CHECKER}} m}/k \rfloor$ **then**
            $\mathcal{C}^i =$ CHECKER.INITIALIZE($\bar{\boldsymbol{f}}, \varepsilon, \beta_{\text{CHECKER}}$) for $i \in [k\varepsilon_{\text{step}}^{-1}]$
        if i is a multiple of $\lfloor 0.5\alpha^{1/4}/k - 1 \rfloor$ **then** ▷ Update important edges when $\mathcal{L}.\boldsymbol{r}^0$ expires
            $\mathcal{L}$.BATCHUPDATE($\emptyset$)
        $\bar{\boldsymbol{f}}, \mu =$ MULTISTEP($\bar{\boldsymbol{f}}, \mu$)
        if i is a multiple of $\widehat{T}$ **then**
            $Z = \emptyset$
            for $e \in E$ **do**
                $\bar{s}_e^+ = u_e - \bar{f}_e$, $\bar{s}_e^- = \bar{f}_e$
                if $\bar{s}_e^+ \not\approx_{\varepsilon_{\text{solve}}/16} \mathcal{L}.s_e^+$ or $\bar{s}_e^- \not\approx_{\varepsilon_{\text{solve}}/16} \mathcal{L}.s_e^-$ **then**
                    $\mathcal{C}^i$.UPDATE($e, \bar{\boldsymbol{f}}$) for $i \in [k\varepsilon_{\text{step}}^{-1}]$
                    $Z = Z \cup \{e\}$
            $\mathcal{L}$.BATCHUPDATE($Z, \bar{\boldsymbol{f}}$)
        else
            for $e \in E$ **do**
                $\bar{s}_e^+ = u_e - \bar{f}_e$, $\bar{s}_e^- = \bar{f}_e$
                if $\bar{s}_e^+ \not\approx_{\varepsilon_{\text{solve}}/8} \mathcal{L}.s_e^+$ or $\bar{s}_e^- \not\approx_{\varepsilon_{\text{solve}}/8} \mathcal{L}.s_e^-$ **then**
                    $\mathcal{C}^i$.UPDATE($e, \bar{\boldsymbol{f}}$) for $i \in [k\varepsilon_{\text{step}}^{-1}]$
                    $\mathcal{L}$.UPDATE($e, \bar{\boldsymbol{f}}$)
        $i = i + 1$
    return ROUND($G, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{u}, \bar{\boldsymbol{f}}$) ▷ Lemma 3.13

Lemma 3.10 (MinCostFlow). *Let $\mathcal{L}$ be an $(\alpha, \beta, \varepsilon) -$ Locator, $\boldsymbol{f}$ be a μ-central flow where $\mu = \text{poly}(m)$, and $k \in \left[m^{1/316}\right]$, $\beta \geq \widetilde{\Omega}\left(k^3/m^{1/4}\right)$, $\widehat{T} \in \left[\widetilde{O}\left(m^{1/2}/k\right)\right]$ be some parameters. There is an algorithm that with high probability computes a μ'-central flow $\boldsymbol{f}'$, where $\mu' \leq m^{-10}$. Additionally, the algorithm runs in time $\widetilde{O}\left(m^{3/2}/k\right)$, plus*

- $\widetilde{O}\left(k^3\beta^{-1/2}\right)$ *calls to* $\mathcal{L}$.Initialize,
- $\widetilde{O}\left(m^{1/2}k^3\right)$ *calls to* $\mathcal{L}$.Solve,
- $\widetilde{O}\left(m^{1/2}\left(k^6\widehat{T} + k^{15}\right)\right)$ *calls to* $\mathcal{L}$.Update,
- $\widetilde{O}\left(m^{1/2}\alpha^{-1/4}\right)$ *calls to* $\mathcal{L}$.BatchUpdate($\emptyset$), *and*
- $\widetilde{O}\left(m^{1/2}k^{-1}\widehat{T}^{-1}\right)$ *calls to* $\mathcal{L}$.BatchUpdate($Z, \bar{\boldsymbol{f}}$) *for some $Z \neq \emptyset, \bar{\boldsymbol{f}}$. Additionally, the sum of $|Z|$ over all such calls is $\widetilde{O}\left(mk^3\beta^{1/2}\right)$.*

The proof appears in Appendix C.4. Its main ingredient is the following lemma, which easily follows from Lemma 3.5 and essentially shows how k steps of the interior point method can be performed in $\widetilde{O}(m)$ instead of $\widetilde{O}(mk)$. Its proof appears in Appendix C.3.

Lemma 3.11 (MultiStep). *Let $k \in \{1, \ldots, \sqrt{m}/10\}$. We are given $\boldsymbol{f}(\mu)$, an $(\alpha, \beta, \varepsilon/2)$-Locator $\mathcal{L}$, and an ε-PerfectChecker $\mathcal{C}$, such that*

- $\mathcal{L}.\boldsymbol{r} = \mathcal{C}.\boldsymbol{r}$ *are $(\mu, 1 + \varepsilon_{\text{solve}}/8)$-central resistances, and*
- $\mathcal{L}.\boldsymbol{r}^0$ *are $(\mu^0, 1 + \varepsilon_{\text{solve}}/8)$-central resistances, where $\mu^0 \leq \mu \cdot (1 + \varepsilon_{\text{step}}/\sqrt{m})^{\widehat{T}}$ and $\widehat{T} = (0.5\alpha^{1/4} - k)\varepsilon_{\text{step}}^{-1}$. Additionally, for any resistances $\widehat{\boldsymbol{r}}$ that $\mathcal{L}$ had at any point since the last call to $\mathcal{L}$.*BatchUpdate*, $\widehat{\boldsymbol{r}}$ are $(\hat{\mu}, 1.1)$-central for some $\hat{\mu} \in [\mu, \mu^0]$.*

*Then, there is an algorithm that with high probability computes $\boldsymbol{f}(\mu')$, where $\mu' = \mu/(1 + \varepsilon_{\text{step}}/\sqrt{m})^{k\varepsilon_{\text{step}}^{-1}}$. The algorithm runs in time $\widetilde{O}(m)$, plus $O(k^{16})$ calls to $\mathcal{L}$.*Update*, $O(k^4)$ calls to $\mathcal{L}$.*Solve*, and $O(k^{16})$ calls to $\mathcal{C}$.*Update *and* $\mathcal{C}$.*Check*. Additionally, $\mathcal{L}.\boldsymbol{r}$ and $\mathcal{C}$ are unmodified.*

3.5 Proof of Theorem 1.1

Correctness. First of all, we apply capacity and cost scaling [Gab83] to make sure that $\|\boldsymbol{c}\|_\infty, \|\boldsymbol{u}\|_\infty = \text{poly}(m)$. These incur an extra factor of $\log(U + W)$ in the runtime.

We first get an initial solution to the interior point method by using the following lemma:

Lemma 3.12 (Interior point method initialization, Appendix A in [AMV20]). *Given a min cost flow instance $\mathcal{I} = (G(V, E), \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{u})$, there exists an algorithm that runs in time $O(m)$ and produces a new min cost flow instance $\mathcal{I}' = \left(G'(V', E'), \boldsymbol{c}', \boldsymbol{d}', \boldsymbol{u}'\right)$, where $|V'| = O(|V|)$ and $|E'| = O(|E|)$, as well as a flow $\boldsymbol{f}$ such that*

- $\boldsymbol{f}$ *is μ-central for $\mathcal{I}'$ for some $\mu = \Theta\left(\|\boldsymbol{c}\|_2\right)$*

Algorithm 2 MultiStep

procedure MULTISTEP($\boldsymbol{f}, \mu$) ▷ Makes equivalent progress to k interior point method steps
  $\widehat{\boldsymbol{r}} = \mathcal{L}.\boldsymbol{r}$ ▷ Save resistances to restore later
  for $i = 1, \ldots, k\varepsilon_{\text{step}}^{-1}$ **do**
    $Z = \mathcal{L}.$SOLVE()
    for $e \in Z$ **do** ▷ Z: Set of edges with sufficiently changed flow
      $\widetilde{f}_e = \mathcal{C}^i.$CHECK($e$)
      if $\widetilde{f}_e \neq 0$ **then**
        $f_e = f_e + \varepsilon_{\text{step}} \widetilde{f}_e$
        $\mathcal{L}.$UPDATE($e, \boldsymbol{f}$)
        $\mathcal{C}^j.$TEMPORARYUPDATE($e, \boldsymbol{f}$) for $j \in \left[i+1, k\varepsilon_{\text{step}}^{-1}\right]$
  $\mu = \mu / (1 + \varepsilon_{\text{step}}/\sqrt{m})^{k\varepsilon_{\text{step}}^{-1}}$
  $\boldsymbol{f} = $ RECENTER($\boldsymbol{f}, \mu$) ▷ Lemma 3.6
  for $e \in E$ **do**
    if $\mathcal{L}.r_e \neq \widehat{r}_e$ **then**
      $\mathcal{L}.$UPDATE($e, \widehat{\boldsymbol{r}}$) ▷ Return LOCATOR resistances to their original state
  Call $\mathcal{C}^i.$ROLLBACK() to undo all TEMPORARYUPDATES for all $\mathcal{C}^i$
  return $\boldsymbol{f}, \mu$

- *Given an optimal solution for $\mathcal{I}'$, an optimal minimum cost flow solution for $\mathcal{I}$ can be computed in $O(m)$*

Therefore we now have a poly(m)-central solution for an instance $\mathcal{I}$. We can now apply Lemma 3.10 to get a μ'-central solution with $\mu' \leq m^{-10}$. Then we can apply the following lemma to round the solution, which follows from Lemma 5.4 in [AMV20].

Lemma 3.13 (Interior point method rounding)**.** *Given a min cost flow instance $\mathcal{I}$ and a μ-central flow $\boldsymbol{f}$ for $\mu \leq m^{-10}$, there is an algorithm that runs in time $\widetilde{O}(m)$ and returns an optimal integral flow.*

By Lemma 3.12, this solution can be turned into an exact solution for the original instance. As Lemma 3.10 succeeds with high probability, the whole algorithm does too.

Runtime. To determine the final runtime, we analyze each operation in Algorithm 1 separately.

The INITIALIZE (Lemma 3.12) and ROUND (Lemma 3.13) operations take time $\widetilde{O}(m)$. Now, the runtime of Lemma 3.10 is $\widetilde{O}\left(m^{3/2}/k\right)$ plus the runtime incurred because of calls to the locator $\mathcal{L}$. We will use the runtimes per operation from Lemma 3.8.

$\mathcal{L}$.SOLVE: This operation is run $\widetilde{O}\left(m^{1/2}k^3\right)$ times, and each of these costs $\widetilde{O}\left(\frac{\beta m}{\varepsilon^2}\right) = \widetilde{O}\left(mk^{12}\beta\right)$. Therefore in total $\widetilde{O}\left(m^{3/2}k^{15}\beta\right)$.

We pick β by $m^{3/2}k^{15}\beta \leq m^{3/2}/k$ as $\beta = k^{-16}$, so the runtime is $\widetilde{O}\left(m^{3/2}/k\right)$. Note that this satisfies the constraint $\beta \geq \widetilde{\Omega}\left(k^3/m^{1/4}\right)$ as long as $k \leq \widetilde{\Omega}\left(m^{1/76}\right)$.

$\mathcal{L}$.BATCHUPDATE: This is run $\widetilde{O}\left(m^{1/2}/\alpha^{1/4}\right)$ times with empty arguments, each of which takes time $\widetilde{O}\left(m/\varepsilon^2\right) = \widetilde{O}\left(mk^{12}\right)$. The total runtime because of these is $\widetilde{O}\left(m^{3/2}k^{12}\alpha^{-1/4}\right)$. As we need this to be below $\widetilde{O}\left(m^{3/2}/k\right)$, we set $\alpha = k^{52}$.

This operation is also run $\widetilde{O}\left(m^{1/2}k^{-1}\widehat{T}^{-1}\right)$ times with some non-empty argument Z, each of which takes time $\widetilde{O}\left(m/\varepsilon^2 + |Z|/(\varepsilon^2\beta^2)\right) = \widetilde{O}\left(mk^{12} + k^{44}|Z|\right)$. As by Lemma 3.10 the total sum of $|Z|$ over all calls is $\widetilde{O}\left(mk^3\beta^{1/2}\right) = \widetilde{O}\left(mk^{-5}\right)$, we get a runtime of

$$\widetilde{O}\left(m^{1/2}k^{-1}\widehat{T}^{-1} \cdot mk^{12} + k^{44} \cdot mk^{-5}\right) = \widetilde{O}\left(m^{3/2}k^{11}\widehat{T}^{-1} + mk^{39}\right) .$$

In order to set the first term to be at most $\widetilde{O}\left(m^{3/2}/k\right)$, we set $\widehat{T} = k^{12}$.

Therefore the total runtime of this operation is $\widetilde{O}\left(m^{3/2}/k + mk^{39}\right)$.

$\mathcal{L}$.UPDATE: This is run $\widetilde{O}\left(m^{1/2}\left(k^6\widehat{T} + k^{15}\right)\right) = \widetilde{O}\left(m^{1/2}k^{18}\right)$ times and the amortized cost per operation is

$$\begin{aligned}
&\widetilde{O}\left(m \cdot \frac{\widehat{\varepsilon}\alpha^{1/2}}{\varepsilon^3} + \widehat{\varepsilon}^{-4}\varepsilon^{-2}\beta^{-8} + \widehat{\varepsilon}^{-2}\varepsilon^{-4}\alpha^2\beta^{-6}\right) \\
&= \widetilde{O}\left(m \cdot k^{44}\widehat{\varepsilon} + k^{140}\widehat{\varepsilon}^{-4} + k^{224}\widehat{\varepsilon}^{-2}\right) ,
\end{aligned}$$

so in total

$$m^{3/2}k^{62}\widehat{\varepsilon} + m^{1/2}k^{158}\widehat{\varepsilon}^{-4} + m^{1/2}k^{242}\widehat{\varepsilon}^{-2} .$$

As we need the first term to be $\widetilde{O}\left(m^{3/2}/k\right)$, we set $\widehat{\varepsilon} = k^{-63}$. Therefore the total runtime is

$$\widetilde{O}\left(m^{3/2}/k + m^{1/2}k^{410} + m^{1/2}k^{368}\right) = \widetilde{O}\left(m^{3/2}/k + m^{1/2}k^{410}\right) .$$

$\mathcal{L}$.INITIALIZE: This is run $\widetilde{O}\left(k^3\beta^{-1/2}\right) = k^{11}$ times in total, and the runtime for each run is

$$\widetilde{O}\left(m \cdot \left(\widehat{\varepsilon}^{-4}\beta^{-8} + \widehat{\varepsilon}^{-2}\varepsilon^{-2}\alpha^2\beta^{-4}\right)\right) = \widetilde{O}\left(m \cdot \left(k^{380} + k^{306}\right)\right) = \widetilde{O}\left(m \cdot k^{380}\right) ,$$

so in total $\widetilde{O}\left(mk^{380}\right)$.

Therefore, for the whole algorithm, we get $\widetilde{O}\left(m^{3/2}/k + m^{1/2}k^{410} + mk^{380}\right)$ which after balancing gives $k = m^{1/762}$.

4 An Efficient $(\alpha, \beta, \varepsilon)$-LOCATOR

In this section we will show how to implement an $(\alpha, \beta, \varepsilon)$-LOCATOR, as defined in Definition 3.7. In order to maintain the approximate electrical flow $\widetilde{\boldsymbol{f}}$ required by Lemma 3.8 we will keep a vertex sparsifier in the form of a sparsified Schur complement onto some vertex set C. As in [GLP21], we choose C to be a *congestion reduction subset.*

Definition 4.1 (Congestion reduction subset [GLP21])**.** *Given a graph $G(V, E)$ with resistances $\boldsymbol{r}$ and any parameter $\beta \in (0, 1)$, a vertex subset $C \subseteq V$ is called a β-*congestion reduction subset *(or just* congestion reduction subset*) if:*

- $|C| \leq O(\beta m)$
- *For any $u \in V$, a random walk starting from u that visits $\widetilde{\Omega}\left(\beta^{-1}\log n\right)$ distinct vertices hits C with high probability*

- *If we generate* $\deg(u)$ *random walks from each* $u \in V \backslash C$*, the expected number of these that hit some fixed* $v \in V \backslash C$ *before* C *is* $\widetilde{O}\left(1/\beta^2\right)$*. Concretely:*

$$\sum_{u \in V} \deg(u) \cdot p_v^{C \cup \{v\}}(u) \leq \widetilde{O}\left(1/\beta^2\right) . \tag{8}$$

The following lemma shows that such a vertex subset can be constructed efficiently:

Lemma 4.2 (Construction of congestion reduction subset [GLP21])**.** *Given a graph* $G(V, E)$ *with resistances* $\boldsymbol{r}$ *and a parameter* $\beta \in (0, 1)$*, there is an algorithm that generates a* β*-congestion reduction subset in time* $\widetilde{O}\left(m/\beta^2\right)$*.*

Intuitively, (8) says that "not too many" random walks go through a given vertex before reaching C. This property is crucial for ensuring that when inserting a new vertex into C, the data structure will not have to change too much. As we will see in Section 4.1, this property plays an even more central role when general demands are introduced, as it allows us to show that the demands outside C can be pushed to C. Additionally, in Section 4.2 we will use it to show that edges that are too far from C in effective resistance metric are not *important*, in the sense that neither can they get congested, nor can their demand congest anything else.

4.1 Moving demands to the sparsifier

The goal of this section is to show that if C is a congestion reduction subset, then any demand of the form $\boldsymbol{d} = \boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}$ for some $\boldsymbol{q} \in [-1, 1]^m$ can be approximated by $\boldsymbol{\pi}^C(\boldsymbol{d})$, i.e. its demand projection onto C (Definition 2.8). This allows us to move all demands to the sublinear-sized C and thus enables us to work with the Schur complement of G onto C.

Lemma 4.3. *Consider a graph* $G(V, E)$ *with resistances* $\boldsymbol{r}$ *and Laplacian* $\boldsymbol{L}$*, a* β*-congestion reduction subset* C*, and a demand* $\boldsymbol{d} = \delta \boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}$ *for some* $\delta > 0$ *and* $\boldsymbol{q} \in [-1, 1]^m$*. Then, the potential embedding defined as*

$$\boldsymbol{\phi} = \boldsymbol{L}^+ \left(\boldsymbol{d} - \boldsymbol{\pi}^C(\boldsymbol{d})\right)$$

has congestion $\delta \cdot \widetilde{O}\left(1/\beta^2\right)$*, i.e.* $\left\|\frac{\boldsymbol{B}\boldsymbol{\phi}}{\sqrt{\boldsymbol{r}}}\right\|_\infty \leq \delta \cdot \widetilde{O}\left(1/\beta^2\right)$*.*

We first prove a restricted version of the lemma where $\boldsymbol{d}$ is an $s - t$ demand. Then, Lemma 4.3 follows trivially by applying (8).

Lemma 4.4. *Consider a graph* $G(V, E)$ *with resistances* $\boldsymbol{r}$ *and Laplacian* $\boldsymbol{L}$*, a* β*-congestion reduction subset* C*, and a demand* $\boldsymbol{d} = \delta \boldsymbol{B}^\top \frac{\boldsymbol{1}_{st}}{\sqrt{\boldsymbol{r}}}$ *for some* $\delta > 0$ *and* $(s, t) \in E \backslash E(C)$*. Then, for the potential embedding defined as*

$$\boldsymbol{\phi} = \boldsymbol{L}^+ \left(\boldsymbol{d} - \boldsymbol{\pi}^C(\boldsymbol{d})\right)$$

it follows that for any $e = (u, v) \in E$ *we have*

$$\left|\frac{(\boldsymbol{B}\boldsymbol{\phi})_e}{\sqrt{r_e}}\right| \leq 2\delta \cdot \left(p_u^{C \cup \{u\}}(s) + p_v^{C \cup \{v\}}(s) + p_u^{C \cup \{u\}}(t) + p_v^{C \cup \{v\}}(t)\right) .$$

The proof of Lemma 4.4 appears in Appendix D.1.

4.2 ε-Important edges

In this section we will show that the effect of edges that are "far" from the congestion reduction subset C is negligible, as both their congestion and the congestion incurred because of their demands are small. More specifically, given a demand $\boldsymbol{d}$ supported on C with energy ≤ 1, i.e. $\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{d}\right) \leq 1$, the congestion $\boldsymbol{\rho} = \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{d}_C$ that it induces satisfies:

$$\begin{aligned}
|\rho_e| &= \left|\left\langle \mathbf{1}_e, \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{d}\right\rangle\right| \\
&= \left|\left\langle \boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}, \boldsymbol{L}^{+}\boldsymbol{d}\right\rangle\right| \\
&= \left|\left\langle \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right), SC^{+}\boldsymbol{d}_C\right\rangle\right| \\
&\leq \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right)\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{d}\right)} \\
&\leq \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right)}\,.
\end{aligned}$$

For the last equality we used Fact 2.9, for the first inequality we applied Cauchy-Schwarz, and for the second one we used the upper bound on the energy required to route $\boldsymbol{d}$. Therefore, if we bound the energy of the projection of $\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}$ onto C, we can also bound the congestion of e. This is done in the following lemma, whose proof appears in Appendix D.2.

Lemma 4.5. *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$ and $C \subseteq V$. Then, for all $e \in E\backslash E(C)$ we have*

$$\sqrt{\mathcal{E}_r\left(\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{r_e}}\right)\right)} \leq 6 \cdot \sqrt{\frac{r_e}{R_{eff}(C,e)}}\,.$$

This is the consequence of the following lemma, which bounds the magnitude of the projection on a specific vertex, based on its effective resistance distance from e, as well as hitting probabilities from e to C. The proof appears in Appendix E.1.

Lemma 4.6. *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$ and a subset of vertices $C \subseteq V$. For any vertex $v \in V\backslash C$ we have that*

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right| \leq \left(p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)\right) \cdot \frac{\sqrt{r_e}}{R_{eff}(v,e)}\,.$$

This lemma is complementary to the more immediate property

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right| \leq \left(p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)\right) \cdot \frac{1}{\sqrt{r_e}}\,,$$

and they are both used in Section 5 in order to estimate demand projections. In fact, the just by multiplying these two, we get the following lemma, which is nice because it doesn't depend on r_e:

Lemma 4.7. *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$ and a subset of vertices $C \subseteq V$. For any vertex $v \in V \backslash C$ we have that*

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right| \le (p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)) \cdot \frac{1}{\sqrt{R_{eff}(v,e)}}\,.$$

By the previous discussion, Lemma 4.6 implies that if $|\rho_e| \ge \varepsilon$, then $R_{eff}(C,e) \le r_e \cdot \frac{36}{\varepsilon^2}$. This motivates the following definition of ε-*important edges.*

Definition 4.8 (Important edges)**.** *An edge $e \in E$ is called* ε-important *(or just* important*) if $R_{eff}(C,e) \le r_e/\varepsilon^2$.*

Now it is time for the main lemma of this section, which uses Lemma 4.5 to show that if our goal is to detect edges with congestion $\ge \varepsilon$, it is sufficient to restrict to computing demand projections of $\Omega(\varepsilon)$-important edges. Its proof appears in Appendix D.3.

Lemma 4.9 (Localization lemma)**.** *Let $\boldsymbol{\phi}^*$ be any solution of*

$$\boldsymbol{L}\boldsymbol{\phi}^* = \delta \cdot \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{p}}{\sqrt{\boldsymbol{r}}}\right)\,,$$

where $\boldsymbol{r}$ are any resistances, $\boldsymbol{p} \in [-1,1]^m$, and $C \subseteq V$. Then, for any $e \in E$ that is not ε-important we have $\left|\frac{\boldsymbol{B}\boldsymbol{\phi}^*}{\sqrt{\boldsymbol{r}}}\right|_e \le 6\varepsilon$.

4.3 Proving Lemma 3.8

Before moving to the description of how Locator works and its proof, we will provide a lemma which bounds how fast a demand projection changes.

We will use the following observation, which states that if our congestion reduction subset C contains an βm-sized uniformly random edge subset, then with high probability, effective resistance neighborhoods that are disjoint from C only have $\widetilde{O}\left(\beta^{-1}\right)$ edges. Note that this is will be true throughout the algorithm as long as the resistances do not depend on the randomness of C. This is true, as resistance updates are only ever given as inputs to Locator.

Lemma 4.10 (Few edges in a small neighborhood)**.** *Let $\beta \in (0,1)$ be a parameter and C be a vertex set which contains a subset of βm edges sampled at random. Then with high probability, for any $v \in V\backslash C$ we have that $|N_E(v, R_{eff}(C,v)/2)| \le 10\beta^{-1}\ln m$, where*

$$N_E(v,R) := \{e \in E \mid R_{eff}(e,v) \le R\}\,.$$

Proof. Suppose that for some vertex $v \in V\backslash C$, $|N_E(u, R_{eff}(C,v)/2)| \ge 10\beta^{-1}\ln m$. Since by construction C contains a random edge subset of size βm, with high probability $N_E(u, R_{eff}(C,u)/2) \cap C \ne \emptyset$, so there exists $u \in C$ such that $R_{eff}(u,v) \le R_{eff}(C,v)/2$. This is a contradiction since $u \in C$ implies $R_{eff}(C,v) \le R_{eff}(u,v)$. Union bounding over all v yields the claim. □

Using this fact, we can now show that the change of the demand projection (measured in energy) is quite mild. The proof of the following lemma can be found in Appendix D.4.

Lemma 4.11 (Projection change)**.** *Consider a graph $G(V, E)$ with resistances $\boldsymbol{r}$, $\boldsymbol{q} \in [-1, 1]^m$, and a β-congestion reduction subset C. Then, with high probability,*

$$\sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right) - \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right)\right)} \leq \widetilde{O}\left(\beta^{-2}\right) .$$

The above lemma can be applied over multiple vertex insertions and resistance changes, to bound the overall energy change. This is shown in the following lemma, which is proved in Appendix D.5:

Lemma 4.12. *Consider a graph $G(V, E)$ with resistances $\boldsymbol{r}^0$, $\boldsymbol{q}^0 \in [-1, 1]^m$, a β-congestion reduction subset C^0, and a fixed sequence of updates, where the i-th update $i \in \{0, \ldots, T-1\}$ is of the following form:*

- AddTerminal*(v^i): Set $C^{i+1} = C^i \cup \{v^i\}$ for some $v^i \in V \backslash C^i$, $q_e^{i+1} = q_e^i, r_e^{i+1} = r_e^i$*
- Update*($e^i, \boldsymbol{q}, \boldsymbol{r}$) Set $C^{i+1} = C^i$, $q_e^{i+1} = q_e$ $r_e^{i+1} = r_e$, where $e^i \in E(C^i)$*

Then, with high probability,

$$\sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{\pi}^{C^0,\boldsymbol{r}^0}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^0}{\sqrt{\boldsymbol{r}^0}}\right) - \boldsymbol{\pi}^{C^T,\boldsymbol{r}^T}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^T}{\sqrt{\boldsymbol{r}^T}}\right)\right)} \leq \widetilde{O}\left(\max_{i\in\{0,\ldots,T-1\}} \left\|\frac{\boldsymbol{r}^T}{\boldsymbol{r}^i}\right\|_\infty^{1/2} \beta^{-2}\right) \cdot T .$$

We are now ready to describe the Locator data structure. We will give an outline here, and defer the full proof to Appendix D.6. The goal of an $(\alpha, \beta, \varepsilon)$-Locator is, given some flow $\boldsymbol{f}$ with slacks $\boldsymbol{s}$ and resistances $\boldsymbol{r}$, to compute all $e \in E$ such that $\sqrt{r_e}\left|\widetilde{f}_e^*\right| \geq \varepsilon$, where

$$\widetilde{\boldsymbol{f}}^* = \delta g(\boldsymbol{s}) - \delta \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{B}^\top g(\boldsymbol{s})$$

($\boldsymbol{L} = \boldsymbol{B}^\top\boldsymbol{R}^{-1}\boldsymbol{B}$), where $\delta = 1/\sqrt{m}$.

If we set $\rho_e^* = \sqrt{r_e}\widetilde{f}_e^*$, we can equivalently write

$$\boldsymbol{\rho}^* = \delta\sqrt{\boldsymbol{r}}g(\boldsymbol{s}) - \delta\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{B}^\top g(\boldsymbol{s}) \,,$$

and require to find all the entries of $\boldsymbol{\rho}^*$ with magnitude at least ε. As $\delta\left\|\sqrt{\boldsymbol{r}}g(\boldsymbol{s})\right\|_\infty \leq \delta \leq \varepsilon/100$, we can concentrate on the second term, and denote

$$\boldsymbol{\rho}'^* = \delta\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{B}^\top g(\boldsymbol{s})$$

for convenience.

First, we use Lemma 4.3 to show that

$$\delta\left\|\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\left(g(\boldsymbol{s}) - \boldsymbol{\pi}^C(g(\boldsymbol{s}))\right)\right\|_\infty \leq \delta\cdot\widetilde{O}\left(\beta^{-2}\right) \leq \varepsilon/100 \,.$$

Now, let's set $\boldsymbol{\pi}_{old} = \boldsymbol{\pi}^{C^0}\left(g(\boldsymbol{s}^0)\right)$, where C^0 was the vertex set of the sparsifier and $\boldsymbol{s}^0$ the slacks after the last call to BatchUpdate. As we will be calling BatchUpdate at least every T calls to Update for some $T \geq 1$, Lemma 4.12 implies that

$$\delta\left\|\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\left(\boldsymbol{\pi}^C(g(\boldsymbol{s})) - \boldsymbol{\pi}_{old}\right)\right\|_\infty \leq \delta\cdot\widetilde{O}\left(\alpha\beta^{-2}\right)T \leq \varepsilon/100 \,,$$

as long as $T \leq \varepsilon\sqrt{m}/\widetilde{O}\left(\alpha\beta^{-2}\right)$.

Importantly, we will never be *removing* vertices from C, so $C^0 \subseteq C$. This implies that it suffices to find the large entries of

$$\delta \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{\pi}_{old}\,.$$

Now, note that for any edge e that was *not* $\varepsilon/(100\alpha)$-important for C^0 and corresponding resistances $\boldsymbol{r}^0$, we have

$$\begin{aligned}
&\delta\left|\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{\pi}_{old}\right|_e\\
&\leq \delta\sqrt{\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{\pi}^{C^0}(\boldsymbol{B}^{\top}\frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}))}\sqrt{\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{\pi}_{old})}\\
&\leq \delta\cdot\sqrt{\alpha}\frac{\varepsilon}{100\alpha}\cdot\sqrt{2\alpha m}\\
&= \varepsilon/50\,,
\end{aligned}$$

where we used Lemma 4.5 and the fact that $\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{\pi}^{C^0}(g(\boldsymbol{s}^0))) \leq 2\mathcal{E}_{\boldsymbol{r}}(g(\boldsymbol{s}^0)) \leq 2\alpha m$. Therefore it suffices to approximate

$$\delta \boldsymbol{I}_S\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^{+}\boldsymbol{\pi}_{old}\,,$$

where S was the set of $\frac{\varepsilon}{100\alpha}$-important edges last computed during the last call to BatchUpdate.

Now, we will use the sketching lemma from (Lemma 5.1, [GLP21] v2), which shows that in order to find all $\Omega(\varepsilon)$ large entries of this vector, it suffices to compute the inner products

$$\delta\left\langle \boldsymbol{\pi}^{C}\left(\boldsymbol{B}^{\top}\frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}}\right), SC^{+}\boldsymbol{\pi}_{old}\right\rangle$$

for $i \in [\widetilde{O}\left(\varepsilon^{-2}\right)]$ up to $O(\varepsilon)$ accuracy. Here SC is the Schur complement onto C.

Based on this, there are two types of quantities that we will maintain:

- $\widetilde{O}\left(1/\varepsilon^2\right)$ approximate demand projections $\widetilde{\boldsymbol{\pi}}^{C}\left(\boldsymbol{B}^{\top}\frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}}\right)$, and
- an approximate Schur complement $\widetilde{SC}$ of G onto C.

For the latter, we will directly use the dynamic Schur complement data structure DynamicSC that was also used by [GLP21] and is based on [DGGP19]. For completeness, we present this data structure in Appendix A.

For the former, we will need $\widetilde{O}\left(1/\varepsilon^2\right)$ data structures for maintaining demand projections onto C, under vertex insertions to C. The guarantees of each such a data structure, that we call an $(\alpha,\beta,\varepsilon)$-DemandProjector, are as follows.

