Computing Maximum Flow with Augmenting Electrical Flows

Aleksander Mądry[*]

MIT

madry@mit.edu

Abstract

We present an $\widetilde{O}\left(m^{\frac{10}{7}}U^{\frac{1}{7}}\right)$-time algorithm for the maximum s-t flow problem (and the minimum s-t cut problem) in directed graphs with m arcs and largest integer capacity U. This matches the running time of the $\widetilde{O}\left((mU)^{\frac{10}{7}}\right)$-time algorithm of Mądry [43] in the unit-capacity case, and improves over it, as well as over the $\widetilde{O}\left(m\sqrt{n}\log U\right)$-time algorithm of Lee and Sidford [38], whenever U is moderately large and the graph is sufficiently sparse.

By well-known reductions, this also gives us an $\widetilde{O}\left(m^{\frac{10}{7}}B^{\frac{1}{7}}\right)$-time algorithm for the maximum-cardinality bipartite $\boldsymbol{b}$-matching problem in which the largest integer demand is B. This, again, matches the $\widetilde{O}\left((mB)^{\frac{10}{7}}\right)$-time algorithm of Mądry [43], when $B = 1$, which corresponds to the maximum-cardinality bipartite matching problem, and outperforms it, as well as the $\widetilde{O}\left(m\sqrt{n}\log B\right)$-time algorithm of Lee and Sidford [38], for moderate values of B and sufficiently sparse graphs.

One of the advantages of our algorithm is that it is significantly simpler than the ones presented in [43] and [38]. In particular, these algorithms employ a sophisticated interior-point method framework, while our algorithm is cast directly in the classic augmenting path setting that almost all the combinatorial maximum flow algorithms use. At a high level, the presented algorithm takes a primal dual approach in which each iteration uses electrical flows computations both to find an augmenting s-t flow in the current residual graph and to update the dual solution. We show that by maintain certain careful coupling of these primal and dual solutions we are always guaranteed to make significant progress.

1 Introduction

The maximum s-t flow problem and its dual, the minimum s-t cut problem, are two of the most fundamental and extensively studied graph problems in combinatorial optimization [51, 1, 50]. They have a wide range of applications (see [2]), are often used as subroutines in other algorithms (see, e.g., [4, 53]), and a number of other important problems – e.g., bipartite matching problem [10] – can be reduced to them. Furthermore, these two problems were often a testbed for development of fundamental algorithmic tools and concepts. Most prominently, the Max-Flow Min-Cut theorem [14, 17] constitutes the prototypical primal-dual relation.

Several decades of extensive work resulted in a number of developments on these problems (see Goldberg and Rao [22] for an overview) and many of their generalizations and special cases. Still, despite all this effort, the basic problem of computing maximum s-t flow and minimum s-t cut in general graphs resisted progress for a long time. In particular, for a number of years, the best running time bound for the problem was an $O(m \min\{m^{\frac{1}{2}}, n^{\frac{2}{3}}\} \log(n^2/m) \log U)$ (with U denoting the largest integer arc capacity) bound established in a a breakthrough paper by Goldberg and Rao [22] and this bound, in turn, matched the $O(m \min\{m^{\frac{1}{2}}, n^{\frac{2}{3}}\})$ bound for unit-capacity graphs due to Even and Tarjan [15] – and, independently, Karzanov [25] – that were put forth more than 40 years ago.

The above bounds were improved only fairly recently. Specifically, in 2013, Mądry [43] presented an interior-point method based framework for flow computations that gave an $\widetilde{O}\left(m^{\frac{10}{7}}\right)$-time[1] algorithm for the unit-capacity case of the maximum s-t flow and minimum s-t cut problems. This finally broke the long-standing $\widetilde{O}\left(n^{\frac{3}{2}}\right)$ running time barrier for sparse graphs, i.e., for $m = O(n)$. Later on, Lee and Sidford [38] developed a variant of interior-point method that enabled them to obtain improvement for the regime of dense graphs. In particular, their algorithm is able to compute the (general) maximum s-t flow and minimum s-t cut in $\widetilde{O}\left(m\sqrt{n}\log U\right)$ time and thus improve over the Goldberg-Rao bound whenever the input graph is sufficiently dense.

It is also worth mentioning that, as a precursor to the above developments, substantial progress was made in the context of $(1-\varepsilon)$-approximate variant of the maximum s-t flow problem in undirected graphs. In 2011, Christiano et al. [7] developed an algorithm that allows one to compute a $(1+\varepsilon)$-approximation to the undirected maximum s-t flow (and the minimum s-t cut) problem in $\widetilde{O}\left(mn^{\frac{1}{3}}\varepsilon^{-11/3}\right)$ time. Their result relies on devising a new approach to the problem that combines electrical flow computations with multiplicative weights update method (see [4]). Later, Lee et al. [37] presented a quite different – but still electrical-flow-based – algorithm that employs purely gradient-descent-type view to obtain an $\widetilde{O}\left(mn^{1/3}\varepsilon^{-2/3}\right)$-time $(1+\varepsilon)$-approximation for the case of unit capacities. Very recently, this line of work was culminated by Sherman [54] and Kelner et al. [26] who independently showed how to integrate non-Euclidean gradient-descent methods with fast poly-logarithmic-approximation algorithms for cut problems of Mądry [41] to get an $O(m^{1+o(1)}\varepsilon^{-2})$-time $(1+\varepsilon)$-approximation to the undirected maximum flow problem. Then, Peng [48] built on these works to obtain a truly nearly-linear, i.e., $\widetilde{O}\left(m\varepsilon^{-2}\right)$, running time.

Finally, we note that, in parallel to the above work that is focused on designing weakly-polynomial algorithms for the maximum s-t flow and minimum s-t cut problems, there is also a considerable interest in obtaining running time bounds that are strongly-polynomial, i.e., that do

---

[1]We recall that $\widetilde{O}(f)$ denotes $O(f \log^c f)$, for some constant c.

not depend on the values of arc capacities. The current best such bound is $O(mn)$ and it follows by combining the algorithms of King et al. [28] and Orlin [47].

Bipartite Matching Problem. Another problem that is related to the maximum s-t problem – and, in fact, can be reduced to it – is the (maximum-cardinality) bipartite matching problem. This problem is a fundamental assignment task with numerous applications (see, e.g., [1, 40]) and long history that has its roots in the works of Frobenius [18, 19] and König [29, 30, 31] from the early 20th century (see [52]). Already in 1931, König [32] and Egerváry [13] provided first constructive characterization of maximum matchings in bipartite graphs. This characterization can be turned into a polynomial-time algorithm. Then, in 1973, Hopcroft and Karp [24] and, independently, Karzanov [25], devised the celebrated $O(m\sqrt{n})$-time algorithm. For 40 years this bound remained the best one known in the regime of relatively sparse graphs. Only recently Mądry [43] obtained an improved, $\widetilde{O}\left(m^{10/7}\right)$ running time. It turns out, however, that whenever the input graph is dense, i.e., when m is close to n^2 even better bounds can be obtain. In this setting, one can combine the algebraic approach of Rabin and Vazirani [49] – that itself builds on the work of Tutte [57] and Lovász [39] – with matrix-inversion techniques of Bunch and Hopcroft [6] to get an algorithm that runs in $O(n^\omega)$ time (see [45]), where $\omega \leq 2.3727$ is the exponent of matrix multiplication [9, 59]. Also, later on, Alt et al. [3], and Feder and Motwani [16] developed combinatorial algorithms that offer a slight improvement – by a factor of, roughly, $\log n / \log \frac{n^2}{m}$ – over the $O(m\sqrt{n})$ bound of Hopcroft and Karp whenever the graph is sufficiently dense.

Finally, a lot of developments has been done in the context of the (maximum-cardinality) matching problem in general, i.e., not necessarily bipartite, graphs. Starting with the pioneering work of Edmonds [11], these developments led to bounds that essentially match the running time guarantees that were previously known only for bipartite case. More specifically, the running time bound of $O(m\sqrt{n})$ for the general-graph case was obtained by Micali and Vazirani [44, 60] (see also [20] and [21]). Then, Mucha and Sankowski [46] gave an $O(n^\omega)$-time algorithms for general graphs that builds on the algebraic characterization of the problem due to Rabin and Vazirani [49]. This result was later significantly simplified by Harvey [23].

1.1 Our Contribution

In this paper, we put forth a new algorithm for solving the maximum s-t flow and the minimum s-t cut problems in directed graphs. More precisely, we develop an algorithm that computes the maximum s-t flow of an input graph in time $\widetilde{O}\left(m^{\frac{10}{7}}U^{\frac{1}{7}}\right)$, where m denotes the number of arcs of that graph and U its largest integer capacity. Known reductions imply similar running time bounds for the minimum s-t cut problem as well as for the maximum-cardinality bipartite **b**-matching problem, a natural generalization of the maximum bipartite matching problem in which each vertex v has a degree demand b_v. For that problem, our algorithm yields an $\widetilde{O}\left(m^{\frac{10}{7}}B^{\frac{1}{7}}\right)$-time algorithm, with B being the largest (integer) vertex demand.

In the light of the above, for the unit-capacity/demand cases, the resulting algorithms match the performance of the algorithm of Mądry [43]. The latter algorithm, however, runs in $\widetilde{O}\left(mU^{\frac{10}{7}}\right)$ time (which translates into an $\widetilde{O}\left((mB)^{\frac{10}{7}}\right)$ running time for the bipartite **b**-matching problem) in the case of arbitrary capacities/demands. Consequently, the significantly better dependence of the running time of our algorithm on the largest capacity U/ largest demand B makes it much more

favorable in that setting. In fact, even though that dependence on U/B is still polynomial it enables our algorithm to remain competitive, for a non-trivial range of parameters, with the best existing algorithms that run in time that is logarithmic in U/B, such as the $\widetilde{O}\left(m\sqrt{n}\log U\right)$-time algorithm of Lee and Sidford [38].

Even more crucially, the key advantage of our algorithm is that it is significantly simpler than both the algorithm of Mądry [43] and that of Lee and Sidford [38]). Both these algorithms rely heavily on the interior-point method framework. Specifically, [43] designed a certain new variant of path-following interior-point method algorithm for the near-perfect bipartite $\boldsymbol{b}$-matching problem that encoded the input maximum s-t flow instance. It then used electrical flow computations to converge to the near-optimal solution for that problem. In order to break the bottlenecking $\widetilde{O}\left(m^{\frac{1}{2}}\right)$ iteration bound, however, Mądry [43] needed to, first, develop an extensive toolkit for perturbing and preconditioning the underlying electrical flow computation and, then, to combine this machinery with a very careful and delicate analysis of the resulting dynamics.

Our algorithm also relies on electrical flow computations but it abandons the above methodology and works instead fully within the classic augmenting path framework that almost all the previous combinatorial maximum s-t flow algorithms used. In this framework, developed by Ford and Fulkerson [17] (see also [14]), the flow is built in stages. Each stage corresponds to finding a so-called augmenting flow in the current residual graph, which is a directed graph that encodes the solution found so far. The algorithm terminates when the residual graph admits no more augmenting flows, i.e., there is no path from s to t in it, since in this case the solution found so far has to be already optimal.

The chief bottleneck in the running time analysis of augmenting path based algorithms is ensuring that each flow augmentation stage makes sufficient progress. Specifically, one wants to obtain a good trade off between the amount of flow pushed in each augmentation step and the time needed to implement each such flow push. One simple approach is to just use here s-t path computations. This is a nearly-linear time procedure but it only guarantees pushing one unit of flow each time. A much more sophisticated primitive developed in this context are blocking flow computations. Combining this primitive with a simple duality argument enabled Goldberg and Rao [22], who built on the work of Even and Tarjan [15], to obtain an $O(m\min\{m^{\frac{1}{2}}, n^{\frac{2}{3}}\}\log U)$-time maximum flow that remained the best known algorithm for nearly two decades. Unfortunately, trying to improve such blocking flow-based approaches turned out to be extremely difficult and no progress was made here so far, even in the unit-capacity case for which the best known bounds were established over 40 years ago.

One of the key contributions of this paper is bringing a new type of primitive: electrical flows to the augmenting path framework; and showing how to successfully use it to outperform the blocking flow-based methods. Specifically, our algorithm finds augmenting flows by computing electrical flows in certain symmetrization of the current residual graph – see Section 3.3 for more details. (Note that performing such a symmetrization is necessary as residual graphs are inherently directed while electrical flows are inherently undirected.) The key difficulty that arises in this context, however, is ensuring that this symmetrized residual graph can still support a significant fraction of the s-t capacity of the original residual graph. It is not hard to see that, in general, this might not be the case. To address this problem we introduce a certain careful coupling of the primal and dual solutions, which is inspired by the so-called centrality condition arising in interior-point method based maximum flow algorithms (see [43]). We then show that maintaining this coupling and applying a simple preconditioning technique let us guarantee that looking only for the flows in

3

the symmetrized version of the residual graph still provides sufficient progress in each iteration and, in particular, immediately delivers a $\widetilde{O}\left(m^{\frac{3}{2}}\log U\right)$-time algorithm.

We then (see Section 4) build on that basic algorithm and develop an ℓ_p-geometric understanding of its running time analysis. This understanding guides us towards a simple electrical flow perturbation technique – akin to the perturbation techniques used in [7] and [43] – that enables us to break the $\Omega(\sqrt{m})$ iterations bottleneck that all the blocking flow-based algorithms were suffering from, and thus get the final, improved result.

We believe that further study of this new augmenting flow based framework will deliver even faster and simpler algorithms.

1.2 Organization of the Paper

We begin the technical part of the paper in Section 2 by presenting some preliminaries on the maximum flow problem and the notion of electrical flows. Then, in Section 3, we present our framework and demonstrate how it yields an $\widetilde{O}\left(m^{\frac{3}{2}}\log U\right)$-time maximum s-t flow algorithm. Finally, in Section 4, we show how to refine our basic framework to obtain the improved running time of $\widetilde{O}\left(m^{\frac{10}{7}}U^{\frac{1}{7}}\right)$.

2 Preliminaries

Throughout this paper, we will be viewing graphs as having both lower and upper capacities. Specifically, we will denote by $G = (V, E, \boldsymbol{u})$ a directed graph with a vertex set V, an arc set E (we allow parallel arcs), and two (non-negative) integer capacities u_e^- and u_e^+, for each arc $e \in E$. (We will explain the role of these capacities below.) Usually, m will denote the number $|E|$ of arcs of the graph in question and $n = |V|$ will be the number of its vertices. We view each arc e of G as an ordered pair (u, v), where u is its *tail* and v is its *head*.

Observe that this perspective enables us to view undirected graphs as directed ones in which the ordered pair $(u, v) \in E$ is an (undirected) *edge* (u, v) and the order just specifies the *orientation* of that edge (from u to v).

Maximum Flow Problem. The basic notion of this paper is the notion of a *flow*. Given a graph G, we view a flow in G as a vector $\boldsymbol{f} \in \mathbb{R}^m$ that assigns a value f_e to each arc e of G. If this value is negative we interpret it as having a flow of $|f_e|$ flowing in the direction opposite to the arc orientation. (This convention is especially useful when discussing flows in undirected graphs.)