Definition 4.13 ($(\alpha,\beta,\varepsilon)$-DemandProjector)**.** *Let $\widehat{\varepsilon} \in (0,\varepsilon)$ be a tradeoff parameter. Given a graph $G(V,E)$, resistances $\boldsymbol{r}$, and a vector $\boldsymbol{q} \in [-1,1]^m$, an $(\alpha,\beta,\varepsilon)$-DemandProjector is a data structure that maintains a vertex subset $C \subseteq V$ and an approximation to the demand projection $\boldsymbol{\pi}^{C}\left(\boldsymbol{B}^{\top}\frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right)$, with high probability under oblivious adversaries. The following operations are supported:*

- Initialize$(C, \boldsymbol{r}, \boldsymbol{q}, S, \mathcal{P})$*: Initialize the data structure in order to maintain an approximation of* $\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$*, where* $C \subseteq V$ *is a* β*-congestion reduction subset,* $\boldsymbol{r}$ *are resistances,* $\boldsymbol{q} \in [-1,1]^m$*, and* $S \subseteq E$ *is a subset of* γ*-important edges.* $\mathcal{P} = \{\mathcal{P}^{u,e,i} \mid u \in V, e \in E, u \in e, i \in [h]\}$ *for some* $h \in \mathbb{Z}_{\geq 1}$*, is a set of independent random walks from* u *to* C *for any* u*.*

- AddTerminal$(v, \widetilde{R}_{eff}(C, v))$*: Insert* v *into* C*. Also,* $\widetilde{R}_{eff}(C, v)$ *is an estimate of* $R_{eff}(C, v)$ *such that* $\widetilde{R}_{eff}(C, v) \approx_2 R_{eff}(C, v)$*. Returns an estimate*

$$\widetilde{\pi}_v^{C \cup \{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right)$$

  for the demand projection of $\boldsymbol{q}$ *onto* $C \cup \{v\}$ *at coordinate* v *such that*

$$\left|\widetilde{\pi}_v^{C \cup \{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C \cup \{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right)\right| \leq \frac{\widehat{\varepsilon}}{\sqrt{R_{eff}(v, C)}}\,.$$

- Update$(e, \boldsymbol{r}', \boldsymbol{q}')$*: Set* $r_e = r'_e$ *and* $q_e = q'_e$*, where* $e \in E(C)$*, and* $q'_e \in [-1,1]$*. Furthermore,* r'_e *satisfies the inequality* $r_e^{\max}/\alpha \leq r'_e \leq \alpha \cdot r_e^{\min}$*, where* $r_e^{\min}$ *and* $r_e^{\max}$ *represent the minimum, respectively the maximum values that the resistance of* e *has had since the last call to* Initialize*.*

- Output()*: Output* $\widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$ *such that such that after* $T \leq n^{O(1)}$ *calls to* AddTerminal*, for any fixed vector* $\boldsymbol{\phi}$*,* $E_{\boldsymbol{r}}(\boldsymbol{\phi}) \leq 1$*, with high probability*

$$\left|\left\langle \widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right), \boldsymbol{\phi}\right\rangle\right| \leq \widehat{\varepsilon} \cdot \sqrt{\alpha} \cdot T\,.$$

We will implement such a data structure in Section 5, where we will prove the following lemma:

Lemma 4.14 (Demand projection data structure)**.** *For any graph* $G(V, E)$ *and parameters* $\widehat{\varepsilon} \in (0, \varepsilon)$*,* $\beta \in (0,1)$*, there exists an* $(\alpha, \beta, \varepsilon)$*-*DemandProjector *for* G *which, given access to* $h = \widetilde{\Theta}(\widehat{\varepsilon}^{-4}\beta^{-6} + \widehat{\varepsilon}^{-2}\beta^{-2}\gamma^{-2})$ *precomputed independent random walks from* u *to* C *for each* $e \in E$*,* $u \in e$*, has the following runtimes per operation:*

- Initialize*:* $\widetilde{O}(m)$*.*
- AddTerminal*:* $\widetilde{O}\left(\widehat{\varepsilon}^{-4}\beta^{-8} + \widehat{\varepsilon}^{-2}\beta^{-6}\gamma^{-2}\right)$*.*
- Update*:* $O(1)$*.*
- Output*:* $O(\beta m + T)$*, where* T *is the number of calls made to* AddTerminal *after the last call to* Initialize*.*

Now we describe the way we will use the DemandProjectors and DynamicSC to get an $(\alpha, \beta, \varepsilon)$-Locator $\mathcal{L}$.

$\mathcal{L}$**.Initialize**: Every time $\mathcal{L}$.Initialize is called, we first generate a β-congestion reduction subset C based on Lemma 4.2 (takes time $\widetilde{O}\left(m/\beta^2\right)$), then a sketching matrix $\boldsymbol{Q}$ and its rows

Algorithm 3 LOCATOR $\mathcal{L}$.INITIALIZE

procedure $\mathcal{L}$.INITIALIZE($\boldsymbol{f}$)
  $\boldsymbol{s}^+ = \boldsymbol{u} - \boldsymbol{f}$, $\boldsymbol{s}^- = \boldsymbol{f}$, $\boldsymbol{r} = \frac{1}{(\boldsymbol{s}^+)^2} + \frac{1}{(\boldsymbol{s}^-)^2}$
  $\boldsymbol{Q}$ = Sketching matrix produced by (Lemma 5.1, [GLP21] v2)
  DYNAMICSC = DYNAMICSC.INITIALIZE($\boldsymbol{G}, \emptyset, \boldsymbol{r}, \varepsilon, \beta$)
  C = DYNAMICSC.C ▷ β-congestion reduction subset
  Estimate $\widetilde{R}_{eff}(C, e) \approx_4 R_{eff}(C, e)$ using Lemma B.2
  $S = \left\{ e \in E \mid \widetilde{R}_{eff}(C, e) \leq r_e \cdot \left(\frac{100\alpha}{\varepsilon}\right)^2 \right\}$
  $h = \widetilde{\Theta}\left(\widehat{\varepsilon}^{-4}\beta^{-6} + \widehat{\varepsilon}^{-2}\varepsilon^{-2}\alpha^2\beta^{-2}\right)$
  Sample walks $\mathcal{P}^{u,e,i}$ from u to C for $e \in E \setminus E(C)$, $u \in e$, $i \in [h]$ (Lemma 5.15, [GLP21] v2)
  DP^i = DEMANDPROJECTOR.INITIALIZE($C, \boldsymbol{r}, \boldsymbol{q}^i, S, \mathcal{P}$) for all rows $\boldsymbol{q}^i$ of $\boldsymbol{Q}$
  $\mathcal{L}$.BATCHUPDATE($\emptyset$)

$\boldsymbol{q}^i$ for $i \in \left[\widetilde{O}\left(1/\varepsilon^2\right)\right]$ as in (Lemma 5.1, [GLP21] v2) (takes time $\widetilde{O}\left(m/\varepsilon^2\right)$), and finally random walks $\mathcal{P}^{u,e,i}$ from u to C for each $u \in V$, $e \in E \backslash E(C)$ with $u \in e$, and $i \in [h]$, where $h = \widetilde{O}\left(\widehat{\varepsilon}^{-4}\beta^{-6} + \widehat{\varepsilon}^{-2}\varepsilon^{-2}\alpha^2\beta^{-2}\right)$ as in (Lemma 5.15, [GLP21] v2) (takes time $\widetilde{O}\left(h/\beta^2\right)$ for each (u, e)).

We also compute $\widetilde{R}_{eff}(C, u) \approx_2 R_{eff}(C, u)$ for all $u \in V$ as described in Lemma B.2 so that we can let S be a subset of $\varepsilon/(100\alpha)$-important edges that contains all $\varepsilon/(200\alpha)$-important edges. This takes time $\widetilde{O}(m)$. Then, we call DYNAMICSC.INITIALIZE($G, C, \boldsymbol{r}, O(\varepsilon), \beta$) (from Appendix A) to initialize the dynamic Schur complement onto C, with error tolerance $O(\varepsilon)$, which takes time $\widetilde{O}\left(m \cdot \frac{1}{\varepsilon^4\beta^4}\right)$, as well as DEMANDPROJECTOR.INITIALIZE($C, \boldsymbol{r}, \boldsymbol{q}, S, \mathcal{P}$) for the $\widetilde{O}\left(1/\varepsilon^2\right)$ DEMANDPROJECTORS, i.e. one for each $\boldsymbol{q} \in \{\boldsymbol{q}^i \mid i \in [\widetilde{O}\left(1/\varepsilon^2\right)]\}$. Also, we compute

$$\boldsymbol{\pi}^{old} = \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right),$$

which takes $\widetilde{O}(m)$ as in DEMANDPROJECTOR.INITIALIZE. All of this takes $\widetilde{O}\left(m \cdot \left(\frac{1}{\widehat{\varepsilon}^4\beta^8} + \frac{\alpha^2}{\widehat{\varepsilon}^2\varepsilon^2\beta^4}\right)\right)$.

$\mathcal{L}$.UPDATE: Now, whenever $\mathcal{L}$.UPDATE is called on an edge e, either $e \in E(C)$ or $e \notin E(C)$. In the first case we simply call UPDATE on DYNAMICSC and all DEMANDPROJECTORS.

In the second case, we first call DYNAMICSC.ADDTERMINAL on one endpoint v of e. After doing this we can also get an estimate $\widetilde{R}_{eff}(C, v) \approx_2 R_{eff}(C, v)$ by looking at the edges between C and v in the sparsified Schur complement. By the guarantees of the expander decomposition used inside DYNAMICSC [GLP21], the number of expanders containing v, amortized over all calls to DYNAMICSC.ADDTERMINAL, is $O(\text{poly}\log(n))$. As the sparsified Schur complement contains $\widetilde{O}\left(1/\varepsilon^2\right)$ neighbors of v from each expander, the amortized number of neighbors of v in the sparsified Schur complement is $\widetilde{O}\left(1/\varepsilon^2\right)$, and the amortized runtime to generate them (by random sampling) is $\widetilde{O}\left(1/\varepsilon^2\right)$.

Given the resistances $r_1, \ldots, r_l$ of these edges, setting $\widetilde{R}_{eff}(C, v) = \left(\sum_{i=1}^{l} r_i^{-1}\right)^{-1}$ we guarantee that $\widetilde{R}_{eff}(C, v) \approx_{1+O(\varepsilon)} R_{eff}(C, v)$, by the fact that DYNAMICSC maintains an $(1 + O(\varepsilon))$-approximate sparsifier of the Schur complement. Then, we call ADDTERMINAL($v, \widetilde{R}_{eff}(C, v)$) on all DEMANDPROJECTORS.

Algorithm 4 LOCATOR $\mathcal{L}$.UPDATE and $\mathcal{L}$.BATCHUPDATE

procedure UPDATE($e = (u, w), \boldsymbol{f}$)
    $s_e^+ = u_e - f_e$, $s_e^- = f_e$, $r_e = \frac{1}{(s_e^+)^2} + \frac{1}{(s_e^-)^2}$
    $\widetilde{R}_{eff}(C, u) =$ DYNAMICSC.ADDTERMINAL(u)
    $\widetilde{R}_{eff}(C \cup \{u\}, w) =$ DYNAMICSC.ADDTERMINAL(w)
    $C = C \cup \{u, w\}$
    for $i = 1, \ldots, \widetilde{O}\left(1/\varepsilon^2\right)$ **do**
        DP^i.ADDTERMINAL($u, \widetilde{R}_{eff}(C, u)$)
        DP^i.ADDTERMINAL($w, \widetilde{R}_{eff}(C \cup \{u\}, w)$)
    DYNAMICSC.UPDATE(e, r_e)
    for $i = 1 \ldots \widetilde{O}\left(1/\varepsilon^2\right)$ **do**
        DP^i.UPDATE($e, \boldsymbol{r}, \boldsymbol{q}^i$)
procedure BATCHUPDATE($Z, \boldsymbol{f}$)
    $\boldsymbol{s}^+ = \boldsymbol{u} - \boldsymbol{f}$, $\boldsymbol{s}^- = \boldsymbol{f}$, $\boldsymbol{r} = \frac{1}{(\boldsymbol{s}^+)^2} + \frac{1}{(\boldsymbol{s}^-)^2}$
    Estimate $\widetilde{R}_{eff}(C, e) \approx_4 R_{eff}(C, e)$ using Lemma B.2
    $S = \left\{ e \in E \mid \widetilde{R}_{eff}(C, e) \leq r_e \cdot \left(\frac{100\alpha}{\varepsilon}\right)^2 \right\}$ ▷ $\frac{\varepsilon}{100\alpha}$-important edges
    for $e = (u, w) \in Z$ **do**
        DYNAMICSC.ADDTERMINAL(u)
        DYNAMICSC.ADDTERMINAL(w)
        $C = C \cup \{u, w\}$
        DYNAMICSC.UPDATE(e, r_e)
    for $i = \left[\widetilde{O}\left(1/\varepsilon^2\right)\right]$ **do**
        DP^i.INITIALIZE($C, \boldsymbol{r}, \boldsymbol{q}^i, S, \mathcal{P}$)
    $\boldsymbol{\pi}_{old} = \frac{1}{\sqrt{m}} \cdot \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-}}{\boldsymbol{r}} \right)$ ▷ Compute exactly using Laplacian solve

After repeating the same process for the other endpoint of e, we finally call UPDATE on DYNAMICSC and all DEMANDPROJECTORS. This takes time $\widetilde{O}\left(\frac{1}{\varepsilon^2\beta^2}\right)$ because of the Schur complement and amortized $\widetilde{O}\left(m \cdot \frac{\widehat{\varepsilon}\alpha^{1/2}}{\varepsilon} + \frac{1}{\widehat{\varepsilon}^4\beta^8} + \frac{\alpha^2}{\widehat{\varepsilon}^2\varepsilon^2\beta^{-6}}\right)$ for each of the demand projectors, so the total amortized runtime is $\widetilde{O}\left(m \cdot \frac{\widehat{\varepsilon}\alpha^{1/2}}{\varepsilon^3} + \frac{1}{\widehat{\varepsilon}^4\varepsilon^2\beta^8} + \frac{\alpha^2}{\widehat{\varepsilon}^2\varepsilon^4\beta^6}\right)$.

$\mathcal{L}$.BATCHUPDATE: When $\mathcal{L}$.BATCHUPDATE is called on a set of edges Z, we add them one by one in the DYNAMICSC data structure following the same process as in $\mathcal{L}$.UPDATE. For the demand projectors, we first manually insert the endpoints of these edges into C and then re-initialize all DEMANDPROJECTORS, by calling INITIALIZE with a new subset S of $\frac{\varepsilon}{200\alpha}$-important edges that contains all $\frac{\varepsilon}{100\alpha}$-important edges. Such a set can be computed by estimating $R_{eff}(C, u)$ for all $u \in V \backslash C$ up to a constant factor and, by Lemma B.2, takes time $\widetilde{O}(m)$. Also, we compute

$$\boldsymbol{\pi}^{old} = \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right) ,$$

which takes $\widetilde{O}(m)$ as in DEMANDPROJECTOR.INITIALIZE. The total runtime of this is $\widetilde{O}\left(m/\varepsilon^2 + |Z|/(\beta^2\varepsilon^2)\right)$.

Algorithm 5 LOCATOR $\mathcal{L}$.SOLVE

```
procedure SOLVE()
    SC~ = DYNAMICSC.SC~()
    φ_old = SC~^+ π_old
    v = 0
    for i = 1, ..., Õ(1/ε²) do
        π~^i = DP^i.OUTPUT()
        v_i = ⟨π~^i, φ_old⟩
    Z = RECOVER(v, ε/100)        ▷ Recovers all ε/2-congested edges (Lemma 5.1, [GLP21] v2)
    return Z
```

$\mathcal{L}$.SOLVE: When $\mathcal{L}$.SOLVE is called, we set $\widetilde{SC} = $ DYNAMICSC.$SC()$, call OUTPUT on all DEMANDPROJECTORS to obtain vectors $\widetilde{\boldsymbol{\pi}}^i$ which are estimators for $\boldsymbol{\pi}^C(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}})$ in the sense of Definition 4.13. Then we compute $v_i = \langle \widetilde{\boldsymbol{\pi}}^i, \widetilde{SC}^+ \boldsymbol{\pi}_{old}\rangle$ where $\boldsymbol{\pi}_{old}$ is the demand projection that was computed exactly the last time BATCHUPDATE was called. These computed terms represent an approximation to the update in $(\boldsymbol{Q\rho})_i$ between two consecutive calls of $\mathcal{L}$.SOLVE. As we will show in the appendix, $\langle \widetilde{\boldsymbol{\pi}}^i, \widetilde{SC}^+ \boldsymbol{\pi}_{old}\rangle$ is an ε-additive approximation of $\langle \boldsymbol{\pi}^C(\boldsymbol{B}^\top \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}}}), \boldsymbol{L}^+ \boldsymbol{\pi}^C(\boldsymbol{B}^\top g(\boldsymbol{s}))\rangle$ for all $i \in \left[\widetilde{O}\left(1/\varepsilon^2\right)\right]$. The key fact that makes this approximation feasible is that although updates to the demand projection are hard to approximate with few samples, when hitting them with the deterministic vector $\boldsymbol{\pi}_{old}$, the resulting inner products strongly concentrate. The runtime of this is $\widetilde{O}\left(\beta m/\varepsilon^2\right)$.

Using these computed values with the ℓ_2 heavy hitter data structure (Lemma 5.1, [GLP21] v2) we get all edges with congestion more than ε. The total runtime is $\widetilde{O}\left(\beta m/\varepsilon^2\right)$.

5 The Demand Projection Data Structure

The main goal of this section is to construct an $(\alpha, \beta, \varepsilon)$-DemandProjector, as defined in Definition 4.13, and thus prove Lemma 4.14. The most important operation that needs to be implemented in order to prove Lemma 4.14 is to maintain the demand projection after inserting a vertex $v \in V \backslash C$ to C. In order to do this, we use the following identity from [GLP21]:

$$\boldsymbol{\pi}^{C\cup\{v\}}(\boldsymbol{d}) = \boldsymbol{\pi}^{C}(\boldsymbol{d}) + \pi_v^{C\cup\{v\}}(\boldsymbol{d}) \cdot \left(\mathbf{1}_v - \boldsymbol{\pi}^C(\mathbf{1}_v)\right) , \tag{9}$$

where $\boldsymbol{d}$ is any demand (in our case, we have $\boldsymbol{d} = \boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}$ for some $S \subseteq E$). For this, we need to compute approximations to $\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$ and $\boldsymbol{\pi}^C(\mathbf{1}_v)$.

In Section 5.1, we will show that if S is a subset of γ-important edges, we can efficiently estimate $\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$ up to additive accuracy $\frac{\widehat{\varepsilon}}{R_{eff}(C,v)}$ by sampling random walks to C starting only from edges with relatively high resistance. For the remaining edges, the γ-importance property will imply that we are not losing much by ignoring them.

Then, in Section 5.2 we will show how to approximate $\mathbf{1}_v - \boldsymbol{\pi}^C(\mathbf{1}_v)$. This is equivalent to estimating the hitting probabilities from v to C. The guarantee that we would ideally like to get is on the error to route

$$\mathcal{E}_{\boldsymbol{r}}\left(\widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v) - \boldsymbol{\pi}^C(\mathbf{1}_v)\right) \le \widehat{\varepsilon}^2 R_{eff}(C,v) . \tag{10}$$

Note that this is not possible to do efficiently for general C. For example, suppose that the hitting distribution is uniform. In this case, $\Omega(|C|)$ random walks are required to get a bound similar to (10). However, it might still be possible to guarantee it by using the structure of C, and this would simplify some parts of our analysis. Instead, we are going to work with the following weaker approximation bound: For any fixed potential vector $\boldsymbol{\phi} \in \mathbb{R}^n$ with $E_{\boldsymbol{r}}(\boldsymbol{\phi}) \le 1$, we have w.h.p.

$$\left|\left\langle \widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v) - \boldsymbol{\pi}^C(\mathbf{1}_v), \boldsymbol{\phi}\right\rangle\right| \le \widehat{\varepsilon}\sqrt{R_{eff}(C,v)} . \tag{11}$$

Now, using these estimation lemmas, we will bound how our demand projection degrades when inserting a new vertex into C. This is stated in the following lemma and proved in Appendix E.6.

Lemma 5.1 (Inserting a new vertex to C)**.** *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$, $\boldsymbol{q} \in [-1,1]^m$, a β-congestion reduction subset C, and $v \in V\backslash C$. We also suppose that we have an estimate of the $C-v$ effective resistance such that $\widetilde{R}_{eff}(C,v) \approx_2 R_{eff}(C,v)$, as well as to independent random walks $\mathcal{P}^{u,e,i}$ for each $u \in V\backslash C$, $e \in E\backslash E(C)$ with $u \in e$, $i \in [h]$, where each random walk starts from u and ends at C.*

If we let S be a subset of γ-important edges for $\gamma > 0$, then for any error parameter $\widehat{\varepsilon} > 0$ we can compute $\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) \in \mathbb{R}$ and $\widetilde{\pi}^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) \in \mathbb{R} \in \mathbb{R}^n$ such that with high probability

$$\left|\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\right| \le \frac{\widehat{\varepsilon}}{\sqrt{R_{eff}(C,v)}} ,$$

as long as $h = \widetilde{\Omega}\left(\widehat{\varepsilon}^{-4}\beta^{-6} + \widehat{\varepsilon}^{-2}\beta^{-2}\gamma^{-2}\right)$. Furthermore, for any fixed $\boldsymbol{\phi}$, $E_{\boldsymbol{r}}(\boldsymbol{\phi}) \le 1$, after T insertions after the last call to Initialize*, with high probability*

$$\left|\left\langle \widetilde{\boldsymbol{\pi}}^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \boldsymbol{\pi}^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right), \boldsymbol{\phi}\right\rangle\right| \le \widehat{\varepsilon}T ,$$

as long as $h = \widetilde{\Omega}\left(\widehat{\varepsilon}^{-2}\beta^{-4}\gamma^{-2}\right)$.

Algorithm 6 DemandProjector DP.AddTerminal

procedure DP.AddTerminal($v, \widetilde{R}_{eff}(C, v)$)
  if $v \in C$ **then**
    return
  $t = t + 1$
  $\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) = 0$
  for $u, e \in S, i$ such that $\mathcal{P}^{u,e,i} \ni v$ and $\widetilde{R}_{eff}(C, v) \leq \frac{1}{\left(\min\{\widehat{\varepsilon}/\widetilde{O}(\beta^{-2}), \gamma/4\}\right)^2} r_e$ **do**
    if $e = (u, *)$ **then**
      $\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) = \widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) + \frac{1}{h}\frac{q_e}{\sqrt{r_e}}$
    else
      $\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) = \widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \frac{1}{h}\frac{q_e}{\sqrt{r_e}}$
    Shortcut $\mathcal{P}^{u,e,i}$ at v
  $h' = \widetilde{O}\left(\widehat{\varepsilon}^{-2}\beta^{-4}\gamma^{-2}\right)$
  $\widetilde{\boldsymbol{\pi}}^C(\boldsymbol{1}_v) = \boldsymbol{0}$
  for $i = 1, \dots h'$ **do**
    Run random walk from v to C with probabilities prop. to $\boldsymbol{r}^{-1}$, let u be the last vertex
    $\widetilde{\pi}_u^C(\boldsymbol{1}_v) = \widetilde{\pi}_u^C(\boldsymbol{1}_v) + \frac{1}{h'}$
  $\widetilde{\boldsymbol{\pi}}^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}) = \widetilde{\boldsymbol{\pi}}^C(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}) + \widetilde{\pi}_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}) \cdot (\boldsymbol{1}_v - \widetilde{\pi}^C(\boldsymbol{1}_v))$
  $C = C \cup \{v\}$, $F = F \backslash \{v\}$

5.1 Estimating $\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$

There is a straightforward algorithm to estimate $\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$. For each edge $e = (u, w) \in E \backslash E(C)$, sample a number of random walks from u and w until they hit $C \cup \{v\}$. Then, add to the estimate $\frac{q_e}{\sqrt{r_e}}$ times the fraction of the random walks starting from u that contain v, minus $\frac{q_e}{\sqrt{r_e}}$ times the fraction of the random walks starting from w that contain v. [GLP21] uses this sampling method together with the following concentration bound, to get a good estimate if the resistances of all congested edges are sufficiently large.

Lemma 5.2 (Concentration inequality 1 [GLP21])**.** *Let $S = X_1 + \cdots + X_n$ be the sum of n independent random variables. The range of X_i is $\{0, a_i\}$ for $a_i \in [-M, M]$. Let t, E be positive numbers such that $t \leq E$ and $\sum_{i=1}^{n} |\mathbb{E}[X_i]| \leq E$. Then*

$$\Pr\left[|S - \mathbb{E}[S]| > t\right] \leq 2\exp\left(-\frac{t^2}{6EM}\right).$$

Unfortunately, in our setting there is no reason to expect these resistances to be large, so the variance of this estimate might be too high. We have already introduced the concept of important edges in order to alleviate this problem, and proved that we only need to look at important edges. Even if all edges of which the demand projection is estimated are important (i.e. close to C), however, v can still be far from C. This is an issue, since we don't directly estimate projections onto C, but instead estimate the projection onto $C \cup \{v\}$ and then from v onto C.

Intuitively, however, if v is far from C, it should also be far from the set of important edges, so the insertion of v should not affect their demand projection too much. As the distance upper bound between an important edge and C is relative to the scale of the resistance of that edge, this statement needs be more fine-grained in order to take the resistances of important edges into account.

More concretely, in the following lemma, which is proved in Appendix E.2, we show that if we only compute demand projection estimates for edges e such that $r_e \geq c^2 R_{eff}(C, v)$ for some appropriately chosen $c > 0$, then we can guarantee a good bound on the number of random walks we need to sample.

For the remaining edges, we will show that the energy of their contributions to the projection is negligible, so that we can reach to our desired statement in Lemma 5.4.

Lemma 5.3. *Consider a graph $G(V, E)$ with resistances $\boldsymbol{r}$, $\boldsymbol{q} \in [-1, 1]^n$, a β-congestion reduction subset C, as well as $v \in V \backslash C$. If for some $c > 0$ we are given a set of edges*

$$S' \subseteq \left\{ e \in E \backslash E(C) \mid R_{eff}(C, v) \leq \frac{1}{c^2} r_e \right\} ,$$

then for any $\delta_1' > 0$ we can compute $\widetilde{\pi}_v^{C \cup \{v\}} \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}} \right) \in \mathbb{R}$ such that with high probability

$$\left| \widetilde{\pi}_v^{C \cup \{v\}} \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}} \right) - \pi_v^{C \cup \{v\}} \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}} \right) \right| \leq \frac{\delta_1'}{\beta c \sqrt{R_{eff}(C, v)}} .$$

The algorithm requires access to $\widetilde{O} \left(\delta_1'^{-2} \log n \log \frac{1}{\beta} \right)$ independent random walks from u to C for each $u \in V \backslash C$ and $e \in E \backslash E(C)$ with $u \in e$.

This leads us to the desired statement for this section, whose proof appears in Appendix E.3.

Lemma 5.4 (Estimating $\pi_v^{C \cup \{v\}} \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}} \right)$)**.** *Consider a graph $G(V, E)$ with resistances $\boldsymbol{r}$, $\boldsymbol{q} \in [-1, 1]^n$, a β-congestion reduction subset C, as well as $v \in V \backslash C$. If we are given a set S of γ-important edges for some $\gamma \in (0, 1)$ and an estimate $\widetilde{R}_{eff}(C, v) \approx_2 R_{eff}(C, v)$, then for any $\delta_1 \in (0, 1)$ we can compute $\widetilde{\pi}_v^{C \cup \{v\}} \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}} \right) \in \mathbb{R}$ such that with high probability*

$$\left| \widetilde{\pi}_v^{C \cup \{v\}} \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}} \right) - \pi_v^{C \cup \{v\}} \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}} \right) \right| \leq \frac{\delta_1}{\sqrt{R_{eff}(C, v)}} . \tag{12}$$

The algorithm requires $\widetilde{O} \left(\delta_1^{-4} \beta^{-6} + \delta_1^{-2} \beta^{-2} \gamma^{-2} \right)$ independent random walks from u to C for each $u \in V \backslash C$ and $e \in E \backslash E(C)$ with $u \in e$.

Additionally, we have

$$\left| \pi_v^{C \cup \{v\}} \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}} \right) \right| \leq \frac{1}{\gamma \sqrt{R_{eff}(C, v)}} \cdot \widetilde{O} \left(\frac{1}{\beta^2} \right) .$$

5.2 Estimating $\boldsymbol{\pi}^C(\mathbf{1}_v)$

In contrast to the quantity $\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right)$, where there are cancellations between its two components $\pi_v^{C\cup\{v\}}\left(\sum\limits_{e=(u,w)\in E}\frac{q_e}{\sqrt{r_e}}\mathbf{1}_u\right)$ and $\pi_v^{C\cup\{v\}}\left(\sum\limits_{e=(u,w)\in E}-\frac{q_e}{\sqrt{r_e}}\mathbf{1}_w\right)$ (as $\boldsymbol{B}^\top\frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}$ sums up to $\mathbf{0}$), in $\boldsymbol{\pi}^C\left(\mathbf{1}_v\right)$ there are no cancellations. The goal is to simply estimate the hitting probabilities from v to the vertices of C, which can be done by sampling a number of random walks from v to C.

As discussed before, even though ideally we would like to have an error bound of the form $\sqrt{\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)-\boldsymbol{\pi}^C(\mathbf{1}_v))}\leq\delta_2\sqrt{R_{eff}(C,v)}$, our analysis is only able to guarantee that for any fixed potential vector $\boldsymbol{\phi}$ with $E_{\boldsymbol{r}}(\boldsymbol{\phi})\leq 1$, with high probability $\left|\langle\boldsymbol{\phi},\widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)-\boldsymbol{\pi}^C(\mathbf{1}_v)\rangle\right|\leq\delta_2\sqrt{R_{eff}(C,v)}$. However, this is still sufficient for our purposes.

In Appendix E.4 we prove the following general concentration inequality, which basically states that we can estimate the desired hitting probabilities as long as we have a bound on the ℓ_2 norm of the potentials $\boldsymbol{\phi}$ weighted by the hitting probabilities.

Lemma 5.5 (Concentration inequality 2)**.** *Let $\boldsymbol{\pi}$ be a probability distribution over $[n]$ and $\widetilde{\boldsymbol{\pi}}$ an empirical distribution of Z samples from $\boldsymbol{\pi}$. For any $\bar{\boldsymbol{\phi}}\in\mathbb{R}^n$ with $\left\|\bar{\boldsymbol{\phi}}\right\|_{\boldsymbol{\pi},2}^2\leq\mathcal{V}$, we have*

$$\Pr\left[\left|\langle\widetilde{\boldsymbol{\pi}}-\boldsymbol{\pi},\bar{\boldsymbol{\phi}}\rangle\right|>t\right]\leq\frac{1}{n^{100}}+\widetilde{O}\left(\log\left(n\cdot\mathcal{V}/t\right)\right)\exp\left(-\frac{Zt^2}{\widetilde{O}\left(\mathcal{V}\log^2 n\right)}\right).$$

We will apply it for $\bar{\boldsymbol{\phi}}=\boldsymbol{\phi}-\phi_v\cdot\mathbf{1}$, and it is important to note that $\mathcal{E}_{\boldsymbol{r}}(\bar{\boldsymbol{\phi}})=\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{\phi})$. In order to get a bound on $\left\|\bar{\boldsymbol{\phi}}\right\|_{\boldsymbol{\pi}^C(\mathbf{1}_v),2}^2$, we use the following lemma, which is proved in Appendix E.5.

Lemma 5.6 (Bounding the second moment of potentials)**.** *For any graph G, resistances $\boldsymbol{r}$, potentials $\boldsymbol{\phi}$ with $E_{\boldsymbol{r}}(\boldsymbol{\phi})\leq 1$, $C\subseteq V$ and $v\in V\backslash C$ we have $\|\boldsymbol{\phi}-\phi_v\mathbf{1}\|_{\boldsymbol{\pi}^C(\mathbf{1}_v),2}^2\leq 8\cdot R_{eff}(C,v)$.*

To give some intuition on this, consider the case when $V=C\cup\{v\}=\{1,\dots,k\}\cup\{v\}$, and there are edges $e_1,\dots,e_k$ between C and v, one for each vertex of C. Then, we have $\pi_i^C(\mathbf{1}_v)=(r_{e_i})^{-1}/\sum\limits_{i=1}^k(r_{e_i})^{-1}$, and so

$$\left\|\bar{\boldsymbol{\phi}}\right\|_{\boldsymbol{\pi}^C(\mathbf{1}_v),2}^2=\sum_{i=1}^k\frac{(\phi_i-\phi_v)^2}{r_{e_i}}\cdot\left(\sum_{i=1}^k(r_{e_i})^{-1}\right)^{-1}\leq\mathcal{E}_{\boldsymbol{r}}(\bar{\boldsymbol{\phi}})\cdot R_{eff}(C,v)\leq R_{eff}(C,v)\,.$$

We finally arrive at the desired statement about estimating $\boldsymbol{\pi}^C(\mathbf{1}_v)$.

Lemma 5.7 (Estimating $\boldsymbol{\pi}^C(\mathbf{1}_v)$)**.** *Consider a graph $G(V,E)$ with resistances $\boldsymbol{r}$, a β-congestion reduction subset C, as well as $v\in V\backslash C$. Then, for any $\delta_2>0$, we can compute $\widetilde{\boldsymbol{\pi}}^C\left(\mathbf{1}_v\right)\in\mathbb{R}^n$ such that with high probability*

$$\left|\langle\boldsymbol{\phi},\widetilde{\boldsymbol{\pi}}^C\left(\mathbf{1}_v\right)-\boldsymbol{\pi}^C\left(\mathbf{1}_v\right)\rangle\right|\leq\delta_2\cdot\sqrt{R_{eff}(C,v)}\,,\tag{13}$$

where $\boldsymbol{\phi}\in\mathbb{R}^n$ is a fixed vector with $E_{\boldsymbol{r}}(\boldsymbol{\phi})\leq 1$. The algorithm computes $\widetilde{O}\left(\frac{\log n}{\delta_2^2}\right)$ random walks from v to C.