We say that a flow $\boldsymbol{f}$ is an $\boldsymbol{\sigma}$-*flow*, for some *demands* $\boldsymbol{\sigma} \in \mathbb{R}^n$ iff it satisfies *flow conservation constraints* with respect to that demands. That is, we have that

$$\sum_{e \in E^+(v)} f_e - \sum_{e \in E^-(v)} f_e = \sigma_v, \quad \text{for each vertex } v \in V. \tag{1}$$

Here, $E^+(v)$ (resp. $E^-(v)$) is the set of arcs of G that are entering (resp. leaving) vertex v. Intuitively, these constraints enforce that the net balance of the total in-flow into vertex v and the total out-flow out of that vertex is equal to σ_v, for every $v \in V$. (Observe that this implies, in particular, that $\sum_v \sigma_v = 0$.)

Furthermore, we say that a $\boldsymbol{\sigma}$-flow $\boldsymbol{f}$ is *feasible* in G iff $\boldsymbol{f}$ obeys the *the capacity constraints*:

$$-u_e^- \le f_e \le u_e^+, \quad \text{for each arc } e \in E. \tag{2}$$

In other words, we want each arc e to have a flow that is at most u_e^+ if it flows in the direction of e's orientation (i.e., $f_e \ge 0$), and at most u_e^-, if it flows in the opposite direction (i.e., $f_e < 0$). Note that setting all u_e^-s be equal to zero recovers the standard notion of flow feasibility in directed graphs.

One type of flows that will be of special interest to us are *s-t flows*, where s (the *source*) and t (the *sink*) are two distinguish vertices of G. Formally, an *s-t flow* is a $\boldsymbol{\sigma}$-flow whose demand vector $\boldsymbol{\sigma}$ is equal to $F \cdot \boldsymbol{\chi}_{s,t}$, where $F \ge 0$ is called the *value* of $\boldsymbol{f}$ and $\boldsymbol{\chi}_{s,t}$ is a demand vector that has -1 (resp. 1) at the coordinate corresponding to s (resp. t) and zeros everywhere else.

Now, the *maximum flow problem* corresponds to a task in which we are given a (directed) graph $G = (V, E, \boldsymbol{u})$ with integer capacities as well as a source vertex s and a sink vertex t and want to find a *feasible* (in the sense of (2)) *s-t* flow of maximum value. We will denote this maximum value as F^*.

Residual Graphs. A fundamental object in many maximum flow algorithms (including ours) is the notion of a residual graph. Given a graph $G = (V, E, \boldsymbol{u})$ and a feasible $\boldsymbol{\sigma}$-flow $\boldsymbol{f}$ in that graph (it is useful to think $\boldsymbol{\sigma} = F \cdot \boldsymbol{\chi}_{s,t}$), we define the *residual graph* $G_{\boldsymbol{f}}$ (of G with respect to $\boldsymbol{f}$) as a graph $G_{\boldsymbol{f}} = (V, E, \widehat{\boldsymbol{u}}(\boldsymbol{f}))$ over the same vertex and arc set as G and such that, for each arc $e = (u, v)$ of G, its lower and upper capacities are defined as

$$\widehat{u}_e^+(\boldsymbol{f}) := u_e^+ - f_e \quad \text{and} \quad \widehat{u}_e^-(\boldsymbol{f}) := u_e^- + f_e. \tag{3}$$

We will refer to $\widehat{u}_e^+(\boldsymbol{f})$ (resp. $\widehat{u}_e^-(\boldsymbol{f})$) as *forward residual capacity* (resp. *backward residual capacity*) of e and also define the *residual capacity* $\widehat{u}_e(\boldsymbol{f})$ of e as the minimum of these two, i.e., $\widehat{u}_e(\boldsymbol{f}) := \min\{\widehat{u}_e^-(\boldsymbol{f}), \widehat{u}_e^+(\boldsymbol{f})\}$. Note that the value of residual capacity depends on the flow $\boldsymbol{f}$ but we will ensure that it is always clear from the context with respect to which flow the residual capacity is measured. Also, observe that feasibility of $\boldsymbol{f}$ implies that all residual capacities are always non-negative (cf. (2)).

The main reason why residual graphs are useful in computing maximum flows is that they constitute a very convenient representation of the progress made so far. Specifically, we have the following important fact. (Again, it is useful to think here of the maximum *s-t* flow setting, in which $\boldsymbol{\sigma} = F^* \boldsymbol{\chi}_{s,t}$.)

Fact 2.1. *Let $\boldsymbol{\sigma}$ be some demand and $G = (V, E, \boldsymbol{u})$ be a graph in which a demand of $\boldsymbol{\sigma}$ can be routed, i.e., there exists a $\boldsymbol{\sigma}$-flow $\boldsymbol{f}^*$ that is feasible in G. Also, for any $0 \le \alpha \le 1$, let $\boldsymbol{f}$ be a feasible $\alpha \boldsymbol{\sigma}$-flow in G, and $G_{\boldsymbol{f}} = (V, E, \widehat{\boldsymbol{u}}(\boldsymbol{f}))$ be the residual graph of G with respect to $\boldsymbol{f}$. We have that*

(a) one can route a demand of $(1 - \alpha)\boldsymbol{\sigma}$ in $G_{\boldsymbol{f}}$;

(b) if $\boldsymbol{f}'$ is a feasible $\alpha' \boldsymbol{\sigma}$-flow in $G_{\boldsymbol{f}}$, for some α', then $\boldsymbol{f} + \boldsymbol{f}'$ is a feasible $(\alpha + \alpha')\boldsymbol{\sigma}$-flow in G.

Intuitively, the above fact enables us to reduce the task of routing a demand $\boldsymbol{\sigma}$ in G to a sequence of computations of augmenting $\alpha' \boldsymbol{\sigma}$-flows in the residual graph $G_{\boldsymbol{f}}$. We know that as long as we have not yet computed a feasible $\boldsymbol{\sigma}$-flow in G, $G_{\boldsymbol{f}}$ can route a demand of $(1 - \alpha)\boldsymbol{\sigma}$-flow, where $(1 - \alpha) > 0$ is the fraction of routed demand that we are still "missing", and each new augmenting $\alpha' \boldsymbol{\sigma}$-flow found in $G_{\boldsymbol{f}}$ brings us closer to routing $\boldsymbol{\sigma}$ in full in G. (Note that initially $G_{\boldsymbol{f}}$ is equal to G and $G_{\boldsymbol{f}}$ is changing after each new augmenting $\alpha' \boldsymbol{\sigma}$-flow is found.)

Electrical Flows and Vertex Potentials. Another notion that will play a fundamental role in this paper is the notion of electrical flows. Here, we just briefly review some of the key properties that we will need later. For an in-depth treatment we refer the reader to [5].

Consider a graph G and a vector of resistances $\boldsymbol{r} \in \mathbb{R}^m$ that assigns to each edge e its *resistance* $r_e > 0$. For a given $\boldsymbol{\sigma}$-flow $\boldsymbol{f}$ in G, let us define its *energy* (with respect to resistances $\boldsymbol{r}$) $\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f})$ to be

$$\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f}) := \sum_e r_e f_e^2. \tag{4}$$

For a given demand vector $\boldsymbol{\sigma}$ and a vector of resistances $\boldsymbol{r}$, we define the *electrical $\boldsymbol{\sigma}$-flow* in G (that is *determined* by resistances $\boldsymbol{r}$) to be the flow that minimizes the energy $\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f})$ among all flows with demand $\boldsymbol{\sigma}$ in G. As energy is a strictly convex function, one can easily see that such a flow is unique. (It is important to keep in mind that such flow is *not* required to be feasible with respect to capacities of G, in the sense of (2).)

A very useful property of electrical flows is that they can be characterized in terms of vertex potentials inducing them. Namely, one can show that a flow $\boldsymbol{f}$ with demands $\boldsymbol{\sigma}$ in G is an electrical $\boldsymbol{\sigma}$-flow determined by resistances $\boldsymbol{r}$ iff there exist *vertex potentials* ϕ_v (that we collect into a vector $\boldsymbol{\phi} \in \mathbb{R}^n$) such that, for any edge $e = (u, v)$ in G,

$$f_e = \frac{\phi_v - \phi_u}{r_e}. \tag{5}$$

In other words, a $\boldsymbol{f}$ with demands $\boldsymbol{\sigma}$ is an electrical $\boldsymbol{\sigma}$-flow iff it is *induced* via (5) by some vertex potential $\boldsymbol{\phi}$. (Note that the orientation of edges matters in this definition.) The above equation corresponds to the Ohm's law known from physics.

Note that we are able to express the energy $\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f})$ (see (4)) of an electrical $\boldsymbol{\sigma}$-flow $\boldsymbol{f}$ in terms of the potentials $\boldsymbol{\phi}$ inducing it as

$$\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f}) = \sum_{e=(u,v)} \frac{(\phi_v - \phi_u)^2}{r_e}. \tag{6}$$

One of the consequences of the above is that one can develop a dual characterization of the energy of an electrical $\boldsymbol{\sigma}$-flow in terms of optimization over vertex potentials. Namely, we have the following lemma whose proof can be found, e.g., in [43] Lemma 2.1.

Lemma 2.2. *For any graph $G = (V, E)$, any vector of resistances $\boldsymbol{r}$, and any demand vector $\boldsymbol{\sigma}$,*

$$\frac{1}{\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f}^*)} = \min_{\boldsymbol{\phi} | \boldsymbol{\sigma}^T \boldsymbol{\phi} = 1} \sum_{e=(u,v) \in E} \frac{(\phi_v - \phi_u)^2}{r_e},$$

where $\boldsymbol{f}^*$ is the electrical $\boldsymbol{\sigma}$-flow determined by $\boldsymbol{r}$ in G. Furthermore, if $\boldsymbol{\phi}^*$ are the vertex potentials corresponding to $\boldsymbol{f}^*$ then the minimum is attained by taking $\boldsymbol{\phi}$ to be equal to $\widehat{\boldsymbol{\phi}} := \boldsymbol{\phi}^* / \mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f}^*)$.

Note that the above lemma provides a convenient way of lowerbounding the energy of an electrical $\boldsymbol{\sigma}$-flow. One just needs to expose any vertex potentials $\boldsymbol{\phi}$ such that $\boldsymbol{\sigma}^T \boldsymbol{\phi} = 1$ and this will immediately constitute an energy lowerbound.

Laplacian Solvers. The fact that the electrical $\boldsymbol{\sigma}$-flow determined by resistances $\boldsymbol{r}$ is the only flow with demands $\boldsymbol{\sigma}$ that can be induced by vertex potentials (cf. (5)) has an important consequence. It enables us to reduce electrical $\boldsymbol{\sigma}$-flow computations to solving a linear system. In fact, the task of finding vertex potentials that induce that flow can be cast as a *Laplacian* linear system. That is, a linear system in which the constraint matrix corresponds to a Laplacian of the underlying graph with weights given by the (inverses of) the resistances $\boldsymbol{r}$.

Now, from the algorithmic point of view, the crucial property of Laplacian systems is that we can solve them, up to a very good approximation, very efficiently. Namely, there is a long line of work [56, 33, 34, 27, 8, 35, 36] that builds on an earlier work of Vaidya [58] and Spielman and Teng [55] and gives us a number of Laplacian system solvers that run in only nearly-linear time and, in case of more recent variants, are conceptually fairly simple. In particular, this line of work establishes the following theorem.

Theorem 2.3. *For any $\varepsilon > 0$, any graph G with n vertices and m edges, any demand vector $\boldsymbol{\sigma}$, and any resistances $\boldsymbol{r}$, one can compute in $\widetilde{O}\left(m \log \varepsilon^{-1}\right)$ time vertex potentials $\tilde{\boldsymbol{\phi}}$ such that $\|\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}^*\|_{\boldsymbol{L}} \le \varepsilon \|\boldsymbol{\phi}^*\|_{\boldsymbol{L}}$, where $\boldsymbol{L}$ is the Laplacian of G, $\boldsymbol{\phi}^*$ are potentials inducing the electrical $\boldsymbol{\sigma}$-flow determined by resistances $\boldsymbol{r}$, and $\|\boldsymbol{\phi}\|_{\boldsymbol{L}} := \sqrt{\boldsymbol{\phi}^T \boldsymbol{L} \boldsymbol{\phi}}$.*

Even though the solutions delivered by the above Laplacian solvers are only approximate, the quality of approximation that it delivers is more than sufficient for our purposes. Therefore, in the rest of this paper we assume that these solutions are exact. (See, e.g., [43] for discussion how to deal with inexactness of the electrical flows computed.)

Vector Norms. We will find it useful to work with various ℓ_p-norms of vectors. To this end, for any $p > 0$, we define the ℓ_p-*norm* $\|\boldsymbol{h}\|_p$ of a vector $\boldsymbol{h}$ as

$$\|\boldsymbol{h}\|_p := \left(\sum_i |h_i|^p\right)^{\frac{1}{p}}. \tag{7}$$

In particular, the ℓ_∞-norm is given by $\|\boldsymbol{h}\|_\infty := \max_i |h_i|$.

3 Augmenting Residual Graphs with Electrical Flows

In this section, we put forth the general framework we will use to solve the maximum s-t problem.[2] In particular, we demonstrate how this framework enables us to solve the maximum s-t flow problem in $\widetilde{O}\left(m^{\frac{3}{2}} \log U\right)$ time, where $m = |E|$ is the number of arcs in of the input graph and U is its largest (integer) capacity.

More precisely, for any maximum s-t flow instance $G = (V, E, \boldsymbol{u})$ and any value $F \ge 0$, our algorithm will work in $\widetilde{O}\left(m^{\frac{3}{2}} \log U\right)$ time and either: compute a feasible s-t flow of value F in G; or conclude that the maximum s-t flow value F^* of G is strictly smaller than F.

---

[2]Observe that our treatment can be immediately extended to routing arbitrary demands since this problem can always be reduced to the directed maximum flow problem by adding a super-source and super-sink and connecting all the surpluses in the original graph to the former and all the deficits in that graph to the latter via arcs with appropriate capacities.

Note that such procedure can be turned into a "classic" maximum s-t flow by applying binary search over values of F and incurring a multiplicative running time overhead of only $O(\log Un)$. (In fact, a standard use of capacity scaling technique [12] enables one to keep the overall running time of the resulting algorithm be only linear, instead of quadratic, in $\log U$.)

As mentioned earlier, our algorithm follows the primal dual augmenting paths based framework. At a high level, in each iteration (see Section 3.2), we use electrical flow computations to compute an augmenting flow as well as an update to the dual solution. To ensure that each augmenting iteration makes enough progress, we maintain a careful coupling of the primal and dual solution. We describe it below.

3.1 Primal Dual Coupling

Let us fix our target flow value F and, for notational convenience, , for any $0 \le \alpha \le 1$, let us denote by $\boldsymbol{\chi}_\alpha$ the demand vector $\alpha F \boldsymbol{\chi}_{s,t}$, i.e., the demand corresponding to routing α-fraction of the target flow value F of the s-t flow. Also, let us define $\boldsymbol{\chi}$ to be the demand equal to $\boldsymbol{\chi}_1$.

Again, our algorithm will be inherently primal dual in nature. That is, in addition to maintaining a primal solution: a $\boldsymbol{\chi}_\alpha$-flow $\boldsymbol{f}$, for some $0 \le \alpha \le 1$, that is feasible in G, it will also maintain a dual solution $\boldsymbol{y} \in \mathbb{R}^n$, which should be viewed as an embedding of all the vertices of the graph G into a line.

Consequently, our goal will be to either to make $\boldsymbol{f}$ be a feasible flow with demands $\boldsymbol{\chi}_\alpha$ and $\alpha = 1$, which corresponds to routing the target s-t flow value F in full, or to make the dual solution $\boldsymbol{y}$ certify that the target demand $\boldsymbol{\chi}_1 = \boldsymbol{\chi}$ cannot be fully routed in G and thus $F > F^*$.