Proof. Because both $\widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)$ and $\boldsymbol{\pi}^C(\mathbf{1}_v)$ are probability distributions, the quantity (13) doesn't change when a multiple of $\mathbf{1}$ is added to $\boldsymbol{\phi}$, and so we can replace it by $\bar{\boldsymbol{\phi}} = \boldsymbol{\phi} - \phi_v \mathbf{1}$.

Now, $\widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)$ will be defined as the empirical hitting distribution that results from sampling Z random walks from v to C. Directly applying the concentration bound in Lemma 5.5 and setting $Z = \widetilde{O}\left(\frac{\log n}{\delta_2^2}\right)$, together with the fact that $\|\bar{\boldsymbol{\phi}}\|^2_{\boldsymbol{\pi}^C(\mathbf{1}_v),2} \leq 8 \cdot R_{eff}(C, v)$ by Lemma 5.6 and $\log\log R_{eff}(C,v) \leq O(\log\log n)$, we get

$$\Pr\left[\left|\langle \widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v) - \boldsymbol{\pi}^C(\mathbf{1}_v), \bar{\boldsymbol{\phi}}\rangle\right| > \delta_2 \cdot \sqrt{R_{eff}(C,v)}\right] < \frac{1}{n^{10}}.$$

□

5.3 Proof of Lemma 4.14

We are now ready for the proof of Lemma 4.14.

Proof of Lemma 4.14. Let DP be a demand projection data structure. We analyze its operations one by one.

Algorithm 7 DemandProjector DP.Initialize

procedure DP.Initialize($C, \boldsymbol{r}, \boldsymbol{q}, S, \mathcal{P}$)
  Initialize $C, \boldsymbol{r}, \boldsymbol{q}, S, \mathcal{P}$
  $F = V \setminus C$
  $h = \widetilde{O}\left(\widehat{\varepsilon}^{-4}\beta^{-6} + \widehat{\varepsilon}^{-2}\beta^{-4}\gamma^{-2}\right)$ ▷ #random walks for each pair $u \in V$, $e \in E$ with $u \in e$
  $t = 0$ ▷ #calls to AddTerminal since last call to UpdateFull
  $\boldsymbol{\phi} = \boldsymbol{L}_{FF}^{+}\left[\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right]_F$
  $\widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) = \left[\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right]_C - \boldsymbol{L}_{CF}\boldsymbol{\phi}$

DP.Initialize($C, \boldsymbol{r}, \boldsymbol{q}, S, \mathcal{P}$): We initialize the values of $C, \boldsymbol{r}, \boldsymbol{q}, S, \mathcal{P}$. Then we exactly compute the demand projection, i.e. $\widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) = \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$, which takes time $\widetilde{O}(m)$ as shown in [GLP21]. More specifically, we have $\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) = \left(\boldsymbol{I} \quad \boldsymbol{L}_{CF}\boldsymbol{L}_{FF}^{-1}\right)\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}$ which only requires applying the operators $\boldsymbol{L}_{FF}^{-1}$ and $\boldsymbol{L}_{CF}$.

DP.AddTerminal(v, $\widetilde{R}_{eff}(C, v)$): We will serve this operation by applying Lemma 5.1. It is important to note that the error guarantee for the Output procedure increases with every call to AddTerminal, so in general we have a bounded budget for the number of calls to thus procedure before having to call again Initialize.

We apply Lemma 5.1 to obtain $\widetilde{\pi}_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}})$, and update the estimate $\widetilde{\boldsymbol{\pi}}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$. The former can be achieved with $h = \widetilde{O}\left(\widehat{\varepsilon}^{-4}\beta^{-6} + \widehat{\varepsilon}^{-2}\beta^{-2}\gamma^{-2}\right)$ random walks. Note that these random walks are already stored in $\mathcal{P}$, so accessing each of them takes time $\widetilde{O}(1)$. Using the congestion reduction property of C, we see that the running time of the procedure, which is dominated by shortcutting the random walks is $\widetilde{O}\left(h\beta^{-2}\right)$, which gives the claimed bound. The latter can be achieved with $h' = \widetilde{O}\left(\widehat{\varepsilon}^{-2}\beta^{-4}\gamma^{-2}\right)$ fresh random walks. Due to the congestion reduction property, simulating each of these requires $\widetilde{O}\left(\beta^{-2}\right)$ time.

Algorithm 8 DemandProjector DP.Update and DP.Output

procedure DP.Update($e, \boldsymbol{r}', \boldsymbol{q}'$)
    if $e \in S$ **then**
        $\widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) = \widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) + \left(\frac{q'_e}{\sqrt{r'_e}} - \frac{q_e}{\sqrt{r_e}}\right) \cdot \boldsymbol{B}^\top \boldsymbol{1}_e$
    $q_e = q'_e$, $r_e = r'_e$
procedure DP.Output()
    return $\widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$

DP.Update($e, \boldsymbol{r}', \boldsymbol{q}'$): We update the values of r_e, q_e. We also update the projection, by noting that since $e \in E(C)$,

$$\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}'}{\sqrt{\boldsymbol{r}'}}\right) = \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right) + \left(\frac{q'_e}{\sqrt{r'_e}} - \frac{q_e}{\sqrt{r_e}}\right) \boldsymbol{B}^\top \boldsymbol{1}_e\,,$$

so we change $\widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}\right)$ by the same amount, which takes time $O(1)$ and does not introduce any additional error in our estimate.

DP.Output(): We output our estimate $\widetilde{\boldsymbol{\pi}}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$. Per Lemma 5.1 we see that each of the previous T calls to AddTerminal add an error to our estimate of at most $\widehat{\varepsilon}$ in the sense that if $\boldsymbol{\Delta}^t$ were the true change in the demand projection at the t^{th} insertion, and $\widetilde{\boldsymbol{\Delta}}^t$ were the update made to our estimate, then

$$\left|\left\langle \widetilde{\boldsymbol{\Delta}}^t - \boldsymbol{\Delta}^t, \boldsymbol{\phi}\right\rangle\right| \le \widehat{\varepsilon},$$

w.h.p. for any fixed $\boldsymbol{\phi}$ such that $E_{\boldsymbol{r}^t}(\boldsymbol{\phi}) \le 1$, where $\boldsymbol{r}^t$ represents the resistances when t^{th} call to AddTerminal is made. Equivalently, for any nonzero $\boldsymbol{\phi}$,

$$\frac{1}{\sqrt{E_{\boldsymbol{r}^t}(\boldsymbol{\phi})}}\left|\left\langle \widetilde{\boldsymbol{\Delta}}^t - \boldsymbol{\Delta}^t, \boldsymbol{\phi}\right\rangle\right| \le \widehat{\varepsilon},$$

By the invariant satisfied by the resistances passed as parameters to the AddTerminal routine, we have that $\boldsymbol{r}^t \le \alpha \cdot \boldsymbol{r}^T$ for all t. Therefore $1/E_{\boldsymbol{r}^T}(\boldsymbol{\phi}) \le \alpha / E_{\boldsymbol{r}^t}(\boldsymbol{\phi})$. So we have that

$$\frac{1}{\sqrt{E_{\boldsymbol{r}^T}(\boldsymbol{\phi})}}\left|\left\langle \widetilde{\boldsymbol{\Delta}}^t - \boldsymbol{\Delta}^t, \boldsymbol{\phi}\right\rangle\right| \le \widehat{\varepsilon} \cdot \sqrt{\alpha}\,.$$

Summing up over T insertions, we obtain the desired error bound. Furthermore, note that returning the estimate takes time proportional to $|C|$, which is $\widetilde{O}\left(\beta m + T\right)$.

□

A Maintaining the Schur Complement

Following the scheme from [GLP21] we maintain a dynamic Schur complement of the graph onto a subset of terminals C. The approach follows rather directly from [GLP21] and leverages the recent work of [BBG+20] to dynamically maintain an edge sparsifier of the Schur complement of the graph onto C. Compared to [GLP21] we do not require a parameter that depends on the adaptivity of the adversary. In addition, when adding a vertex to C we also return a $(1+\varepsilon)$-approximation of the effective resistance $R_{eff}(v, C)$, which gets returned by the function call.

Lemma A.1 (DynamicSC (Theorem 4, [GLP21])). *There is a* DynamicSC *data structure supporting the following operations with the given runtimes against oblivious adversaries, for constants* $0 < \beta, \varepsilon < 1$*:*

- Initialize$(G, C^{(init)}, \boldsymbol{r}, \varepsilon, \beta)$*: Initializes a graph G with resistances $\boldsymbol{r}$ and a set of safe terminals $C^{(safe)}$. Sets the terminal set $C = C^{(safe)} \cup C^{(init)}$. Runtime:* $\widetilde{O}\left(m\beta^{-4}\varepsilon^{-4}\right)$.

- AddTerminal$(v \in V(G))$*: Returns $\widetilde{R}_{eff}(C, v) \approx_2 R_{eff}(C, v)$ and adds v as a terminal. Runtime: Amortized* $\widetilde{O}\left(\beta^{-2}\varepsilon^{-2}\right)$.

- TemporaryAddTerminals$(\Delta C \subseteq V(G))$*: Adds all vertices in the set ΔC as (temporary) terminals. Runtime: Worst case $\widetilde{O}\left(K^2\beta^{-4}\varepsilon^{-4}\right)$, where K is the total number of terminals added by all of the* TemporaryAddTerminals *operations that have not been rolled back using* Rollback. *All* TemporaryAddTerminals *operations should be rolled back before the next call to* AddTerminals.

- Update(e, r)*: Under the guarantee that both endpoints of e are terminals, updates $r_e = r$. Runtime: Worst case* $\widetilde{O}(1)$.

- $\widetilde{SC}()$*: Returns a spectral sparsifier $\widetilde{SC} \approx_{1+\varepsilon} SC(G, C)$ (with respect to resistances $\boldsymbol{r}$) with $\widetilde{O}\left(|C|\varepsilon^{-2}\right)$ edges. Runtime: Worst case $\widetilde{O}\left(\left(\beta m + (K\beta^{-2}\varepsilon^{-2})^2\right)\varepsilon^{-2}\right)$ where K is the total number of terminals added by all of the* TemporaryAddTerminals *operations that have not been rolled back.*

- Rollback()*: Rolls back the last* Update, AddTerminals, *or* TemporaryAddTerminals *if it exists. The runtime is the same as the original operation.*

Finally, all calls return valid outputs with high probability. The size of C should always be $O(\beta m)$.

This data structure is analyzed in detail in [GLP21]. Additionally, let us show that an approximation to $R_{eff}(v, C)$ can be efficiently computed along with the AddTerminal operation. To get an estimate we simply inspect the neighbors of v in the sparsified Schur complement of $C \cup \{v\}$ and compute the inverse of the sum of their inverses. This is indeed a $1 + O(\varepsilon)$-approximation, as effective resistances are preserved within a $1 + O(\varepsilon)$ factor in the sparsifier.

To show that this operation takes little amortized time, we note that by the proof appearing in [GLP21, Lemma 6.2], vertex v appears in amortized $\widetilde{O}(1)$ expanders maintained dynamically. As the dynamic sparsifier keeps $\widetilde{O}\left(\varepsilon^{-2}\right)$ neighbors of v from each expander, the number of neighbors to inspect with each call is $\widetilde{O}\left(\varepsilon^{-2}\right)$, which also bounds the time necessary to approximate the resistance.

B Auxiliary Lemmas

Lemma 2.10. *Let $\boldsymbol{d}$ be a demand vector, let $\boldsymbol{r}$ be resistances, and let $C \subseteq V$ be a subset of vertices. Then*

$$\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^C(\boldsymbol{d})\right) \leq \mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{d}\right) .$$

Proof. Letting $F = V \setminus C$, and $\boldsymbol{L}$ be the Laplacian of the underlying graph, we can write

$$\boldsymbol{\pi}^C(\boldsymbol{d}) = \boldsymbol{d}_C - \boldsymbol{L}_{CF}\boldsymbol{L}_{FF}^{-1}\boldsymbol{d}_F .$$

By factoring $\boldsymbol{L}^+$ as

$$\boldsymbol{L}^+ = \begin{bmatrix} I & 0 \\ -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} & I \end{bmatrix} \begin{bmatrix} SC(\boldsymbol{L},C)^+ & 0 \\ 0 & \boldsymbol{L}_{FF}^{-1} \end{bmatrix} \begin{bmatrix} I & -\boldsymbol{L}_{CF}\boldsymbol{L}_{FF}^{-1} \\ 0 & I \end{bmatrix}$$

we can write

$$\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{d}) = \boldsymbol{d}^\top \boldsymbol{L}^+ \boldsymbol{d} = \begin{bmatrix} \boldsymbol{\pi}^C(\boldsymbol{d}) \\ \boldsymbol{d}_F \end{bmatrix}^\top \begin{bmatrix} SC(\boldsymbol{L},C)^+ & 0 \\ 0 & \boldsymbol{L}_{FF}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\pi}^C(\boldsymbol{d}) \\ \boldsymbol{d}_F \end{bmatrix} = \|\boldsymbol{\pi}^C(\boldsymbol{d})\|_{SC(\boldsymbol{L},C)^+}^2 + \|\boldsymbol{d}_F\|_{\boldsymbol{L}_{FF}^{-1}}^2 .$$

Furthermore, we can use the same factorization to write

$$\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{\pi}^C(\boldsymbol{d})) = \begin{bmatrix} \boldsymbol{\pi}^C(\boldsymbol{d}) \\ 0 \end{bmatrix}^\top \begin{bmatrix} SC(\boldsymbol{L},C)^+ & 0 \\ 0 & \boldsymbol{L}_{FF}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\pi}^C(\boldsymbol{d}) \\ 0 \end{bmatrix} = \|\boldsymbol{\pi}^C(\boldsymbol{d})\|_{SC(\boldsymbol{L},C)^+}^2 ,$$

which proves the claim. □

Lemma B.1. *For any $\mu \in (1/\text{poly}(m), \text{poly}(m))$, we have $\|\boldsymbol{r}(\mu)\|_\infty \leq m^{\widetilde{O}(\log m)}$.*

Proof. By Appendix A in [AMV20], for some $\mu_0 = \Theta(\|\boldsymbol{c}\|_2)$, the solution $\boldsymbol{f} = \boldsymbol{u}/2$ has

$$\left\|\boldsymbol{C}^\top \left(\frac{\boldsymbol{c}}{\mu_0} + \frac{\boldsymbol{1}}{\boldsymbol{s}^+} - \frac{\boldsymbol{1}}{\boldsymbol{s}^-}\right)\right\|_{(\boldsymbol{C}^\top \boldsymbol{R}\boldsymbol{C})^+} \leq 1/10 .$$

This implies that $\min_e \{s_e(\mu_0)^+, s_e(\mu_0)^-\} \geq \min_e u_e/4 \geq 1/4$, and so $\|\boldsymbol{r}(\mu_0)\|_\infty \leq O(1)$. Additionally, $\|\boldsymbol{c}\|_\infty \in [1, \text{poly}(m)]$, so $\mu_0 = \Theta\left(\text{poly}(m)\right)$.

Now, for any integer $i \geq 0$ we let $\mu_{i+1} = \mu_i \cdot (1 - 1/\sqrt{m})^{\sqrt{m}/10}$. By Lemma C.4 we have that $\boldsymbol{r}\left(\mu_{i+1}\right) \approx_{m^2} \boldsymbol{r}\left(\mu_i\right)$, and so

$$\boldsymbol{r}\left(\frac{1}{\text{poly}(m)}\right) = \boldsymbol{r}(\mu_{\widetilde{O}(\log m)}) \leq \left(\frac{9}{100}m^2\right)^{\widetilde{O}(\log m)} \boldsymbol{r}(\mu_0) \leq m^{\widetilde{O}(\log m)}\boldsymbol{r}(\mu_0) \leq m^{\widetilde{O}(\log m)} .$$

□

Lemma B.2. *Given a graph $G(V,E)$ with resistances $\boldsymbol{r}$ and any parameter $\varepsilon > 0$, there exists an algorithm that runs in time $\widetilde{O}\left(m/\varepsilon^2\right)$ and produces a matrix $\boldsymbol{Q} \in \mathbb{R}^{\widetilde{O}\left(1/\varepsilon^2\right) \times n}$ such that with high probability for any $u, v \in V$,*

$$R_{eff}(u,v) \approx_{1+\varepsilon} \|\boldsymbol{Q}\boldsymbol{1}_u - \boldsymbol{Q}\boldsymbol{1}_v\|_2^2$$

C Deferred Proofs from Section 3

C.1 Central path stability bounds

Lemma C.1 (Central path energy stability). *Consider a minimum cost flow instance on a graph $G(V,E)$. For any $\mu > 0$ and $\mu' = \mu/(1+1/\sqrt{m})^k$ for some $k \in (0, \sqrt{m}/10)$, we have*

$$\sum_{e\in E}\left(\frac{1}{s_e(\mu)^+ \cdot s_e(\mu')^+} + \frac{1}{s_e(\mu)^- \cdot s_e(\mu')^-}\right)\left(f_e(\mu') - f_e(\mu)\right)^2 \leq 2k^2$$

Proof of Lemma C.1. We let $\delta = 1/\sqrt{m}$, $\boldsymbol{f} = \boldsymbol{f}(\mu)$, $\boldsymbol{s} = \boldsymbol{s}(\mu)$, $\boldsymbol{r} = \boldsymbol{r}(\mu)$, $\boldsymbol{f}' = \boldsymbol{f}(\mu')$, $\boldsymbol{s}' = \boldsymbol{s}(\mu')$, and $\boldsymbol{r}' = \boldsymbol{r}(\mu')$. We also set $\widetilde{\boldsymbol{f}} = \boldsymbol{f}' - \boldsymbol{f}$. By definition of centrality we have

$$\begin{aligned}\boldsymbol{C}^\top\left(\frac{1}{\boldsymbol{s}^-} - \frac{1}{\boldsymbol{s}^+}\right) &= \boldsymbol{C}^\top\frac{\boldsymbol{c}}{\mu}\\ \boldsymbol{C}^\top\left(\frac{1}{\boldsymbol{s}^- + \widetilde{\boldsymbol{f}}} - \frac{1}{\boldsymbol{s}^+ - \widetilde{\boldsymbol{f}}}\right) &= \boldsymbol{C}^\top\frac{\boldsymbol{c}}{\mu'}\,,\end{aligned}$$

which, after subtracting, give

$$\begin{aligned}&\boldsymbol{C}^\top\left(\frac{1}{\boldsymbol{s}^- + \widetilde{\boldsymbol{f}}} - \frac{1}{\boldsymbol{s}^-} - \frac{1}{\boldsymbol{s}^+ - \widetilde{\boldsymbol{f}}} + \frac{1}{\boldsymbol{s}^+}\right) = \boldsymbol{C}^\top\left(\frac{\boldsymbol{c}}{\mu'} - \frac{\boldsymbol{c}}{\mu}\right)\\ \Leftrightarrow\ &\boldsymbol{C}^\top\left(\left(\frac{1}{\boldsymbol{s}^-(\boldsymbol{s}^- + \widetilde{\boldsymbol{f}})} + \frac{1}{\boldsymbol{s}^+(\boldsymbol{s}^+ - \widetilde{\boldsymbol{f}})}\right)\widetilde{\boldsymbol{f}}\right) = -\left((1+\delta)^k - 1\right)\boldsymbol{C}^\top\frac{\boldsymbol{c}}{\mu}\,.\end{aligned}$$

As $\widetilde{\boldsymbol{f}} = \boldsymbol{C}\boldsymbol{x}$ for some $\boldsymbol{x}$, after taking the inner product of both sides with $\boldsymbol{x}$ we get

$$\left\langle \widetilde{\boldsymbol{f}}, \left(\frac{1}{\boldsymbol{s}^-(\boldsymbol{s}^- + \widetilde{\boldsymbol{f}})} + \frac{1}{\boldsymbol{s}^+(\boldsymbol{s}^+ - \widetilde{\boldsymbol{f}})}\right)\widetilde{\boldsymbol{f}}\right\rangle = -\left((1+\delta)^k - 1\right)\left\langle\frac{\boldsymbol{c}}{\mu}, \widetilde{\boldsymbol{f}}\right\rangle\,. \tag{14}$$

We will now prove that $-\left\langle \frac{\boldsymbol{c}}{\mu}, \widetilde{\boldsymbol{f}}\right\rangle \leq k\sqrt{m}$. First of all, by differentiating the centrality condition

$$\boldsymbol{C}^\top\left(\frac{\boldsymbol{c}}{\nu} + \frac{\mathbf{1}}{\boldsymbol{s}(\nu)^+} - \frac{\mathbf{1}}{\boldsymbol{s}(\nu)^-}\right) = \mathbf{0}$$

with respect to ν we get

$$\boldsymbol{C}^\top\left(-\frac{\boldsymbol{c}}{\nu^2} + \left(\frac{\mathbf{1}}{(\boldsymbol{s}(\nu)^+)^2} + \frac{\mathbf{1}}{(\boldsymbol{s}(\nu)^-)^2}\right)\frac{d\boldsymbol{f}(\nu)}{d\nu}\right) = \mathbf{0}\,,$$

or equivalently

$$\boldsymbol{C}^\top\left(\boldsymbol{r}(\nu)\frac{d\boldsymbol{f}(\nu)}{d\nu}\right) = -\frac{1}{\nu}\boldsymbol{C}^\top\left(\frac{\mathbf{1}}{\boldsymbol{s}(\nu)^+} - \frac{\mathbf{1}}{\boldsymbol{s}(\nu)^-}\right)\,.$$

If we set $g(\boldsymbol{s}) = \frac{\frac{\mathbf{1}}{\boldsymbol{s}^+} - \frac{\mathbf{1}}{\boldsymbol{s}^-}}{\boldsymbol{r}}$, this can also be equivalently written as

$$\frac{d\boldsymbol{f}(\nu)}{d\nu} = -\frac{1}{\nu}\left(g(\boldsymbol{s}(\nu)) - (\boldsymbol{R}(\nu))^{-1}\boldsymbol{B}(\boldsymbol{B}^\top(\boldsymbol{R}(\nu)^{-1})\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s}(\nu))\right)\,.$$

We have

$$
\begin{aligned}
-\left\langle \frac{\boldsymbol{c}}{\mu}, \widetilde{\boldsymbol{f}} \right\rangle &= -\int_{\nu=\mu}^{\mu'} \left\langle \frac{\boldsymbol{c}}{\mu}, d\boldsymbol{f}(\nu) \right\rangle \\
&= \frac{1}{\mu}\int_{\nu=\mu}^{\mu'} \left\langle \frac{\nu}{\boldsymbol{s}(\nu)^-} - \frac{\nu}{\boldsymbol{s}(\nu)^+}, \frac{1}{\nu}\left(g(\boldsymbol{s}(\nu)) - (\boldsymbol{R}(\nu))^{-1}\boldsymbol{B}(\boldsymbol{B}^\top(\boldsymbol{R}(\nu))^{-1}\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s}(\nu))\right)\right\rangle d\nu \\
&= -\frac{1}{\mu}\int_{\nu=\mu}^{\mu'} \left\langle \sqrt{\boldsymbol{r}(\nu)}g(\boldsymbol{s}(\nu)), \boldsymbol{\Pi}_{\ker(\boldsymbol{B}^\top(\boldsymbol{R}(\nu))^{-1/2})}\sqrt{\boldsymbol{r}(\nu)}g(\boldsymbol{s}(\nu))\right\rangle d\nu \\
&= \frac{1}{\mu}\int_{\nu=\mu'}^{\mu} \left\|\boldsymbol{\Pi}_{\ker(\boldsymbol{B}^\top(\boldsymbol{R}(\nu))^{-1/2})}\sqrt{\boldsymbol{r}(\nu)}g(\boldsymbol{s}(\nu))\right\|_2^2 d\nu \\
&\le \frac{1}{\mu}\int_{\nu=\mu'}^{\mu} \left\|\sqrt{\boldsymbol{r}(\nu)}g(\boldsymbol{s}(\nu))\right\|_2^2 d\nu \\
&\le \frac{1}{\mu}\int_{\nu=\mu'}^{\mu} m d\nu \\
&= m\frac{\mu-\mu'}{\mu} \\
&= m(1-(1+\delta)^{-k}) \\
&\le \delta k m \\
&= k\sqrt{m}\,,
\end{aligned}
$$

where $\boldsymbol{\Pi}_{\ker(\boldsymbol{B}^\top(\boldsymbol{R}(\nu))^{-1/2})} = \boldsymbol{I} - (\boldsymbol{R}(\nu))^{-1/2}\boldsymbol{B}(\boldsymbol{B}^\top(\boldsymbol{R}(\nu))^{-1}\boldsymbol{B})^+\boldsymbol{B}^\top(\boldsymbol{R}(\nu))^{-1/2}$ is the orthogonal projection onto the kernel of $\boldsymbol{B}^\top(\boldsymbol{R}(\nu))^{-1/2}$.

Plugging this into (14) and using the fact that $(1+\delta)^k \le 1 + 1.1\delta k = 1 + 1.1k/\sqrt{m}$, we get

$$
\sum_{e\in E}\left(\frac{1}{s_e(\mu)^+\cdot s_e(\mu')^+} + \frac{1}{s_e(\mu)^-\cdot s_e(\mu')^-}\right)\left(f_e(\mu') - f_e(\mu)\right)^2 \le 2k^2\,.
$$

□

We give an auxiliary lemma which converts between different kinds of slack approximations.

Lemma C.2. *We consider flows $\boldsymbol{f}, \boldsymbol{f}'$ with slacks $\boldsymbol{s}, \boldsymbol{s}'$ and resistances $\boldsymbol{r}, \boldsymbol{r}'$. Then,*

$$
\max\left\{\left|\frac{s_e'^+ - s_e^+}{s_e^+}\right|, \left|\frac{s_e'^- - s_e^-}{s_e^-}\right|\right\} \le \sqrt{r_e}\left|f_e' - f_e\right| \le \sqrt{2}\max\left\{\left|\frac{s_e'^+ - s_e^+}{s_e^+}\right|, \left|\frac{s_e'^- - s_e^-}{s_e^-}\right|\right\}
$$

and if $r_e \not\approx_{1+\gamma} r_e'$ for some $\gamma \in (0,1)$, then $\sqrt{r_e}\,|f_e' - f_e| \ge \gamma/6$.

Proof. For the first one, note that

$$
r_e = \frac{1}{(s_e^+)^2} + \frac{1}{(s_e^-)^2} \in \left[\max\left\{\frac{1}{(s_e^+)^2}, \frac{1}{(s_e^-)^2}\right\}, 2\max\left\{\frac{1}{(s_e^+)^2}, \frac{1}{(s_e^-)^2}\right\}\right]\,.
$$

Together with the fact that $|f_e' - f_e| = |s_e'^+ - s_e^+| = |s_e'^- - s_e^-|$, it implies the first statement.

For the second one, without loss of generality let $s_e^+ \le s_e^-$, so by the previous statement we have $\sqrt{r_e}\,|f_e' - f_e| \ge \frac{|s_e'^+ - s_e^+|}{s_e^+}$. If this is $< \gamma/6$ then $(1-\gamma/6)s_e^+ \le s_e'^+ \le (1+\gamma/6)s_e^+$, so $s_e'^+ \approx_{1+\gamma/3} s_e^+$.

However, we also have that $\frac{|s'^-_e - s^-_e|}{s^-_e} \le \frac{|s'^+_e - s^+_e|}{s^+_e} \le \gamma/6$, so $s'^-_e \approx_{1+\gamma/3} s^-_e$. Therefore, $r'_e = \frac{1}{(s'^+_e)^2} + \frac{1}{(s'^-_e)^2} \approx_{1+\gamma} \frac{1}{(s^+_e)^2} + \frac{1}{(s^-_e)^2} = r_e$, a contradiction. $\square$

The following lemma is a fine-grained explanation of how resistances can change.

Lemma C.3. *Consider a minimum cost flow instance on a graph $G(V,E)$ and parameters $\mu > 0$ and $\mu' \ge \mu/(1+1/\sqrt{m})^k$, where $k \in (0, \sqrt{m}/10)$. For any $e \in E$ and $\gamma \in (0,1)$ we let* change(e,γ) *be the largest integer $t(e) \ge 0$ such that there are real numbers $\mu = \mu_1(e) > \mu_2(e) > \cdots > \mu_{t(e)+1}(e) = \mu'$ with $\sqrt{r_e(\mu_i)}\,|f_e(\mu_{i+1}) - f_e(\mu_i)| \ge \gamma$ for all $i \in [t(e)]$.*

Then, $\sum_{e\in E} (\mathrm{change}(e,\gamma))^2 \le O(k^2/\gamma^2)$.

Proof of Lemma C.3. First, we assume that without loss of generality, $r_e(\mu_{i+1}(e)) \approx_{(1+6\gamma)^2} r_e(\mu_i(e))$ for all $e \in E$ and $i \in [t(e)]$. If this is not true, then by continuity there exists a $\nu \in (\mu_{i+1}, \mu_i)$ such that $r_e(\mu_{i+1}(e)) \not\approx_{1+6\gamma} r_e(\nu)$ and $r_e(\nu) \not\approx_{1+6\gamma} r_e(\mu_i(e))$. By Lemma C.2, this implies that $\sqrt{r_e(\mu_{i+1}(e))}\,|f_e(\nu) - f_e(\mu_{i+1}(e))| \ge \gamma$ and $\sqrt{r_e(\nu)}\,|f_e(\mu_i(e)) - f_e(\nu)| \ge \gamma$. Therefore we can break the interval (μ_{i+1}, μ_i) into (μ_{i+1}, ν) and (ν, μ_i) and make the statement stronger.

Similarly, we also assume that $r_e(\nu) \approx_{(1+6\gamma)^3} r_e(\mu_i(e))$ for all $e \in E$, $i \in [t(e)]$, and $\nu \in (\mu_{i+1}, \mu_i)$. If this is not the case, then by using the fact that $r_e(\mu_{i+1}(e)) \approx_{(1+6\gamma)^2} r_e(\mu_i(e))$, we also get that $r_e(\mu_{i+1}) \not\approx_{1+6\gamma} r_e(\nu)$, and so we can again break the interval as before and obtain a stronger statement.