Well-coupled Solutions. Our primal dual scheme will be enforcing a very specific coupling of the primal solution $\boldsymbol{f}$ and the dual solution $\boldsymbol{y}$. More precisely, let us define for each arc $e = (u, v)$

$$\Delta_e(\boldsymbol{y}) := y_v - y_u, \tag{8}$$

to be the "stretch" of the arc e in the embedding given by $\boldsymbol{y}$. Also, let $G_{\boldsymbol{f}}$ be the residual graph of G with respect to $\boldsymbol{f}$ and let us define, for a given arc e, a potential function

$$\Phi_e(\boldsymbol{f}) := \frac{1}{\widehat{u}_e^+(\boldsymbol{f})} - \frac{1}{\widehat{u}_e^-(\boldsymbol{f})}, \tag{9}$$

where we recall that $\widehat{u}_e^+(\boldsymbol{f}) := u_e^+ - f_e$ (resp. $\widehat{u}_e^-(\boldsymbol{f}) := u_e^- + f_e$) are forward (resp. backward) residual capacities of the arc e. (See preliminaries, i.e., Section 2, for details.)

Then, our intention is to maintain the following relation between $\boldsymbol{f}$ and $\boldsymbol{y}$:

$$\Delta_e(\boldsymbol{y}) = \Phi_e(\boldsymbol{f}) \quad \text{for each arc } e. \tag{10}$$

Intuitively, this condition ensures that the line embedding $\boldsymbol{y}$ stretches each arc e in the direction of the smaller of the residual capacities, and that this stretch is inversely proportional to the value of that capacity. (Note that if G was undirected and thus the initial capacities u_e^+ and u_e^- were equal, the direction of smaller residual capacity is also the direction in which the flow $\boldsymbol{f}$ flows through the arc e.) It is worth pointing out that this coupling condition is directly inspired by (and, in fact, can be directly derived from) a certain variant of centrality condition used by interior-point method based maximum flow algorithms (see [43]).

Even though condition (10) expresses our intended coupling, it will be more convenient to work with a slightly relaxed condition that allows us to have small violations of that ideal coupling. Specifically, we say that a primal dual solution $(\boldsymbol{f}, \boldsymbol{y})$ is $\boldsymbol{\gamma}$-*coupled* iff

$$|\Delta_e(\boldsymbol{y}) - \Phi_e(\boldsymbol{f})| \leq \frac{\gamma_e}{\widehat{u}_e(\boldsymbol{f})} \quad \text{for all arcs } e = (u, v). \tag{11}$$

Here, $\widehat{u}_e(\boldsymbol{f}) = \min\{\widehat{u}_e^+(\boldsymbol{f}), \widehat{u}_e^-(\boldsymbol{f})\}$ is the (symmetrized) residual capacity of the arc e, and $\boldsymbol{\gamma} \in \mathbb{R}^m$ is the *violation vector* that we intend to keep very small. In particular, we say that a primal dual solution $(\boldsymbol{f}, \boldsymbol{y})$ is *well-coupled* iff its violation vector $\boldsymbol{\gamma}$ has its ℓ_2-norm (see (7)) be at most $\frac{1}{100}$, i.e., $\|\boldsymbol{\gamma}\|_2 \leq \frac{1}{100}$.

One of the key consequences of maintaining a well-coupled primal dual pair of solutions $(\boldsymbol{f}, \boldsymbol{y})$ is that it enables us to use $\boldsymbol{y}$ as a dual certificate for inability to route certain demands in G. The following lemma gives us a concrete criterion for doing that.

Lemma 3.1. *Let $(\boldsymbol{f}, \boldsymbol{y})$ be a well-coupled primal dual solution with $\boldsymbol{f}$ being a $\boldsymbol{\chi}_\alpha$-flow, for some $0 \leq \alpha < 1$. If*

$$\boldsymbol{\chi}^T \boldsymbol{y} > \frac{2m}{(1 - \alpha)}$$

then the demand $\boldsymbol{\chi}$ cannot be routed in G, i.e., $F > F^*$.

Note that the choice of the constant 2 in the above lemma is fairly arbitrary. In principle, any constant strictly larger than 1 would suffice.

Proof. Assume for the sake of contradiction that $\boldsymbol{\chi}^T \boldsymbol{y} > \frac{2m}{(1-\alpha)}$ but the demand $\boldsymbol{\chi}$ can still be routed in G. As $\boldsymbol{f}$ is an $\boldsymbol{\chi}_\alpha$-flow that is feasible in G, by Fact 2.1, there has to exist a $\boldsymbol{\chi}_{(1-\alpha)}$-flow $\boldsymbol{f}'$ that is feasible in the residual graph $G_{\boldsymbol{f}}$. Observe that, by (1) and (8), we have in that case that

$$
\begin{aligned}
(\boldsymbol{f}')^T \Delta(\boldsymbol{y}) &= \sum_e f'_e \Delta_e(\boldsymbol{y}) = \sum_{e=(u,v)} f'_e (y_v - y_u) = \sum_v y_v \left(\sum_{e \in E^+(v)} f'_e - \sum_{e \in E^-(v)} f'_e \right) \tag{12} \\
&= (1 - \alpha) F (y_t - y_s) = (1 - \alpha) \boldsymbol{\chi}^T \boldsymbol{y} > 2m.
\end{aligned}
$$

We want now to use the feasibility of $\boldsymbol{f}'$ to derive an upper bound on the inner produce $(\boldsymbol{f}')^T \Delta(\boldsymbol{y})$ that will violate the above lower bound and thus deliver the desired contradiction.

To this end, let us consider some particular arc e. Assume wlog that $f'_e \geq 0$ (handling of the case of $f'_e < 0$ is completely analogous). We have that the contribution of that arc to the inner product $(\boldsymbol{f}')^T \Delta(\boldsymbol{y})$ is, by (11), at most

$$f'_e \Delta_e(\boldsymbol{y}) \leq f'_e \Phi_e(\boldsymbol{f}) + \frac{\gamma_e f'_e}{\widehat{u}_e(\boldsymbol{f})} \leq f'_e \left(\frac{1}{\widehat{u}_e^+(\boldsymbol{f})} - \frac{1}{\widehat{u}_e^-(\boldsymbol{f})} \right) + \frac{\gamma_e f'_e}{\widehat{u}_e(\boldsymbol{f})},$$

where we recall that $\boldsymbol{\gamma}$ is the violation vector of $(\boldsymbol{f}, \boldsymbol{y})$ (cf. (11)).

As $\boldsymbol{f}'$ is feasible in $G_{\boldsymbol{f}}$, we need to have that $f'_e \leq \widehat{u}_e^+(\boldsymbol{f})$. So, if $\widehat{u}_e^+(\boldsymbol{f}) \leq \widehat{u}_e^-(\boldsymbol{f})$, i.e., $\widehat{u}_e^+(\boldsymbol{f}) = \widehat{u}_e(\boldsymbol{f})$, then

$$f'_e \Delta_e(\boldsymbol{y}) \leq f'_e \left(\frac{1}{\widehat{u}_e^+(\boldsymbol{f})} - \frac{1}{\widehat{u}_e^-(\boldsymbol{f})} \right) + \frac{\gamma_e f'_e}{\widehat{u}_e(\boldsymbol{f})} \leq \frac{f'_e (1 + \gamma_e)}{\widehat{u}_e(\boldsymbol{f})} \leq (1 + \gamma_e).$$

9

Otherwise, we have that

$$f'_e \Delta_e(\boldsymbol{y}) \le f'_e \left(\frac{1}{\widehat{u}_e^+(\boldsymbol{f})} - \frac{1}{\widehat{u}_e^-(\boldsymbol{f})} \right) + \frac{\gamma_e f'_e}{\widehat{u}_e(\boldsymbol{f})} \le f'_e \left(\frac{1}{\widehat{u}_e^+(\boldsymbol{f})} - \frac{1-\gamma_e}{\widehat{u}_e(\boldsymbol{f})} \right) \le 1 - \frac{(1-\gamma_e)f'_e}{\widehat{u}_e(\boldsymbol{f})} \le 1,$$

as $\gamma_e \le \|\boldsymbol{\gamma}\|_\infty \le \|\boldsymbol{\gamma}\|_2 \le \frac{1}{2}$. So, in either case, this contribution is at most $(1 + \gamma_e)$. Summing these contributions over all arcs we get that

$$(\boldsymbol{f}')^T \Delta(\boldsymbol{y}) \le \sum_e f'_e \Delta_e(\boldsymbol{y}) \le \sum_e (1 + \gamma_e) \le m + \sqrt{m}\|\boldsymbol{\gamma}\|_2 \le 2m,$$

which indeed contradicts the lower bound (12). The lemma follows. $\qquad\square$

In the light of the above discussion, for a given well-coupled primal dual solution $(\boldsymbol{f}, \boldsymbol{y})$, we should view the value of α as a measure of our primal progress, while the value of the inner product $\boldsymbol{\sigma}^T \boldsymbol{y}$ can be seen as a measure of our dual progress.

Initialization. The coupling condition (11) ties the primal and dual solutions $\boldsymbol{f}$ and $\boldsymbol{y}$ fairly tightly. In fact, coming up with some primal dual solutions that are well-coupled, which we need to initialize our framework, might be difficult.

Fortunately, finding such a pair of initial well-coupled solutions turns out to be easy, if our input graph G is undirected. In that case, just taking $\boldsymbol{f}$ to be a trivial zero flow and $\boldsymbol{y}$ to be a trivial all-zeros embedding makes condition (11) satisfied (with γ_es being all zero). After all, the residual graph $G_{\boldsymbol{f}}$ with respect to such zero flow $\boldsymbol{f}$ is just the graph G itself and thus $\widehat{u}_e^+(\boldsymbol{f}) = u_e^+ = u_e^- = \widehat{u}_e^-(\boldsymbol{f})$.

Furthermore, even though we are interested in solving directed instances too, every such instance can be reduced to an undirected one. Specifically, we have the following lemma.

Lemma 3.2. *Let G be an instance of the maximum s-t flow problem with m arcs and the maximum capacity U, and let F be the corresponding target flow value. In $\widetilde{O}(m)$ time, one can construct an instance G' of undirected maximum s-t flow problem that has $O(m)$ arcs and the maximum capacity U, as well as target flow value F' such that:*

(a) if there exists a feasible s-t flow of value F in G then a feasible s-t flow of value F' exists in G';

(b) given a feasible s-t flow of value F' in G' one can construct in $\widetilde{O}(m)$ time a feasible s-t flow of value F in G.

The proof of the above lemma boils down to a known reduction of the directed maximum flow problem to its undirected version – see, e.g., Theorem 3.6.1 in [42] for details. Consequently, from now on we can assume without loss of generality that we always have a well-coupled primal dual pair to initialize our framework.

3.2 Progress Steps

Once we described our basic framework and how to initialize it, we are ready to put forth its main ingredient: progress steps that enable us to gradually improve the primal dual solutions that we maintain. To this end, let us fix a well-coupled primal dual solutions $(\boldsymbol{f}, \boldsymbol{y})$ with $\boldsymbol{f}$ being a $\boldsymbol{\chi}_\alpha$-flow, for some $0 \le \alpha < 1$, that is feasible in G. Our goal in this section will be to use $(\boldsymbol{f}, \boldsymbol{y})$ to compute, in nearly-linear time, another pair of well-coupled primal dual solutions $(\boldsymbol{f}^+, \boldsymbol{y}^+)$ that bring us closer

10

to the optimal solutions. The flow $\boldsymbol{f}^+$ we obtain will be a $\boldsymbol{\chi}_{\alpha'}$-flow feasible in G, for $\alpha' > \alpha$. So, the resulting flow update $\boldsymbol{f}^+ - \boldsymbol{f}$ is an augmenting flow that is feasible in our current residual graph $G_{\boldsymbol{f}}$ and pushes $(\alpha' - \alpha)$-fraction of the target s-t flow.

We will compute $(\boldsymbol{f}^+, \boldsymbol{y}^+)$ in two stages. First, in the *augmentation step*, we obtain a pair of solutions $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}})$, with $\widehat{\boldsymbol{f}}$ being a $\boldsymbol{\chi}_{\alpha'}$-flow, for $\alpha' > \alpha$, that is feasible in G. These solutions make progress toward the optimal solutions but might end up being not well-coupled. Then, in the *fixing step*, we correct $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}})$ slightly by adding a carefully chosen flow circulation, i.e., a flow with all-zeros demands, to $\widehat{\boldsymbol{f}}$ and an dual update to $\widehat{\boldsymbol{y}}$ so as to make the resulting solutions $(\boldsymbol{f}^+, \boldsymbol{y}^+)$ be well-coupled, as desired.

The key primitive in both these steps will be electrical flow computation. As we will see, the crucial property of electrical flows we will rely on here is their"self-duality". That is, the fact that each electrical flow computation gives us both the flow and the corresponding vertex potentials that are coupled to it via Ohm's law (5). This enables us not only to update our primal and dual solutions with that flow and vertex potentials, respectively, but also, much more crucially, this coupling introduced by Ohm's law will be exactly what will allow us to (approximately) maintain our desired primal dual coupling property (11).

Augmentation Step. To perform an augmentation step we compute first an electrical $\boldsymbol{\chi}$-flow $\widetilde{\boldsymbol{f}}$ in G with the resistances $\boldsymbol{r}$ defined as

$$r_e := \frac{1}{(\widehat{u}_e^+(\boldsymbol{f}))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{f}))^2}, \tag{13}$$

for each arc e. Note that the resistance r_e is proportional, roughly, to the inverse of the square of the residual capacity $\widehat{u}_e(\boldsymbol{f})$ of that arc. So, in particular, it becomes very large whenever residual capacity of the arc e is small, and vice versa. As we will see shortly, this correspondence will allow us to control the amount of flow that $\widetilde{\boldsymbol{f}}$ sends over each edge and thus ensure that the respective residual capacities are not violated.

Let $\widetilde{\boldsymbol{\phi}}$ be the vertex potentials inducing $\widetilde{\boldsymbol{f}}$ (via the Ohm's law (6)). Then, we obtain the new primal and dual solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}})$ as follows:

$$\begin{aligned} \widehat{f}_e &:= f_e + \delta \widetilde{f}_e && \text{for each arc } e \\ \widehat{y}_v &:= y_v + \delta \widetilde{\phi}_v && \text{for each vertex } v, \end{aligned} \tag{14}$$

where δ is the desired *step size*. Observe that this update is exactly an augmentation of our current flow $\boldsymbol{f}$ with the (scaled) electrical flow $\delta \widetilde{\boldsymbol{f}}$ and adding to our dual solution the (scaled) vertex potentials $\delta \widetilde{\boldsymbol{\phi}}$. This, in particular, means that the new flow $\widehat{\boldsymbol{f}}$ we obtain here is a $\boldsymbol{\chi}_{\alpha'}$-flow with

$$\alpha' = \alpha + \delta. \tag{15}$$

The step size δ, however, will have to be carefully chosen. On one hand, as we see in (15), the larger it is the more progress we make. On the other hand, though, it has to be small enough so as to keep the flow $\delta \widetilde{\boldsymbol{f}}$ feasible in $G_{\boldsymbol{f}}$ (and thus, by Fact 2.1, to make the flow $\widehat{\boldsymbol{f}} + \boldsymbol{f}$ feasible in G).