Now, we look at the following integral:

$$\mathcal{E} := \int_{\nu=\mu}^{\mu'} \sum_{e\in E} r_e(\nu) \left(\frac{df_e(\nu)}{d\nu}\right)^2 |d\nu| \,,$$

where $df_e(\nu)$ is the differential of the flow $f_e(\nu)$ with respect to the centrality parameter. Similarly to Lemma C.1, we use the following equation that describes how the flow changes:

$$\frac{d\boldsymbol{f}(\nu)}{d\nu} = -\frac{1}{\nu}\left(g(\boldsymbol{s}(\nu)) - (\boldsymbol{R}(\nu))^{-1}\boldsymbol{B}(\boldsymbol{B}^\top(\boldsymbol{R}(\nu)^{-1})\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s}(\nu))\right).$$

This implies that

$$\begin{aligned}
\left\|\sqrt{\boldsymbol{r}(\nu)}\frac{d\boldsymbol{f}(\nu)}{d\nu}\right\|_2^2 &= \frac{1}{\nu^2}\left\|\sqrt{\boldsymbol{r}(\nu)}g(\boldsymbol{s}(\nu)) - (\boldsymbol{R}(\nu))^{-1/2}\boldsymbol{B}(\boldsymbol{B}^\top(\boldsymbol{R}(\nu)^{-1})\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s}(\nu))\right\|_2^2 \\
&\le \frac{1}{\nu^2}\left\|\left(I - (\boldsymbol{R}(\nu))^{-1/2}\boldsymbol{B}(\boldsymbol{B}^\top(\boldsymbol{R}(\nu)^{-1})\boldsymbol{B})^+\boldsymbol{B}^\top(\boldsymbol{R}(\nu))^{-1/2}\right)\sqrt{\boldsymbol{r}}g(\boldsymbol{s}(\nu))\right\|_2^2 \\
&\le \frac{1}{\nu^2}\left\|\sqrt{\boldsymbol{r}}g(\boldsymbol{s}(\nu))\right\|_2^2 \\
&\le \frac{m}{\nu^2}\,,
\end{aligned}$$

and so

$$\mathcal{E} \le \int_{\nu=\mu}^{\mu'} \frac{m}{\nu^2}\,|d\nu| = m\left(\frac{1}{\mu'} - \frac{1}{\mu}\right) \le m\frac{1.1\delta k}{\mu} = 1.1k\sqrt{m}/\mu\,. \tag{15}$$

On the other hand, for any $e \in E$ and $i \in [t(e)]$ we have

$$\begin{aligned}
\int_{\nu=\mu_i(e)}^{\mu_{i+1}(e)} r_e(\nu)\left(\frac{df_e(\nu)}{d\nu}\right)^2 |d\nu| &\geq \frac{r_e(\mu_i(e))}{(1+6\gamma)^3}\int_{\nu=\mu_i(e)}^{\mu_{i+1}(e)}\left(\frac{df_e(\nu)}{d\nu}\right)^2|d\nu| \\
&\geq \frac{r_e(\mu_i(e))}{(1+6\gamma)^3}\frac{\left(\int_{\nu=\mu_i(e)}^{\mu_{i+1}(e)}\left|\frac{df_e(\nu)}{d\nu}\right||d\nu|\right)^2}{\int_{\nu=\mu_i(e)}^{\mu_{i+1}(e)}|d\nu|} \\
&= \frac{r_e(\mu_i(e))}{(1+6\gamma)^3(\mu_i(e)-\mu_{i+1}(e))}\left(f(\mu_i(e))-f(\mu_{i+1}(e))\right)^2 \\
&\geq \frac{\gamma^2}{36(1+6\gamma)^3(\mu_i(e)-\mu_{i+1}(e))},
\end{aligned}$$

where we used the Cauchy-Schwarz inequality. Now, note that

$$\begin{aligned}
\int_{\nu=\mu_1(e)}^{\mu_{t(e)+1}(e)} r_e(\nu)\left(\frac{df_e(\nu)}{d\nu}\right)^2|d\nu| &\geq \sum_{i=1}^{t(e)}\frac{\gamma^2}{(1+6\gamma)^3(\mu_i(e)-\mu_{i+1}(e))} \\
&\geq \frac{\gamma^2(t(e))^2}{(1+6\gamma)^3(\mu-\mu')} \\
&\geq \frac{\gamma^2(t(e))^2\sqrt{m}}{(1+6\gamma)^3 k\mu},
\end{aligned}$$

where remember that $t(e) = \text{change}(e,\gamma)$ and we again used Cauchy-Schwarz. Summing this up for all $e \in E$ and combining with (15), we get that $\sum\limits_{e\in E}(\text{change}(e,\gamma))^2 \leq O(k^2/\gamma^2)$. □

Lemma C.4 (Central path ℓ_∞ slack stability)**.** *Consider a minimum cost flow instance on a graph $G(V,E)$. For any $\mu > 0$ and $\mu' = \mu/(1+1/\sqrt{m})^k$ for some $k \in (0, \sqrt{m}/10)$, we have*

$$\boldsymbol{s}(\mu') \approx_{3k^2} \boldsymbol{s}(\mu).$$

Proof of Lemma C.4. By Lemma C.1, for any $e \in E$ we have that

$$\left(\frac{1}{s_e(\mu)^+\cdot s_e(\mu')^+}+\frac{1}{s_e(\mu)^-\cdot s_e(\mu')^-}\right)\left(f_e(\mu')-f_e(\mu)\right)^2 \leq 2k^2. \tag{16}$$

If $s_e(\mu')^+ = (1+c)\cdot s_e(\mu)^+$ for some $c \geq 0$, then

$$(f_e(\mu')-f_e(\mu))^2 = c^2(s_e(\mu)^+)^2$$

and

$$s_e(\mu)^+\cdot s_e(\mu')^+ = (1+c)(s_e(\mu)^+)^2,$$

so by (16) we have that $c \leq 3k^2$.

Similarly, if $s_e(\mu')^+ = (1+c)^{-1}\cdot s_e(\mu)^+$ for some $c \geq 0$, then

$$(f_e(\mu')-f_e(\mu))^2 = c^2(s_e(\mu')^+)^2$$

and

$$s_e(\mu)^+ \cdot s_e(\mu')^+ = (1+c)(s_e(\mu')^+)^2\,,$$

so by (16) we have that $c \leq 3k^2$.

We have proved that $s_e(\mu')^+ \approx_{3k^2} s_e(\mu)^+$ and by symmetry we also have $s_e(\mu')^- \approx_{3k^2} s_e(\mu)^-$. □

C.2 Proof of Lemma 3.5

Our goal is to keep track of how close $\boldsymbol{f}^*$ remains to centrality (in ℓ_2 norm) and how close $\boldsymbol{f}$ remains to $\boldsymbol{f}^*$ in ℓ_∞ norm. From these two we can conclude that at all times $\boldsymbol{f}$ is close in ℓ_∞ to the central flow. We first prove the following lemma, which bounds how the distance of $\boldsymbol{f}^*$ to centrality (measured in energy of the residual) degrades when taking a progress step.

Lemma C.5. *Let $\boldsymbol{f}^*$ be a flow with slacks $\boldsymbol{s}^*$ and resistances $\boldsymbol{r}^*$, and $\boldsymbol{f}$ be a flow with slacks $\boldsymbol{s}$ and resistances $\boldsymbol{r}$, where $\boldsymbol{s} \approx_{1+\varepsilon_{\mathrm{solve}}} \boldsymbol{s}^*$ for some $\varepsilon_{\mathrm{solve}} \in (0, 0.1)$. We define $\boldsymbol{f}'^* = \boldsymbol{f}^* + \varepsilon_{\mathrm{step}}\widetilde{\boldsymbol{f}}^*$ for some $\varepsilon_{\mathrm{step}} \in (0, 0.1)$ (and the new slacks $\boldsymbol{s}'^*$), where*

$$\widetilde{\boldsymbol{f}}^* = \delta g(\boldsymbol{s}) - \delta \boldsymbol{R}^{-1}\boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s})\,, \tag{17}$$

$\delta = \frac{1}{\sqrt{m}}$, and $g(\boldsymbol{s}) := \frac{\frac{1}{s^+} - \frac{1}{s^-}}{\boldsymbol{r}}$. If we let $\boldsymbol{h} = \frac{\boldsymbol{c}}{\mu} + \frac{1}{\boldsymbol{s}^{*+}} - \frac{1}{\boldsymbol{s}^{*-}}$ and $\boldsymbol{h}' = \frac{\boldsymbol{c}(1+\varepsilon_{\mathrm{step}}\delta)}{\mu} + \frac{1}{\boldsymbol{s}'^{*+}} - \frac{1}{\boldsymbol{s}'^{*-}}$ be the residuals of $\boldsymbol{f}^*$ and $\boldsymbol{f}'^*$ for some $\mu > 0$, then

$$\left\|\boldsymbol{C}^\top \boldsymbol{h}'\right\|_{\overline{\boldsymbol{H}}^+} \leq (1+\varepsilon_{\mathrm{step}}\delta)\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_{\overline{\boldsymbol{H}}^+} + 5\left\|\frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}}\right\|_\infty^{1/2} \varepsilon_{\mathrm{solve}}\cdot\varepsilon_{\mathrm{step}} + 2\left\|\frac{\boldsymbol{r}'^*}{\bar{\boldsymbol{r}}}\right\|_\infty^{1/2}\varepsilon_{\mathrm{step}}^2\,,$$

where $\bar{\boldsymbol{r}}$ are some arbitrary resistances and $\overline{\boldsymbol{H}} = \boldsymbol{C}^\top \overline{\boldsymbol{R}}\boldsymbol{C}$.

Proof. Let $\boldsymbol{\rho}^+ = \varepsilon_{\mathrm{step}}\widetilde{\boldsymbol{f}}^*/\boldsymbol{s}^{*+}$ and $\boldsymbol{\rho}^- = -\varepsilon_{\mathrm{step}}\widetilde{\boldsymbol{f}}^*/\boldsymbol{s}^{*-}$. First of all, it is easy to see that

$$\begin{aligned}
\|\boldsymbol{\rho}\|_2 &\leq \left\|\frac{\boldsymbol{s}}{\boldsymbol{s}^*}\right\|_\infty \left\|\frac{\boldsymbol{s}^*}{\boldsymbol{s}}\boldsymbol{\rho}\right\|_2 \\
&\leq \varepsilon_{\mathrm{step}}(1+\varepsilon_{\mathrm{solve}})\left\|\widetilde{\boldsymbol{f}}^*\right\|_{\boldsymbol{r},2} \\
&= \varepsilon_{\mathrm{step}}\delta(1+\varepsilon_{\mathrm{solve}})\left\|\sqrt{\boldsymbol{r}}g(\boldsymbol{s}) - \boldsymbol{R}^{-1/2}\boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{R}^{-1}\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s})\right\|_2 \\
&= \varepsilon_{\mathrm{step}}\delta(1+\varepsilon_{\mathrm{solve}})\left\|\left(\boldsymbol{I} - \boldsymbol{R}^{-1/2}\boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{R}^{-1}\boldsymbol{B})^+\boldsymbol{B}^\top\boldsymbol{R}^{-1/2}\right)\sqrt{\boldsymbol{r}}g(\boldsymbol{s})\right\|_2 \\
&\leq \varepsilon_{\mathrm{step}}\delta(1+\varepsilon_{\mathrm{solve}})\left\|\sqrt{\boldsymbol{r}}g(\boldsymbol{s})\right\|_2 \\
&= \varepsilon_{\mathrm{step}}\delta(1+\varepsilon_{\mathrm{solve}})\left\|\frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-}}{\sqrt{\frac{1}{(\boldsymbol{s}^+)^2} + \frac{1}{(\boldsymbol{s}^-)^2}}}\right\|_2 \\
&\leq \varepsilon_{\mathrm{step}}\delta(1+\varepsilon_{\mathrm{solve}})\sqrt{m} \\
&= \varepsilon_{\mathrm{step}}(1+\varepsilon_{\mathrm{solve}})\,.
\end{aligned}$$

We bound the energy to route the residual of $\boldsymbol{f}'^*$ as

$$\begin{aligned}
&\left\|\boldsymbol{C}^\top \boldsymbol{h}'\right\|_{\overline{\boldsymbol{H}}^+}\\
&= \left\|\boldsymbol{C}^\top \boldsymbol{h} + \boldsymbol{C}^\top\left(\frac{\varepsilon_{\text{step}}\delta \boldsymbol{c}}{\mu} + \frac{1}{\boldsymbol{s}'^{*+}} - \frac{1}{\boldsymbol{s}^{*+}} - \frac{1}{\boldsymbol{s}'^{*-}} + \frac{1}{\boldsymbol{s}^{*-}}\right)\right\|_{\overline{\boldsymbol{H}}^+}\\
&= \left\|\boldsymbol{C}^\top \boldsymbol{h} + \boldsymbol{C}^\top\left(\frac{\varepsilon_{\text{step}}\delta \boldsymbol{c}}{\mu} + \frac{\boldsymbol{\rho}^+}{\boldsymbol{s}'^{*+}} - \frac{\boldsymbol{\rho}^-}{\boldsymbol{s}'^{*-}}\right)\right\|_{\overline{\boldsymbol{H}}^+}\\
&= \left\|\boldsymbol{C}^\top \boldsymbol{h} + \boldsymbol{C}^\top\left(\frac{\varepsilon_{\text{step}}\delta \boldsymbol{c}}{\mu} + \frac{\boldsymbol{\rho}^+}{\boldsymbol{s}^{*+}} - \frac{\boldsymbol{\rho}^-}{\boldsymbol{s}^{*-}}\right) + \boldsymbol{C}^\top\left(\frac{(\boldsymbol{\rho}^+)^2}{\boldsymbol{s}'^{*+}} - \frac{(\boldsymbol{\rho}^-)^2}{\boldsymbol{s}'^{*-}}\right)\right\|_{\overline{\boldsymbol{H}}^+}.
\end{aligned}$$

Now, using (17) we get that $\boldsymbol{r}\widetilde{\boldsymbol{f}}^* = \delta \boldsymbol{r} g(\boldsymbol{s}) - \delta \boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B})^+\boldsymbol{B}^\top g(\boldsymbol{s})$ and so $\boldsymbol{C}^\top\left(\boldsymbol{r}\widetilde{\boldsymbol{f}}^*\right) = \delta \boldsymbol{C}^\top(\boldsymbol{r}g(\boldsymbol{s}))$, which follows by the fact that for any i, $\boldsymbol{1}_i^\top \boldsymbol{C}^\top \boldsymbol{B} = \left(\boldsymbol{B}^\top \boldsymbol{C}\boldsymbol{1}_i\right)^\top = \boldsymbol{0}$, since $\boldsymbol{C}\boldsymbol{1}_i$ is a circulation by definition of $\boldsymbol{C}$. As $\boldsymbol{r}\widetilde{\boldsymbol{f}}^* = \left(\frac{1}{(\boldsymbol{s}^+)^2} + \frac{1}{(\boldsymbol{s}^-)^2}\right)\widetilde{\boldsymbol{f}}^* = \varepsilon_{\text{step}}^{-1}\frac{\boldsymbol{s}^{*+}}{(\boldsymbol{s}^+)^2}\boldsymbol{\rho}^+ - \varepsilon_{\text{step}}^{-1}\frac{\boldsymbol{s}^{*-}}{(\boldsymbol{s}^-)^2}\boldsymbol{\rho}^-$, we have $\varepsilon_{\text{step}}\delta \boldsymbol{C}^\top(\boldsymbol{r}g(\boldsymbol{s})) = \boldsymbol{C}^\top\left(\frac{\boldsymbol{s}^{*+}}{(\boldsymbol{s}^+)^2}\boldsymbol{\rho}^+ - \frac{\boldsymbol{s}^{*-}}{(\boldsymbol{s}^-)^2}\boldsymbol{\rho}^-\right)$ and so

$$\begin{aligned}
&\left\|\boldsymbol{C}^\top \boldsymbol{h} + \boldsymbol{C}^\top\left(\frac{\varepsilon_{\text{step}}\delta \boldsymbol{c}}{\mu} + \frac{\boldsymbol{\rho}^+}{\boldsymbol{s}^{*+}} - \frac{\boldsymbol{\rho}^-}{\boldsymbol{s}^{*-}}\right) + \boldsymbol{C}^\top\left(\frac{(\boldsymbol{\rho}^+)^2}{\boldsymbol{s}'^{*+}} - \frac{(\boldsymbol{\rho}^-)^2}{\boldsymbol{s}'^{*-}}\right)\right\|_{\overline{\boldsymbol{H}}^+}\\
&= \left\|\boldsymbol{C}^\top \boldsymbol{h} + \boldsymbol{C}^\top\left(\frac{\varepsilon_{\text{step}}\delta \boldsymbol{c}}{\mu} + \varepsilon_{\text{step}}\delta \boldsymbol{r}g(\boldsymbol{s}) - \frac{\boldsymbol{s}^{*+}}{(\boldsymbol{s}^+)^2}\boldsymbol{\rho}^+ + \frac{\boldsymbol{s}^{*-}}{(\boldsymbol{s}^-)^2}\boldsymbol{\rho}^- + \frac{\boldsymbol{\rho}^+}{\boldsymbol{s}^{*+}} - \frac{\boldsymbol{\rho}^-}{\boldsymbol{s}^{*-}}\right) + \boldsymbol{C}^\top\left(\frac{(\boldsymbol{\rho}^+)^2}{\boldsymbol{s}'^{*+}} - \frac{(\boldsymbol{\rho}^-)^2}{\boldsymbol{s}'^{*-}}\right)\right\|_{\overline{\boldsymbol{H}}^+}\\
&= \Bigg\|\boldsymbol{C}^\top \boldsymbol{h} + \varepsilon_{\text{step}}\delta \boldsymbol{C}^\top\left(\frac{\boldsymbol{c}}{\mu} + \boldsymbol{r}g(\boldsymbol{s})\right) + \boldsymbol{C}^\top\left(\left(\boldsymbol{1} - \left(\frac{\boldsymbol{s}^{*+}}{\boldsymbol{s}^+}\right)^2\right)\frac{\boldsymbol{\rho}^+}{\boldsymbol{s}^{*+}} - \left(\boldsymbol{1} - \left(\frac{\boldsymbol{s}^{*-}}{\boldsymbol{s}^-}\right)^2\right)\frac{\boldsymbol{\rho}^-}{\boldsymbol{s}^{*-}}\right)\\
&+ \boldsymbol{C}^\top\left(\frac{(\boldsymbol{\rho}^+)^2}{\boldsymbol{s}'^{*+}} - \frac{(\boldsymbol{\rho}^-)^2}{\boldsymbol{s}'^{*-}}\right)\Bigg\|_{\overline{\boldsymbol{H}}^+}\\
&\leq (1+\varepsilon_{\text{step}}\delta)\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_{\overline{\boldsymbol{H}}^+} + 5\left\|\frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}}\right\|_\infty^{1/2}\varepsilon_{\text{solve}}\cdot\varepsilon_{\text{step}} + 2\left\|\frac{\boldsymbol{r}'^*}{\bar{\boldsymbol{r}}}\right\|_\infty^{1/2}\varepsilon_{\text{step}}^2
\end{aligned}$$

where we have used the triangle inequality, the fact that

$$
\begin{aligned}
&\varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \left(\frac{\boldsymbol{c}}{\mu} + \boldsymbol{r} g(\boldsymbol{s}) \right) \right\|_{\overline{\boldsymbol{H}}^+} \\
&= \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \left(\frac{\boldsymbol{c}}{\mu} + \frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-} \right) \right\|_{\overline{\boldsymbol{H}}^+} \\
&\le \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \left(\frac{\boldsymbol{c}}{\mu} + \frac{1}{\boldsymbol{s}^{*+}} - \frac{1}{\boldsymbol{s}^{*-}} \right) \right\|_{\overline{\boldsymbol{H}}^+} + \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \left(\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^{*+}} - \frac{1}{\boldsymbol{s}^-} + \frac{1}{\boldsymbol{s}^{*-}} \right) \right\|_{\overline{\boldsymbol{H}}^+} \\
&\le \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \left(\frac{\boldsymbol{c}}{\mu} + \frac{1}{\boldsymbol{s}^{*+}} - \frac{1}{\boldsymbol{s}^{*-}} \right) \right\|_{\overline{\boldsymbol{H}}^+} + \varepsilon_{\text{step}}\delta \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \left\| \boldsymbol{C}^\top \left(\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^{*+}} - \frac{1}{\boldsymbol{s}^-} + \frac{1}{\boldsymbol{s}^{*-}} \right) \right\|_{(\boldsymbol{C}^\top \boldsymbol{R}^* \boldsymbol{C})^+} \\
&\le \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \boldsymbol{h} \right\|_{\overline{\boldsymbol{H}}^+} + \varepsilon_{\text{step}}\delta \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \left\| \frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^{*+}} - \frac{1}{\boldsymbol{s}^-} + \frac{1}{\boldsymbol{s}^{*-}}}{\left(\frac{1}{(\boldsymbol{s}^{*+})^2} + \frac{1}{(\boldsymbol{s}^{*-})^2} \right)^{1/2}} \right\|_2 \\
&\le \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \boldsymbol{h} \right\|_{\overline{\boldsymbol{H}}^+} + \varepsilon_{\text{step}}\delta \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \left(\left\| \frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^{*+}}}{\left(\frac{1}{(\boldsymbol{s}^{*+})^2} \right)^{1/2}} \right\|_2 + \left\| \frac{\frac{1}{\boldsymbol{s}^-} - \frac{1}{\boldsymbol{s}^{*-}}}{\left(\frac{1}{(\boldsymbol{s}^{*-})^2} \right)^{1/2}} \right\|_2 \right) \\
&= \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \boldsymbol{h} \right\|_{\overline{\boldsymbol{H}}^+} + \varepsilon_{\text{step}}\delta \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \left(\left\| \frac{\boldsymbol{s}^{*+}}{\boldsymbol{s}^+} - \boldsymbol{1} \right\|_2 + \left\| \frac{\boldsymbol{s}^{*-}}{\boldsymbol{s}^-} - \boldsymbol{1} \right\|_2 \right) \\
&\le \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \boldsymbol{h} \right\|_{\overline{\boldsymbol{H}}^+} + \varepsilon_{\text{step}}\delta \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} 2\varepsilon_{\text{solve}}\sqrt{m} \text{ (as } \boldsymbol{s} \approx_{1+\varepsilon_{\text{solve}}} \boldsymbol{s}^*) \\
&= \varepsilon_{\text{step}}\delta \left\| \boldsymbol{C}^\top \boldsymbol{h} \right\|_{\overline{\boldsymbol{H}}^+} + 2 \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \varepsilon_{\text{step}}\varepsilon_{\text{solve}}
\end{aligned}
$$

and similarly, using $\left\| 1 - \left(\frac{\boldsymbol{s}^*}{\boldsymbol{s}}\right)^2 \right\|_\infty \le \varepsilon_{\text{solve}}(2 + \varepsilon_{\text{solve}})$, the fact that

$$
\begin{aligned}
&\left\| \boldsymbol{C}^\top \left(\left(\boldsymbol{1} - \left(\frac{\boldsymbol{s}^{*+}}{\boldsymbol{s}^+} \right)^2 \right) \frac{\boldsymbol{\rho}^+}{\boldsymbol{s}^{*+}} - \left(\boldsymbol{1} - \left(\frac{\boldsymbol{s}^{*-}}{\boldsymbol{s}^-} \right)^2 \right) \frac{\boldsymbol{\rho}^-}{\boldsymbol{s}^{*-}} \right) + \boldsymbol{C}^\top \left(\frac{(\boldsymbol{\rho}^+)^2}{\boldsymbol{s}'^{*+}} + \frac{(\boldsymbol{\rho}^-)^2}{\boldsymbol{s}'^{*-}} \right) \right\|_{\overline{\boldsymbol{H}}^+} \\
&\le \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \varepsilon_{\text{solve}}(2 + \varepsilon_{\text{solve}}) \left\| \boldsymbol{\rho} \right\|_2 + \left\| \frac{\boldsymbol{r}'^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \left\| \boldsymbol{\rho} \right\|_4^2 \\
&\le \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \varepsilon_{\text{step}}\varepsilon_{\text{solve}}(2 + \varepsilon_{\text{solve}})(1 + \varepsilon_{\text{solve}}) + \left\| \frac{\boldsymbol{r}'^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \varepsilon_{\text{step}}^2 (1 + \varepsilon_{\text{solve}})^2 \\
&\le 3 \left\| \frac{\boldsymbol{r}^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \varepsilon_{\text{step}}\varepsilon_{\text{solve}} + 2 \left\| \frac{\boldsymbol{r}'^*}{\bar{\boldsymbol{r}}} \right\|_\infty^{1/2} \varepsilon_{\text{step}}^2 .
\end{aligned}
$$

□

We will also use the following lemma, which is standard [AMV20].

Lemma C.6 (Small residual implies ℓ_∞ closeness)**.** *Given a flow* $\boldsymbol{f} = \boldsymbol{f}^0 + \boldsymbol{C}\boldsymbol{x}$ *with slacks* $\boldsymbol{s}$ *and resistances* $\boldsymbol{r}$*, if* $\left\| \boldsymbol{C}^\top \left(\frac{\boldsymbol{c}}{\mu} + \frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-} \right) \right\|_{(\boldsymbol{C}^\top \boldsymbol{R}\boldsymbol{C})^+} \le 1/1000$ *then* $\boldsymbol{f}$ *is* $(\mu, 1.01)$*-central.*

Applying Lemma C.5 for $T = \frac{k}{\varepsilon_{\text{step}}}$ iterations, we get the lemma below, which measures the closeness of $\boldsymbol{f}^*$ to the central path in ℓ_2 after T iterations.

Lemma C.7 (Centrality of $\boldsymbol{f}^*$)**.** *Let $\boldsymbol{f}^{*1}, \dots, \boldsymbol{f}^{*T+1}$ be flows with slacks $\boldsymbol{s}^{*1}, \dots, \boldsymbol{s}^{*T+1}$ and resistances $\boldsymbol{r}^{*1}, \dots, \boldsymbol{r}^{*T+1}$, and $\boldsymbol{f}^1, \dots, \boldsymbol{f}^{T+1}$ be flows with slacks $\boldsymbol{s}^1, \dots, \boldsymbol{s}^{T+1}$ and resistances $\boldsymbol{r}^1, \dots, \boldsymbol{r}^{T+1}$, such that $\boldsymbol{s}^t \approx_{1+\varepsilon_{\text{solve}}} \boldsymbol{s}^{*t}$ for all $t \in [T]$, where $T = \frac{k}{\varepsilon_{\text{step}}}$ for some $k \leq \sqrt{m}/10$, $\varepsilon_{\text{step}} \in (0, 0.1)$ and $\varepsilon_{\text{solve}} \in (0, 0.1)$. Additionally, we have that*

- *$\boldsymbol{f}^{*1}$ is μ-central*
- *For all $t \in [T]$, $\boldsymbol{f}^{*t+1} = \boldsymbol{f}^{*t} + \varepsilon_{\text{step}} \cdot \widetilde{\boldsymbol{f}}^t$, where*

$$\left\| \sqrt{\boldsymbol{r}^t} \left(\widetilde{\boldsymbol{f}}^{*t} - \widetilde{\boldsymbol{f}}^t \right) \right\|_\infty \leq \varepsilon \,,$$

$$\widetilde{\boldsymbol{f}}^{*t} = \delta g(\boldsymbol{s}^t) - \delta (\boldsymbol{R}^t)^{-1} \boldsymbol{B} \left(\boldsymbol{B}^\top (\boldsymbol{R}^t)^{-1} \boldsymbol{B} \right)^+ \boldsymbol{B}^\top g(\boldsymbol{s}^t)$$

  and $\delta = \frac{1}{\sqrt{m}}$.

Then, $\boldsymbol{f}^{*T+1}$ is $(\mu/(1+\varepsilon_{\text{step}}\delta)^T, 1.01)$-central, as long as we set $\varepsilon_{\text{step}} \leq 10^{-5}k^{-3}$ and $\varepsilon_{\text{solve}} \leq 10^{-5}k^{-3}$.

Proof. For all $t \in [T+1]$, we denote the residual of $\boldsymbol{f}^{*t}$ as $\boldsymbol{h}^t = \frac{\boldsymbol{c}(1+\varepsilon_{\text{step}}\delta)^{t-1}}{\mu} + \frac{1}{\boldsymbol{s}^{+,*t}} - \frac{1}{\boldsymbol{s}^{-,*t}}$. Note that $\boldsymbol{C}^\top \boldsymbol{h}^1 = \boldsymbol{0}$ as $\boldsymbol{f}^{*1}$ is μ-central.

We assume that the statement of the lemma is not true, and let $\widehat{T}$ be the smallest $t \in [T+1]$ such that $\boldsymbol{f}^{*t}$ is not $(\mu/(1+\varepsilon_{\text{step}}\delta)^{t-1}, 1.01)$-central. Obviously $\widehat{T} > 1$. This means that $\boldsymbol{f}^{*t}$ is $(\mu/(1+\varepsilon_{\text{step}}\delta)^{t-1}, 1.01)$-central for all $t \in [\widehat{T}-1]$, i.e. $\boldsymbol{s}^{*t} \approx_{1.01} \boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^{t-1}\right)$.

Also, note that by Lemma C.4 about slack stability, and since $(1+\varepsilon_{\text{step}}\delta)^{|\widehat{T}-t|} \leq (1+\delta)^{1.1k}$, we have $\boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^{t-1}\right) \approx_{3.7k^2} \boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^{\widehat{T}-1}\right)$ for all $t \in [T+1]$. Additionally, note that, as shown in proof of Lemma C.5, we have

$$\left\| \widetilde{\boldsymbol{f}}^{*\widehat{T}-1} \right\|_{\boldsymbol{r}^{\widehat{T}-1},\infty} \leq \left\| \widetilde{\boldsymbol{f}}^{*\widehat{T}-1} \right\|_{\boldsymbol{r}^{\widehat{T}-1},2} \leq 1 \,,$$

so

$$\begin{aligned}
\left\| \frac{\boldsymbol{s}^{*\widehat{T}}}{\boldsymbol{s}^{*\widehat{T}-1}} - \boldsymbol{1} \right\|_\infty &= \varepsilon_{\text{step}} \left\| \frac{\widetilde{\boldsymbol{f}}^{\widehat{T}-1}}{\boldsymbol{s}^{*\widehat{T}-1}} \right\|_\infty \\
&\leq \varepsilon_{\text{step}}(1+\varepsilon_{\text{solve}}) \left\| \sqrt{\boldsymbol{r}^{\widehat{T}-1}} \widetilde{\boldsymbol{f}}^{\widehat{T}-1} \right\|_\infty \\
&\leq \varepsilon_{\text{step}}(1+\varepsilon_{\text{solve}}) \left(\left\| \sqrt{\boldsymbol{r}^{\widehat{T}-1}} \widetilde{\boldsymbol{f}}^{*\widehat{T}-1} \right\|_\infty + \varepsilon \right) \\
&\leq \varepsilon_{\text{step}}(1+\varepsilon_{\text{solve}})(1+\varepsilon) \\
&\leq 1.3\varepsilon_{\text{step}} \,.
\end{aligned}$$

From this we conclude that $\boldsymbol{s}^{*\widehat{T}} \approx_{1+2.6\varepsilon_{\text{step}}} \boldsymbol{s}^{*\widehat{T}-1}$, and from the previous discussion we get that

$$\boldsymbol{s}^{*\widehat{T}} \approx_{1+2.6\varepsilon_{\text{step}}} \boldsymbol{s}^{*\widehat{T}-1} \approx_{1.01} \boldsymbol{s}(\mu/(1+\varepsilon_{\text{step}}\delta)^{\widehat{T}-1}) \approx_{3.7k^2} \boldsymbol{s}(\mu/(1+\varepsilon_{\text{step}}\delta)^{t-1}) \approx_{1.01} \boldsymbol{s}^{*t}\,,$$

so $\boldsymbol{s}^{*\widehat{T}} \approx_{4k^2} \boldsymbol{s}^{*t}$ for all $t \in [\widehat{T}-1]$.

On the other hand, if we apply Lemma C.5 $\widehat{T}-1$ times with $\bar{\boldsymbol{r}} = \boldsymbol{r}^{*\widehat{T}}$, we get

$$\begin{aligned}
&\left\|\boldsymbol{C}^\top \boldsymbol{h}^{\widehat{T}}\right\|_{\left(\boldsymbol{C}^\top \boldsymbol{R}^{\widehat{T}} \boldsymbol{C}\right)^+}\\
&= \left\|\boldsymbol{C}^\top \boldsymbol{h}^{\widehat{T}}\right\|_{\overline{\boldsymbol{H}}^+}\\
&\le (1+\varepsilon_{\text{step}}\delta)\left\|\boldsymbol{C}^\top \boldsymbol{h}^{\widehat{T}-1}\right\|_{\overline{\boldsymbol{H}}^+} + 5\left\|\frac{\boldsymbol{r}^{*\widehat{T}-1}}{\bar{\boldsymbol{r}}}\right\|_\infty^{1/2} \varepsilon_{\text{step}}\cdot\varepsilon_{\text{solve}} + 2\left\|\frac{\boldsymbol{r}^{*\widehat{T}}}{\bar{\boldsymbol{r}}}\right\|_\infty^{1/2} \varepsilon_{\text{step}}^2\\
&\dots\\
&\le 5\sum_{t=1}^{\widehat{T}-1}(1+\varepsilon_{\text{step}}\delta)^{\widehat{T}-t-1}\left\|\frac{\boldsymbol{r}^{*t}}{\boldsymbol{r}^{*\widehat{T}}}\right\|_\infty^{1/2} \varepsilon_{\text{step}}\cdot\varepsilon_{\text{solve}} + 2\sum_{t=1}^{\widehat{T}-1}(1+\varepsilon_{\text{step}}\delta)^{\widehat{T}-t-1}\left\|\frac{\boldsymbol{r}^{*t+1}}{\boldsymbol{r}^{*\widehat{T}}}\right\|_\infty^{1/2} \varepsilon_{\text{step}}^2\\
&\le 6T \max_{t\in[\widehat{T}-1]}\left\|\frac{\boldsymbol{r}^{*t}}{\boldsymbol{r}^{*\widehat{T}}}\right\|_\infty^{1/2} \varepsilon_{\text{step}}\cdot\varepsilon_{\text{solve}} + 2.4T \max_{t\in[\widehat{T}-1]}\left\|\frac{\boldsymbol{r}^{*t+1}}{\boldsymbol{r}^{*\widehat{T}}}\right\|_\infty^{1/2} \varepsilon_{\text{step}}^2\\
&\le 24Tk^2\varepsilon_{\text{step}}\cdot\varepsilon_{\text{solve}} + 10Tk^2\varepsilon_{\text{step}}^2\\
&= 24k^3\varepsilon_{\text{solve}} + 10k^3\varepsilon_{\text{step}}\\
&\le 1/1000\,,
\end{aligned}$$

where we used the fact that $(1+\varepsilon_{\text{step}}\delta)^{\widehat{T}} \le e^{\varepsilon_{\text{step}}\delta T} = e^{\delta k} \le 1.2$ and our setting of $\varepsilon_{\text{solve}} \le 10^{-5}k^{-3}$ and $\varepsilon_{\text{step}} \le 10^{-5}k^{-3}$. By Lemma C.6 this implies that $\boldsymbol{f}^{*\widehat{T}}$ is $(\mu/(1+\varepsilon_{\text{step}}\delta)^{\widehat{T}-1}, 1.01)$-central, a contradiction.

□

We are now ready to prove the following lemma, which is the goal of this section:

Lemma 3.5. *Let $\boldsymbol{f}^1,\dots,\boldsymbol{f}^{T+1}$ be flows with slacks $\boldsymbol{s}^t$ and resistances $\boldsymbol{r}^t$ for $t \in [T+1]$, where $T = \frac{k}{\varepsilon_{\text{step}}}$ for some $k \le \sqrt{m}/10$ and $\varepsilon_{\text{step}} = 10^{-5}k^{-3}$, such that*

- *$\boldsymbol{f}^1$ is $(\mu, 1+\varepsilon_{\text{solve}}/8)$-central for $\varepsilon_{\text{solve}} = 10^{-5}k^{-3}$*
- *For all $t \in [T]$, $\boldsymbol{f}^{t+1} = \begin{cases}\boldsymbol{f}(\mu) + \varepsilon_{\text{step}}\sum_{i=1}^{t}\widetilde{\boldsymbol{f}}^i & \text{if } \exists i \in [t] : \widetilde{\boldsymbol{f}}^i \neq \boldsymbol{0}\\ \boldsymbol{f}^1 & \text{otherwise}\end{cases}$, where*

$$\widetilde{\boldsymbol{f}}^{*t} = \delta g(\boldsymbol{s}^t) - \delta(\boldsymbol{R}^t)^{-1}\boldsymbol{B}\left(\boldsymbol{B}^\top(\boldsymbol{R}^t)^{-1}\boldsymbol{B}\right)^+\boldsymbol{B}^\top g(\boldsymbol{s}^t)$$

  for $\delta = \frac{1}{\sqrt{m}}$ and

$$\left\|\sqrt{\boldsymbol{r}^t}\left(\widetilde{\boldsymbol{f}}^{*t} - \widetilde{\boldsymbol{f}}^t\right)\right\|_\infty \le \varepsilon$$

  for $\varepsilon = 10^{-6}k^{-6}$.

Then, setting $\varepsilon_{\text{step}} = \varepsilon_{\text{solve}} = 10^{-5}k^{-3}$ *and* $\varepsilon = 10^{-6}k^{-6}$ *we get that* $\boldsymbol{s}^{T+1} \approx_{1.1} \boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^{k\varepsilon_{\text{step}}^{-1}}\right)$.

Proof. We set $\boldsymbol{f}^{*1} = \boldsymbol{f}(\mu)$ and for each $t \in [T]$,

$$\boldsymbol{f}^{*t+1} = \boldsymbol{f}^{*t} + \varepsilon_{\text{step}}\widetilde{\boldsymbol{f}}^{*t},$$

and the corresponding slacks $\boldsymbol{s}^{*t}$ and resistances $\boldsymbol{r}^{*t}$. Let $\widehat{T}$ be the first $t \in [T+1]$ such that $\boldsymbol{s}^{\widehat{T}} \not\approx_{1+\varepsilon_{\text{solve}}} \boldsymbol{s}^{*\widehat{T}}$. Obviously $t > 1$ as $\boldsymbol{s}^1 \approx_{1+\varepsilon_{\text{solve}}/8} \boldsymbol{s}(\mu) = \boldsymbol{s}^{*1}$.

Now, for all $t \in [T]$ we have

$$\left\|\sqrt{\boldsymbol{r}^t}\left(\widetilde{\boldsymbol{f}}^{*t} - \widetilde{\boldsymbol{f}}^t\right)\right\|_\infty \le \varepsilon.$$

Fix some $e \in E$. If $\widetilde{f}_e^t = \mathbf{0}$ for all $t \in [\widehat{T}-1]$, then we have $\sqrt{r_e^{\widehat{T}}}\left|\widetilde{f}_e^{*t}\right| = \sqrt{r_e^t}\left|\widetilde{f}_e^{*t}\right| \le \varepsilon$ for all such t. This means that

$$\begin{aligned}
\sqrt{r_e^{\widehat{T}}}\left|f_e^{*\widehat{T}} - f_e^{\widehat{T}}\right| &\le \sqrt{r_e^{\widehat{T}}}\left|f_e^{*\widehat{T}} - f_e^{*1}\right| + \sqrt{r_e^{\widehat{T}}}\left|f_e^{*1} - f_e^{\widehat{T}}\right| \\
&= \sqrt{r_e^{\widehat{T}}}\left|f_e^{*\widehat{T}} - f_e^{*1}\right| + \sqrt{r_e^{\widehat{T}}}\left|f_e^{*1} - f_e^1\right| \\
&\le \varepsilon_{\text{step}}\sqrt{r_e^{\widehat{T}}}\sum_{t=1}^{\widehat{T}-1}\left|\widetilde{f}_e^{*t}\right| + \sqrt{r_e^{\widehat{T}}}\left|f_e^{*1} - f_e^1\right| \\
&\le \widehat{T}\varepsilon_{\text{step}}\varepsilon + \sqrt{r_e^{\widehat{T}}}\left|f_e^{*1} - f_e^1\right| \\
&\le k\varepsilon + \sqrt{2}\varepsilon_{\text{solve}}/8 \\
&\le \varepsilon_{\text{solve}}/2,
\end{aligned}$$

as long as $\varepsilon \le \varepsilon_{\text{solve}}/(2k) = O(1/k^4)$. In the second to last inequality we used Lemma C.2.

Otherwise, there exists $t \in [\widehat{T}-1]$ such that $\widetilde{f}_e^t \ne \mathbf{0}$, and by definition $f_e^{\widehat{T}} = f_e(\mu) + \varepsilon_{\text{step}}\sum_{t=1}^{\widehat{T}-1}\widetilde{f}_e^t$, so

$$\begin{aligned}
\sqrt{r_e^{*\widehat{T}}}\left|f_e^{*\widehat{T}} - f_e^{\widehat{T}}\right| &\le \sqrt{r_e^{*\widehat{T}}}\left|f_e^{*1} - f_e(\mu)\right| + \varepsilon_{\text{step}}\sum_{t=1}^{\widehat{T}-1}\sqrt{r_e^{*\widehat{T}}}\left|\widetilde{f}_e^{*t} - \widetilde{f}_e^t\right| \\
&\le 3k^2\varepsilon_{\text{step}}\sum_{t=1}^{\widehat{T}-1}\sqrt{r_e^{*t}}\left|\widetilde{f}_e^{*t} - \widetilde{f}_e^t\right| \\
&\le 3k^2\varepsilon_{\text{step}}(1+\varepsilon_{\text{solve}})\sum_{t=1}^{\widehat{T}-1}\sqrt{r_e^t}\left|\widetilde{f}_e^{*t} - \widetilde{f}_e^t\right| \\
&\le 3k^2\varepsilon_{\text{step}}(1+\varepsilon_{\text{solve}})T\varepsilon \\
&\le 4k^3\varepsilon,
\end{aligned}$$

where we have used Lemma C.4 and the fact that $\boldsymbol{s}^t \approx_{1+\varepsilon_{\text{solve}}} \boldsymbol{s}^{*t}$ for all $t \in [\widehat{T}-1]$ which also implies that $\sqrt{\boldsymbol{r}^t} \approx_{1+\varepsilon_{\text{solve}}} \sqrt{\boldsymbol{r}^{*t}}$. Setting $\varepsilon = \frac{\varepsilon_{\text{solve}}}{8k^3} = O\left(\frac{1}{k^6}\right)$, this becomes $\le \varepsilon_{\text{solve}}/2$.

Therefore we have proved that $\left\|\sqrt{\boldsymbol{r}^{*\widehat{T}}}\left(\boldsymbol{f}^{*\widehat{T}}-\boldsymbol{f}^{\widehat{T}}\right)\right\|_{\infty} \leq \varepsilon_{\text{solve}}/2$, and so $\boldsymbol{s}^{*\widehat{T}} \approx_{1+\varepsilon_{\text{solve}}} \boldsymbol{s}^{\widehat{T}}$, a contradiction. Therefore we conclude that $\boldsymbol{s}^t \approx_{1+\varepsilon_{\text{solve}}} \boldsymbol{s}^{*t}$ for all $t \in [T+1]$.

Now, as long as $\varepsilon_{\text{step}}, \varepsilon_{\text{solve}} \leq 10^{-5}k^{-3}$, we can apply Lemma C.7, which guarantees that $\boldsymbol{s}^{*T+1} \approx_{1.01} \boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^T\right)$, and so $\boldsymbol{s}^{T+1} \approx_{1.1} \boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^T\right)$. Therefore we set $\varepsilon_{\text{step}} = \varepsilon_{\text{solve}} = 10^{-5}k^{-3}$ and $\varepsilon = 10^{-6}k^{-6} \leq \frac{\varepsilon_{\text{solve}}}{8k^3}$. □

C.3 Proof of Lemma 3.11

Proof. We will apply Lemma 3.5 with $\boldsymbol{f}^1$ being the flow corresponding to the resistances $\mathcal{L}.\boldsymbol{r} = \mathcal{C}.\boldsymbol{r}$, and $T = k\varepsilon_{\text{step}}^{-1}$. Note that it is important to maintain the invariant $\mathcal{L}.\boldsymbol{r} = \mathcal{C}.\boldsymbol{r}$ throughout the algorithm so that both data structures correspond to the same electrical flow problem. For each $t \in [T]$, for the t-th iteration, Lemma 3.5 requires an estimate $\widetilde{\boldsymbol{f}}^t$ such that $\left\|\sqrt{\boldsymbol{r}^t}\left(\widetilde{\boldsymbol{f}}^{*t}-\widetilde{\boldsymbol{f}}^t\right)\right\|_{\infty} \leq \varepsilon$, where

$$\widetilde{\boldsymbol{f}}^{*t} = \delta g(\boldsymbol{s}^t) - \delta(\boldsymbol{R}^t)^{-1}\boldsymbol{B}\left(\boldsymbol{B}^\top(\boldsymbol{R}^t)^{-1}\boldsymbol{B}\right)^+\boldsymbol{B}^\top g(\boldsymbol{s}^t)$$

and $\delta = 1/\sqrt{m}$.

We claim that such an estimate can be computed for all t by using $\mathcal{L}$ and $\mathcal{C}$. We apply the following process for each $t \in [T]$:

- Let Z be the edge set returned by $\mathcal{L}$.Solve().
- Call $\mathcal{C}$.Check(e) for each $e \in Z$ to obtain flow values $\widetilde{f}_e^t$.
- Compute $\boldsymbol{f}^t$ and its slacks $\boldsymbol{s}^{t+1}$ and resistances $\boldsymbol{r}^{t+1}$ as in Lemma 3.5, i.e.

$$f_e^{t+1} = \begin{cases} f_e(\mu) + \varepsilon_{\text{step}}\sum_{i=1}^{t}\widetilde{f}_e^i & \text{if } \exists i \in [t] : \widetilde{f}_e^i \neq 0 \\ f_e^1 & \text{otherwise} \end{cases}.$$

  This can be computed in $O(|Z|)$ by adding either $\varepsilon_{\text{step}}\widetilde{f}_e^i$ or $f_e(\mu) - f_e^1 + \varepsilon_{\text{step}}\widetilde{f}_e^i$ to f_e^t for each $e \in Z$.

- Call $\mathcal{L}$.Update($e, \boldsymbol{f}^{t+1}$) and $\mathcal{C}$.Update($e, \boldsymbol{f}^{t+1}$) for all e in the support of $\widetilde{\boldsymbol{f}}^t$. Note that $\mathcal{L}$.Update works as long as

$$r_e^{\max}/\alpha \leq r_e^{t+1} \leq \alpha \cdot r_e^{\min},$$

  where $r_e^{\max}, r_e^{\min}$ are the maximum and minimum values of $\mathcal{L}.r_e$ since the last call to $\mathcal{L}$.BatchUpdate. After this, we have $\mathcal{L}.\boldsymbol{r} = \mathcal{C}.\boldsymbol{r} = \boldsymbol{r}^{t+1}$.

In the above process, when $\mathcal{L}$.Solve() is called we have $\mathcal{L}.\boldsymbol{r} = \boldsymbol{r}^t$ (for $t = 1$ this is true because $\mathcal{L}.\boldsymbol{r}$ are the resistances corresponding to $\boldsymbol{f}^1$). By the $(\alpha, \beta, \varepsilon/2)$-Locator guarantees in Definition 3.7, with high probability Z contains all the edges e such that $\sqrt{r_e^t}\left|\widetilde{f}_e^{*t}\right| \geq \varepsilon/2$. Now, for each $e \in Z$, $\mathcal{C}$.Check(e) returns a flow value $\widetilde{f}_e^t$ such that:

- $\sqrt{r_e^t}\left|\widetilde{f}_e^t - \widetilde{f}_e^{*t}\right| \leq \varepsilon$
- if $\sqrt{r_e^t}\left|\widetilde{f}_e^{*t}\right| < \varepsilon/2$, then $\widetilde{f}_e^t = 0$.

Therefore, the condition that

$$\sqrt{\boldsymbol{r}^t} \left\| \widetilde{\boldsymbol{f}}^t - \widetilde{\boldsymbol{f}}^{*t} \right\|_\infty \le \varepsilon$$

is satisfied. Additionally $\widetilde{\boldsymbol{f}}^t$ is independent of the randomness of $\mathcal{L}$, because (the distribution of) $\widetilde{\boldsymbol{f}}^t$ would be the same if $\mathcal{C}$.CHECK was run for **all** edges e.

It remains to show that the LOCATOR requirement

$$\boldsymbol{r}^{\max}/\alpha \le \boldsymbol{r}^{t+1} \le \alpha \cdot \boldsymbol{r}^{\min}$$

is satisfied. Consider the minimum value of t for which this is not satisfied. By Lemma 3.5, we have that

$$\boldsymbol{s}^{\tau+1} \approx_{1.1} \boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^\tau\right) \tag{18}$$

for any $\tau \in [t]$.

Now let $\widehat{\boldsymbol{r}}$ be the resistances of $\mathcal{L}$ at any point since the last call to $\mathcal{L}$.BATCHUPDATE. By the lemma statement and (18), we know that $\widehat{\boldsymbol{r}} \approx_{1.1^2} \boldsymbol{r}(\hat{\mu})$ for some $\hat{\mu} \in [\mu/(1+\varepsilon_{\text{step}}\delta)^t, \mu^0]$. However, we also know that $\mu^0 \le \mu \cdot (1+\varepsilon_{\text{step}}\delta)^{(0.5\alpha^{1/4}-k)\varepsilon_{\text{step}}^{-1}}$ and so

$$\frac{\hat{\mu}}{\mu/(1+\varepsilon_{\text{step}}\delta)^t} \le (1+\varepsilon_{\text{step}}\delta)^{0.5\alpha^{1/4}\varepsilon_{\text{step}}^{-1}} \le (1+\delta)^{0.5\alpha^{1/4}},$$

so by Lemma C.4 we have

$$\boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^t\right) \approx_{0.75\alpha^{1/2}} \boldsymbol{s}\left(\hat{\mu}\right).$$

As $\boldsymbol{s}^{t+1} \approx_{1.1} \boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^t\right)$, we have that $\boldsymbol{s}^{t+1} \approx_{0.825\alpha^{1/2}} \boldsymbol{s}(\hat{\mu})$, and so

$$\boldsymbol{r}^{t+1} \approx_{0.825\alpha} \boldsymbol{r}(\hat{\mu}) \approx_{1.1^2} \widehat{\boldsymbol{r}}.$$

This means that $\boldsymbol{r}^{t+1} \approx_\alpha \widehat{\boldsymbol{r}}$ and is a contradiction.

We conclude that the requirements of $\mathcal{L}$ are met for all t, and as a result Lemma 3.5 shows that $\boldsymbol{s}^{T+1} \approx_{1.1} \boldsymbol{s}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^{k\varepsilon_{\text{step}}^{-1}}\right)$. By Lemma 3.6, we can now obtain $\boldsymbol{f}\left(\mu/(1+\varepsilon_{\text{step}}\delta)^{k\varepsilon_{\text{step}}^{-1}}\right)$. Finally, we return $\mathcal{L}.\boldsymbol{r}$ and $\mathcal{C}$ to their original states.

Success probability. We note that all the outputs of $\mathcal{C}$ are independent of the randomness of $\mathcal{L}$, and $\mathcal{L}$ is only updated based on these outputs. As each operation of $\mathcal{L}$ succeeds with high probability, the whole process succeeds with high probability as well.

Runtime. The recentering operation in Lemma 3.6 takes $\widetilde{O}(m)$. Additionally, we call $\mathcal{L}$.SOLVE $k\varepsilon_{\text{step}}^{-1} = O(k^4)$ times and, as $|Z| = O(1/\varepsilon^2)$, the total number of times $\mathcal{L}$.UPDATE, $\mathcal{C}$.UPDATE, and $\mathcal{C}$.CHECK are called is $O(k\varepsilon_{\text{step}}^{-1}\varepsilon^{-2}) = O(k^{16})$.

□

C.4 Proof of Lemma 3.10

Proof. Let $\delta = 1/\sqrt{m}$. Over a number of $T = \widetilde{O}\left(m^{1/2}/k\right)$ iterations, we will repeatedly apply MULTISTEP (Lemma 3.11). We will also replace the oracle from Definition 3.9 by the CHECKER data structure in Section F.

Initialization. We first initialize the Locator with error $\varepsilon/2$, by calling $\mathcal{L}.\textsc{Initialize}(\boldsymbol{f})$. Let $\boldsymbol{s}^t$ be the slacks $\mathcal{L}.\boldsymbol{s}$ before the t-th iteration and $\boldsymbol{r}^t$ the corresponding resistances, and $\boldsymbol{s}^{0t}$ be the slacks $\mathcal{L}.\boldsymbol{r}^0$ before the t-th iteration and $\boldsymbol{r}^{0t}$ the corresponding resistances, for $t \in [T]$. Also, we let $\mu_t = \mu / \left(1+\varepsilon_{\text{step}}\delta\right)^{(t-1)k\varepsilon_{\text{step}}^{-1}}$. We will maintain the invariant that $\boldsymbol{s}^t \approx_{1+\varepsilon_{\text{solve}}/8} \boldsymbol{s}\left(\mu_t\right)$, which is a requirement in order to apply Lemma 3.11.

As in [GLP21], we will also need to maintain $O(k^4)$ Checkers $\mathcal{C}^i$ for $i \in [O(k^4)]$, so we call $\mathcal{C}^i.\textsc{Initialize}(\boldsymbol{f}, \varepsilon, \beta_{\textsc{Checker}})$ for each one of these. Note that in general $\beta_{\textsc{Checker}} \neq \beta$, as the vertex sparsifiers $\mathcal{L}$ and $\mathcal{C}^i$ will not be on the same vertex set. As in Lemma 3.11, we will maintain the invariant that $\mathcal{L}.\boldsymbol{r} = \mathcal{C}^i.\boldsymbol{r}$ for all i.

Resistance updates. Assuming that all the requirements of Lemma 3.11 (MultiStep) are satisfied at the t-th iteration, that lemma computes a flow $\bar{\boldsymbol{f}} = \boldsymbol{f}\left(\mu_{t+1}\right)$ with slacks $\bar{\boldsymbol{s}}$. In order to guarantee that $\boldsymbol{s}^{t+1} \approx_{1+\varepsilon_{\text{solve}}/8} \boldsymbol{s}\left(\mu_{t+1}\right)$, we let Z be the set of edges such that either

$$s_e^{+,t} \not\approx_{1+\varepsilon_{\text{solve}}/8} \bar{s}_e^{+} \text{ or } s_e^{-,t} \not\approx_{1+\varepsilon_{\text{solve}}/8} \bar{s}_e^{-}$$

and then call $\mathcal{L}.\textsc{Update}(e, \bar{\boldsymbol{f}})$ for all $e \in Z$. This guarantees that $s_e^{+,t+1} = \bar{s}_e^{+}$ and $s_e^{-,t+1} = \bar{s}_e^{-}$ for all $e \in Z$ and so $\boldsymbol{s}^{t+1} \approx_{1+\varepsilon_{\text{solve}}/8} \bar{\boldsymbol{s}} = \boldsymbol{s}\left(\mu_{t+1}\right)$. We also apply the same updates to the $\mathcal{C}^i$'s using $\mathcal{C}^i.\textsc{Update}$, in order to ensure that they have the same resistances with $\mathcal{L}$.

Batched resistance updates. The number of times $\mathcal{L}.\textsc{Update}$ is called can be quite large because of multiple edges on which error slowly accumulates. This is because in general $\Omega(m)$ resistances will be updated throughout the algorithm. As Locator.Update is only slightly sublinear, this would lead to an $\Omega(m^{3/2})$-time algorithm. For this reason, as in [GLP21], we occasionally (every $\widehat{T}$ iterations for some $\widehat{T} \geq 1$ to be defined later) perform batched updates by calling $\mathcal{L}.\textsc{BatchUpdate}(Z, \bar{\boldsymbol{f}})$, where Z is the set of edges such that either

$$s_e^{+,t} \not\approx_{1+\varepsilon_{\text{solve}}/16} \bar{s}_e^{+} \text{ or } s_e^{-,t} \not\approx_{1+\varepsilon_{\text{solve}}/16} \bar{s}_e^{-}\,.$$

This again guarantees that $s_e^{+,t+1} = \bar{s}_e^{+}$ and $s_e^{-,t+1} = \bar{s}_e^{-}$ for all $e \in Z$ and so $\boldsymbol{s}^{t+1} \approx_{1+\varepsilon_{\text{solve}}/16} \bar{\boldsymbol{s}} = \boldsymbol{s}\left(\mu_{t+1}\right)$. Note that after updating $\mathcal{L}.\boldsymbol{s}$ and $\mathcal{L}.\boldsymbol{r}$, this operation also sets $\mathcal{L}.\boldsymbol{r}^0 = \mathcal{L}.\boldsymbol{r}$. We perform the same resistance updates to the $\mathcal{C}^i$'s in the regular (i.e. not batched) way, using $\mathcal{C}^i.\textsc{Update}$.

Locator requirements. What is left is to ensure that the requirements of Lemma 3.11 are satisfied at the t-th iteration, as well as that the requirements of $\mathcal{L}.\textsc{Update}$ and $\mathcal{L}.\textsc{BatchUpdate}$ from Definition 3.7 are satisfied.

The requirements are as follows:

1. Lemma 3.11: $\boldsymbol{s}^{0t} \approx_{1+\varepsilon_{\text{solve}}/8} \boldsymbol{s}\left(\mu^0\right)$ for some $\mu^0 \leq \mu_t \cdot \left(1+\varepsilon_{\text{step}}\delta\right)^{(0.5\alpha^{1/4}-k)\varepsilon_{\text{step}}^{-1}}$.

   Note that $\mathcal{L}.\boldsymbol{s}^0$ is updated every time $\mathcal{L}.\textsc{BatchUpdate}$ is called, and after the call we have $\mathcal{L}.\boldsymbol{s}^0 = \mathcal{L}.\boldsymbol{s} \approx_{1+\varepsilon_{\text{solve}}/16} \boldsymbol{s}(\mu^0)$ for some $\mu^0 > 0$. To ensure that it is called often enough, we call $\mathcal{L}.\textsc{BatchUpdate}(\emptyset)$ every $(0.5\alpha^{1/4}/k - 1)\varepsilon_{\text{step}}^{-1}$ iterations. Because of this, we have $\mu^0 \leq \mu_t \cdot \left(1+\varepsilon_{\text{step}}\delta\right)^{(0.5\alpha^{1/4}/k-1)\varepsilon_{\text{step}}^{-1}\cdot k} = \mu_t \cdot \left(1+\varepsilon_{\text{step}}\delta\right)^{(0.5\alpha^{1/4}-k)\varepsilon_{\text{step}}^{-1}}$. Additionally, for any resistances $\widehat{\boldsymbol{r}}$ that $\mathcal{L}$ had at any point since the last call to $\mathcal{L}.\textsc{BatchUpdate}$, it is immediate that

   $$\widehat{\boldsymbol{r}} \approx_{(1+\varepsilon_{\text{solve}}/8)^2} \boldsymbol{r}(\hat{\mu})$$

for some $\hat{\mu} \in [\mu_t, \mu^0]$, as this is exactly the invariant that our calls to $\mathcal{L}$.UPDATE maintain. Therefore, $\widehat{\boldsymbol{r}} \approx_{1.1^2} \boldsymbol{r}(\hat{\mu})$.

2. $\mathcal{L}$.UPDATE: $r_e^{\max}/\alpha \le r_e^{t+1} \le \alpha \cdot r_e^{\min}$, where $r_e^{\min}, r_e^{\max}$ are the minimum and maximum values that $\mathcal{L}.r_e$ has had since the last call to $\mathcal{L}$.BATCHUPDATE.

   Let $\widehat{\boldsymbol{r}}$ be any value of $\mathcal{L}.\boldsymbol{r}$ since the last call to $\mathcal{L}$.BATCHUPDATE. Because of the invariant maintained by resistance updates (including inside MULTISTEP), we have that $\widehat{\boldsymbol{r}}$ are $(\hat{\mu}, 1.1)$-central resistances for some $\hat{\mu}$ such that

$$\mu_{t+1} \le \hat{\mu} \le \mu_{t+1} \cdot (1+\varepsilon_{\text{step}}\delta)^{(0.5\alpha^{1/4}-k)\varepsilon_{\text{step}}^{-1}} \le \mu_{t+1} \cdot (1+\delta)^{0.5\alpha^{1/4}}\,.$$

   As in the previous item, we have that and $\boldsymbol{s}^{0t}$ are $(\mu^0, 1+\varepsilon_{\text{solve}}/8)$-central. By Lemma C.4 this implies

$$\boldsymbol{s}(\mu_{t+1}) \approx_{0.75\alpha^{1/2}} \boldsymbol{s}(\hat{\mu})\,,$$

   and since $\widehat{\boldsymbol{r}} \approx_{1.1^2} \boldsymbol{r}(\hat{\mu})$, we conclude that $\boldsymbol{r}(\mu_{t+1}) \approx_\alpha \widehat{\boldsymbol{r}}$.

3. $\mathcal{L}$.BATCHUPDATE: Between any two successive calls to $\mathcal{L}$.INITIALIZE, the number of edges updated (number of calls to $\mathcal{L}$.UPDATE plus the sum of $|Z|$ for all calls to $\mathcal{L}$.BATCHUPDATE) is $O(\beta m)$.

   We make sure that this is satisfied by calling $\mathcal{L}$.INITIALIZE($\bar{\boldsymbol{f}}$) every $\varepsilon_{\text{solve}}\sqrt{\beta m}/k$ iterations, where $\bar{\boldsymbol{f}} = \boldsymbol{f}(\mu_t)$, at the beginning of the t-th iteration.

   Consider any two successive initializations at iterations t^{init} and t^{end} respectively. Let ℓ be the number of edges e that have potentially been updated, i.e. such that either

$$s_e(\mu_{t^{init}})^+ \not\approx_{1+\varepsilon_{\text{solve}}/16} s_e(\mu_i)^+ \text{ or } s_e(\mu_{t^{init}})^- \not\approx_{1+\varepsilon_{\text{solve}}/16} s_e(\mu_i)^-$$

   for some $i \in \left[t^{init}, t^{end}\right]$. First, note that this implies that

$$\sqrt{r_e(\mu_{t^{init}})}\,|f_e(\mu_{t^{init}}) - f_e(\mu_i)| > \frac{\varepsilon_{\text{solve}}/16}{1+\varepsilon_{\text{solve}}/16} > \varepsilon_{\text{solve}}/17\,.$$

   Now, by the fact that $t^{end} - t^{init} \le \varepsilon_{\text{solve}}\sqrt{\beta m}/k$, we have that

$$\mu_{t^{init}} \le \mu_i \cdot (1+\varepsilon_{\text{step}}\delta)^{k\varepsilon_{\text{step}}^{-1}\cdot\varepsilon_{\text{solve}}\sqrt{\beta m}/k} \le \mu_i \cdot (1+\delta)^{\varepsilon_{\text{solve}}\sqrt{\beta m}}\,.$$

   By applying Lemma C.3 with $k = \varepsilon_{\text{solve}}\sqrt{\beta m}$ and $\gamma = \varepsilon_{\text{solve}}/17$, we get that $\ell \le O(\beta m)$. Therefore the statement follows.

We will also need to show how to implement the PERFECTCHECKER used in MULTISTEP using the $\mathcal{C}^i$'s, as well as how to satisfy all CHECKER requirements.

CHECKER requirements.

1. Implementing ε-PERFECTCHECKER inside MULTISTEP.

   We follow almost the same procedure as in [GLP21], other than the fact that we also need to provide some additional information to $\mathcal{C}^i$.SOLVE. Each call to PERFECTCHECKER.UPDATE translates to calls to $\mathcal{C}^i$.TEMPORARYUPDATE for all i. In addition, the i-th batch of calls to PerfectChecker.Check inside MultiStep (i.e. that corresponding to a single set of edges returned by $\mathcal{L}$.Solve) is only run on $\mathcal{C}^i$ using $\mathcal{C}^i$.Check. As each call to $\mathcal{C}^i$.Check is independent of previous calls to it, we can get correct outputs with high probability even when we run it multiple times (one for each edge returned by $\mathcal{L}$.Solve).

In order to guarantee that we have a vector $\boldsymbol{\pi}_{old}^i$ as required by $\mathcal{C}^i$.Check, once every k^4 calls to MultiStep (i.e. if t is a multiple of k^4) we compute

$$\boldsymbol{\pi}_{old}^i = \boldsymbol{\pi}^{C^{i,t}}\left(\boldsymbol{B}^\top g(\boldsymbol{s}(\mu_t))\right)$$

for all $i \in [O(k^4)]$, where $C^{i,t}$ is the vertex set of the Schur complement data structure stored internally by the $\mathcal{C}^i$ right before the t-th call to MultiStep. This can be computed in $\widetilde{O}(m)$ for each i as in DemandProjector.Initialize in Lemma 4.14. Now, the total number of $\mathcal{C}^i$.TemporaryUpdates that have not been rolled back is $O(k^{16})$, and the total number of $\mathcal{C}^i$.Updates over k^4 calls to MultiStep by Lemma C.3 is $O(k^{10}/\varepsilon_{\text{solve}}^2) = O(k^{16})$. This means that the total number of terminal insertions to $C^{i,t}$ as well as resistance changes is $O(k^{16})$. By Lemma 4.12, if C^i is the current state of the vertex set of the Schur complement of $\mathcal{C}^i$ and $\boldsymbol{s}$ are the current slacks,

$$\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}_{old}^i - \boldsymbol{\pi}^{C^i}(g(\boldsymbol{s}))\right) \le \widetilde{O}\left(\alpha' \beta_{\text{Checker}}^{-4}\right) \cdot k^{32},$$

where α' is the largest possible multiplicative change of some $\mathcal{C}^i.r_e$ since the computation of $\boldsymbol{\pi}_{old}^i$. Furthermore, note that $\boldsymbol{\pi}_{old}$ is supported on C^i. This is because $C^{i,t} \subseteq C^i$ and $C^{i,t}$ does not contain any temporary terminals.

Now, as we have already proved in Lemma 3.11, at any point inside the t-th call to MultiStep, $\mathcal{C}^i.\boldsymbol{r}$ are $(\hat{\mu}, 1.1)$-central resistances for some $\hat{\mu} \in [\mu_{t+1}, \mu_t]$.

Fix $\hat{\mu} \in [\mu_{t+1}, \mu_t], \hat{\mu}' \in [\mu_{t'+1}, \mu_{t'}]$, as well as the corresponding resistances of $\mathcal{C}^i$, $\widehat{\boldsymbol{r}}$, $\widehat{\boldsymbol{r}}'$, where $t' \ge t$. Now, note that since we are computing $\boldsymbol{\pi}_{old}^i$ every k^4 calls to MultiStep, we have that

$$\frac{\hat{\mu}}{\hat{\mu}'} \le \frac{\mu_t}{\mu_{t'+1}} \le (1 + \varepsilon_{\text{step}}\delta)^{k\varepsilon_{\text{step}}^{-1}\cdot(t'-t+1)} \le (1+\delta)^{O(k^5)},$$

so Lemma C.4 implies that

$$\boldsymbol{s}(\hat{\mu}) \approx_{O(k^{10})} \boldsymbol{s}(\hat{\mu}').$$

As $\widehat{\boldsymbol{r}} \approx_{1.1^2} \boldsymbol{r}(\hat{\mu})$ and $\widehat{\boldsymbol{r}}' \approx_{1.1^2} \boldsymbol{r}(\hat{\mu}')$, we get that $\widehat{\boldsymbol{r}} \approx_{O(k^{20})} \widehat{\boldsymbol{r}}'$. Therefore, $\alpha' \le O(k^{20})$. Setting $\beta_{\text{Checker}} \ge \widetilde{\Omega}\left(\alpha'^{1/4} k^8 \varepsilon^{-1/2} m^{-1/4}\right) = \widetilde{\Omega}\left(k^{16}/m^{1/4}\right)$, we get that

$$\widetilde{O}\left(\alpha' \beta_{\text{Checker}}^{-4}\right) \cdot k^{32} \le \varepsilon^2 m/4,$$

as required by $\mathcal{C}^i$.Check.

Finally, at the end of MultiStep we bring all $\mathcal{C}^i$ to their original state before calling MultiStep, by calling $\mathcal{C}^i$.Rollback. We also update all the resistances of $\mathcal{L}$ to their original state by calling $\mathcal{L}$.Update.