Note that, a priori, we have no direct control over neither the directions in which the electrical $\boldsymbol{\chi}$-flow $\widetilde{\boldsymbol{f}}$ is flowing thorough each arc nor the amount of that flow. So, in order to establish a grasp on what is the right setting of δ, it is useful to define a *congestion vector* $\boldsymbol{\rho}$ given by

$$\rho_e := \frac{\widetilde{f}_e}{\widehat{u}_e(\boldsymbol{f})}, \tag{16}$$

11

for each arc e. One can view ρ_e as a normalized measure of how much the electrical flow $\widetilde{f}$ overflows the residual capacity $\widehat{u}_e(f)$ of the arc e and in what direction. In other words, the sign of ρ_e encodes the direction of the flow $\widetilde{f}_e$.

It is now not hard to see that to ensure that $\delta\widetilde{f}$ is feasible in G_f, i.e., that no residual capacity is violated by the update (14), it suffices that $\delta|\rho_e| \leq \frac{1}{4}$, for all arcs e, or, equivalently, that

$$\delta \leq \frac{1}{4\|\boldsymbol{\rho}\|_\infty},\tag{17}$$

where $\|\cdot\|_\infty$ is the standard ℓ_∞-norm.

It is also worth pointing out that the congestion vector $\boldsymbol{\rho}$ turns out to capture (up to a small multiplicative factor) the contribution of each arc e to the energy $\mathcal{E}_r(\widetilde{f})$ of $\widetilde{f}$. In particular, we have the following simple but important observation.

Lemma 3.3. *For any arc e, $\rho_e^2 \leq r_e\widetilde{f}_e^2 \leq 2\rho_e^2$ and*

$$\|\boldsymbol{\rho}\|_2^2 \leq \mathcal{E}_r(\widetilde{f}) \leq 2\|\boldsymbol{\rho}\|_2^2,$$

where $\|\cdot\|_2$ is the standard ℓ_2-norm.

Proof. Note that, by definition of the resistance r_e (13), (16), and the definition of the residual capacity $\widehat{u}_e$, we have that

$$\rho_e^2 \leq \frac{(\rho_e\widehat{u}_e(f))^2}{\widehat{u}_e(f)^2} \leq \frac{\widetilde{f}_e^2}{\widehat{u}_e(f)^2} \leq \left((\widehat{u}_e^+(f))^{-2} + (\widehat{u}_e^-(f))^{-2}\right)\widetilde{f}_e^2 = r_e\widetilde{f}_e^2 \leq \frac{2\widetilde{f}_e^2}{\widehat{u}_e(f)^2} = 2\rho_e^2.$$

We thus also have that

$$\|\boldsymbol{\rho}\|_2^2 = \sum_e \rho_e^2 \leq \sum_e r_e\widetilde{f}_e^2 = \mathcal{E}_r(\widetilde{f}) \leq \sum_e 2\rho_e^2 = 2\|\boldsymbol{\rho}\|_2^2,$$

as desired. $\qquad\square$

This link between the energy-minimizing nature of the electrical $\boldsymbol{\sigma}$-flow $\widetilde{f}$ and the ℓ_2-norm of the congestion vector $\boldsymbol{\rho}$ will end up being very important. One reason for that is the fact that ℓ_∞-norm is always bounded by the ℓ_2-norm. Consequently, we can use this connection to control the ℓ_∞-norm of the vector $\boldsymbol{\rho}$ and thus the value of δ needed to satisfy the feasibility condition (17).

It turns out, however, that just ensuring that our augmenting flow is feasible is not enough for our purposes. Specifically, we also need to control the coupling of our primal dual solutions, and the feasibility bound (17) might be not sufficiently strong for this purpose. We thus have to develop analyze the impact of the update (14) on the coupling condition (11) more closely.

To this end, let us first notice the following fact that stems from a standard application of the Taylor's theorem. Its proof appears in Appendix A.1

Fact 3.4. *For any $u_1, u_2 > 0$ and x such as $|x| \leq \frac{u}{4}$, where $u = \min\{u_1, u_2\}$, we have that*

$$\left(\frac{1}{u_1 - x} - \frac{1}{u_2 + x}\right) = \frac{1}{u_1} - \frac{1}{u_2} + \left(\frac{1}{u_1^2} + \frac{1}{u_2^2}\right)x + x^2\zeta,$$

where $|\zeta| \leq \frac{5}{u^3}$.

12

Now, the above approximation bound enables us to get an accurate estimate of how the coupling condition evolves during the augmentation step (14). That is, for any arc e, applying Fact 3.4 with $u_1 = \widehat{u}_e^+(\boldsymbol{f})$, $u_2 = \widehat{u}_e^-(\boldsymbol{f})$ and $x = \delta \widetilde{f}_e$, gives us that

$$\Phi_e(\widehat{\boldsymbol{f}}) = \frac{1}{\widehat{u}_e^+(\boldsymbol{f}) - \delta \widetilde{f}_e} - \frac{1}{\widehat{u}_e^-(\boldsymbol{f}) + \delta \widetilde{f}_e} = \Phi_e(\boldsymbol{f}) + \left(\frac{1}{(\widehat{u}_e^+(\boldsymbol{f}))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{f}))^2} \right) \delta \widetilde{f}_e + (\delta \widetilde{f}_e)^2 \zeta_e, \quad (18)$$

with $|\zeta_e| \leq \frac{5}{\widehat{u}_e(\boldsymbol{f})^3}$.

Observe that the above expression tells us that the first order approximation of the change in the primal contribution of the arc e to the coupling condition (11) caused by the update (14) is exactly

$$\Phi_e(\widehat{\boldsymbol{f}}) - \Phi_e(\boldsymbol{f}) \approx \left(\frac{1}{(\widehat{u}_e^+(\boldsymbol{f}))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{f}))^2} \right) \delta \widetilde{f}_e = r_e \delta \widetilde{f}_e,$$

where we also used (13). (In fact, the choice of the resistances $\boldsymbol{r}$ was made exactly to make the above statement true.)

Furthermore, by Ohm's law (5) and the definition of our augmentation step (14), we have that

$$r_e \delta \widetilde{f}_e = \delta \left(\widetilde{\phi}_v - \widetilde{\phi}_u \right) = \Delta_e(\widehat{\boldsymbol{y}}) - \Delta_e(\boldsymbol{y}),$$

which is exactly the change in the dual contribution of the arc $e = (u, v)$ to the coupling condition (11) caused by the augmentation step update (14).

So, up to first order approximation, these two contributions cancel out, leaving the coupling (11) intact. Consequently, any increase in the violation of the coupling condition must come from the second-order terms in the approximation (18). The following lemma makes this precise and is proved in Appendix A.2.

Lemma 3.5. *Let $0 < \delta \leq (4\|\boldsymbol{\rho}\|_\infty)^{-1}$ and the primal dual solution $(\boldsymbol{f}, \boldsymbol{y})$ be $\boldsymbol{\gamma}$-coupled. Then, we have that, for any arc $e = (u, v)$,*

$$\left| \Delta_e(\widehat{\boldsymbol{y}}) - \Phi_e(\widehat{\boldsymbol{f}}) \right| \leq \frac{\frac{4}{3}\gamma_e + 7(\delta\rho_e)^2}{\widehat{u}_e(\widehat{\boldsymbol{f}})}.$$

Fixing Step. Although Lemma 3.5 enables us to bound the deterioration of the primal dual coupling during the augmentation step, we cannot prevent this effect altogether. Therefore, we need to introduce a *fixing step* that deals with this problem. More precisely, we develop a procedure that uses a single electrical flow computation to significantly reduce that violation, provided it was not too large to begin with. This is formalized by the following lemma, whose proof appears in Appendix A.3.

Lemma 3.6. *Let $(\boldsymbol{g}, \boldsymbol{z})$ be a $\boldsymbol{\varsigma}$-coupled primal dual solution, with $\boldsymbol{g}$ being a feasible $\boldsymbol{\chi}_{\alpha'}$-flow and $\|\boldsymbol{\varsigma}\|_2 \leq \frac{1}{50}$. In $\widetilde{O}(m)$ time, we can compute a primal dual solution $(\bar{\boldsymbol{g}}, \bar{\boldsymbol{z}})$ that is well-coupled and in which $\bar{\boldsymbol{g}}$ is still a $\boldsymbol{\chi}_{\alpha'}$-flow.*

Now, after putting Lemmas 3.5 and 3.6 together, we are finally able to state the condition that δ in the update (14) has to satisfy in order to ensure that the solutions $(\boldsymbol{f}^+, \boldsymbol{y}^+)$ we obtain after performing the augmentation and fixing step is still well-coupled.

Lemma 3.7. $(\boldsymbol{f}^+, \boldsymbol{y}^+)$ *is a well-coupled primal dual solution with* $\boldsymbol{f}^+$ *being a* $\boldsymbol{\chi}_{\alpha'}$*-flow that is feasible in G whenever*

$$\delta \leq (33\|\boldsymbol{\rho}\|_4)^{-1},$$

The above lemma tells us that the step size δ of our augmentation step (14) should be governed by the ℓ_4-norm (see (7)) of the congestion vector (16). Observe that the ℓ_4-norm of a vector is always upper bounding its ℓ_∞-norm. So, the condition (17) is subsumed by this ℓ_4-norm bound.

Proof. Note that we always have that $\|\boldsymbol{\rho}\|_4 \geq \|\boldsymbol{\rho}\|_\infty$. So, the condition (17) is automatically satisfied and the flow $\boldsymbol{f}^+$ is indeed a $\boldsymbol{\chi}_{\alpha'}$-flow that is feasible in G.

Now, to argue about well-coupling of $(\boldsymbol{f}^+, \boldsymbol{y}^+)$, notice that, in the light of Lemma 3.6, it suffices to argue that the primal dual solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}})$ obtained after executing the augmentation step (14) is $\widehat{\boldsymbol{\gamma}}$-coupled with $\|\widehat{\boldsymbol{\gamma}}\|_2 \leq \frac{1}{50}$.

To this end, observe that by Lemma 3.5 and the fact that $(\boldsymbol{f}, \boldsymbol{y})$ are well-coupled, i.e., $\boldsymbol{\gamma}$-coupled with $\|\boldsymbol{\gamma}\|_2 \leq \frac{1}{100}$, we have that

$$
\begin{aligned}
\|\widehat{\boldsymbol{\gamma}}\|_2 &= \sqrt{\sum_e \widehat{\gamma}_e^2} \leq \sqrt{\sum_e \left(\frac{4}{3}\gamma_e + 7(\delta\rho_e)^2\right)^2} \\
&\leq \frac{4}{3}\|\boldsymbol{\gamma}\|_2 + 7\delta^2 \sqrt{\sum_e \rho_e^4} \leq \frac{4}{300} + 7\delta^2\|\boldsymbol{\rho}\|_4^2 \leq \frac{2}{150} + \frac{7}{1089} < \frac{1}{50}.
\end{aligned}
$$

The lemma follows. $\qquad\square$

3.3 Analysis of the Algorithm

We want now to analyze the overall running time of our algorithm. Recall that given our target demand $\boldsymbol{\chi}$ that corresponds to sending F units of flow from the source s to the sink t, our overarching goal is to either route this demand fully in G or provide a dual certificate that it is impossible to route $\boldsymbol{\chi}$ in G.

We aim to achieve this goal by maintaining and gradually improving a primal dual solution $(\boldsymbol{f}, \boldsymbol{y})$. In this solution, $\boldsymbol{f}$ is a $\boldsymbol{\chi}_\alpha$-flow (which corresponds to routing an α fraction of the desired demand $\boldsymbol{\chi}$) that is feasible in G and $\boldsymbol{f}$ and $\boldsymbol{y}$ are well-coupled, i.e., tied to each other via condition (11) with the violation vector $\widehat{\gamma}$ having sufficiently small ℓ_2-norm. As described in Section 3.2, each iteration runs in nearly-linear time and boils down to employing electrical flow computations to find an augmenting flow in the current residual graph $G_{\boldsymbol{f}}$ (as well as to update the dual solution to maintain well-coupling).

Consequently, all we need to do now is to lower bound the amount of progress that each of these iteration makes. Ideally, we would like to prove that in each iteration in which $\boldsymbol{f}$ already routed α-fraction of the desired flow, i.e., $\boldsymbol{f}$ is a feasible $\boldsymbol{\chi}_\alpha$-flow, the step size δ (see (14)) can be taken to be at least

$$\delta \geq (1-\alpha)\hat{\delta}, \tag{19}$$

for some fixed $\hat{\delta} > 0$. Observe that if such a lower bound was established then, by (15), it would imply that each iteration finds an augmenting flow that routes at least $\hat{\delta}$-fraction of the amount of flow still to be routed. As a result, after executing at most $O(\hat{\delta}^{-1} \log mU)$ iterations, the remaining value of flow to be routed would be at most 1 and thus a simple flow rounding and augmenting path

finding would yield the final answer (see, e.g., [43]), making the overall running time be at most $\widetilde{O}\left(\hat{\delta}^{-1} m \log U\right)$.

Unfortunately, a priori, it is difficult to provide any such non-trivial unconditional lower bound on the amount of primal progress we make in each iteration. After all, it could be the case that the target flow cannot be even routed in G. More importantly though, even if the target flow could be routed in G, and thus the residual graph always admitted augmenting flows of sufficiently large value, it is still not clear that our flow augmenting procedure would be able to find them. (It is worth noting that this problem is by no means specific to our algorithm. In fact, in *all* the maximum flow algorithms that rely on the augmenting paths framework ensuring that each iteration makes a sufficient primal progress is a chief bottleneck in the analysis.)

The root of the problem here is that our flow augmenting procedure is based on electrical flows and these are undirected in nature. Consequently, the augmenting flows that it finds have to come from a fairly restricted class: *s-t* flows that are feasible in a certain "symmetrized" version of the residual graph.

To make it precise, given a residual graph $G_{\boldsymbol{f}}$, let us define its *symmetrization* $\widehat{G}_{\boldsymbol{f}}$ to be an undirected graph in which each arc e has its forward and backward capacity equal to $\widehat{u}_e(\boldsymbol{f})$, i.e., to the minimum of the forward $\widehat{u}_e^+(\boldsymbol{f})$ and backward $\widehat{u}_e^-(\boldsymbol{f})$ residual capacities in $G_{\boldsymbol{f}}$. Observe now that each (electrical) augmenting flow $\delta \widetilde{\boldsymbol{f}}$ found in the augmentation step (cf. (14)) is not only feasible in the residual graph $G_{\boldsymbol{f}}$ but also in its symmetrization $\widehat{G}_{\boldsymbol{f}}$ – this is exactly what the condition (17) enforces.

However, not all augmenting *s-t* flows that are feasible in $G_{\boldsymbol{f}}$ have to be feasible in $\widehat{G}_{\boldsymbol{f}}$ too. In fact, it can happen that a large maximum *s-t* flow value that the residual graph $G_{\boldsymbol{f}}$ supports mostly vanishes in its symmetrization $\widehat{G}_{\boldsymbol{f}}$, and thus prevents our algorithm from making a sufficient good primal progress. (Again, a difficulty of a exactly the same nature arises in the analysis of the classic flow augmenting algorithms such as [15, 25, 22].)

Preconditioning Arcs.   It turns out, however, that there is a fairly simple way to circumvent the above difficulty and ensure that the kind of direct, "primal-only" analysis we hoped for above can indeed be performed. Namely, we just need to "precondition" our input graph by adding to it a large number of *s-t* arcs of sufficiently large capacities.