2. Between any two successive calls to $\mathcal{C}^i$.Initialize, the total number of edges updated at any point (via $\mathcal{C}^i$.Update or $\mathcal{C}^i$.TemporaryUpdate that have not been rolled back) is $O(\beta_{\text{Checker}} m)$.

For Update, we can apply a similar analysis as in the Locator case to show that if we call $\mathcal{C}^i$.Initialize every $\varepsilon_{\text{solve}}\sqrt{\beta_{\text{Checker}} m}/k$ iterations, the total number of updates never exceeds $O(\beta_{\text{Checker}} m)$. For TemporaryUpdate, note that at any time there are at most $O(k^{16})$ of these that have not been rolled back (this is inside MultiStep). Therefore, as long as $k^{16} \le O(\beta_{\text{Checker}} m) \Leftrightarrow \beta_{\text{Checker}} \ge \Omega(k^{16}/m)$, the requirement is met.

Output Guarantee. After the application of Lemma 3.11 at the last iteration, we will have $\bar{\boldsymbol{f}} = \boldsymbol{f}(\mu_{T+1})$, where $\mu_{T+1} = \mu/(1+\varepsilon_{\text{step}}\delta)^{\widetilde{O}(m^{1/2}\varepsilon_{\text{step}}^{-1})} = \mu/\text{poly}(m) \le m^{-10}$.

Success probability. Note that all operations of Locator and Checker work with high probability. Regarding the interaction of the randomness of these data structures and the fact that they work against oblivious adversaries, we defer to [GLP21], where there is a detailed discussion of why this works.

In short, note that outside of MultiStep, all updates are deterministic (as they only depend on the central path), and in MultiStep the updates to Locator and Checker only depend on outputs of a Checker. As each time we are getting the output from a different Checker, the inputs to $\mathcal{C}^i$.Check are independent of the randomness of $\mathcal{C}^i$, and thus succeed with high probability. Finally, note that the output of Locator is only passed onto $\mathcal{C}^i$.Check, whose output is then independent of the inputs received by Locator. Therefore, Locator does not "leak" any randomness.

Our only deviation from [GLP21] in Checker has to do with the extra input of Checker.Check ($\boldsymbol{\pi}^i_{old}$). However, note that this is computed outside of MultiStep, and as such the only randomness it depends on is the β_{Checker}-congestion reduction subset C^i generated when calling $\mathcal{C}^i$.Initialize. As such, it only depends on the internal randomness of $\mathcal{C}^i$. As we mentioned, the output of $\mathcal{C}^i$.Check is never fed back to $\mathcal{C}^i$, and thus the operation works with high probability.

Runtime (except Checker). Each call to MultiStep (Lemma 3.11) takes time $\widetilde{O}(m)$ plus $O(k^{16})$ calls to $\mathcal{L}$.Update and $O(k^4)$ calls to $\mathcal{L}$.Solve. As the total number of iterations is $\widetilde{O}\left(m^{1/2}/k\right)$, the total time because of calls to MultiStep is $\widetilde{O}\left(m^{3/2}/k\right)$, plus $\widetilde{O}\left(m^{1/2}k^{15}\right)$ calls to $\mathcal{L}$.Update and $\widetilde{O}\left(m^{1/2}k^3\right)$ calls to $\mathcal{L}$.Solve.

Now, the total number of calls to $\mathcal{L}$.Initialize is $\widetilde{O}\left(\frac{m^{1/2}/k}{\varepsilon_{\text{solve}}\sqrt{\beta m}/k}\right) = \widetilde{O}\left(k^3\beta^{-1/2}\right)$.

The total number of calls to $\mathcal{L}$.BatchUpdate($\emptyset$) is $\widetilde{O}\left(\frac{m^{1/2}/k}{0.5\alpha^{1/4}/k}\right) = \widetilde{O}\left(m^{1/2}\alpha^{-1/4}\right)$ and the total number of calls to $\mathcal{L}$.BatchUpdate($Z, \boldsymbol{f}$) is $\widetilde{O}\left(\frac{m^{1/2}}{k\widehat{T}}\right)$. Regarding the size of Z, let us focus on the calls to $\mathcal{L}$.BatchUpdate($Z, \boldsymbol{f}$) between two successive calls to $\mathcal{L}$.Initialize. We already showed that the sum of $|Z|$ over all calls during this interval is $O(\beta m)$. Therefore the total sum of $|Z|$ over all iterations of the algorithm is $\widetilde{O}\left(mk^3\beta^{1/2}\right)$.

In order to bound the number of calls to $\mathcal{L}$.Update, we concentrate on those between two successive calls to $\mathcal{L}$.BatchUpdate($Z, \boldsymbol{f}$) in iterations t^{old} and $t^{new} > t^{old}$. After the call to $\mathcal{L}$.BatchUpdate($Z, \boldsymbol{f}$) in iteration t^{old} we have $\mathcal{L}.\boldsymbol{s} \approx_{1+\varepsilon_{\text{solve}}/16} \boldsymbol{s}(\mu_{t^{old}})$. Fix $\mu \in \left[\mu(t^{new}), \mu(t^{old})\right]$ and let ℓ be the number of $e \in E$ such that $s_e(\mu) \not\approx_{1+\varepsilon_{\text{solve}}/8} s_e^{t^{old}}$. As $s_e^{t^{old}} \approx_{1+\varepsilon_{\text{solve}}/16} s_e(\mu_{t^{old}})$ by the guarantees of $\mathcal{L}$.BatchUpdate, this implies that $s_e(\mu) \not\approx_{1+\varepsilon_{\text{solve}}/16} s_e(\mu_{t^{old}})$, and so

$$\sqrt{r_e(\mu^{t^{old}})}\,|f_e(\mu_{t^{old}}) - f_e(\mu)| \ge \frac{\varepsilon_{\text{solve}}/16}{1+\varepsilon_{\text{solve}}/16} > \varepsilon_{\text{solve}}/17\,.$$

As

$$\mu_{t_{old}} \leq \mu \cdot (1+\varepsilon_{\text{step}}\delta)^{k\varepsilon_{\text{step}}^{-1}\cdot \widehat{T}} \leq \mu \cdot (1+\delta)^{k\cdot \widehat{T}} ,$$

by applying Lemma C.3 with $k = (1+\delta)^{k\cdot\widehat{T}}$ and $\gamma = \varepsilon_{\text{solve}}/17$, we get that $\ell \leq O(k^2\widehat{T}^2\varepsilon_{\text{solve}}^{-2})$. As there are $\widetilde{O}\left(\frac{m^{1/2}}{k\widehat{T}}\right)$ calls to $\mathcal{L}$.BatchUpdate$(Z, \boldsymbol{f})$, the total number of calls to $\mathcal{L}$.Update is

$$\widetilde{O}\left(m^{1/2}\widehat{T}\varepsilon_{\text{solve}}^{-2}\right)\widetilde{O}\left(m^{1/2}k^6\widehat{T}\right) .$$

We conclude that we have runtime $\widetilde{O}\left(m^{3/2}/k\right)$, plus

- $\widetilde{O}\left(k^3\beta^{-1/2}\right)$ calls to $\mathcal{L}$.Initialize,
- $\widetilde{O}\left(m^{1/2}k^3\right)$ calls to $\mathcal{L}$.Solve,
- $\widetilde{O}\left(m^{1/2}\left(k^6\widehat{T}+k^{15}\right)\right)$ calls to $\mathcal{L}$.Update,
- $\widetilde{O}\left(m^{1/2}\alpha^{-1/4}\right)$ calls to $\mathcal{L}$.BatchUpdate$(\emptyset)$, and
- $\widetilde{O}\left(m^{1/2}k^{-1}\widehat{T}^{-1}\right)$ calls to $\mathcal{L}$.BatchUpdate$(Z, \boldsymbol{f})$.

Runtime of Checker. We look at each operation separately. We begin with the runtime of Checker.Check. We have

$$\widetilde{O}\left(\underbrace{m^{1/2}/k}_{\#\text{ calls to MultiStep}} \cdot \underbrace{k^{16}}_{\#\text{ calls in each MultiStep}} \cdot \underbrace{(\beta_{\text{Checker}}m + (k^{16}\beta_{\text{Checker}}^{-2}\varepsilon^{-2})^2))\varepsilon^{-2}}_{\text{runtime per call}}\right) .$$

To make the first term $\widetilde{O}\left(m^{3/2}/k\right)$, we set $\beta_{\text{Checker}} = k^{-28}$. Note that this satisfies our previous requirements that $\beta_{\text{Checker}} \geq \widetilde{\Omega}\left(k^{16}/m\right)$ and $\beta_{\text{Checker}} \geq \widetilde{\Omega}\left(k^{16}/m^{1/4}\right)$ as long as $k \leq m^{1/176}$. Therefore the total runtime because of this operation is $\widetilde{O}\left(m^{3/2}/k + m^{1/2}k^{195}\right)$.

For Checker.Initialize, we have

$$\widetilde{O}\left(\underbrace{k^4}_{\#\text{Checkers}} \cdot \underbrace{k^3\beta_{\text{Checker}}^{-1/2}}_{\#\text{times initialized}} \cdot \underbrace{m\beta_{\text{Checker}}^{-4}\varepsilon^{-4}}_{\text{runtime per init}}\right) = \widetilde{O}\left(mk^{157}\right) .$$

For Checker.Update, similarly with the analysis of Locator but noting that there are no batched updates, we have

$$\widetilde{O}\left(\underbrace{k^4}_{\#\text{Checkers}} \cdot \underbrace{m\varepsilon_{\text{solve}}^{-2}}_{\#\text{calls per Checker}} \cdot \underbrace{\beta_{\text{Checker}}^{-2}\varepsilon^{-2}}_{\text{runtime per call}}\right) = \widetilde{O}\left(mk^{78}\right) .$$

For Checker.TemporaryUpdate, we have

$$\widetilde{O}\left(\underbrace{k^4}_{\#\text{Checkers}} \cdot \underbrace{m^{1/2}/k}_{\#\text{ calls to MultiStep}} \cdot \underbrace{k^{16}}_{\#\text{ calls per MultiStep}} \cdot \underbrace{(k^{16}\beta_{\text{Checker}}^{-2}\varepsilon^{-2})^2}_{\text{runtime per call}}\right) = \widetilde{O}\left(m^{1/2}k^{187}\right) .$$

Finally, note that, by definition, computing the vectors $\boldsymbol{\pi}_{old}^i$ takes $\widetilde{O}\left(m^{3/2}/k\right)$, as we do it once per k^4 calls to MultiStep and it takes $\widetilde{O}\left(mk^4\right)$.

As long as $k \leq m^{1/316}$, the total runtime because of Checker is $\widetilde{O}\left(m^{3/2}/k\right)$. □

D Deferred Proofs from Section 4

D.1 Proof of Lemma 4.4

We first provide a helper lemma for upper bounding escape probabilities in terms of the underlying graph's resistances.

Lemma D.1 (Bounding escape probabilities)**.** *Let a graph with resistances $\boldsymbol{r}$, and consider a random walk which at each step moves from the current vertex u to an adjacent vertex v sampled with probability proportional to $1/r_{uv}$. Let $p_u^{\{u,t\}}(s)$ represent the probability that a walk starting at s hits u before t. Then*

$$p_u^{\{u,t\}}(s) = \frac{R_{eff}(s,t)}{R_{eff}(u,t)} \cdot p_s^{\{s,t\}}(u) \leq \frac{R_{eff}(s,t)}{R_{eff}(u,t)} \leq \frac{r_{st}}{R_{eff}(u,t)}\,.$$

Proof. Using standard arguments we can prove that if $\boldsymbol{L}$ is the Laplacian associated with the underlying graph, then

$$p_u^{\{u,t\}}(s) = \frac{(\mathbf{1}_s - \mathbf{1}_t)^\top \boldsymbol{L}^+ (\mathbf{1}_u - \mathbf{1}_t)}{R_{eff}(u,t)}\,.$$

This immediately yields the claim as we can further write it as

$$p_u^{\{u,t\}}(s) = \frac{R_{eff}(s,t)}{R_{eff}(u,t)} \cdot \frac{(\mathbf{1}_u - \mathbf{1}_t)^\top \boldsymbol{L}^+ (\mathbf{1}_s - \mathbf{1}_t)}{R_{eff}(s,t)} = \frac{R_{eff}(s,t)}{R_{eff}(u,t)} \cdot p_s^{\{s,t\}}(u) \leq \frac{R_{eff}(s,t)}{R_{eff}(u,t)}\,,$$

where we crucially used the symmetry of $\boldsymbol{L}$. The final inequality is due to the fact that $R_{eff}(s,t) \leq r_{st}$.

Now let us prove the claimed identity for escape probabilities. Let $\boldsymbol{\psi}$ be the vector defined by $\psi_i = p_u^{\{u,t\}}(i)$ for all $i \in V$, which clearly satisfies $\psi_u = 1$ and $\psi_t = 0$. Furthermore, for all $i \notin \{u,t\}$ we have

$$\psi_i = \sum_{j \sim i} \frac{r_{ij}^{-1}}{\sum_{k \sim i} r_{ik}^{-1}} \psi_j\,,$$

which can be written in short as

$$(\boldsymbol{L}\boldsymbol{\psi})_i = 0 \quad \text{for all } i \notin \{s,t\}\,.$$

Now we solve the corresponding linear system. We interpret $\boldsymbol{\psi}$ as electrical potentials corresponding to routing $1/R_{eff}(u,t)$ units of electrical flow from u to t. Indeed, by Ohm's law, this corresponds to a potential difference $\psi_u - \psi_t = 1$. Furthermore, this shows that

$$\psi_s - \psi_t = (\mathbf{1}_s - \mathbf{1}_t)^\top \boldsymbol{L}^+ (\mathbf{1}_u - \mathbf{1}_t) \cdot \frac{1}{R_{eff}(u,t)}\,,$$

which concludes the proof. □

Now we are ready to prove the main statement.

Proof of Lemma 4.4. Note that the demand can be decomposed as $\boldsymbol{d} - \boldsymbol{\pi}^C(\boldsymbol{d}) = \boldsymbol{d}^1 - \boldsymbol{d}^2$, where $\boldsymbol{d}^1 = \frac{\mathbf{1}_s}{\sqrt{r_{st}}} - \boldsymbol{\pi}^C(\frac{\mathbf{1}_s}{\sqrt{r_{st}}})$ and $\boldsymbol{d}^2 = \frac{\mathbf{1}_t}{\sqrt{r_{st}}} - \boldsymbol{\pi}^C(\frac{\mathbf{1}_t}{\sqrt{r_{st}}})$. Now let p^1 be the probability distribution of $s-C$ random walks obtained via a random walk from s with transition probabilities proportional to inverse resistances. Similarly, let p^2 be the probability distribution of $t-C$ random walks obtained by running the same process starting from t.

Now, it is well known that an electrical flow is the sum of these random walks, i.e.

$$\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{d}^1 = \frac{1}{\sqrt{r_{st}}} \cdot \mathbb{E}_{P\sim p^1}\left[net(P)\right]$$

and similarly for $\boldsymbol{d}^2$

$$\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{d}^2 = \frac{1}{\sqrt{r_{st}}} \cdot \mathbb{E}_{P\sim p^2}\left[net(P)\right] ,$$

where $net(P) \in \mathbb{R}^m$ is a flow vector whose e-th entry is the net number of times the edge $e = (u, v)$ is used by P. Therefore we can write:

$$\begin{aligned} \left|\frac{\phi_u - \phi_v}{\sqrt{r_{uv}}}\right| &= \sqrt{r_{uv}}\left|\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{d}\right|_{uv} \\ &= \sqrt{\frac{r_{uv}}{r_{st}}}\left|\mathbb{E}_{P^1\sim p^1}\left[net_e(P^1)\right] - \mathbb{E}_{P^2\sim p^2}\left[net_e(P^2)\right]\right| . \end{aligned}$$

Let us also subdivide e by inserting an additional vertex w in the middle (i.e. $r_{uw} = r_{wv} = r_{uv}/2$). This has no effect in the random walks, but will be slightly more convenient in terms of notation. The first expectation term can be expressed as

$$\begin{aligned} \mathbb{E}_{P^1\sim p^1}\left[net_e(P^1)\right] = &\Pr_{P^1\sim p^1}\left[P^1 \text{ visits } t \text{ before } C \cup \{w\}\right] \cdot \mathbb{E}_{P^1\sim p^1}\left[net_e(P^1) \mid P^1 \text{ visits } t \text{ before } C\cup\{w\}\right] \\ &+ \Pr_{P^1\sim p^1}\left[P^1 \text{ visits } w \text{ before } C\cup\{t\}\right] \cdot \mathbb{E}_{P^1\sim p^1}\left[net_e(P^1) \mid P^1 \text{ visits } t \text{ before } C\cup\{t\}\right] . \end{aligned}$$

Now, note that

$$\mathbb{E}_{P^1\sim p^1}\left[net_e(P^1) \mid P^1 \text{ visits } t \text{ before } C\cup\{w\}\right] = \mathbb{E}_{P^2\sim p^2}\left[net_e(P^2)\right] .$$

Additionally,

$$\begin{aligned} &\Pr_{P^1\sim p^1}\left[P^1 \text{ visits } w \text{ before } C\cup\{t\}\right] \\ &= \Pr_{P^1\sim p^1}\left[P^1 \text{ visits } w \text{ before } C\right] \cdot \Pr_{P^1\sim p^1}\left[P^1 \text{ visits } w \text{ before } t \mid P^1 \text{ visits } w \text{ before } C\right] \end{aligned}$$

The first term of the product is $p_w^{C\cup\{w\}}(s)$. For the second term, we define a new graph $\widehat{G}$ by deleting C, and denote the hitting probabilities in $\widehat{G}$ by $\widehat{p}$. Then, the second term is equal to $\widehat{p}_w^{\{t,w\}}(s)$.

We have concluded that

$$\mathbb{E}_{P^1\sim p^1}\left[net_e(P^1)\right] \leq \mathbb{E}_{P^2\sim p^2}\left[net_e(P^2)\right] + p_w^{C\cup\{w\}}(s) \cdot \widehat{p}_w^{\{t,w\}}(s) .$$

Combining this with the symmetric argument for p^2 shows that

$$\left|\mathbb{E}_{P^1\sim p^1}\left[net_e(P^1)\right]-\mathbb{E}_{P^2\sim p^2}\left[net_e(P^2)\right]\right|\leq p_w^{C\cup\{w\}}(s)\cdot\widehat{p}_w^{\{t,w\}}(s)+p_w^{C\cup\{w\}}(t)\cdot\widehat{p}_w^{\{s,w\}}(t)\,.$$

Using Lemma D.1 and the fact that $\widehat{R}_{eff}(w,t)\geq r_{uv}/4$ ($\widehat{R}_{eff}$ are the effective resistances in $\widehat{G}$), we can bound

$$\widehat{p}_w^{\{t,w\}}(s)\leq\min\{1,\frac{r_{st}}{\widehat{R}_{eff}(w,t)}\}\leq\min\{1,4\frac{r_{st}}{r_{uv}}\}\,,$$

and the same upper bound holds for $\widehat{p}_w^{\{s,w\}}(t)$. Therefore,

$$\begin{aligned}&\left|\mathbb{E}_{P^1\sim p^1}\left[net_e(P^1)\right]-\mathbb{E}_{P^2\sim p^2}\left[net_e(P^2)\right]\right|\\&\leq\min\{1,4\frac{r_{st}}{r_{uv}}\}\left(p_w^{C\cup\{w\}}(s)+p_w^{C\cup\{w\}}(t)\right)\\&\leq 2\sqrt{\frac{r_{st}}{r_{uv}}}\left(p_w^{C\cup\{w\}}(s)+p_w^{C\cup\{w\}}(t)\right)\\&\leq 2\sqrt{\frac{r_{st}}{r_{uv}}}\left(p_u^{C\cup\{u\}}(s)+p_v^{C\cup\{v\}}(s)+p_u^{C\cup\{u\}}(t)+p_v^{C\cup\{v\}}(t)\right)\,.\end{aligned}$$

Putting everything together, we have that

$$\left|\frac{\phi_u-\phi_v}{\sqrt{r_{uv}}}\right|\leq 2\left(p_u^{C\cup\{u\}}(s)+p_v^{C\cup\{v\}}(s)+p_u^{C\cup\{u\}}(t)+p_v^{C\cup\{v\}}(t)\right)\,.$$

□

D.2 Proof of Lemma 4.5

Proof. Let $\boldsymbol{d}=\boldsymbol{B}^\top\frac{\boldsymbol{1}_e}{\sqrt{\boldsymbol{r}}}$ and $e=(u,w)$. Note that $\mathcal{E}_r(\boldsymbol{d})\leq r_e\cdot(\frac{1}{\sqrt{r_e}})^2=1$, therefore the case that remains is

$$R_{eff}(C,e)\geq 36\cdot r_e\,. \tag{19}$$

For each $v\in C$ by Lemma 4.6 we have that

$$\left|\pi_v^C(\boldsymbol{d})\right|\leq(p_v^C(u)+p_v^C(w))\cdot\frac{\sqrt{r_e}}{R_{eff}(v,e)}\,.$$

Now, we would like to bound the energy of routing $\boldsymbol{\pi}^C(\boldsymbol{d})$ by the energy to route it via w. For each $v\in C$ we let $\boldsymbol{d}^v$ be the following demand:

$$\boldsymbol{d}^v=\pi_v^C(\boldsymbol{d})\cdot(\boldsymbol{1}_v-\boldsymbol{1}_w)\,.$$

Note that $\boldsymbol{\pi}^C(\boldsymbol{d}) = \sum_{v\in C} \boldsymbol{d}^v$. We have,

$$\begin{aligned}
\sqrt{\mathcal{E}_{\boldsymbol{r}}(\pi^C(\boldsymbol{d}))} &= \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\pi^C\left(\sum_{v\in C}\boldsymbol{d}^v\right)\right)} \\
&\leq \sum_{v\in C}\sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\pi^C\left(\boldsymbol{d}^v\right)\right)} \\
&\leq \sum_{v\in C}(R_{eff}(v,w))^{1/2}\left|\pi_v^C(\boldsymbol{d})\right| \\
&\leq \sum_{v\in C}(R_{eff}(v,w))^{1/2}\cdot(p_v^C(u)+p_v^C(w))\cdot\frac{\sqrt{r_e}}{R_{eff}(v,e)}\,.
\end{aligned}$$

Now, note that, because R_{eff} is a metric,

$$\begin{aligned}
R_{eff}(v,u) &\geq R_{eff}(v,w) - |R_{eff}(v,w) - R_{eff}(v,u)| \\
&\geq R_{eff}(v,w) - r_e \\
&\geq R_{eff}(v,w) - \frac{1}{36}R_{eff}(C,e) \\
&\geq \frac{35}{36}R_{eff}(v,w)\,,
\end{aligned}$$

where we used the triangle inequality twice, (19), and also the fact that $R_{eff}(v,w) \geq R_{eff}(C,e)$ because $v \in C$ and $w \in e$. Now, note also that

$$R_{eff}(v,e) \geq \frac{1}{2}\min\{R_{eff}(v,w), R_{eff}(v,u) \geq \frac{1}{2}\cdot\frac{35}{36}R_{eff}(v,w)\}\,,$$

so

$$\begin{aligned}
&2\sum_v(R_{eff}(v,w))^{1/2}\cdot(p_v^C(u)+p_v^C(w))\cdot\frac{\sqrt{r_e}}{R_{eff}(v,e)} \\
&\leq 2\sqrt{2\cdot\frac{36}{35}}\sum_v(p_v^C(u)+p_v^C(w))\cdot\sqrt{\frac{r_e}{R_{eff}(v,e)}} \\
&\leq 6\cdot\sqrt{\frac{r_e}{R_{eff}(C,e)}}\,,
\end{aligned}$$

where we used the fact that $\sum_{v\in C}(p_v^C(u)+p_v^C(w)) = 2$ and $R_{eff}(v,e) \geq R_{eff}(C,e)$ because $v \in C$. $\square$

D.3 Proof of Lemma 4.9

Proof. By definition of the fact that e is not ε-important, we have

$$R_{eff}(C,e) > r_e/\varepsilon^2\,.$$

Using the fact that the demand $\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{p}}{\sqrt{\boldsymbol{r}}}\right)$ is supported on C and Lemma 4.5, we get

$$\begin{aligned}
\left|\left\langle \mathbf{1}_e, \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{\phi}^*\right\rangle\right| &= \left|\left\langle \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right), \boldsymbol{\phi}^*_C\right\rangle\right| \\
&\leq \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right)} \cdot \sqrt{E_{\boldsymbol{r}}(\boldsymbol{\phi}^*)} \\
&= \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\mathbf{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right)} \cdot \delta\sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top \frac{\boldsymbol{p}}{\sqrt{\boldsymbol{r}}}\right)\right)} \\
&\leq 6\sqrt{\frac{r_e}{R_{eff}(C,e)}} \cdot \delta\sqrt{m} \\
&\leq 6\varepsilon\,.
\end{aligned}$$

□

D.4 Proof of Lemma 4.11

Proof. We note that

$$\begin{aligned}
&\sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}) - \boldsymbol{\pi}^C(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}})\right)} \\
&= \left|\pi_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}})\right| \sqrt{R_{eff}(C,v)} \\
&\leq \sum_{e=(u,w)\in E} \left|\pi_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{q_e}{\sqrt{r_e}}\mathbf{1}_e)\right| \sqrt{R_{eff}(C,v)} \\
&\leq \sum_{e=(u,w)\in E} (p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)) \cdot \min\left\{\sqrt{\frac{R_{eff}(C,v)}{r_e}}, \frac{\sqrt{r_e R_{eff}(C,v)}}{R_{eff}(e,v)}\right\}\,,
\end{aligned}$$

where we used Lemma 4.6. For some sufficiently large c to be defined later, we partition E into X and Y, where $X = \{e \in E \mid R_{eff}(C,v) \leq c^2 \cdot r_e \text{ or } r_e R_{eff}(C,v) \leq c^2 \cdot (R_{eff}(e,v))^2\}$ and $Y = E\backslash X$. We first note that

$$\begin{aligned}
&\sum_{e=(u,w)\in X} (p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)) \cdot \min\left\{\sqrt{\frac{R_{eff}(C,v)}{r_e}}, \frac{\sqrt{r_e R_{eff}(C,v)}}{R_{eff}(e,v)}\right\} \\
&\leq c \cdot \sum_{e=(u,w)\in X} (p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)) \\
&\leq c \cdot \widetilde{O}\left(\beta^{-2}\right)\,,
\end{aligned}$$

where the last inequality follows by the congestion reduction property.

Now, let $e = (u,w) \in Y$. We will prove that both u and w are much closer to v than C. This, in turn, will imply that their hitting probabilities on v are roughly the same, and so they mostly cancel out in the projection.

First of all, we let $R_{eff}(C,v) = c_1^2 \cdot r_e$ and $r_e R_{eff}(C,v) = c_2^2 \cdot (R_{eff}(e,v))^2$, for some $c_1, c_2 > 0$, where by definition $c_1, c_2 \geq c$. Now, we assume without loss of generality that $R_{eff}(u,v) \leq R_{eff}(w,v)$, and so

$$R_{eff}(u,v) \leq 2R_{eff}(e,v) = \frac{2}{c_2}\sqrt{r_e R_{eff}(C,v)} = \frac{2}{c_1 c_2} R_{eff}(C,v) \leq \frac{2}{c_1 c_2}\left(R_{eff}(C,u) + R_{eff}(u,v)\right),$$

so

$$R_{eff}(u,v) \leq \frac{\frac{2}{c_1 c_2}}{1 - \frac{2}{c_1 c_2}} R_{eff}(C,u) = \frac{2}{c_1 c_2 - 2} R_{eff}(C,u) \leq \frac{3}{c_1 c_2} R_{eff}(C,u)\,. \tag{20}$$

Futhermore, note that

$$R_{eff}(w,v) \leq R_{eff}(u,v) + r_e \leq \left(\frac{2}{c_1 c_2} + \frac{1}{c_1^2}\right) R_{eff}(C,v) \leq \left(\frac{2}{c_1 c_2} + \frac{1}{c_1^2}\right)\left(R_{eff}(C,w) + R_{eff}(w,v)\right),$$

and so we have

$$R_{eff}(w,v) \leq \frac{\frac{2}{c_1 c_2} + \frac{1}{c_1^2}}{1 - \frac{2}{c_1 c_2} - \frac{1}{c_1^2}} R_{eff}(C,w) \leq 3\left(\frac{1}{c_1 c_2} + \frac{1}{c_1^2}\right) R_{eff}(C,w)\,. \tag{21}$$

Now, by Lemma D.1 together with (20) we have

$$p_v^{C\cup\{v\}}(u) \geq 1 - \frac{3}{c_1 c_2}$$

and with (21) we have

$$p_v^{C\cup\{v\}}(w) \geq 1 - 3\left(\frac{1}{c_1 c_2} + \frac{1}{c_1^2}\right),$$

therefore

$$\left|p_v^{C\cup\{v\}}(u) - p_v^{C\cup\{v\}}(w)\right| \leq 6\left(\frac{1}{c_1 c_2} + \frac{1}{c_1^2}\right).$$

So,

$$\begin{aligned}
\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{q_e}{\sqrt{r_e}}\mathbf{1}_e\right)\right| \sqrt{R_{eff}(C,v)} &= \left|p_v^{C\cup\{v\}}(u) - p_v^{C\cup\{v\}}(w)\right| \sqrt{\frac{R_{eff}(C,v)}{r_e}} \\
&\leq 6\left(\frac{1}{c_1 c_2} + \frac{1}{c_1^2}\right) \cdot c_1 \\
&= 6\left(\frac{1}{c_2} + \frac{1}{c_1}\right) \\
&= O\left(\frac{1}{c}\right).
\end{aligned}$$

Now, we will apply Lemma 4.10 to prove that with high probability $|Y| \leq \widetilde{O}\left(\beta^{-1}\right)$. The reason we can apply the lemma is that for any $e = (u,w) \in Y$, we have

$$R_{eff}(e,v) = \frac{1}{c_1 c_2} R_{eff}(C,v) \leq \frac{1}{2} R_{eff}(C,v),$$

and so $Y \subseteq N_E(v, R_{eff}(C,v)/2)$. Therefore, we get that

$$\sum_{e=(u,w)\in Y} \left| \pi_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{q_e}{\sqrt{r_e}} \mathbf{1}_e) \right| \sqrt{R_{eff}(C,v)} \leq c^{-1} \cdot \widetilde{O}\left(\beta^{-1}\right) .$$

Overall, we conclude that

$$\sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}}) - \boldsymbol{\pi}^{C}(\boldsymbol{B}^\top \frac{\boldsymbol{q}}{\sqrt{\boldsymbol{r}}})\right)} \leq (c + c^{-1})\widetilde{O}\left(\beta^{-2}\right) = \widetilde{O}\left(\beta^{-2}\right) ,$$

by setting c to be a large enough constant. □

D.5 Proof of Lemma 4.12

Proof. We write

$$\begin{aligned}
&\boldsymbol{\pi}^{C^T,\boldsymbol{r}^T}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^T}{\sqrt{\boldsymbol{r}^T}}\right) - \boldsymbol{\pi}^{C^0,\boldsymbol{r}^0}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^0}{\sqrt{\boldsymbol{r}^0}}\right) \\
&= \underbrace{\sum_{i \text{ is an } \textsc{AddTerminal}} \pi_{v^i}^{C^{i+1},\boldsymbol{r}^i}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}^i}}\right) \cdot \left(\mathbf{1}_{v^i} - \boldsymbol{\pi}^{C^i,\boldsymbol{r}^i}\left(\mathbf{1}_{v^i}\right)\right)}_{\boldsymbol{d}_{Add}} \\
&+ \underbrace{\sum_{i \text{ is an } \textsc{Update}} \boldsymbol{\pi}^{C^i,\boldsymbol{r}^i}\left(\boldsymbol{B}^\top \left(\frac{\boldsymbol{q}^{i+1}}{\sqrt{\boldsymbol{r}^{i+1}}} - \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}^i}}\right)\mathbf{1}_{e^i}\right)}_{\boldsymbol{d}_{Upd}},
\end{aligned}$$

which implies that

$$\sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{\pi}^{C^T,\boldsymbol{r}^T}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^T}{\sqrt{\boldsymbol{r}^T}}\right) - \boldsymbol{\pi}^{C^0,\boldsymbol{r}^0}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^0}{\sqrt{\boldsymbol{r}^0}}\right)\right)} \leq \sqrt{\mathcal{E}_{\boldsymbol{r}^T}(\boldsymbol{d}_{Add})} + \sqrt{\mathcal{E}_{\boldsymbol{r}^T}(\boldsymbol{d}_{Upd})}\,.$$

We bound each of these terms separately. For the second one, we have that

$$\begin{aligned}
&\sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{d}_{Upd}\right)} \\
&\leq \sum_{i \text{ is an } \textsc{Update}} \sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{\pi}^{C^i,\boldsymbol{r}^i}\left(\boldsymbol{B}^\top \left(\frac{\boldsymbol{q}^{i+1}}{\sqrt{\boldsymbol{r}^{i+1}}} - \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}^i}}\right)\mathbf{1}_{e^i}\right)\right)} \\
&= \sum_{i \text{ is an } \textsc{Update}} \sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{B}^\top \left(\frac{\boldsymbol{q}^{i+1}}{\sqrt{\boldsymbol{r}^{i+1}}} - \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}^i}}\right)\mathbf{1}_{e^i}\right)} \\
&\leq \sum_{i \text{ is an } \textsc{Update}} \left(\sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^{i+1}}{\sqrt{\boldsymbol{r}^{i+1}}}\mathbf{1}_{e^i}\right)} + \sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}^i}}\mathbf{1}_{e^i}\right)}\right) \\
&\leq \max_i \left\|\frac{\boldsymbol{r}^T}{\boldsymbol{r}^i}\right\|_\infty^{1/2} \sum_{i \text{ is an } \textsc{Update}} \left(\sqrt{\mathcal{E}_{\boldsymbol{r}^{i+1}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^{i+1}}{\sqrt{\boldsymbol{r}^{i+1}}}\mathbf{1}_{e^i}\right)} + \sqrt{\mathcal{E}_{\boldsymbol{r}^i}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}^i}}\mathbf{1}_{e^i}\right)}\right) \\
&\leq 2\max_i \left\|\frac{\boldsymbol{r}^T}{\boldsymbol{r}^i}\right\|_\infty^{1/2} T\,.
\end{aligned}$$

For the first one, we have

$$
\begin{aligned}
&\sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\boldsymbol{d}_{Add}\right)} \\
&\leq \sum_{i \text{ is an } \textsc{AddTerminal}} \left|\pi_{v^i}^{C^{i+1},\boldsymbol{r}^i}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}^i}}\right)\right| \cdot \sqrt{\mathcal{E}_{\boldsymbol{r}^T}\left(\mathbf{1}_{v^i} - \boldsymbol{\pi}^{C^i,\boldsymbol{r}^i}\left(\mathbf{1}_{v^i}\right)\right)} \\
&\leq \left\|\frac{\boldsymbol{r}^T}{\boldsymbol{r}^i}\right\|_\infty^{1/2} \sum_{i \text{ is an } \textsc{AddTerminal}} \left|\pi_{v^i}^{C^{i+1},\boldsymbol{r}^i}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}^i}{\sqrt{\boldsymbol{r}^i}}\right)\right| \cdot \sqrt{\mathcal{E}_{\boldsymbol{r}^i}\left(\mathbf{1}_{v^i} - \boldsymbol{\pi}^{C^i,\boldsymbol{r}^i}\left(\mathbf{1}_{v^i}\right)\right)} \\
&\leq \widetilde{O}\left(\max_i \left\|\frac{\boldsymbol{r}^T}{\boldsymbol{r}^i}\right\|_\infty^{1/2} \beta^{-2}\right) \cdot T\,,
\end{aligned}
$$

where in the last inequality we used Lemma 4.11. The desired statement now follows immediately. □

D.6 Proof of Lemma 3.8

Proof. **Initialize**($\boldsymbol{f}$): We set $\boldsymbol{s}^+ = \boldsymbol{u} - \boldsymbol{f}$, $\boldsymbol{s}^- = \boldsymbol{f}$, $\boldsymbol{r}^0 = \boldsymbol{r} = \frac{1}{(\boldsymbol{s}^+)^2} + \frac{1}{(\boldsymbol{s}^-)^2}$.

We first initialize a β-congestion reduction subset C based on Lemma 4.2, which takes time $\widetilde{O}\left(m\beta^{-2}\right)$, and a data structure DynamicSC for maintaining the sparsified Schur Complement onto C, as described in Appendix A, which takes time $\widetilde{O}\left(m\beta^{-4}\varepsilon^{-4}\right)$. We also set $C^0 = C$.

Then, we generate an $\widetilde{O}\left(\varepsilon^{-2}\right) \times m$ sketching matrix $\boldsymbol{Q}$ as in (Lemma 5.1, [GLP21] v2), which takes time $\widetilde{O}\left(m\varepsilon^{-2}\right)$, and let its rows be $\boldsymbol{q}^i$ for $i \in [\widetilde{O}\left(\varepsilon^{-2}\right)]$.

In order to compute the set of important edges, we use Lemma B.2 after contracting C, which shows that we can compute all resistances of the form $R_{eff}(C, u)$ for $u \in V \backslash C$ up to a factor of 2 in $\widetilde{O}(m)$. From these, we can get 4-approximate estimates of $R_{eff}(C, e)$ for $e \in E \backslash E(C)$, using the fact that

$$
\min\{R_{eff}(C, u), R_{eff}(C, w)\} \approx_2 R_{eff}(C, e)\,.
$$

Then, in $O(m)$ time, we can easily compute a set of edges S such that

$$
\{e \mid e \text{ is } \frac{\varepsilon\beta}{\alpha}\text{-important }\} \subseteq S \subseteq \{e \mid e \text{ is } \frac{\varepsilon\beta}{4\alpha}\text{-important}\}
$$

We also need to sample the random walks that will be used inside the demand projection data structures. We use (Lemma 5.15, [GLP21] v2) to sample $h = \widetilde{O}\left(\widehat{\varepsilon}^{-4}\beta^{-6} + \widehat{\varepsilon}^{-2}\beta^{-2}\gamma^{-2}\right)$ random walks for each $u \in V \backslash C$ and $e \in E \backslash E(C)$ with $u \in e$, where we set $\gamma = \frac{\varepsilon}{4\alpha}$ so that S is a subset of γ-important edges. Note that, by Definition 3.7, a γ-important edge will always remain γ-important until the Locator is re-initialized, as any edge's resistive distance to C can only decrease, and its own resistance is constant. Therefore S can be assumed to always be a subset of γ-important edges.

The runtime to sample the set $\mathcal{P}$ of these random walks is

$$
\widetilde{O}\left(mh\beta^{-2}\right) = \widetilde{O}\left(m(\widehat{\varepsilon}^{-4}\beta^{-8} + \widehat{\varepsilon}^{-2}\beta^{-4}\gamma^{-2})\right) = \widetilde{O}\left(m(\widehat{\varepsilon}^{-4}\beta^{-8} + \widehat{\varepsilon}^{-2}\varepsilon^{-2}\alpha^2\beta^{-4})\right)\,.
$$

In order to be able to detect congested edges, we will initialize $\widetilde{O}\left(\varepsilon^{-2}\right)$ demand projection data structures, with the guarantees from Lemma 4.14. We will maintain an approximation to $\boldsymbol{\pi}^C(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}})$ for all $i \in \widetilde{O}\left(\varepsilon^{-2}\right)$, where $\boldsymbol{q}^i$ are the rows of the sketching matrix that we have generated, as well as $\boldsymbol{\pi}^{old} := \boldsymbol{\pi}^{C^0,\boldsymbol{r}^0}\left(\boldsymbol{B}^\top \frac{\boldsymbol{p}^0}{\sqrt{\boldsymbol{r}^0}}\right)$, where $\boldsymbol{p}^0 = \sqrt{\boldsymbol{r}^0} g(\boldsymbol{s}^0) = \frac{\frac{1}{\boldsymbol{s}^{+,0}} - \frac{1}{\boldsymbol{s}^{-,0}}}{\sqrt{\boldsymbol{r}^0}}$.

Specifically, we call

$$\mathrm{DP}^i.\textsc{Initialize}(C, \boldsymbol{r}, \boldsymbol{q}^i, S, \mathcal{P})$$

for all $i \in [\widetilde{O}\left(\varepsilon^{-2}\right)]$, and also exactly compute $\boldsymbol{\pi}^{old}$, which can be done by calling

$$\textsc{DemandProjector}.\textsc{Initialize}(C, \boldsymbol{r}, \boldsymbol{p}, [m], \mathcal{P})\,.$$

The total runtime for this operation is dominated by the random walk generation, and is

$$\widetilde{O}\left(m(\widehat{\varepsilon}^{-4}\beta^{-8} + \widehat{\varepsilon}^{-2}\varepsilon^{-2}\alpha^2\beta^{-4})\right)\,.$$

Update$(e, \boldsymbol{f})$: We set $s_e^+ = u_e - f_e$, $s_e^- = f_e$, and $r_e = \frac{1}{(s_e^+)^2} + \frac{1}{(s_e^-)^2}$. Then, we also set $p_e = \frac{\frac{1}{s_e^+} - \frac{1}{s_e^-}}{\sqrt{r_e}}$. We distinguish two cases:

- $e \in E(C)$:

  In this case, we can simply call

  $$\textsc{DynamicSC}.\textsc{Update}(e, r_e)$$

  and

  $$\mathrm{DP}^i.\textsc{Update}(e, \boldsymbol{r}, \boldsymbol{q})$$

  for all $i \in [\widetilde{O}\left(\varepsilon^{-2}\right)]$. Note that we can do this as DP^i was initialized with resistances $\boldsymbol{r}^0$ and $\boldsymbol{r}^0 \approx_\alpha \boldsymbol{r}$.

- $e \in E \backslash E(C)$:

  We let $e = (u, w)$. We want to insert u and w into C, but for doing that DP^i's require constant factor estimates of the resistances $R_{eff}(C, u)$ and $R_{eff}(C \cup \{u\}, w)$. In order to get these estimates, we will use DynamicSC.

  We first call

  $$\textsc{DynamicSC}.\textsc{AddTerminal}(u)\,,$$

  which takes time $\widetilde{O}\left(\beta^{-2}\varepsilon^{-2}\right)$ and returns $\widetilde{R}_{eff}(C, u) \approx_2 R_{eff}(C, u)$. Given this estimate, we can call

  $$\mathrm{DP}^i.\textsc{AddTerminal}(u, \widetilde{R}_{eff}(C, u))\,,$$

  for all $i \in [\widetilde{O}\left(\varepsilon^{-2}\right)]$, each of which takes time

  $$\widetilde{O}\left(\widehat{\varepsilon}^{-4}\beta^{-8} + \widehat{\varepsilon}^{-2}\beta^{-6}\gamma^{-2}\right) = \widetilde{O}\left(\widehat{\varepsilon}^{-4}\beta^{-8} + \widehat{\varepsilon}^{-2}\varepsilon^{-2}\alpha^2\beta^{-6}\right)\,.$$

  Now, we can set $C = C \cup \{u\}$ and repeat the same process for w.

  Finally, to update the resistance, note that we now have $e \in E(C)$, so we apply the procedure from the first case.

Finally, if the total number of calls to $\mathrm{DP}^i.\textsc{AddTerminal}$ for some fixed i since the last call to $\mathcal{L}.\textsc{BatchUpdate}(\emptyset)$ exceeds $\frac{\varepsilon}{\widehat{\varepsilon}\alpha^{1/2}}$ (note that the number of calls is actually the same for all i), we call $\mathcal{L}.\textsc{BatchUpdate}(\emptyset)$ in order to re-initialize the demand projections.

We conclude that the total runtime is

$$\widetilde{O}\left(m\frac{\widehat{\varepsilon}\alpha^{1/2}}{\varepsilon^3}+\widehat{\varepsilon}^{-4}\varepsilon^{-2}\beta^{-8}+\widehat{\varepsilon}^{-2}\varepsilon^{-4}\alpha^2\beta^{-6}\right),$$

where the first term comes from amortizing the calls to $\mathcal{L}$.BatchUpdate($\emptyset$), each of which, as we will see, takes $\widetilde{O}\left(m\varepsilon^{-2}\right)$.
BatchUpdate$(Z,\boldsymbol{f})$: First, for each $e\in Z$, we set $s_e^+=u_e-f_e$, $s_e^-=f_e$, $r_e^0=r_e=\frac{1}{(s_e^+)^2}+\frac{1}{(s_e^-)^2}$, and $p_e^0=p_e=\frac{\frac{1}{s_e^+}-\frac{1}{s_e^-}}{\sqrt{r_e}}$.

For each $e=(u,w)\in Z$, we call

$$\text{DynamicSC.AddTerminal}(u)$$

and

$$\text{DynamicSC.AddTerminal}(w)$$

(if u and w are not already in C), and then we call

$$\text{DynamicSC.Update}(e,r_e)\,.$$

Then, we set $C^0=C=C\cup(\cup_{(u,w)\in Z}\{u,w\})$. Additionally, we re-compute $\boldsymbol{\pi}^{old}$ based on the new values of $C^0,\boldsymbol{r}^0,\boldsymbol{p}^0$. All of this takes time $\widetilde{O}\left(m+|Z|\beta^{-2}\varepsilon^{-2}\right)$.

Now, to pass these updates to the DemandProjectors, we first have to re-compute the set of important edges S (with the newly updated resistances) as any set such that

$$\{e\mid e\text{ is }\frac{\varepsilon}{\alpha}\text{-important }\}\subseteq S\subseteq\{e\mid e\text{ is }\frac{\varepsilon}{4\alpha}\text{-important}\}\,.$$

As we have already argued, this takes $\widetilde{O}(m)$.

Now, finally, we re-initialize all the DemandProjectors by calling

$$\text{DP}^i.\text{Initialize}(C,\boldsymbol{r},\boldsymbol{q},S,\mathcal{P})\,.$$

for all $i\in[\widetilde{O}\left(\varepsilon^{-2}\right)]$, where each call takes $\widetilde{O}(m)$.

We conclude with a total runtime of

$$\widetilde{O}\left(m\varepsilon^{-2}+|Z|\beta^{-2}\varepsilon^{-2}\right)\,.$$

Solve(): This operation performs the main task of the locator, which is to detect congested edges. We will do that by using the approximate demand projections that we have been maintaining.

We remind that the congestion vector we are trying to approximate to $O(\varepsilon)$ additive accuracy is

$$\boldsymbol{\rho}^*=\delta\sqrt{\boldsymbol{r}}g(\boldsymbol{s})-\delta\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{B}^\top g(\boldsymbol{s})\,.$$

We will first reduce the problem of finding the entries of $\boldsymbol{\rho}^*$ with magnitude $\geq\Omega(\varepsilon)$, to the problem of computing an $O(\varepsilon)$-additive approximation to

$$v_i^*=\delta\cdot\left\langle\boldsymbol{\pi}^C\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}}\right),\widetilde{SC}^+\boldsymbol{\pi}^{C^0,\boldsymbol{r}^0}\left(\boldsymbol{B}^\top\frac{\boldsymbol{p}^0}{\sqrt{\boldsymbol{r}^0}}\right)\right\rangle$$

for all $i \in [\widetilde{O}\left(\varepsilon^{-2}\right)]$, where $\widetilde{SC}$ is the approximate Schur complement maintained in DynamicSC. Then, we will see how to approximate v_i^* to additive accuracy $O(\varepsilon)$ using the demand projection data structures.

First of all, note that, by definition of $g(\boldsymbol{s}) = \frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-}}{\boldsymbol{r}}$,

$$\left\|\delta\sqrt{\boldsymbol{r}}g(\boldsymbol{s})\right\|_\infty \leq \delta \leq \varepsilon\,,$$

so this term can be ignored. Using Lemma 4.3, we get that

$$\delta\left\|\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+(\boldsymbol{B}^\top g(\boldsymbol{s}) - \boldsymbol{\pi}^C(\boldsymbol{B}^\top g(\boldsymbol{s})))\right\|_\infty \leq \delta \cdot \widetilde{O}\left(\beta^{-2}\right) \leq \varepsilon/2\,.$$

This means that the entries of the vector

$$\delta\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{\pi}^C(\boldsymbol{B}^\top g(\boldsymbol{s}))$$

that have magnitude $\leq \varepsilon$ do not correspond to the $\Omega(\varepsilon)$-congested edges that we are looking for.

Now, we set $T = |C \backslash C^0|$, where C^0 was the congestion reduction subset during the last call to BatchUpdate, and apply Lemma 4.12. This shows that

$$\sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{old} - \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right)\right)} \leq \widetilde{O}\left(\alpha^{1/2}\beta^{-2}\right) \cdot T\,.$$

Therefore, if we define

$$\boldsymbol{\rho} = -\delta\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{\pi}^{old}\,,$$

we conclude that

$$\|\boldsymbol{\rho} - \boldsymbol{\rho}^*\|_\infty \leq O(\varepsilon) + \delta\left\|\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\left(\boldsymbol{\pi}^{old} - \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right)\right)\right\|_\infty \leq O(\varepsilon) + \delta T \cdot \widetilde{O}\left(\alpha^{1/2}\beta^{-2}\right) \leq O(\varepsilon)\,,$$

where we used the fact that $T = \frac{\varepsilon}{\widetilde{\varepsilon}\alpha^{1/2}} \leq \frac{\varepsilon}{\delta\beta^{-2}\alpha^{1/2}}$. Therefore it suffices to estimate $\boldsymbol{\rho}$ up to $O(\varepsilon)$-additive accuracy.

Now, note that, by definition, no edge $e \in E \backslash S$ is ε/α-important with respect to $\boldsymbol{r}^0$ and C^0. By using Lemma 4.5, for each such edge we get

$$\begin{aligned}
&\delta\left|\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{\pi}^{old}\right|_e \\
&\leq \delta\sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{C^0}\left(\boldsymbol{B}^\top\frac{\boldsymbol{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right)}\sqrt{\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{\pi}^{old})} \\
&\leq \delta\alpha\sqrt{\mathcal{E}_{\boldsymbol{r}^0}\left(\boldsymbol{\pi}^{C^0}\left(\boldsymbol{B}^\top\frac{\boldsymbol{1}_e}{\sqrt{\boldsymbol{r}}}\right)\right)}\sqrt{\mathcal{E}_{\boldsymbol{r}^0}(\boldsymbol{\pi}^{old})} \\
&\leq \delta\alpha \cdot \frac{\varepsilon}{\alpha} \cdot O(\sqrt{m}) \\
&= O(\varepsilon)\,,
\end{aligned}$$

where we also used the fact that $\mathcal{E}_{\boldsymbol{r}^0}(\boldsymbol{\pi}^{C^0,\boldsymbol{r}^0}(g(\boldsymbol{s}^0))) \leq O(\mathcal{E}_{\boldsymbol{r}^0}(g(\boldsymbol{s})))$. Therefore it suffices to approximate

$$\boldsymbol{\rho}' = \delta\boldsymbol{I}_S\boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{\pi}^{old}\,.$$

Note that here we can replace $\boldsymbol{L}^{+}$ by $\begin{pmatrix} -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} \\ \boldsymbol{I} \end{pmatrix} \widetilde{SC}^{+}$ where $\widetilde{SC} \approx_{1+\varepsilon} SC$ and only lose another additive ε error, as

$$\begin{aligned}
&\delta \left| \left\langle \boldsymbol{1}_e, \boldsymbol{R}^{-1/2} \boldsymbol{B} \boldsymbol{L}^{+} \boldsymbol{\pi}^{old} \right\rangle - \delta \left\langle \boldsymbol{1}_e, \boldsymbol{R}^{-1/2} \boldsymbol{B} \begin{pmatrix} -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} \\ \boldsymbol{I} \end{pmatrix} \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle \right| \\
&= \delta \left| \left\langle \boldsymbol{\pi}^{C} \left(\boldsymbol{B}^{\top} \frac{\boldsymbol{1}_e}{\sqrt{\boldsymbol{r}}} \right), \left(SC^{+} - \widetilde{SC}^{+} \right) \boldsymbol{\pi}^{old} \right\rangle \right| \\
&\leq O(\delta \varepsilon \cdot \sqrt{m}) \\
&\leq O(\varepsilon) \,,
\end{aligned}$$

where we used the fact that

$$(1-\varepsilon) SC \preceq \widetilde{SC} \preceq (1+\varepsilon) SC \Rightarrow -O(\varepsilon)\widetilde{SC}^{+} \preceq SC^{+} - \widetilde{SC}^{+} \preceq O(\varepsilon)\widetilde{SC}^{+} \,.$$

and that

$$\begin{aligned}
\sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{old}\right)} &\leq \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{C}\left(\boldsymbol{B}^{\top} g(\boldsymbol{s})\right)\right)} + \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{old} - \boldsymbol{\pi}^{C}\left(\boldsymbol{B}^{\top} g(\boldsymbol{s})\right)\right)} \\
&\leq O(m) + \widetilde{O}\left(\alpha^{1/2}\beta^{-2}\right) \cdot T \\
&\leq O(m) \,,
\end{aligned}$$

where we used the fact that $T = \frac{\varepsilon}{\widehat{\varepsilon}\alpha^{1/2}} \leq \frac{\varepsilon}{\delta\beta^{-2}\alpha^{1/2}} \leq \frac{\sqrt{m}}{\beta^{-2}\alpha^{1/2}}$.

Now, we will use the sketching lemma (Lemma 5.1, [GLP21] v2), which shows that in order to find all entries of

$$\boldsymbol{I}_S \boldsymbol{R}^{-1/2} \boldsymbol{B} \begin{pmatrix} -\boldsymbol{L}_{FF}\boldsymbol{L}_{FC} \\ \boldsymbol{I} \end{pmatrix} \widetilde{SC}^{+} \boldsymbol{\pi}^{old}$$

with magnitude $\Omega(\varepsilon)$, it suffices to compute the inner products

$$\begin{aligned}
&\delta \left\langle \boldsymbol{B}^{\top} \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}}, \begin{pmatrix} -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} \\ \boldsymbol{I} \end{pmatrix} \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle \\
&= \delta \left\langle \boldsymbol{\pi}^{C} \left(\boldsymbol{B}^{\top} \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}} \right), \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle
\end{aligned}$$

for $i \in [\widetilde{O}\left(\varepsilon^{-2}\right)]$, up to additive accuracy

$$\varepsilon \cdot \left\| \delta \boldsymbol{I}_S \boldsymbol{R}^{-1/2} \boldsymbol{B} \begin{pmatrix} -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} \\ \boldsymbol{I} \end{pmatrix} \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\|_2^{-1} \geq \Omega(\varepsilon) \,,$$

where we used the fact that

$$\begin{aligned}
&\left\|\delta \boldsymbol{I}_S \boldsymbol{R}^{-1/2} \boldsymbol{B} \begin{pmatrix} -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} \\ \boldsymbol{I} \end{pmatrix} \widetilde{SC}^{+} \boldsymbol{\pi}^{old}\right\|_2^2 \\
&= \delta^2 \langle \widetilde{SC}^{+} \boldsymbol{\pi}^{old}, \begin{pmatrix} -\boldsymbol{L}_{CF}\boldsymbol{L}_{FF}^{-1} & \boldsymbol{I} \end{pmatrix} \boldsymbol{L} \begin{pmatrix} -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} \\ \boldsymbol{I} \end{pmatrix} \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \rangle \\
&= \delta^2 \left\langle \widetilde{SC}^{+} \boldsymbol{\pi}^{old}, \begin{pmatrix} -\boldsymbol{L}_{CF}\boldsymbol{L}_{FF}^{-1} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{L}_{FF} & \boldsymbol{L}_{FC} \\ \boldsymbol{L}_{CF} & \boldsymbol{L}_{CC} \end{pmatrix} \begin{pmatrix} -\boldsymbol{L}_{FF}^{-1}\boldsymbol{L}_{FC} \\ \boldsymbol{I} \end{pmatrix} \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle \\
&= \delta^2 \left\langle \widetilde{SC}^{+} \boldsymbol{\pi}^{old}, SC\widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle \\
&\le 2\delta^2 \left\langle \boldsymbol{\pi}^{old}, \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle \\
&\le O(\delta^2 m) \\
&= O(1)\,.
\end{aligned}$$

Now, for the second part of the proof, we would like to compute $\boldsymbol{v}$ such that $\left\|\boldsymbol{v} - \boldsymbol{v}^*\right\|_\infty \le O(\varepsilon)$, where we remind that

$$\boldsymbol{v}^* = \left\langle \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}} \right), \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle .$$

Note that we already have estimates $\widetilde{\boldsymbol{\pi}}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}} \right)$ given by DP^i for all $i \in [\widetilde{O}\left(\varepsilon^{-2}\right)]$. We obtain these estimates by calling

$$\mathrm{DP}^i.\textsc{Output}()$$

each of which takes time $O(\beta m)$. By the guarantees of Definition 4.13, with high probability we have

$$\delta \left| \left\langle \widetilde{\boldsymbol{\pi}}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}} \right) - \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}} \right), \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle \right| \le \widehat{\varepsilon} \sqrt{\alpha} T\,,$$

where we used the fact that

$$E_{\boldsymbol{r}}(\delta \cdot \widetilde{SC}^{+} \boldsymbol{\pi}^{old}) \le O(1)\,.$$

Now, since by definition $\textsc{BatchUpdate}$ is called every $\frac{\varepsilon}{\widehat{\varepsilon}\alpha^{1/2}}$ calls to $\textsc{Update}$, We have $T \le \frac{\varepsilon}{\widehat{\varepsilon}\alpha^{1/2}}$ and so

$$\delta \left| \left\langle \widetilde{\boldsymbol{\pi}}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}} \right) - \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}} \right), \widetilde{SC}^{+} \boldsymbol{\pi}^{old} \right\rangle \right| \le \varepsilon\,.$$

This means that, running the algorithm from (Lemma 5.1, [GLP21] v2), we can obtain an edge set of size $\widetilde{O}\left(\varepsilon^{-2}\right)$ that contains all edges such that $|\rho_e^*| \ge c \cdot \varepsilon$ for some constant $c > 0$. By rescaling ε to get the right constant, we obtain all edges such that $|\rho_e^*| \ge \varepsilon/2$ with high probability. The runtime is dominated by the time to get $\widetilde{SC}$ and apply its inverse, and is $\widetilde{O}\left(\beta m \varepsilon^{-2}\right)$.

Success probability We will argue that $\mathcal{L}$ uses DynamicSC and the DP^i as an oblivious adversary. First of all, note that no randomness is injected into the inputs of DynamicSC, as they are all coming form the inputs of $\mathcal{L}$.

Regarding DP^i, note that its only output is given by the call to DP^i.OUTPUT. However, note that its output is only used to estimate the inner product

$$\left\langle \boldsymbol{\pi}^C \left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S^i}{\sqrt{\boldsymbol{r}}} \right), \widetilde{SC}^+ \boldsymbol{\pi}_{old} \right\rangle ,$$

from which we obtain the set of congested edges and we directly return it from $\mathcal{L}$. Thus, it does not influence the state of $\mathcal{L}$, DYNAMICSC or any future inputs.

□

E Deferred Proofs from Section 5

E.1 Proof of Lemma 4.6

Proof. Let $\mathcal{P}_v(u)$ be a random walk that starts from u and stops when it hits v.

$$\begin{aligned} p_v^{C \cup \{v,w\}}(u) &= \Pr\left[\mathcal{P}_v(u) \cap C = \emptyset \text{ and } w \notin \mathcal{P}_v(u)\right] \\ &= \Pr\left[\mathcal{P}_v(u) \cap C = \emptyset\right] \cdot \Pr\left[w \notin \mathcal{P}_v(u) \mid \mathcal{P}_v(u) \cap C = \emptyset\right] \\ &= p_v^{C \cup \{v\}}(u) \cdot \Pr\left[w \notin \mathcal{P}_v(u) \mid \mathcal{P}_v(u) \cap C = \emptyset\right] \end{aligned}$$

Consider new resistances $\widehat{r}$, where $\widehat{r}_e = r_e$ for all $e \in E$ not incident to C and $\widehat{r}_e = \infty$ for all $e \in E$ incident to C. Also, let $\widehat{p}$ be the hitting probability function for these new resistances. It is easy to see that

$$\Pr\left[w \notin \mathcal{P}_v(u) \mid \mathcal{P}_v(u) \cap C = \emptyset\right] = \widehat{p}_v^{\{v,w\}}(u) .$$

Therefore, we have

$$p_v^{C \cup \{v,w\}}(u) = p_v^{C \cup \{v\}}(u) \cdot \widehat{p}_v^{\{v,w\}}(u) .$$

Now we will bound $\widehat{p}_v^{\{v,w\}}(u)$. Let ψ be electrical potentials for pushing 1 unit of flow from v to w with resistances $\widehat{r}$ and let f be the associated electrical flow. We have that

$$|\psi_u - \psi_w| = |f_e| \widehat{r}_e \le \widehat{r}_e = r_e \tag{22}$$

(because $|f_e| \le 1$ and e is not incident to C) and

$$|\psi_v - \psi_w| = \widehat{R}_{eff}(v,w) \ge R_{eff}(v,w) \tag{23}$$

Additionally, by well known facts that connect electrical potential embeddings with random walks, we have that

$$\psi_u = \psi_w + \widehat{p}_v^{\{v,w\}}(u)(\psi_v - \psi_w) ,$$

or equivalently

$$\widehat{p}_v^{\{v,w\}}(u) = \frac{\psi_u - \psi_w}{\psi_v - \psi_w} .$$

Using (22) and (23), this immediately implies that

$$\widehat{p}_v^{\{v,w\}}(u) \le \frac{r_e}{R_{eff}(v,w)} .$$

So we have proved that

$$p_v^{C\cup\{v,w\}}(u) \le p_v^{C\cup\{v\}}(u)\frac{r_e}{R_{eff}(v,w)}$$

and symmetrically

$$p_v^{C\cup\{v,u\}}(w) \le p_v^{C\cup\{v\}}(w)\frac{r_e}{R_{eff}(v,u)}\,.$$

Now, let's look at $\pi_v^{C\cup\{v\}}(B^\top \mathbf{1}_e) = p_v^{C\cup\{v\}}(u) - p_v^{C\cup\{v\}}(w)$. Note that

$$p_v^{C\cup\{v\}}(u) = p_v^{C\cup\{v,w\}}(u) + p_w^{C\cup\{v,w\}}(u)p_v^{C\cup\{v\}}(w)$$

which we re-write as

$$p_v^{C\cup\{v\}}(u) - p_v^{C\cup\{v\}}(w) = p_v^{C\cup\{v,w\}}(u) - (1-p_w^{C\cup\{v,w\}}(u))p_v^{C\cup\{v\}}(w) \le p_v^{C\cup\{v,w\}}(u)\,.$$

Symmetrically,

$$p_v^{C\cup\{v\}}(w) - p_v^{C\cup\{v\}}(u) \le p_v^{C\cup\{v,u\}}(w)\,.$$

From these we conclude that

$$\begin{aligned}
\left|\pi_v^{C\cup\{v\}}(B^\top \mathbf{1}_e)\right| &= \left|p_v^{C\cup\{v\}}(u) - p_v^{C\cup\{v\}}(w)\right| \\
&\le \max\left\{p_v^{C\cup\{v,w\}}(u), p_v^{C\cup\{v,u\}}(w)\right\} \\
&\le \max\left\{p_v^{C\cup\{v\}}(u)\cdot\frac{r_e}{R_{eff}(v,w)}, p_v^{C\cup\{v\}}(w)\cdot\frac{r_e}{R_{eff}(v,u)}\right\} \\
&\le (p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w))\cdot\max\left\{\frac{r_e}{R_{eff}(v,w)}, \frac{r_e}{R_{eff}(v,u)}\right\}\,,
\end{aligned}$$

which, after dividing by $\sqrt{r_e}$ gives

$$\left|\pi_v^{C\cup\{v\}}(B^\top \mathbf{1}_e)\right| \le (p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w))\cdot\max\left\{\frac{\sqrt{r_e}}{R_{eff}(v,w)}, \frac{\sqrt{r_e}}{R_{eff}(v,u)}\right\}\,.$$

□

E.2 Proof of Lemma 5.3

Proof. For each $u \in V\backslash C$ and $e \in S'$ with $u \in e$, we generate Z random walks $P^1(u), \dots, P^Z(u)$ from u to $C \cup \{v\}$. We set

$$\begin{aligned}
\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right) &= \sum_{e=(u,w)\in S'}\sum_{z=1}^{Z}\frac{1}{Z}\frac{q_e}{\sqrt{r_e}}\left(1_{\{v\in P^z(u)\}} - 1_{\{v\in P^z(w)\}}\right) \\
&= \sum_{e=(u,w)\in S'}\sum_{z=1}^{Z}(X_{e,u,z} - X_{e,w,z})\,,
\end{aligned}$$

where we have set $X_{e,u,z} = \frac{1}{Z}\frac{q_e}{\sqrt{r_e}}1_{\{v\in P^z(u)\}}$ and $X_{e,w,z} = -\frac{1}{Z}\frac{q_e}{\sqrt{r_e}}1_{\{v\in P^z(w)\}}$.