More precisely, we modify our input graph G by adding to it m undirected arcs between the source s and sink t with a forward and backward capacities equal to $2U$ and their orientation being from s to t. We will call these arcs *preconditioning arcs*. Observe that after this modification the number of arcs of our graph as well as its maximum capacity at most doubled, and the maximum *s-t* flow value changed additively by exactly $2mU$. In particular, the preconditioning arcs constitute exactly half of all the arcs and the amount of *s-t* flow that they alone can support is at least twice the *s-t* throughput of the rest of the graph. (Also, as these arcs are undirected they do not interfere with our initialization procedure – cf. Lemma 3.2.[3]) Consequently, we can just focus on analyzing the running time of our algorithm on this preconditioned instance and the bounds we establish will immediately translate over to the original instance.[4]

---

[3]More precisely, we can just first initialize our framework for the original graph, as before, and only precondition the undirected graph resulting from that initialization.

[4]Note that the preconditioning arcs have to be fully saturated in any maximum *s-t* flow. So, simply dropping these arcs and the flow on them will yield the maximum *s-t* flow in the original graph.

As already mentioned, somewhat surprisingly, once such preconditioning arcs are in place and our primal dual solution $(\boldsymbol{f}, \boldsymbol{y})$ is well-coupled, it is always the case that the symmetrization $\widehat{G}_{\boldsymbol{f}}$ of our current residual graph $G_{\boldsymbol{f}}$ retains a constant fraction of the s-t throughput. Intuitively speaking, well-coupling prevents the "shortcutting" preconditioning arc from getting "clogged" too quickly. Instead, their residual capacity is consumed at the same rate as that of the rest of the graph. Consequently, these arcs alone are always able to provide enough of s-t throughput in the symmetrization $\widehat{G}_{\boldsymbol{f}}$ of the residual graph $G_{\boldsymbol{f}}$. This is made precise in the following lemma, whose proof appears in Appendix A.4

Lemma 3.8. *Let $(\boldsymbol{f}, \boldsymbol{y})$ be a well-coupled primal dual solution in the (preconditioned) graph G and let $\boldsymbol{f}$ be a $\boldsymbol{\chi}_\alpha$-flow, for some $0 \le \alpha < 1$, that is feasible in G. We have either that:*

(a) there exists a $\boldsymbol{\chi}_{\frac{(1-\alpha)}{10}}$-flow $\boldsymbol{f}'$ that is feasible in the symmetrization $\widehat{G}_{\boldsymbol{f}}$ of the residual graph $G_{\boldsymbol{f}}$;

(b) or $\boldsymbol{\chi}^T \boldsymbol{y} > \frac{2m}{(1-\alpha)}$ implying that our target demand $\boldsymbol{\chi}$ cannot be routed in G (cf. Lemma 3.1).

Note that if our target demand $\boldsymbol{\chi}$ is exactly the demand $F^* \boldsymbol{\chi}_{s,t}$ of the maximum s-t flow, the second condition cannot ever trigger and thus indeed it is the case that the symmetrization of the (preconditioned) residual graph retains a constant fraction of the original s-t throughput.

Lower Bounding $\hat{\delta}$. Once we proved that the symmetrization $\widehat{G}_{\boldsymbol{f}}$ of the residual graph $G_{\boldsymbol{f}}$ retains most of its s-t flow throughput (see Lemma 3.8), we are finally able to provide an absolute lower bound $\hat{\delta}$ (cf. (19)) on the amount of primal progress each iteration of our algorithm makes. To this end, we upper bound first the energy, or, (almost) equivalently, the ℓ_2-norm of the congestion vector (see Lemma 3.3) of the electrical flow that we use in our augmentation step (see (14)).

Lemma 3.9. *Let $(\boldsymbol{f}, \boldsymbol{y})$ be a well-coupled primal dual solution, with $\boldsymbol{f}$ being a $\boldsymbol{\chi}_\alpha$-flow that is feasible in $G_{\boldsymbol{f}}$, for some $0 \le \alpha < 1$. Let $\widetilde{\boldsymbol{f}}$ be an electrical $\boldsymbol{\chi}$-flow determined by the resistances $\boldsymbol{r}$ given by (13). We have that either:*

(a) $\|\boldsymbol{\rho}\|_2^2 \le \mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}}) \le \frac{C_{\mathcal{E}} m}{(1-\alpha)^2}$, where $\boldsymbol{\rho}$ is the congestion vector defined in (16), and $C_{\mathcal{E}} > 0$ is an explicit constant;

(b) or, $\boldsymbol{\chi}^T \boldsymbol{y} > \frac{2m}{(1-\alpha)}$, i.e., our target demand $\boldsymbol{\chi}$ cannot be routed in G.

Proof. By Lemma 3.8, we either have that $\boldsymbol{\chi}^T \boldsymbol{y} > \frac{2m}{(1-\alpha)}$, i.e., our condition (b) triggers, or there exists a $\boldsymbol{\chi}_{\frac{(1-\alpha)}{10}}$-flow $\boldsymbol{f}'$ that is feasible in the symmetrization $\widehat{G}_{\boldsymbol{f}}$ of the residual graph $G_{\boldsymbol{f}}$.

Let us first bound the energy $\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f}')$ of that flow $\boldsymbol{f}'$. To this end, observe that, by definition (13), we have that

$$\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f}') = \sum_e r_e (f_e')^2 \le \sum_e \left(\frac{1}{(\widehat{u}_e^+(\boldsymbol{f}))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{f}))^2} \right) (f_e')^2 \le 2 \sum_e \left(\frac{f_e'}{\widehat{u}_e(\boldsymbol{f})} \right)^2 \le 2m,$$

where the last inequality follows by the fact that the flow f_e' is feasible in $\widehat{G}_{\boldsymbol{f}}$. (Recall that the forward and backward capacities of an arc e in $\widehat{G}_{\boldsymbol{f}}$ are equal to $\widehat{u}_e(\boldsymbol{f})$.)

Consequently, if we scale $\boldsymbol{f}'$ by $\frac{10}{(1-\alpha)}$ then it will become a $\boldsymbol{\chi}$-flow and its energy will be at most $\frac{200}{(1-\alpha)^2}m$. However, by definition, the energy $\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})$ of the electrical $\boldsymbol{\chi}$-flow $\widetilde{\boldsymbol{f}}$ cannot be larger than the energy of any other $\boldsymbol{\chi}$-flow. We thus can conclude that

$$\|\boldsymbol{\rho}\|_2^2 \leq \mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}}) \leq \frac{100}{(1-\alpha)^2}\mathcal{E}_{\boldsymbol{r}}(\boldsymbol{f}') \leq \frac{200}{(1-\alpha)^2}m,$$

where we also used Lemma 3.3. So, taking $C_{\mathcal{E}} := 200$ concludes the proof. □

Now, we should notice that by Lemma 3.7 it suffices that we always have that

$$\delta \leq \frac{1}{33\|\boldsymbol{\rho}\|_4} \leq \frac{1}{33\|\boldsymbol{\rho}\|_2} \leq \frac{(1-\alpha)}{33\sqrt{C_{\mathcal{E}}m}}, \tag{20}$$

where we used Lemma 3.9 and the simple fact that it is always the case that $\|\boldsymbol{\rho}\|_4 \leq \|\boldsymbol{\rho}\|_2$. Consequently, by (19), we see that we can take $\hat{\delta} := (33\sqrt{C_{\mathcal{E}}m})^{-1}$, which gives us the desired $\widetilde{O}\left(m^{\frac{3}{2}}\log U\right)$ time algorithm.

Finally, we want to emphasize again that even though our above analysis was based solely on analyzing our primal progress[5], maintaining the dual solution and the primal dual coupling (11) was absolutely critical for its success.

4 An Improved $\widetilde{O}\left(m^{\frac{10}{7}}U^{\frac{1}{7}}\right)$-Time Algorithm

In this section, we present an improved maximum s-t flow algorithm that runs in time $\widetilde{O}\left(m^{\frac{10}{7}}U^{\frac{1}{7}}\right)$, where U is the value of the largest (integer) capacity of the graph. In the case of the unit capacities, i.e., $U = 1$, this running time matches the running time of the $\widetilde{O}(m)^{\frac{10}{7}}$-time algorithm of Mądry [43], and improves over the best known bound of $\min\{\widetilde{O}\left((mU)^{\frac{10}{7}}\right), \widetilde{O}(m\sqrt{n}\log U)\}$ that stems for the work of Mądry [43] and Lee and Sidford [38] whenever U is moderately large (and the graph is sufficiently sparse).

To achieve this goal, we will modify the $\widetilde{O}\left(m^{\frac{3}{2}}\log U\right)$-time algorithm presented in Section 3. These modifications will enable us to lower bound the amount of primal progress $\hat{\delta}$ (cf. (19)) this algorithm makes in each iteration. Specifically, we aim to show that

$$\hat{\delta} \geq \left(m^{\frac{1}{2}-\eta}\right)^{-1}, \tag{21}$$

for certain $\eta := \frac{1}{14} - \frac{1}{7}\log_m U - O(\log\log mU)$, whenever that modified algorithm executes progress steps (see Section 3.2). Clearly, as discussed in Section 3.3, establish that lower bound on $\hat{\delta}$ ensures that we need at most $\widetilde{O}\left(m^{\frac{1}{2}-\eta}\log U\right) = \widetilde{O}\left(m^{\frac{10}{7}}U^{\frac{1}{7}}\right)$ progress steps to converge to the optimum solution.

---

[5]In fact, one could perform it even without resorting explicitly to the dual infeasibility certificates $\boldsymbol{\chi}^T\boldsymbol{y}$.

4.1 Modified Algorithm

Before we explain how we modify the $\widetilde{O}\left(m^{\frac{3}{2}}\log U\right)$-time algorithm we presented in Section 3, let us understand first what is its main running time bottleneck.

To this end, recall that the key quantity that captures the progress that each augmentation iteration makes is the ℓ_4-norm $\|\boldsymbol{\rho}\|_4$ of the congestion vector $\boldsymbol{\rho}$ of the electrical $\boldsymbol{\chi}$-flow $\widetilde{\boldsymbol{f}}$ determined by the resistances $\boldsymbol{r}$ (see (13)). More precisely, by Lemma 3.7, we need to always have that

$$(1-\alpha)\hat{\delta} \leq \delta \leq (33\|\boldsymbol{\rho}\|_4)^{-1}.$$

In Section 3.3, we bounded that ℓ_4-norm by simply noticing that it always has to be upper bounded by the ℓ_2-norm of that vector and that, by Lemma 3.3, this ℓ_2-norm is directly tied to the energy $\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})$ of the electrical flow $\widetilde{\boldsymbol{f}}$. Furthermore, since we were able to also prove that this energy is always $O\left(\frac{m}{(1-\alpha)^2}\right)$ (see Lemma 3.9) the final $O(\sqrt{m})$ bound on $\hat{\delta}^{-1}$ followed. (See (20) for details.)

Now, the key observation is that such bounding of the ℓ_4-norm of the vector $\boldsymbol{\rho}$ with its ℓ_2-norm might be wasteful. That is, even though it is not hard to construct examples in which the congestion vector $\boldsymbol{\rho}$ has these two norms be fairly close, such examples are inherently fragile. More precisely, as first pointed out by Christano et al. [7] in the context of an analogous ℓ_∞- vs ℓ_2-norm trade off, whenever $\|\boldsymbol{\rho}\|_4 \approx \|\boldsymbol{\rho}\|_2$, it must be necessarily be the case that most of the energy of the electrical flow $\widetilde{\boldsymbol{f}}$ is contributed by a very small set of arcs. Moreover, if one perturbs such arcs by increasing their resistance, this will result in a great increase of the energy of the corresponding electrical flow. Christano et al. [7], and later Mądry [43] have demonstrated that a careful exploitation of this phenomena can lead to ensuring that such fragile worst-case instances do not appear too often and thus a better bound on the ℓ_4-norm can be achieved.

Arc Boosting. Our improved algorithm will also follow the perturbation paradigm we outlined above. To this end, we introduce an operation of boosting that we will use to perturb high-energy contributing arcs whenever needed.

Formally, let us fix a primal dual well-coupled solution $(\boldsymbol{f}, \boldsymbol{y})$ and a particular arc $e = (u, v)$. Let us assume wlog that $\widehat{u}_e^+(\boldsymbol{f}) = \widehat{u}_e(\boldsymbol{f})$ (the case of $\widehat{u}_e^-(\boldsymbol{f}) = \widehat{u}_e(\boldsymbol{f})$ is completely symmetric). We define a *boost* of e to be an operation that modifies the input graph and the solution $(\boldsymbol{f}, \boldsymbol{y})$ in the following way:

(1) The arc e is replaced by an u-v path consisting of $\beta(e) := 2 + \lceil \frac{2U}{\widehat{u}_e(\boldsymbol{f})} \rceil$ arcs $e_1, \ldots, e_{\beta(e)}$, all oriented towards v.

(2) The first two arcs e_1, e_2 are just copies of the arc e, with $u_{e_1}^+ = u_{e_2}^+ := u_e^+$ and $u_{e_1}^- = u_{e_2}^- := u_e^-$.

(3) For all the remaining $(\beta(e) - 2)$ arcs, we have that $u_{e_i}^+ := +\infty$ and $u_{e_i}^- := \widetilde{u}$, for $2 < i \leq \beta(e)$, where $\widetilde{u} \geq U$ is a value that we will set below.

(4) The flow $\boldsymbol{f}$ pushes the original flow f_e over the created u-v path.

(5) Finally, if $v_0 = u, v_1, \ldots, v_{\beta(e)} = v$ are the consecutive vertices on the created path then:

- $y_{v_0} := y_u$ and $y_{v_{\beta(e)}} := y_v$;

18

- $y_{v_1} := y_v$, and $y_{v_2} := y_{v_1} + \Phi_e(\boldsymbol{f}) = y_v + \Phi_e(\boldsymbol{f})$;

- $y_{v_3}, \ldots, y_{v_{\beta(e)-1}}$ are set so as $\Delta_{e_i}(\boldsymbol{y}) = -\frac{\Delta_{e_2}(\boldsymbol{y})}{\beta(e)-2} = -\frac{\Phi_e(\boldsymbol{f})}{\beta(e)-2}$, for each $2 < i \le \beta(e)$.

Observe that after the above modifications, we have that

$$
\begin{aligned}
\Delta_{e_1}(\boldsymbol{y}) - \Phi_{e_1}(\boldsymbol{f}) &= \Delta_e(\boldsymbol{y}) - \Phi_e(\boldsymbol{f}) \\
\Delta_{e_2}(\boldsymbol{y}) &= \Phi_{e_2}(\boldsymbol{f}) = \Phi_e(\boldsymbol{f}) \\
\Phi_{e_i}(\boldsymbol{f}) &= -\frac{1}{(\widetilde{u} + f_e)} = \Delta_{e_i}(\boldsymbol{y}),
\end{aligned}
$$

for each $2 < i \le \beta(e)$, as long as we set

$$
\widetilde{u} := \frac{\beta(e) - 2}{\Phi_e(\boldsymbol{f})} - f_e.
$$

Therefore, the solution $(\boldsymbol{f}, \boldsymbol{y})$ remains feasible in the (modified) residual graph $G_{\boldsymbol{f}}$ and is well-coupled too. Also, note that since $\Phi_e(\boldsymbol{f}) \le \frac{1}{\widetilde{u}_e(\boldsymbol{f})}$ and $|f_e| \le U$, the fact that $(\beta(e) - 2) \ge \frac{2U}{\widetilde{u}_e(\boldsymbol{f})}$ ensures that $u_{e_i}^- = \widetilde{u} \ge U$, for all $2 < i \le \beta(e)$. Consequently, the effective forward and backward capacities of the created path are the same as the forward and backward capacities of the original arc e, and thus the s-t throughput of our graph remains unchanged.

We thus see that, from our point of view, the boosting operation had only two real consequences: one of them is the intended effect and the other one is an undesired side effect. The intended effect is that the effective resistance with respect to the resistances $\boldsymbol{r}$ (cf. (13)) of the created path is at least twice as large as the original resistance r_e of the arc e. (Note that the arcs e_1 and e_2 have exactly the same residual capacities, and thus the resistances r_{e_1} and r_{e_2} are both equal to the resistance r_e of the original arc e.) The undesired consequence, however, is that the number of arcs in our graph increased by $\beta(e) - 1$.

The key impact of that (at least) doubling of the resistance of the arc e is that, intuitively, it makes the energy of our electrical $\boldsymbol{\chi}$-flow $\widetilde{\boldsymbol{f}}$ increase proportionally to the initial contribution of that arc to the (squared) ℓ_2-norm of the congestion vector $\boldsymbol{\rho}$. This is made precise in the following lemma. (This lemma is stated in a slightly more general form for future reference. For now, one can think of the set S consisting of a single arc e.)

Lemma 4.1. *Let $\widetilde{\boldsymbol{f}}$ be an electrical $\boldsymbol{\chi}$-flow determined by the resistance $\boldsymbol{r}$ in our graph G and let $\boldsymbol{\rho}$ be the corresponding congestion vector defined by (16). If $\widetilde{\boldsymbol{f}}'$ is the electrical $\boldsymbol{\chi}$-flow determined by the resistances $\boldsymbol{r}'$ in the graph G' that was obtained by boosting a set of arcs S then*

$$
\mathcal{E}_{\boldsymbol{r}'}(\widetilde{\boldsymbol{f}}') \ge \left(1 + \frac{1}{8}\left(\sum_{e \in S} \frac{\rho_e^2}{\|\boldsymbol{\rho}\|_2^2}\right)\right) \mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}}).
$$

Proof. Let $\widetilde{\phi}$ be the vertex potentials inducing $\widetilde{\boldsymbol{f}}$, by Lemma 2.2 and (16), we have that

$$
\frac{1}{\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})} = \sum_{e' = (u', v')} \frac{(\widetilde{\phi}_{v'} - \widetilde{\phi}_{u'})^2}{r_{e'} \mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})^2}
$$

and $\boldsymbol{\chi}^T \widetilde{\phi} = \mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})$.

We want now to use the same voltages $\widetilde{\phi}$ to lowerbound the energy $\mathcal{E}_{r'}(\widetilde{f}')$ of the electrical $\boldsymbol{\chi}$-flow $\widetilde{f}'$ after boosting the arcs in S. Observe that the effective resistance $r'_{u,v}$ between the endpoints u and v of some boosted arc $e = (u,v) \in S$ after that boost is at least $2r_e$. So, when we analyze the energy of $\widetilde{f}'$ we can treat the u-v path that was created as just an arc e but with the resistance r'_e equal to $r'_{u,v} \geq 2r_e$.

Therefore, by using Lemma 2.2 again, we can conclude that

$$
\begin{aligned}
\frac{1}{\mathcal{E}_{r'}(\widetilde{f}')} \;\leq\; & \left(\sum_{e'=(u',v')\notin S} \frac{(\widetilde{\phi}_{v'} - \widetilde{\phi}_{u'})^2}{r_{e'}\mathcal{E}_r(\widetilde{f})^2} \right) + \sum_{e=(u,v)\in S} \frac{(\widetilde{\phi}_v - \widetilde{\phi}_u)^2}{r'_e\mathcal{E}_r(\widetilde{f})^2} \\
\leq\; & \left(\sum_{e'=(u',v')} \frac{(\widetilde{\phi}_{v'} - \widetilde{\phi}_{u'})^2}{r_{e'}\mathcal{E}_r(\widetilde{f})^2} \right) - \sum_{e=(u,v)\in S} \frac{(\widetilde{\phi}_v - \widetilde{\phi}_u)^2}{2r_e\mathcal{E}_r(\widetilde{f})^2} = \frac{1}{\mathcal{E}_r(\widetilde{f})}\left(1 - \sum_{e=(u,v)\in S} \frac{(\widetilde{\phi}_v - \widetilde{\phi}_u)^2}{2r_e\mathcal{E}_r(\widetilde{f})} \right) \\
\leq\; & \frac{1}{\mathcal{E}_r(\widetilde{f})}\left(1 - \sum_{e\in S} \frac{r_e\widetilde{f}_e^2}{2\mathcal{E}_r(\widetilde{f})} \right) \leq \frac{1}{\mathcal{E}_r(\widetilde{f})}\left(1 - \sum_{e\in S} \frac{\rho_e^2}{4\|\boldsymbol{\rho}\|_2^2} \right),
\end{aligned}
$$

where we also used Lemma 3.3 and Ohm's law (5). The lemma follows by noticing that $\frac{1}{(1-\varepsilon)} \geq (1 + \frac{\varepsilon}{2})$ whenever $0 \leq \varepsilon \leq \frac{1}{4}$. $\qquad\square$

Boosting Use and Early Termination. Once we introduced the boosting operation, we can state the two modifications of the basic algorithm from Section 3 that we will make.

The first one is relatively minor: we terminate our algorithm whenever

$$(1-\alpha)F \leq m^{\frac{1}{2}-\eta}. \tag{22}$$

That is, once we know that we are within an additive factor of $m^{\frac{1}{2}-\eta}$ of the target flow, we do not execute progress steps anymore. Observe that in that case we can just round our current flow and compute the optimal solution using at most $m^{\frac{1}{2}-\eta}$ augmenting paths computation (see, e.g., [43] for details). So, if we perform these operations, our algorithm will still run within the desired running time bound.

The second modification specifies how and when we apply boosting operations. To this end, given an electrical $\boldsymbol{\chi}$-flow $\widetilde{f}$ and its congestion vector $\boldsymbol{\rho}$, let us define an arc e to be *high-energy* if

$$|\rho_e| \geq \rho^* = \frac{m^{\frac{1}{2}-3\eta}}{C_{\rho^*}(1-\alpha)}, \tag{23}$$

for certain constant $C_{\rho^*} > 0$ that we will set later (see (28) below).

Now, recall that our goal is to ensure that (21) holds, which corresponds to ensuring that, by Lemma 3.7, $\|\boldsymbol{\rho}\|_4 \leq \frac{m^{\frac{1}{2}-\eta}}{33(1-\alpha)}$. However, for reasons that will be clear shortly, we will actually want to have a stronger, ℓ_3-norm bound on $\boldsymbol{\rho}$. Specifically, we want to make a progress step only when

$$\|\boldsymbol{\rho}\|_3 \leq \frac{m^{\frac{1}{2}-\eta}}{33(1-\alpha)}, \tag{24}$$

and then use the step size

$$\delta = \frac{\hat{\delta}}{(1-\alpha)} = \frac{1}{33(1-\alpha)\|\boldsymbol{\rho}\|_3}$$

when making the augmentation step described in Section 3.2. Observe that as $\|\boldsymbol{\rho}\|_3 \geq \|\boldsymbol{\rho}\|_4$, this new condition still satisfies the ℓ_4-norm requirement set forth in Lemma 3.7.

Unfortunately, as mentioned earlier, it can sometime happen that the bound (24) is violated. Therefore, we want to use our boosting technique to enforce that such violations do not happen too often. Specifically, whenever condition (24) does not hold we boost all the arcs in the set S^*, which is defined to be the set of up to $k^* := m^{4\eta}$ high-energy arcs e with highest values of $|\rho_e|$. In other words, we order all the high-energy arcs in a non-increasing order with respect to $|\rho_e|$ and take S^* to consist of the first up to k^* of them. (Note that, in principle, in some iterations there can be only very few, or even no, high-energy arcs.)

Once we boost the set S^*, we simply proceed to the next iteration (in the new version of the graph that our boosting operation created). Note that, at first glance, there is nothing preventing us from needing to execute such a boosting step in each iteration, and thus to never be able to execute progress steps. However, as we argue in the next section, we are actually guaranteed to be able to execute progress steps often enough and thus make sufficiently good progress overall.

4.2 Analysis of the Improved Algorithm

We proceed now to analyzing our modified algorithm and establishing the improved running time bound. This will require tackling two issues.

First one is controlling the increase in the number of arcs resulting from performing of all our boosting operations. Specifically, we want to maintain the following invariant.

Invariant 4.2. *The total increase in the number of arcs is at most $\frac{m}{10}$.*

As we will see, ensuring that this invariant is maintained will be what determines our final bound on the value of η (cf. (21)).

We need to maintain this invariant in order to be able to use the machinery we developed in Section 3. (It is not hard to check that, as long as that invariant is not violated, the whole analysis performed there is still valid, subject to slight adjustment of the corresponding constants.)

The other issue we need to tackle corresponds to bounding the total number of boosting steps we execute. After all, we want to ensure that we execute progress steps often enough. In order to obtain such a bound we will perform a potential based argument with our potential function being the energy $\mathcal{E}_r(\widetilde{\boldsymbol{f}})$ of the electrical $\boldsymbol{\chi}$-flow $\widetilde{\boldsymbol{f}}$ we compute in each iteration. (Note that this electrical flow $\widetilde{\boldsymbol{f}}$ is always determined by the resistances $\boldsymbol{r}$ (cf. (13)) in the current version of the graph G.) One can easily convince oneself that this energy has to be at least $\frac{1}{U^2}$ at the beginning of the algorithm and, by Lemma 3.9 (and provided Invariant 4.2 holds), it is never larger than

$$\mathcal{E}_r(\widetilde{\boldsymbol{f}}) \leq \frac{C_{\mathcal{E}} m}{(1-\alpha)^2} \leq O(m^3 U^2).$$

(Here, the second inequality follows due to the fact that $1 \leq F \leq 2mU$ and we have our termination condition (22).) We want now to establish two claims:

(I) Whenever a boosting step is executed, the energy $\mathcal{E}_r(\widetilde{\boldsymbol{f}})$ increases by a factor of at least $\left(1 + C_B m^{-2\eta}\right)$, for some constant $C_B > 0$. (To prove this statement we will use Lemma 4.1.)

(II) Whenever a progress step is performed, with the condition (24) satisfied, the energy $\mathcal{E}_r(\widetilde{\boldsymbol{f}})$ decreases by at most $\left(1 + C_D m^{-2\eta}\right)$, for some other constant $C_D > 1$. (Proving this statement is where we leverage the fact that we enforce the stronger, ℓ_3-norm condition (24) on $\boldsymbol{\rho}$.)

21

Observe that once the above two claims are established, we can amortize the number of boosting steps against the number of good progress steps. Specifically, a simple calculation shows that unless the number of boosting step is within a $O(\log m \log U)$ factor of the number of the progress steps, the resulting increase of the energy $\mathcal{E}_r(\widetilde{\boldsymbol{f}})$ would violate the absolute upper bound $O(m^3U^2)$ we established. Furthermore, since we execute a progress step only when condition (24) is satisfied, this means that our desired lower bound (21) on $\hat{\delta}$ holds. Therefore, by our reasoning in Section 3.3, we know that the number of progress steps is at most $\widetilde{O}\left(\hat{\delta}^{-1} \log mU\right) = \widetilde{O}\left(m^{\frac{1}{2}-\eta} \log mU\right)$. As each boosting and progress step can be executed in nearly-linear time, our desired improved running time bound will then follow.

In the light of the above, it remains to only argue that claims (I) and (II) are indeed correct and that the Invariant 4.2 is never violated.

Preservation of the Invariant 4.2. To prove that the Invariant 4.2 is indeed preserved, let us recall that each arc boosting operation increases the number of arcs by $\beta(e) - 1 \leq \frac{2U}{\widehat{u}_e(\boldsymbol{f})}$, where $\widehat{u}_e(\boldsymbol{f})$ is the residual capacity of the arc e at the time the boosting occurred. As it turns out, since we only boost arcs that are high-energy, i.e., arcs e with $|\rho_e| \geq \rho^*$ at the time of the boosting, one can use a simple vertex potential based argument to argue that $\beta(e)$ has to be always at most $O(m^{4\eta}U)$.

Lemma 4.3. *Assume that Invariant 4.2 holds, let $\widetilde{\boldsymbol{f}}$ be an electrical $\boldsymbol{\chi}$-flow determined by the resistances $\boldsymbol{r}$ and let $e = (u, v)$ be a high-energy arc that was boosted. Then, the total arc number increase $\beta(e) - 1$ is at most $O(m^{4\eta}U)$.*

Proof. Recall that $\beta(e) - 1 \leq \frac{2U}{\widehat{u}_e(\boldsymbol{f})}$. So, it suffices that we prove that $\widehat{u}_e(\boldsymbol{f}) \geq \Omega(m^{-4\eta})$. To this end, let $\widetilde{\boldsymbol{\phi}}$ be vertex potentials inducing the electrical flow $\widetilde{\boldsymbol{f}}$ and let $R_{s,t}$ be the effective resistance between the source s and sink t in our graph. As $\widetilde{\boldsymbol{f}}$ is an s-t flow of value F, we need to have that

$$F^2 R_{s,t} = \mathcal{E}_r(\widetilde{\boldsymbol{f}}) \leq \frac{C_{\mathcal{E}} m}{(1-\alpha)^2},$$

where we used Lemma 3.9 and the fact that Invariant 4.2 holds.

On the other hand, we know that the vertex potential drop $\widetilde{\phi}_t - \widetilde{\phi}_s$ between s and t is non-negative (as the electrical flow has to flow from s to t) and at most

$$\widetilde{\phi}_t - \widetilde{\phi}_s = F R_{s,t} \leq \frac{C_{\mathcal{E}} m}{(1-\alpha)^2 F} \leq \frac{C_{\mathcal{E}} m^{\frac{1}{2}+\eta}}{(1-\alpha)},$$

where the last inequality follows as $(1-\alpha)F \geq m^{\frac{1}{2}-\eta}$ due to our termination condition (22).

However, as $\widetilde{\boldsymbol{f}}$ is an s-t flow it must be the case that the vertex potential difference between the endpoints of our arc $e = (u, v)$ cannot be larger than such difference between the vertex potentials of s and t. That is, we have that

$$|\widetilde{\phi}_v - \widetilde{\phi}_u| \leq |\widetilde{\phi}_t - \widetilde{\phi}_s| \leq \frac{C_{\mathcal{E}} m^{\frac{1}{2}+\eta}}{(1-\alpha)}. \tag{25}$$

But, by Ohm's law (5) and definition (13) of the resistance $\boldsymbol{r}$, we have that

$$|\widetilde{\phi}_v - \widetilde{\phi}_u| = |\widetilde{f}_e| r_e \geq \frac{|\widetilde{f}_e|}{\widehat{u}_e(\boldsymbol{f})^2} = \frac{|\rho_e|}{\widehat{u}_e(\boldsymbol{f})} \geq \frac{\rho^*}{\widehat{u}_e(\boldsymbol{f})} \geq \frac{m^{\frac{1}{2}-3\eta}}{C_{\rho^*}(1-\alpha)\widehat{u}_e(\boldsymbol{f})},$$

22

where we used the definition (23) of the high-energy arcs.