Note that $\mathop{\mathbb{E}}_{P^z(u)}[X_{e,u,z}] = \frac{1}{Z}\frac{q_e}{\sqrt{r_e}}p_v^{C\cup\{v\}}(u)$ and $\mathop{\mathbb{E}}_{P^z(w)}[X_{e,w,z}] = -\frac{1}{Z}\frac{q_e}{\sqrt{r_e}}p_v^{C\cup\{v\}}(w)$. This implies that our estimate is unbiased, as

$$\mathbb{E}\left[\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right)\right] = \sum_{e=(u,w)\in S'}\frac{q_e}{\sqrt{r_e}}(p_v^{C\cup\{v\}}(u) - p_v^{C\cup\{v\}}(w)) = \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right).$$

We now need to show that our estimate is concentrated around the mean. To apply the concentration bound in Lemma 5.2, we need the following bounds:

$$\sum_{e=(u,w)\in S'}\sum_{z=1}^{Z}(|\mathbb{E}[X_{e,u,z}]| + |\mathbb{E}[X_{e,w,z}]|) = \sum_{e=(u,w)\in S'}\frac{|q_e|}{\sqrt{r_e}}(p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)) := E$$

$$\max_{\substack{e=(u,w)\in S'\\ z\in[Z]}}\max\{|X_{e,u,z}|,|X_{e,w,z}|\} \le \max_{e\in S'}\frac{1}{Z\sqrt{r_e}} := M.$$

So now for any $t\in[0,E]$ we have

$$\begin{aligned}
&\Pr\left[\left|\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C\cup\{v\}}(\boldsymbol{B}^\top\frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}})\right| > t\right]\\
&\le 2\exp\left(-\frac{t^2}{6EM}\right)\\
&= 2\exp\left(-\frac{Zt^2}{6\sum\limits_{e=(u,w)\in S'}\frac{|q_e|}{\sqrt{r_e}}(p_v^{C\cup\{v\}}(u)+p_v^{C\cup\{v\}}(w))\max_{e\in S'}\frac{1}{\sqrt{r_e}}}\right)\\
&\le 2\exp\left(-\frac{Zt^2}{6\sum\limits_{e=(u,w)\in S'}(p_v^{C\cup\{v\}}(u)+p_v^{C\cup\{v\}}(w))\max_{e\in S'}\frac{1}{r_e}}\right)\\
&\le 2\exp\left(-\frac{Zt^2c^2R_{eff}(C,v)}{6\sum\limits_{e=(u,w)\in S'}(p_v^{C\cup\{v\}}(u)+p_v^{C\cup\{v\}}(w))}\right)\\
&\le 2\exp\left(-Zt^2c^2R_{eff}(C,v)/\widetilde{O}\left(\beta^{-2}\right)\right)\\
&\le \frac{1}{n^{100}},
\end{aligned}$$

where the last inequality follows by setting $Z = \widetilde{O}\left(\frac{\log n\log\frac{1}{\beta}}{\delta_1'^2}\right)$ and $t = \frac{\delta_1'}{\beta c\sqrt{R_{eff}(C,v)}}$. Note that we have used the fact that $R_{eff}(C,v)\le r_e/c^2$ for all $e\in S'$, as well as the congestion reduction property (Definition 4.1)

$$\sum_{e=(u,w)\in E\setminus E(C)}(p_v^{C\cup\{v\}}(u)+p_v^{C\cup\{v\}}(w)) \le \widetilde{O}\left(1/\beta^2\right).$$

□

E.3 Proof of Lemma 5.4

Proof. In order to compute $\widetilde{\pi}_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}})$ we use Lemma 5.3 with demand $\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}$ and error parameter $\delta_1' > 0$, where

$$\{e \in S \mid R_{eff}(C,v) \le r_e/(2c^2)\} \subseteq S' \subseteq \{e \in S \mid R_{eff}(C,v) \le r_e/c^2\} ,$$

and $c > 0$ will be defined later. Note that such a set S' can be trivially computed given our effective resistance estimate $\widetilde{R}_{eff}(C,v) \approx_2 R_{eff}(C,v)$. However, algorithmically we do not directly compute S', but instead find its intersection with the edges from which a sampled random walk ends up at v. (Using the congestion reduction property of C, this can be done in $\widetilde{O}\left(\delta_1'^{-2}\beta^{-2}\log n \log\frac{1}{\beta}\right)$ time just by going through all random walks that contain v.)

Now, Lemma 5.3 guarantees that

$$\left|\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right)\right| \le \frac{\delta_1'}{\beta c\sqrt{R_{eff}(C,v)}}$$

given access to $O(\delta'^{-2}\log n \log\frac{1}{\beta})$ random walks for each $u \in V\backslash C$, $e \in S'$ with $u \in e$.

Then, we set $\widetilde{\pi}_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}) := \widetilde{\pi}_v^{C\cup\{v\}}(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}})$, and we have that

$$\begin{aligned}
&\left|\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\right| \\
&\le \left|\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right)\right| + \left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S - \boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right)\right| \\
&\le \frac{\delta_1'}{\beta c\sqrt{R_{eff}(C,v)}} + \left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S - \boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right)\right| .
\end{aligned} \tag{24}$$

Now, to bound the second term, we use Lemma 4.6, which gives

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S - \boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right)\right| \le \sum_{e=(u,w)\in S\backslash S'} \left(p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)\right)\frac{\sqrt{r_e}}{R_{eff}(v,e)} .$$

Now, note that for each $e \in S\backslash S'$, e is close to C, but v is far from C, so $R_{eff}(v,e)$ should be large. Specifically, by Lemma 2.12 we have $R_{eff}(v,e) \ge \frac{1}{2}\min\{R_{eff}(v,u), R_{eff}(v,w)\}$, and by the triangle inequality

$$\min\{R_{eff}(v,u), R_{eff}(v,w)\} \ge R_{eff}(C,v) - \max\{R_{eff}(C,u), R_{eff}(C,w)\} \ge R_{eff}(C,v) - 2R_{eff}(C,e) .$$

By the fact that e is γ-important and that

$$e \notin S' \supseteq \{e \in S \mid R_{eff}(C,v) \le r_e/(2c^2)\} ,$$

we have $R_{eff}(C,e) \le r_e/\gamma^2 \le 2c^2 R_{eff}(C,v)/\gamma^2$, so

$$\begin{aligned}
\frac{\sqrt{r_e}}{R_{eff}(v,e)} &\le \frac{1}{1/2 - 2c^2/\gamma^2}\frac{\sqrt{r_e}}{R_{eff}(C,v)} \\
&\le \frac{c}{1/2 - 2c^2/\gamma^2}\frac{1}{\sqrt{R_{eff}(C,v)}} .
\end{aligned}$$

By using the congestion reduction property (Definition 4.1), we obtain

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S - \boldsymbol{q}_{S'}}{\sqrt{\boldsymbol{r}}}\right)\right| \leq \frac{c}{1/2 - 2c^2/\gamma^2} \frac{1}{\sqrt{R_{eff}(C,v)}} \widetilde{O}\left(\frac{1}{\beta^2}\right). \tag{25}$$

Setting $c = \min\{\delta_1/\widetilde{O}\left(\beta^{-2}\right), \gamma/4\}$ and $\delta_1' = \beta c \cdot \delta_1/2$, (24) becomes

$$\left|\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\right| \leq \frac{\delta_1}{\sqrt{R_{eff}(C,v)}}.$$

Also, the number of random walks needed for each valid pair (u,e) is

$$\widetilde{O}\left(\delta_1'^{-2}\log n \log\frac{1}{\beta}\right) = \widetilde{O}\left(\delta_1^{-2}\beta^{-2}c^{-2}\log n\log\frac{1}{\beta}\right) = \widetilde{O}\left(\left(\delta_1^{-4}\beta^{-6} + \delta_1^{-2}\beta^{-2}\gamma^{-2}\right)\log n\log\frac{1}{\beta}\right)$$

.

For the last part of the lemma, we let $S'' = \{e \in S \mid R_{eff}(C,v) \leq r_e/\left(\gamma/4\right)^2\}$ and write

$$\begin{aligned}&\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\right| \\ &\leq \left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S''}}{\sqrt{\boldsymbol{r}}}\right)\right| + \left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S - \boldsymbol{q}_{S''}}{\sqrt{\boldsymbol{r}}}\right)\right|.\end{aligned}$$

For the first term,

$$\begin{aligned}\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_{S''}}{\sqrt{\boldsymbol{r}}}\right)\right| &\leq \sum_{e=(u,w)\in S''}\left(p_v^{C\cup\{v\}}(u) + p_v^{C\cup\{v\}}(w)\right)\frac{1}{\sqrt{r_e}} \\ &\leq \frac{1}{(\gamma/4)\sqrt{R_{eff}(C,v)}}\widetilde{O}\left(\frac{1}{\beta^2}\right),\end{aligned}$$

and for the second term we have already proved in (25) (after replacing c by $\gamma/4$) that

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S - \boldsymbol{q}_{S''}}{\sqrt{\boldsymbol{r}}}\right)\right| \leq \frac{\gamma/4}{\sqrt{R_{eff}(C,v)}}\widetilde{O}\left(\frac{1}{\beta^2}\right).$$

Putting these together, we conclude that

$$\left|\pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\right| \leq \frac{1}{\gamma\sqrt{R_{eff}(C,v)}} \cdot \widetilde{O}\left(\frac{1}{\beta^2}\right).$$

□

E.4 Proof of Lemma 5.5

Proof. For any $I \subseteq \mathbb{R}$, we define $F_I = \{i \in [n] \mid |\bar{\phi}_i| \in I\}$. For some $0 < a < b$ to be defined later, we partition $[n]$ as

$$[n] = F_{I_0} \cup F_{I_1} \cup \cdots \cup F_{I_K} \cup F_{I_{K+1}},$$

where $I_0 = [0, a)$, $I_{K+1} = [b, \infty)$, and $I_1, \ldots, I_K$ is a partition of $[a, b)$ into $K = O(\log\frac{b}{a})$ intervals such that for all $k \in [K]$ we have $\Phi_k := \max_{i\in F_k}|\bar{\phi}_i| \leq 2 \cdot \min_{i\in F_k}|\bar{\phi}_i|$.

A union bound gives

$$\Pr\left[\left|\langle\widetilde{\boldsymbol{\pi}}-\boldsymbol{\pi},\bar{\boldsymbol{\phi}}\rangle\right|>t\right]\leq\sum_{k=0}^{K+1}\Pr\left[\left|\sum_{i\in I_k}(\widetilde{\pi}_i-\pi_i)\bar{\phi}_i\right|>t/(K+2)\right].$$

We first examine I_0 and I_{K+1} separately. Note that

$$\left|\sum_{i\in F_{I_0}}(\widetilde{\pi}_i-\pi_i)\bar{\phi}_i\right|\leq\|\widetilde{\boldsymbol{\pi}}-\boldsymbol{\pi}\|_1 a\leq 2a$$

and

$$\left|\sum_{i\in F_{I_{K+1}}}(\widetilde{\pi}_i-\pi_i)\bar{\phi}_i\right|\leq\sum_{i\in F_{I_{K+1}}}\widetilde{\pi}_i\left|\bar{\phi}_i\right|+\sum_{i\in F_{I_{K+1}}}\pi_i\left|\bar{\phi}_i\right|\leq\frac{1}{b}\sum_{i\in F_{I_{K+1}}}|\widetilde{\pi}_i-\pi_i|\bar{\phi}_i^2\leq\frac{1}{b}\sum_{i\in F_{I_{K+1}}}\widetilde{\pi}_i|\bar{\phi}_i|+\frac{\|\bar{\phi}\|_{\boldsymbol{\pi},2}^2}{b}$$

But note that by picking $b\geq\max\left\{\frac{(K+2)Var_{\boldsymbol{\pi}}(\bar{\phi})}{t},\sqrt{Var_{\boldsymbol{\pi}}(\bar{\boldsymbol{\phi}})\cdot n^{101}}\right\}$, we have $\frac{Var_{\boldsymbol{\pi}}(\bar{\phi})}{b}\leq t/(K+2)$ and also for any $i\in F_{K+1}$ we have $\pi_i\leq\frac{Var_{\boldsymbol{\pi}}(\bar{\phi})}{b^2}\leq\frac{1}{n^{101}}$. This means that $\Pr[\widetilde{\pi}_i\neq 0]\leq\frac{1}{n^{101}}$, and so by union bound

$$\Pr\left[\sum_{i\in F_{I_{K+1}}}\widetilde{\pi}_i|\bar{\phi}_i|\neq 0\right]\leq\frac{1}{n^{100}}.$$

Now, we proceed to $F_1,\ldots,F_K$. We draw Z samples $x_1,\ldots,x_Z$ from $\boldsymbol{\pi}$. Then, we also define the following random variables for $z\in[Z]$ and $i\in[n]$:

$$X_{z,i}=\begin{cases}1 & \text{if } x_z=i\\ 0 & \text{otherwise}\end{cases}$$

for $i\in[n]$ and

$$Y_{z,k}=\frac{1}{Z}\sum_{i\in F_k}X_{z,i}\bar{\phi}_i$$

This allows us to write $\sum\limits_{i\in F_k}\widetilde{\pi}_i\bar{\phi}_i=\sum\limits_{z=1}^{Z}Y_{z,k}$.

Fix $k\in[K]$. We will apply Lemma 5.2 on the random variable $\sum\limits_{z=1}^{Z}Y_{z,k}$. We first compute

$$\sum_{z=1}^{Z}|\mathbb{E}[Y_{z,k}]|=\left|\sum_{i\in F_k}\pi_i\bar{\phi}_i\right|\leq\sum_{i\in F_k}\pi_i|\bar{\phi}_i|:=E_k$$

and

$$\max_{z\in[Z]}|Y_{z,k}|\leq\frac{\Phi_k}{Z}:=M_k.$$

Therefore we immediately have $E_k M_k \leq \frac{2}{Z} \sum_{i\in F_k} \pi_i \bar{\phi}_i^2 \leq \frac{2}{Z} \cdot Var_\pi(\bar{\phi})$. By Lemma 5.2,

$$\begin{aligned}
&\Pr\left[\left|\sum_{z=1}^{Z} Y_{z,k} - \mathbb{E}\left[\sum_{z=1}^{Z} Y_{z,k}\right]\right| > t/(K+2)\right] \\
&\leq 2\exp\left(-\frac{t^2}{6E_k M_k (K+2)^2}\right) \\
&\leq 2\exp\left(-\frac{Zt^2}{12\cdot Var_\pi(\bar{\phi})(K+2)^2}\right).
\end{aligned}$$

Summarizing, and using the fact that $K = \widetilde{O}\left(\log(n\cdot Var_{\boldsymbol{\pi}}(\bar{\boldsymbol{\phi}})/t^2)\right)$, we get

$$\Pr\left[\left|\langle\widetilde{\boldsymbol{\pi}} - \boldsymbol{\pi}, \bar{\boldsymbol{\phi}}\rangle\right| > t\right] \leq \widetilde{O}\left(\frac{1}{n^{100}}\right) + 2\widetilde{O}\left(\log\left(n\cdot Var_{\boldsymbol{\pi}}(\bar{\boldsymbol{\phi}})/t^2\right)\right)\exp\left(-\frac{Zt^2}{12\cdot\widetilde{O}\left(Var_{\boldsymbol{\pi}}(\bar{\boldsymbol{\phi}})\log^2 n\right)}\right).$$

□

E.5 Proof of Lemma 5.6

Proof. Let $S_0 = \emptyset$ and and for each $k \in \mathbb{N}$ let

$$S_k = \left\{i \in [n]\backslash S_{k-1} \ : \ \phi_i^2 \leq 2^{k+1} R_{eff}(C,v)\right\}.$$

Fix some $k \geq 2$. Note that $\phi_i^2 > 2^k R_{eff}(C,v)$ for all $i \in S_k$, implying $\frac{2^k R_{eff}(C,v)}{R_{eff}(S_k,v)} < E_{\boldsymbol{r}}(\boldsymbol{\phi}) \leq 1$, and so $R_{eff}(S_k,v) > 2^k R_{eff}(C,v) \geq 4R_{eff}(C,v)$. As

$$R_{eff}(C,v) \geq \frac{1}{4}\min\{R_{eff}(S_k,v), R_{eff}(C\backslash S_k,v)\} > \min\{R_{eff}(C,v), \frac{1}{4}R_{eff}(C\backslash S_k,v)\},$$

we have $R_{eff}(C\backslash S_k,v) < 4R_{eff}(C,v)$. This implies that $\left\|\pi_{S_k}^C(\mathbf{1}_v)\right\|_1 \leq \frac{R_{eff}(C\backslash S_k,v)}{R_{eff}(S_k,v)} < \frac{1}{2^{k-2}}$.

So we conclude that $Var_{\boldsymbol{\pi}}(\boldsymbol{\phi}) = \sum_{i\in S_k} \pi_i\phi_i^2 \leq \frac{1}{2^{k-2}}\cdot 2^{k+1}R_{eff}(C,v) = 8R_{eff}(C,v)$. □

E.6 Proof of Lemma 5.1

Proof. The first part of the statement is given by applying Lemma 5.4, and we see that it requires $\widetilde{O}\left(\delta_1^{-4}\beta^{-6} + \delta_1^{-2}\beta^{-2}\gamma^{-2}\right)$ random walks for each $u \in V\backslash C$ and $e \in E\backslash E(C)$ with $u \in e$.

For the second part we use the fact that the change in the demand projection after inserting v into C is given by

$$\boldsymbol{\pi}^{C\cup\{v\}}\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) = \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\cdot\left(\mathbf{1}_v - \boldsymbol{\pi}^C(\mathbf{1}_v)\right),$$

and therefore we can estimate this update via

$$\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top\frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\cdot\left(\mathbf{1}_v - \widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)\right).$$

where $\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)$ is the estimate we computed using Lemma 5.4 and $\widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)$ is obtained by applying Lemma 5.7.

Let us show that this estimation indeed introduces only a small amount of error. For any $\boldsymbol{\phi}$, such that $E_r(\boldsymbol{\phi}) \leq 1$, we can write

$$\begin{aligned}
&\left|\left\langle \widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\cdot(\mathbf{1}_v - \widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)) - \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\cdot(\mathbf{1}_v - \boldsymbol{\pi}^C(\mathbf{1}_v)), \boldsymbol{\phi}\right\rangle\right| \\
&\leq \left|\left\langle \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\cdot(\boldsymbol{\pi}^C(\mathbf{1}_v) - \widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)), \boldsymbol{\phi}\right\rangle\right| \\
&+ \left|\left\langle \left(\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\right)\cdot(\mathbf{1}_v - \boldsymbol{\pi}^C(\mathbf{1}_v)), \boldsymbol{\phi}\right\rangle\right| \\
&+ \left|\left\langle \left(\widetilde{\pi}_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right) - \pi_v^{C\cup\{v\}}\left(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}\right)\right)\cdot(\boldsymbol{\pi}^C(\mathbf{1}_v) - \widetilde{\boldsymbol{\pi}}^C(\mathbf{1}_v)), \boldsymbol{\phi}\right\rangle\right|.
\end{aligned}$$

At this point we can bound these quantities using Lemmas 5.4 and 5.7. It is important to notice that they require that S is a set of γ-important edges, for some parameter γ. Our congestion reduction subset C keeps increasing due to vertex insertions. This, however, means that effective resistances between any vertex in $V \setminus C$ and C can only decrease, and therefore the set of important edges can only increase. Thus we are still in a valid position to apply these lemmas.

Using $E_{\boldsymbol{r}}(\boldsymbol{\phi}) \leq 1$, which allows us to write:

$$\langle \mathbf{1}_v - \boldsymbol{\pi}^C(\mathbf{1}_v)), \boldsymbol{\phi}\rangle \leq \mathcal{E}_{\boldsymbol{r}}\left(\mathbf{1}_v - \boldsymbol{\pi}^C(\mathbf{1}_v)\right) = R_{eff}(v, C),$$

we can continue to upper bound the error by:

$$\begin{aligned}
&\frac{\widetilde{O}\left(\gamma^{-1}\beta^{-2}\right)}{\sqrt{R_{eff}(C,v)}}\cdot \delta_2\sqrt{R_{eff}(C,v)} + \frac{\delta_1}{\sqrt{R_{eff}(C,v)}}\cdot\sqrt{R_{eff}(C,v)} + \frac{\delta_1}{\sqrt{R_{eff}(C,v)}}\cdot\delta_2\sqrt{R_{eff}(C,v)} \\
&= \delta_2\cdot\widetilde{O}\left(\gamma^{-1}\beta^{-2}\right) + \delta_1 + \delta_1\delta_2.
\end{aligned}$$

Setting $\delta_1 = \widehat{\varepsilon}/2$ and $\delta_2 = \widehat{\varepsilon}\beta^2\gamma/\widetilde{O}(1)$, we conclude that w.h.p. each operation introduces at most $\widehat{\varepsilon}$ additive error in the maintained estimate for $\left\langle \widetilde{\boldsymbol{\pi}}^C(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}}), \boldsymbol{\phi}\right\rangle$.

Per Lemma 5.4, estimating one coordinate of the demand projection requires

$$\widetilde{O}\left(\delta_1^{-4}\beta^{-6} + \delta_1^{-2}\beta^{-2}\gamma^{-2}\right) = \widetilde{O}\left(\widehat{\varepsilon}^{-4}\beta^{-6} + \widehat{\varepsilon}^{-2}\beta^{-2}\gamma^{-2}\right)$$

random walks, and estimating $\widetilde{\boldsymbol{\pi}}^C(\boldsymbol{B}^\top \frac{\boldsymbol{q}_S}{\sqrt{\boldsymbol{r}}})$, per Lemma 5.7, requires

$$\widetilde{O}\left(\delta_2^{-2}\right) = \widetilde{O}\left(\widehat{\varepsilon}^{-2}\beta^{-4}\gamma^{-2}\right)$$

random walks. This concludes the proof.

□

F The Checker Data Structure

Theorem F.1 (Theorem 3, [GLP21]). *There is a* Checker *data structure supporting the following operations with the given runtimes against oblivious adversaries, for parameters* $0 < \beta_{\text{Checker}}, \varepsilon < 1$ *such that* $\beta_{\text{Checker}} \geq \widetilde{\Omega}\left(\varepsilon^{-1/2}/m^{1/4}\right)$.

- INITIALIZE($\boldsymbol{f}, \varepsilon, \beta_{\text{Checker}}$): *Initializes the data structure with slacks* $\boldsymbol{s}^+ = \boldsymbol{u} - \boldsymbol{f}$, $\boldsymbol{s}^- = \boldsymbol{f}$, *and resistances* $\boldsymbol{r} = \frac{1}{(\boldsymbol{s}^+)^2} + \frac{1}{(\boldsymbol{s}^-)^2}$. *Runtime:* $\widetilde{O}\left(m\beta_{\text{Checker}}^{-4}\varepsilon^{-4}\right)$.

- UPDATE($e, \boldsymbol{f}'$): *Set* $s_e^+ = u_e - f_e'$, $s_e^- = f_e'$, *and* $r_e = \frac{1}{(s_e^+)^2} + \frac{1}{(s_e^-)^2}$. *Runtime: Amortized* $\widetilde{O}\left(\beta_{\text{Checker}}^{-2}\varepsilon^{-2}\right)$.

- TEMPORARYUPDATE($e, \boldsymbol{f}'$): *Set* $s_e^+ = u_e - f_e'$, $s_e^- = f_e'$, *and* $r_e = \frac{1}{(s_e^+)^2} + \frac{1}{(s_e^-)^2}$. *Runtime: Worst case* $\widetilde{O}\left((K\beta_{\text{Checker}}^{-2}\varepsilon^{-2})^2\right)$, *where* K *is the number of* TEMPORARYUPDATE*s that have not been rolled back using* ROLLBACK. *All* TEMPORARYUPDATE*s should be rolled back before the next call to* UPDATE.

- ROLLBACK(): *Rolls back the last* TEMPORARYUPDATE *if it exists. The runtime is the same as the original operation.*

- CHECK($e, \boldsymbol{\pi}_{old}$): *Returns* $\widetilde{f}_e$ *such that* $\sqrt{r_e}\left|\widetilde{f}_e - \widetilde{f}_e^*\right| \leq \varepsilon$, *where*

$$\widetilde{\boldsymbol{f}}^* = \delta g(\boldsymbol{s}) - \delta \boldsymbol{R}^{-1}\boldsymbol{B}\left(\boldsymbol{B}^\top \boldsymbol{R}^{-1}\boldsymbol{B}\right)^+ \boldsymbol{B}^\top g(\boldsymbol{s}),$$

  for $\delta = 1/\sqrt{m}$. *Additionally, a vector* $\boldsymbol{\pi}_{old}$ *that is supported on* C *such that*

$$\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}_{old} - \boldsymbol{\pi}^C\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right)\right) \leq \varepsilon^2 m/4$$

  is provided, where C *is the vertex set of the dynamic sparsifier in the* DYNAMICSC *that is maintained internally. Runtime: Worst case* $\widetilde{O}\left(\left(\beta_{\text{Checker}}m + (K\beta_{\text{Checker}}^{-2}\varepsilon^{-2})^2\right)\varepsilon^{-2}\right)$, *where* K *is the number of* TEMPORARYUPDATE*s that have not been rolled back. Additionally, the output of* CHECK(e) *is independent of any previous calls to* CHECK.

Finally, all calls to CHECK *return valid outputs with high probability. The total number of* UPDATE*s and* TEMPORARYUPDATE*s that have not been rolled back should always be* $O(\beta_{\text{Checker}}m)$.

This theorem is from [GLP21]. The only difference is in the guarantee of CHECK. We will now show how it can be implemented. Let

$$\widetilde{\boldsymbol{f}}^* = \delta g(\boldsymbol{s}) - \delta \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{B}^\top g(\boldsymbol{s}).$$

Let DYNAMICSC be the underlying Schur complement data structure. We first add the endpoints u, w of e as terminals by calling

$$\text{DYNAMICSC.TEMPORARYADDTERMINALS}(\{u, w\})$$

so that the new Schur complement is on the vertex set $C' = C \cup \{u, w\}$. Then, we set

$$\boldsymbol{\phi} = -\widetilde{SC}^+\boldsymbol{\pi}_{old}$$

and $\widetilde{f}_e = (\phi_u - \phi_w)/\sqrt{r_e}$, where $\widetilde{SC}$ is the output of DYNAMICSC.$\widetilde{SC}$(). Equivalently, note that

$$\widetilde{f}_e = \delta \cdot \boldsymbol{1}_e^\top \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{L}^+\boldsymbol{\pi}_{old}.$$

We will show that $\sqrt{r_e}\left|\widetilde{f}_e - \widetilde{f}^*_e\right| \leq \varepsilon$.

We write

$$\begin{aligned}
&\sqrt{r_e}\left|\widetilde{f}_e - \widetilde{f}^*_e\right| \\
&\leq \left\|\delta\sqrt{\boldsymbol{r}}g(\boldsymbol{s})\right\|_\infty + \left\|\delta \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\left(\boldsymbol{B}^\top g(\boldsymbol{s}) - \boldsymbol{\pi}^{C'}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right)\right)\right\|_\infty \\
&+ \left\|\delta \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\left(\boldsymbol{\pi}^{C'}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right) - \boldsymbol{\pi}^{C}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right)\right)\right\|_\infty + \left\|\delta \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\left(\boldsymbol{\pi}^{C}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right) - \boldsymbol{\pi}_{old}\right)\right\|_\infty .
\end{aligned}$$

For the first term,

$$\left\|\delta\sqrt{\boldsymbol{r}}g(\boldsymbol{s})\right\|_\infty = \left\|\delta \frac{\frac{1}{\boldsymbol{s}^+} - \frac{1}{\boldsymbol{s}^-}}{\sqrt{\frac{1}{(\boldsymbol{s}^+)^2} + \frac{1}{(\boldsymbol{s}^-)^2}}}\right\|_\infty \leq \delta \leq \varepsilon/10 .$$

Now, by the fact that C' is a $\beta_{\textsc{Checker}}$-congestion reduction subset by definition in DynamicSC, Lemma 4.3 immediately implies that the second term is $\leq \delta \cdot \widetilde{O}\left(\beta_{\textsc{Checker}}^{-2}\right) \leq \varepsilon/10$.

For the third term, we apply Lemma 4.12, which shows that

$$\begin{aligned}
&\left\|\delta \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\left(\boldsymbol{\pi}^{C'}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right) - \boldsymbol{\pi}^{C}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right)\right)\right\|_\infty \\
&\leq \delta \cdot \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{C'}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right) - \boldsymbol{\pi}^{C}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right)\right)} \\
&\leq \delta \cdot \widetilde{O}\left(\beta_{\textsc{Checker}}^{-2}\right) \\
&\leq \varepsilon/10 ,
\end{aligned}$$

as the resistances don't change and we only have two terminal insertions from C to C'.

Finally, the fourth term is

$$\left\|\delta \boldsymbol{R}^{-1/2}\boldsymbol{B}\boldsymbol{L}^+\left(\boldsymbol{\pi}^{C}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right) - \boldsymbol{\pi}_{old}\right)\right\|_\infty \leq \delta \cdot \sqrt{\mathcal{E}_{\boldsymbol{r}}\left(\boldsymbol{\pi}^{C}\left(\boldsymbol{B}^\top g(\boldsymbol{s})\right) - \boldsymbol{\pi}_{old}\right)} \leq \delta \cdot \varepsilon\sqrt{m}/2 = \varepsilon/2 .$$

We conclude that $\sqrt{r_e}\left|\widetilde{f}_e - \widetilde{f}^*_e\right| \leq \varepsilon$. Finally, we call DynamicSC.Rollback to undo the terminal insertions.

The runtime of this operation is dominated by the call to DynamicSC.$\widetilde{SC}()$, which takes time $\widetilde{O}\left((\beta_{\textsc{Checker}} m + (K\beta_{\textsc{Checker}}^{-2}\varepsilon^{-2})^2)\varepsilon^{-2}\right)$.

References

[AMV20] Kyriakos Axiotis, Aleksander Mądry, and Adrian Vladu. Circulation control for faster minimum cost flow in unit-capacity graphs. In *61st Annual Symposium on Foundations of Computer Science*, pages 93–104. IEEE, 2020.

[BBG+20] Aaron Bernstein, Jan van den Brand, Maximilian Probst Gutenberg, Danupon Nanongkai, Thatchaphol Saranurak, Aaron Sidford, and He Sun. Fully-dynamic graph sparsifiers against an adaptive adversary. *arXiv preprint arXiv:2004.08432*, 2020.

[CKM+11] Paul Christiano, Jonathan A Kelner, Aleksander Mądry, Daniel A Spielman, and Shang-Hua Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd Annual ACM Symposium on Theory of computing*, pages 273–282, 2011.

[CMSV17] Michael B Cohen, Aleksander Mądry, Piotr Sankowski, and Adrian Vladu. Negative-weight shortest paths and unit capacity minimum cost flow in $\widetilde{O}(m^{10/7} \log W)$ time. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 752–771. SIAM, 2017.

[DGGP19] David Durfee, Yu Gao, Gramoz Goranci, and Richard Peng. Fully dynamic spectral vertex sparsifiers and applications. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 914–925, 2019.

[DS08] Samuel I Daitch and Daniel A Spielman. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the 40th Annual ACM Symposium on Theory of computing*, pages 451–460, 2008.

[Gab83] Harold N Gabow. Scaling algorithms for network problems. In *24th Annual Symposium on Foundations of Computer Science (SFCS 1983)*, pages 248–258. IEEE, 1983.

[GLP21] Yu Gao, Yang P Liu, and Richard Peng. Fully dynamic electrical flows: Sparse maxflow faster than goldberg-rao. *arXiv preprint arXiv:2101.07233*, 2021.

[GR98] Andrew V Goldberg and Satish Rao. Beyond the flow decomposition barrier. *Journal of the ACM (JACM)*, 45(5):783–797, 1998.

[KLOS14] Jonathan A Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete algorithms*, pages 217–226. SIAM, 2014.

[KLS20] Tarun Kathuria, Yang P. Liu, and Aaron Sidford. Unit capacity maxflow in almost $O(m^{4/3})$ time. In *61st Annual Symposium on Foundations of Computer Science*, pages 119–130. IEEE, 2020.

[KPSW19] Rasmus Kyng, Richard Peng, Sushant Sachdeva, and Di Wang. Flows in almost linear time via adaptive preconditioning. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 902–913, 2019.

[LRS13] Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 755–764, 2013.

[LS14] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in o (vrank) iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433. IEEE, 2014.

[LS20] Yang P Liu and Aaron Sidford. Faster energy maximization for faster maximum flow. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 2020.

[Mąd13] Aleksander Mądry. Navigating central path with electrical flows: From flows to matchings, and back. In *54th Annual Symposium on Foundations of Computer Science*, pages 253–262. IEEE, 2013.

[Mądry16] Aleksander Mądry. Computing maximum flow with augmenting electrical flows. In *57th Annual Symposium on Foundations of Computer Science*, pages 593–602. IEEE, 2016.

[Pen16] Richard Peng. Approximate undirected maximum flows in $O(m \text{ polylog } n)$ time. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete algorithms*, pages 1862–1867. SIAM, 2016.

[She13] Jonah Sherman. Nearly maximum flows in nearly linear time. In *54th Annual Symposium on Foundations of Computer Science*, pages 263–269. IEEE, 2013.

[She17a] Jonah Sherman. Area-convexity, ℓ_∞ regularization, and undirected multicommodity flow. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 452–460, 2017.

[She17b] Jonah Sherman. Generalized preconditioning and undirected minimum-cost flow. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 772–780. SIAM, 2017.

[ST18] Aaron Sidford and Kevin Tian. Coordinate methods for accelerating ℓ_∞ regression and faster approximate maximum flow. In *59th Annual Symposium on Foundations of Computer Science*, pages 922–933. IEEE, 2018.

[vdBLL$^+$21] Jan van den Brand, Yin Tat Lee, Yang P Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 859–869, 2021.

[vdBLN$^+$20] Jan van den Brand, Yin-Tat Lee, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Bipartite matching in nearly-linear time on moderately dense graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 919–930. IEEE, 2020.