So, putting this inequality together with the bound (25), gives us the desired lower bound on $\widehat{u}_e(\boldsymbol{f})$. The lemma follows. $\qquad\square$

Observe now that as our boosting operation never boosts more than $k^* = m^{4\eta}$ arcs at a time and we have at most $\widetilde{O}\left(m^{\frac{1}{2}-\eta}\log U\right)$ boosting steps, by Lemma 4.3, the total arc number increase throughout the whole algorithm is at most

$$m^{4\eta} \cdot \widetilde{O}\left(m^{\frac{1}{2}-\eta}\log U\right) \cdot O(m^{4\eta}U) = \widetilde{O}\left(m^{\frac{1}{2}+7\eta}U\right) < \frac{m}{10},$$

provide we set $\eta := \frac{1}{14} - \frac{1}{7}\log_m U - O(\log\log mU)$ with appropriately chosen constant in the $O(\log\log mU)$ term. Thus Invariant 4.2 is indeed never violated.

Establishing Claim (I). We want now to prove that each boosting step results in an increase of the energy $\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})$ by a factor of at least $\left(1 + \Omega(m^{-2\eta})\right)$. In the light of Lemma 4.1, it suffices we show that if S^* is the set of high-energy arcs that got boosted due to the condition (24) not holding then these arcs had to have a large contribution to the (square) of the ℓ_2-norm of the congestion vector $\boldsymbol{\rho}$. That is, we want to argue that

$$\sum_{e \in S^*} \frac{\rho_e^2}{\|\boldsymbol{\rho}\|_2^2} \geq \Omega(m^{-2\eta}). \tag{26}$$

Clearly, once we establish that then the desired energy increase will follow from Lemma 4.1 applied to $S = S^*$.

To prove (26), let us first consider the case that S^* has maximum size, i.e., $|S^*| = k^* = m^{4\eta}$. Note that all the arcs in S^* are high-energy arcs. Therefore, (23) gives us in this case that

$$\sum_{e \in S^*} \frac{\rho_e^2}{\|\boldsymbol{\rho}\|_2^2} \geq m^{4\eta} \cdot \frac{(\rho^*)^2}{\|\boldsymbol{\rho}\|_2^2} \geq \frac{m^{1-2\eta}}{C_{\rho^*}^2(1-\alpha)^2\|\boldsymbol{\rho}\|_2^2} \geq \Omega(m^{-2\eta}),$$

where we used the fact that Lemma 3.9 implies that $\|\boldsymbol{\rho}\|_2^2 \leq \frac{C_{\mathcal{E}}m}{(1-\alpha)^2}$. So, (26) holds in this case and we can focus on the situation when $|S^*| < m^{4\eta}$. Observe that this means that there is no high-energy arcs that are not in S^*.

We want to argue that in such situation the arcs of S^* contribute at least a half of the total (cubed) ℓ_3-norm of the vector $\boldsymbol{\rho}$. That is, that we have that

$$\sum_{e \in S^*} |\rho_e|^3 \geq \left(\frac{m^{\frac{1}{2}-\eta}}{66(1-\alpha)}\right)^3. \tag{27}$$

To this end, observe that the contribution of arcs that are not high energy (and thus not in S^*) to that (cubed) ℓ_3-norm is, by the Cauchy-Schwartz inequality, (23) and Lemma 3.9, at most

$$\left(\sum_{e \notin S^*} |\rho_e|^3\right) \leq \left(\max_{e \notin S^*} |\rho_e|\right)\left(\sum_{e \notin S^*} \rho_e^2\right) \leq \rho^*\|\boldsymbol{\rho}\|_2^2 \leq \frac{C_{\mathcal{E}}m^{\frac{3}{2}-3\eta}}{C_{\rho^*}(1-\alpha)^3} \leq \left(\frac{m^{\frac{1}{2}-\eta}}{66(1-\alpha)}\right)^3, \tag{28}$$

provided the constant C_{ρ^*} is set to be large enough. Therefore, since the condition (24) does not hold, we must have that the inequality (27) is indeed valid.

Now, if there exists an arc $e \in S^*$ such that $|\rho_e| \geq \frac{m^{\frac{1}{2}-\eta}}{(1-\alpha)}$ then the contribution of this arc alone suffices to make (26) true. We can thus assume that $\max_{e \in S^*} |\rho_e| < \frac{m^{\frac{1}{2}-\eta}}{(1-\alpha)}$ and apply the Cauchy-Schwartz inequality to (27) to obtain that

$$\sum_{e \in S^*} \rho_e^2 \geq \frac{\sum_{e \in S^*} |\rho_e|^3}{\max_{e \in S^*} |\rho_e|} \geq \frac{1}{66} \left(\frac{m^{\frac{1}{2}-\eta}}{66(1-\alpha)} \right)^2,$$

which, by Lemma 3.9, establishes (26). Claim (I) is thus proved.

Establishing Claim (II). To prove Claim (II) we will tie the potential decrease of energy during progress steps to the quantity that we already control: the ℓ_3-norm of the congestion vector $\boldsymbol{\rho}$. (In fact, the desire to control the energy decrease is precisely the reason why we enforce the stronger ℓ_3-norm condition (24) instead of the ℓ_4-norm condition that Lemma 3.7 suggests.)

We make this connection precise in the following lemma, whose proof appears in Appendix A.5.

Lemma 4.4. *Let $(\boldsymbol{f}, \boldsymbol{y})$ be a well-coupled feasible primal dual solution and let $\widetilde{\boldsymbol{f}}$ be the corresponding electrical $\boldsymbol{\chi}$-flow determined by the resistances $\boldsymbol{r}$ and congestion vector $\boldsymbol{\rho}$ that satisfies the ℓ_3-norm condition (24). Then, after execution of the progress steps, as described in Section 3.2, the energy $\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})$ decreases by a factor of at most*

$$\left(1 + O \left(\frac{\|\boldsymbol{\rho}\|_3^2}{\|\boldsymbol{\rho}\|_2^2} \right) \right).$$

Observe that, by (24) and Lemma 3.9, we have that

$$\frac{\|\boldsymbol{\rho}\|_3^2}{\|\boldsymbol{\rho}\|_2^2} \leq \frac{m^{1-2\eta}(1-\alpha)^2}{(33)^2 C_{\mathcal{E}} m(1-\alpha)^2} = O(m^{-2\eta}),$$

which immediately establishes Claim (II) and thus concludes the analysis of the improved algorithm.

A Appendix

A.1 Proof of Fact 3.4

Let us define

$$g(x) := \left(\frac{1}{u_1 - x} - \frac{1}{u_2 + x} \right).$$

By Taylor's theorem, we have that

$$g(x) = g(0) + g'(0)x + g''(z)\frac{x^2}{2} = \frac{1}{u_1} - \frac{1}{u_2} + \left(\frac{1}{u_1^2} + \frac{1}{u_2^2} \right) x + x^2 \left(\frac{1}{(u_1 - z)^3} - \frac{1}{(u_2 + z)^3} \right) \quad (29)$$

where z is some value with $|z| \le |x|$. As a result, we can conclude that

$$|\zeta| = \left| \left(\frac{1}{(u_1 - z)^3} - \frac{1}{(u_2 + z)^3} \right) \right| \le \left(\frac{64}{27u_1^3} + \frac{64}{27u_2^3} \right) \le \frac{5}{u^3}, \quad (30)$$

as we wanted to show.

A.2 Proof of Lemma 3.5

By the definition of our update (14) and the approximation bound (18) as well as the coupling condition (11), we have that

$$
\begin{aligned}
\left| \Delta_e(\widehat{\boldsymbol{y}}) - \Phi_e(\widehat{\boldsymbol{f}}) \right| &= \left| \Delta_e(\boldsymbol{y}) + \delta(\widetilde{\phi}_v - \widetilde{\phi}_u) - \Phi(\boldsymbol{f}) - \left(\frac{1}{(\widehat{u}_e^+)^2} + \frac{1}{(\widehat{u}_e^-)^2} \right) \delta\widetilde{f}_e - (\delta\widetilde{f}_e)^2 \zeta_e \right| \\
&\le \left| \delta\left((\widetilde{\phi}_v - \widetilde{\phi}_u) - r_e\widetilde{f}_e \right) - (\delta\widetilde{f}_e)^2 \zeta_e \right| + \frac{\gamma_e}{\widehat{u}_e(\boldsymbol{f})} \\
&= \left| (\delta\widetilde{f}_e)^2 \zeta_e \right| + \frac{\gamma_e}{\widehat{u}_e(\boldsymbol{f})} \le \frac{7\delta^2 \frac{\widetilde{f}_e^2}{\widetilde{u}_e^2} + \frac{4}{3}\gamma_e}{\widehat{u}_e(\widehat{\boldsymbol{f}})} = \frac{7(\delta\rho_e)^2 + \frac{4}{3}\gamma_e}{\widehat{u}_e(\widehat{\boldsymbol{f}})},
\end{aligned}
$$

where the second to last equality follows by Ohm's law (6), the last equality follows by (16) and we also used the fact that $\widehat{u}_e(\widehat{\boldsymbol{f}}) \ge \frac{3}{4}\widehat{u}_e(\boldsymbol{f})$ since $\delta \le (4\|\boldsymbol{\rho}\|_\infty)^{-1}$. The lemma follows.

A.3 Proof of Lemma 3.6

For each arc e, let us define

$$\theta_e := \left(\frac{1}{(\widehat{u}_e^+(\boldsymbol{g}))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{g}))^2} \right)^{-1} (\Delta_e(\boldsymbol{z}) - \Phi_e(\boldsymbol{g})). \quad (31)$$

Intuitively, θ_e is the first-order correction to the coupling of the primal dual solution $(\boldsymbol{g}, \boldsymbol{z})$ with respect to the arc e.

Note that by the fact that $(\boldsymbol{g}, \boldsymbol{z})$ is ς-coupled we have that

$$|\theta_e| \le \left(\frac{1}{(\widehat{u}_e^+(\boldsymbol{g}))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{g}))^2} \right)^{-1} |\Delta_e(\boldsymbol{z}) - \Phi_e(\boldsymbol{g})| \le \widehat{u}_e(\boldsymbol{g})^2 \left(\frac{\varsigma_e}{\widehat{u}_e(\boldsymbol{g})} \right) = \varsigma_e\widehat{u}_e(\boldsymbol{g}). \quad (32)$$

Let us now define a flow $\boldsymbol{g}'$ to be the flow $\boldsymbol{g}$ with the "correcting" flow $\boldsymbol{\theta}$ added. That is,

$$g'_e := g_e + \theta_e,$$

for each arc e. Observe that, as $\|\boldsymbol{\varsigma}\|_\infty \le \|\boldsymbol{\varsigma}\|_2 \le \frac{1}{50}$ and, by (32), $|\theta_e| \le \varsigma_e \widehat{u}_e(\boldsymbol{g})$, this flow $\boldsymbol{g}'$ is feasible in $G_{\boldsymbol{g}}$ and, in fact, $\frac{51}{50}\widehat{u}_e(\boldsymbol{g}) \ge \widehat{u}_e(\boldsymbol{g}') \ge \frac{49}{50}\widehat{u}_e(\boldsymbol{g})$.

Furthermore, if we analyze the coupling of $(\boldsymbol{g}', \boldsymbol{z})$, by Fact 3.4, for each arc e, we get that

$$
\begin{aligned}
\left| \Delta_e(\boldsymbol{z}) - \Phi_e(\boldsymbol{g}') \right| &= \left| \Delta_e(\boldsymbol{z}) - \left(\frac{1}{\widehat{u}_e^+(\boldsymbol{g}) - \theta_e} - \frac{1}{\widehat{u}_e^-(\boldsymbol{g}) + \theta_e} \right) \right| \\
&= \left| \Delta_e(\boldsymbol{z}) - \Phi_e(\boldsymbol{g}) - \left(\frac{1}{(\widehat{u}_e^+(\boldsymbol{g}))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{g}))^2} \right) \theta_e - \theta_e^2 \zeta_e' \right| \\
&= \left| -\theta_e^2 \zeta_e' \right| \le \frac{5\widehat{u}_e(\boldsymbol{g})^2 \varsigma_e^2}{\widehat{u}_e(\boldsymbol{g})^3} = \frac{5\varsigma_e^2}{\widehat{u}_e(\boldsymbol{g})} \le \frac{51\varsigma_e^2}{10\widehat{u}_e(\boldsymbol{g}')},
\end{aligned}
$$

where the third equality follows by the definition of θ_e (31) and the second to last inequality follows by (32).

We thus see that the solutions $(\boldsymbol{g}', \boldsymbol{z})$ are well-coupled. In fact, they are $\widetilde{\boldsymbol{\gamma}}$-coupled with

$$\|\widetilde{\boldsymbol{\gamma}}\|_2 = \sqrt{\sum_e \widetilde{\gamma}_e^2} = \sqrt{\sum_e \left(\frac{11\varsigma_e^2}{2} \right)^2} \le \frac{51}{10}\|\boldsymbol{\varsigma}\|_2^2 \le \frac{51}{25000}. \tag{33}$$

So, these solutions $(\boldsymbol{g}', \boldsymbol{z})$ have all the desired properties except $\boldsymbol{g}'$ is an $(\boldsymbol{\chi}_{\alpha'} + \widehat{\boldsymbol{\sigma}})$-flow, where $\widehat{\boldsymbol{\sigma}}$ are the demands of the "correcting" flow $\boldsymbol{\theta}$ we added – and not a $\boldsymbol{\chi}_{\alpha'}$-flow we need.

To remedy this deficiency, we compute an electrical $(-\widehat{\boldsymbol{\sigma}})$-flow $\widehat{\boldsymbol{\theta}}$ determined by the to resistances

$$\widehat{r}_e := \frac{1}{(\widehat{u}_e^+(\boldsymbol{g}'))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{g}'))^2}, \tag{34}$$

and obtain our desired solutions $(\bar{\boldsymbol{g}}, \bar{\boldsymbol{z}})$ by adding this electrical flow to the flow $\boldsymbol{g}'$, and simultaneously adding the vertex potentials $\widehat{\boldsymbol{\phi}}$ that induce $\widehat{\boldsymbol{\theta}}$ to the dual solution $\boldsymbol{z}$. In other words, we set

$$
\begin{aligned}
\bar{\boldsymbol{g}} &:= \boldsymbol{g}' + \widehat{\boldsymbol{\theta}} \\
\bar{\boldsymbol{z}} &:= \boldsymbol{z} + \widehat{\boldsymbol{\phi}}.
\end{aligned}
\tag{35}
$$

Clearly, $\bar{\boldsymbol{g}}$ is a $\boldsymbol{\chi}_{\alpha'}$-flow, as desired. It thus only remains to ensure that the solutions $(\bar{\boldsymbol{g}}, \bar{\boldsymbol{z}})$ are still well-coupled.

To analyze the coupling of $(\bar{\boldsymbol{g}}, \bar{\boldsymbol{z}})$ one should note first that the procedure we used to obtain these solutions from the solutions $(\boldsymbol{g}', \boldsymbol{z})$ is analogous to the one we used obtain the solutions $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}})$ from the solutions $(\boldsymbol{f}, \boldsymbol{y})$ in our augmentation step. More precisely, the update (35) is an exact analogue of the update (14) with $\delta = 1$.

So, to quantify the change in coupling we can follow the approach we used when analyzing the augmentation step. Specifically, let us define a congestion vector $\widetilde{\boldsymbol{\rho}}$ as

$$\widehat{\rho}_e := \frac{\widehat{\theta}_e}{\widehat{u}_e(\boldsymbol{g}')},$$

for each arc e. By Lemma 3.3, we know that

$$\|\widetilde{\boldsymbol{\rho}}\|_2^2 \leq \mathcal{E}_{\widehat{\boldsymbol{r}}}(\widehat{\boldsymbol{\theta}}) \leq \mathcal{E}_{\widehat{\boldsymbol{r}}}(-\boldsymbol{\theta}),$$

where the second inequality follows by the energy minimizing property of electrical flows. (Note that $-\boldsymbol{\theta}$ is a $(-\widehat{\boldsymbol{\sigma}})$-flow.) We thus have that, by (34),

$$
\begin{aligned}
\mathcal{E}_{\widehat{\boldsymbol{r}}}(-\boldsymbol{\theta}) &= \sum_e \widehat{r}_e(-\theta_e)^2 = \sum_e \left(\frac{1}{(\widehat{u}_e^+(\boldsymbol{g}'))^2} + \frac{1}{(\widehat{u}_e^-(\boldsymbol{g}'))^2} \right) \theta_e^2 \\
&\leq \sum_e \frac{2}{\widehat{u}_e(\boldsymbol{g}')^2} \left(\varsigma_e \widehat{u}_e(\boldsymbol{g}) \right)^2 \leq \sum_e 2 \left(\frac{51\varsigma_e}{50} \right)^2 \leq \frac{21}{20} \|\varsigma\|_2^2 \leq \frac{1}{2000},
\end{aligned}
$$

where the first inequality follows by (32). As a result, we can conclude that

$$\|\widetilde{\boldsymbol{\rho}}\|_2^2 \leq \mathcal{E}_{\widehat{\boldsymbol{r}}}(-\boldsymbol{\theta}) \leq \frac{1}{2000}, \tag{36}$$

which ensures, in particular, that our analogue of condition (17) (for $\delta = 1$) is satisfied.

Consequently, we can use Lemma 3.5 to conclude that the solutions $(\bar{\boldsymbol{g}}, \bar{\boldsymbol{z}})$ is $\widehat{\boldsymbol{\gamma}}$-coupled with

$$\|\widehat{\boldsymbol{\gamma}}\|_2 = \sqrt{\sum_e \widehat{\gamma}_e^2} \leq \sqrt{\sum_e \left(\frac{4}{3}\widetilde{\gamma}_e + 7\widetilde{\rho}_e^2 \right)^2} \leq \frac{4}{3}\|\widetilde{\boldsymbol{\gamma}}\|_2 + 7\|\widetilde{\boldsymbol{\rho}}\|_2^2 \leq \frac{204}{75000} + \frac{7}{2000} < \frac{1}{100},$$

where we used the estimates (33) and (36). So, $(\bar{\boldsymbol{g}}, \bar{\boldsymbol{z}})$ is well-coupled, as desired.

A.4   Proof of Lemma 3.8

Let us assume that $\boldsymbol{\chi}^T \boldsymbol{y} \leq \frac{2m}{(1-\alpha)}$. Otherwise, by Lemma 3.1, we have a dual certificate that the target demand $\boldsymbol{\chi}$ cannot be routed in G, i.e., the condition (b) triggers.

As all the preconditioning arcs have s and t as their endpoints and are oriented from s to t, we have that

$$\Delta_e(\boldsymbol{y}) = y_t - y_s = \frac{\boldsymbol{\chi}^T \boldsymbol{y}}{F} \leq \frac{2m}{(1-\alpha)F}, \tag{37}$$

for each such arc e. Also, all these preconditioning arcs are completely indistinguishable from the point of view of our algorithm. They have the same endpoints, orientation, and initial (residual) capacities and stretch. Consequently, their resistances are always equal when we compute our electrical flows (see, e.g., (13)) and thus, in turn, their forward and backward residual capacities evolve in exactly the same way during each of the augmentation and fixing steps (see Section 3.2).

We want to argue that, for each such arc e,

$$\widehat{u}_e(\boldsymbol{f}) \geq \frac{(1-\alpha)F}{5m}. \tag{38}$$

As the residual capacity $\widehat{u}_e(\boldsymbol{f})$ is exactly the capacity of e in the symmetrization $\widehat{G}_{\boldsymbol{f}}$ of the residual graph $G_{\boldsymbol{f}}$, and preconditioning arc constitute exactly half of all the m arcs, a flow $\boldsymbol{f}'$ that just routes $\frac{(1-\alpha)F}{5m}$ units of flow from s to t on each such arc will be the desired $\boldsymbol{\chi}_{\frac{(1-\alpha)}{10}}$-flow feasible in $\widehat{G}_{\boldsymbol{f}}$, and our proof will follow.

To establish (38), let us fix some representative preconditioning arc e and assume for the sake of contradiction that (38) does not hold. Since the total preconditioning arc capacity constitutes at least $\frac{2}{3}$-fraction of the total s-t capacity F^* of the graph G, we must have that

$$u_e^+ = u_e^- \geq \frac{2}{m} \cdot \frac{2}{3} F^* = \frac{4F^*}{3m}, \tag{39}$$

where u_e^+ and u_e^- are the initial capacities of the preconditioning arcs. As a result, if (38) did not hold, by (3), we know that

$$|f_e| \geq \max\{u_e^+ - \widehat{u}_e^+(\boldsymbol{f}), u_e^- - \widehat{u}_e^-(\boldsymbol{f})\} \geq \frac{4F^*}{3m} - \frac{(1-\alpha)F}{5m} \geq \frac{\frac{5}{2}F^* + \alpha F}{3m} > \frac{2F^*}{3m},$$

where we used the fact that $F \leq \frac{3}{2}F^*$ as otherwise we would be able to immediately tell that it is impossible to route at least $\frac{2}{3}$-fraction of the s-t flow only via preconditioning arcs.

In fact, we also know that $f_e > 0$. This is so, as e is oriented from s to t and thus having $f_e < 0$ would mean that there is a flow of strictly more than $\frac{F^*}{3}$ from t towards s over all the preconditioning arcs. It would, however, be impossible to balance this "backward" t-s flow via an s-t flow over the remaining arcs as they can support an s-t flow of at most $\frac{F^*}{3}$. So, indeed $f_e > 0$.

Furthermore, if (38) would indeed not hold then $f_e > 0$, (3) and (39) would imply that

$$\widehat{u}_e^+ = \widehat{u}_e(\boldsymbol{f}) < \frac{(1-\alpha)F}{3m} \leq \frac{F^*}{2m} \leq \frac{1}{2}u_e^- = \frac{1}{2}(\widehat{u}_e^-(\boldsymbol{f}) - f_e) \leq \frac{\widehat{u}_e^-(\boldsymbol{f})}{2},$$

where we again used the fact that $F \leq \frac{3}{2}F^*$.

Now, combining the above bound with the fact that $(\boldsymbol{f}, \boldsymbol{y})$ is well-coupled (11), we can conclude that, for any preconditioning arc e,

$$
\begin{aligned}
\Delta_e(\boldsymbol{y}) \;\geq\; & \Phi_e(\boldsymbol{f}) - \frac{\gamma_e}{\widehat{u}_e(\boldsymbol{f})} = \frac{1}{\widehat{u}_e^+(\boldsymbol{f})} - \frac{1}{\widehat{u}_e^-(\boldsymbol{f})} - \frac{\gamma_e}{\widehat{u}_e(\boldsymbol{f})} \geq \frac{(1-\gamma_e)}{\widehat{u}_e(\boldsymbol{f})} - \frac{1}{2\widehat{u}_e(\boldsymbol{f})} \\
\;=\; & \frac{(1-2\gamma_e)}{2\widehat{u}_e(\boldsymbol{f})} \geq \frac{5m(1 - \frac{2}{\sqrt{m}})}{2(1-\alpha)F} > \frac{2m}{(1-\alpha)F},
\end{aligned}
$$

where we again used the assumption that (38) does not hold as well as the fact that $\gamma_e \leq \frac{1}{\sqrt{m}}$ since otherwise the contribution of all the $\frac{m}{2}$ preconditioning arcs to the ℓ_2-norm of the violation vector $\boldsymbol{\gamma}$ alone would make this norm violate the well-coupling bound. However, the above lower bound on $\Delta_e(\boldsymbol{y})$ directly violates our bound (37). So, we reached a contradiction that proves (38) and thus our lemma follows.

A.5   Proof of Lemma 4.4

By examining the augmentation and fixing steps that are performed during the progress steps (cf. (14) and (35) in the proof of Lemma 3.6) and applying standard Taylor approximations to the resulting changes in the resistances $\boldsymbol{r}$ (cf. (13)), one obtains that, for each arc e, the resulting arc resistance r_e' change is such that

$$r_e' \leq (1 + O(\delta|\rho_e| + \kappa_e))^{-1} r_e, \tag{40}$$

where $\boldsymbol{\rho}$ is the congestion vector (16), $\delta = (33\|\boldsymbol{\rho}\|_3)^{-1}$ is the step size, and $\boldsymbol{\kappa}$ is a vector such that $\|\boldsymbol{\kappa}\|_2 \leq 1$. (Roughly speaking, the $\delta\rho_e$ term corresponds to augmentation step and the κ_e term corresponds to the fixing step.)

Now, let $\widetilde{\phi}$ be the vertex potentials inducing the electrical $\boldsymbol{\chi}$-flow $\widetilde{\boldsymbol{f}}$. By Lemma 2.2, we know that

$$\frac{1}{\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})} = \sum_{e=(u,v)} \frac{(\widetilde{\phi}_v - \widetilde{\phi}_u)^2}{r_e \mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})^2}.$$

Furthermore, if $\mathcal{E}_{\boldsymbol{r}'}(\widetilde{\boldsymbol{f}}')$ is the energy of an electrical $\boldsymbol{\chi}$-flow $\widetilde{\boldsymbol{f}}'$ determined by the new resistances $\boldsymbol{r}'$, applying Lemma 2.2 again, we obtain that

$$
\begin{aligned}
\frac{1}{\mathcal{E}_{\boldsymbol{r}'}(\widetilde{\boldsymbol{f}}')} &\leq \sum_{e=(u,v)} \frac{(\widetilde{\phi}_v - \widetilde{\phi}_u)^2}{r_e' \mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})^2} \leq \sum_{e=(u,v)} \frac{(1 + O(\delta|\rho_e| + \kappa_e))\,(\widetilde{\phi}_v - \widetilde{\phi}_u)^2}{r_e \mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})^2} \\
&= \frac{1}{\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})} \left(1 + O\left(\sum_e \frac{(\delta\rho_e + \kappa_e)(r_e \widetilde{f}_e^2)}{\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})}\right)\right) \\
&\leq \frac{1}{\mathcal{E}_{\boldsymbol{r}}(\widetilde{\boldsymbol{f}})} \left(1 + O\left(\sum_e \frac{(\delta\rho_e^3 + \kappa_e \rho_e^2)}{\|\boldsymbol{\rho}\|_2^2}\right)\right),
\end{aligned}
$$

where we used (40), Ohm's law (5) and Lemmas 3.3 and 3.9.

Finally, observe that since $\delta = (33\|\boldsymbol{\rho}\|_3)^{-1}$ and $\|\boldsymbol{\kappa}\|_2 \leq 1$, an application of the Cauchy-Schwartz inequality gives us that

$$\sum_e (\delta\rho_e^3 + \kappa_e \rho_e^2) \leq \delta\|\boldsymbol{\rho}\|_3^3 + \|\boldsymbol{\kappa}\|_2\|\boldsymbol{\rho}\|_4^2 \leq \delta\|\boldsymbol{\rho}\|_3^3 + \|\boldsymbol{\kappa}\|_2\|\boldsymbol{\rho}\|_3^2 \leq O(\|\boldsymbol{\rho}\|_3^2),$$

as desired. The lemma thus follows.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, 1993.

[2] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, and M. R. Reddy. *Applications of Network Optimization*, volume 7 of *Handbooks in Operations Research and Management Science*. North-Holland, 1995.

[3] H. Alt, N. Blum, K. Mehlhorn, and M. Paul. Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}\sqrt{m/\log n})$. *Inf. Process. Lett.*, 37(4):237–240, 1991.

[4] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[5] B. Bollobas. *Modern Graph Theory*. Springer, 1998.

[6] J. R. Bunch and J. E. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(125):231–236, 1974.

[7] P. Christiano, J. Kelner, A. Mądry, D. Spielman, and S.-H. Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *STOC'11: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 273–281, 2011.

[8] M. B. Cohen, R. Kyng, G. L. Miller, J. W. Pachocki, R. Peng, A. B. Rao, and S. C. Xu. Solving SDD linear systems in nearly m $\log^{1/2} n$ time. In *STOC'14: Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 343–352, 2014.

[9] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251–280, 1990.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

[11] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[12] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972.

[13] J. Egerváry. Matrixok kombinatorius tulajdonságairól. *Matematikai és Fizikai Lapok*, 38:16–28, 1931.

[14] P. Elias, A. Feinstein, and C. E. Shannon. A note on the maximum flow through a network. *IRE Transactions on Information Theory*, 2, 1956.

[15] S. Even and R. E. Tarjan. Network flow and testing graph connectivity. *SIAM Journal on Computing*, 4(4):507–518, 1975.

[16] T. Feder and R. Motwani. Clique partitions, graph compression and speeding-up algorithms. *Journal of Computer and System Sciences*, 51(2):261Ű–272, 1995.

[17] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[18] F. G. Frobenius. Über matrizen aus nicht negativen elementen. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, pages 456–477, 1912.

[19] F. G. Frobenius. Über zerlegbare determinanten. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, pages 274–277, 1917.

[20] H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for general graph matching problems. *Journal of the ACM*, 38(4):815–853, 1991.

[21] A. V. Goldberg and A. V. Karzanov. Maximum skew-symmetric flows and matchings. *Mathematical Programming*, 100(3):537–568, 2004.

[22] A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 45(5):783–797, 1998.

[23] N. J. A. Harvey. Algebraic algorithms for matching and matroid problems. *SIAM Journal on Computing*, 39(2):679–702, 2009.

[24] J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[25] A. V. Karzanov. O nakhozhdenii maksimal'nogo potoka v setyakh spetsial'nogo vida i nekotorykh prilozheniyakh. *Matematicheskie Voprosy Upravleniya Proizvodstvom*, 5:81–94, 1973. (in Russian; title translation: On finding maximum flows in networks with special structure and some applications).

[26] J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *SODA'14: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 217–226, 2014.

[27] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. In *STOC'13: Proceedings of the 45th Annual ACM Symposium on the Theory of Computing*, pages 911–920, 2013.

[28] V. King, S. Rao, and R. Tarjan. A faster deterministic maximum flow algorithm. *Journal of Algorithms*, 17(3):447–474, 1994.

[29] D. König. Vonalrendszerek és determinánsok. *Mathematikai és Természettudományi Értesitö*, 33:221–229, 1915.

[30] D. König. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen*, 77:453–465, 1916.

[31] D. König. Sur un probléme de la théorie générale des ensembles et la théorie des graphes. *Revue de Métaphysique et de Morale*, 30:443–449, 1923.

[32] D. König. Graphok és matrixok. *Matematikai és Fizikai Lapok*, 38:116–119, 1931.

[33] I. Koutis, G. L. Miller, and R. Peng. Approaching optimality for solving SDD systems. In *FOCS'10: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 235–244, 2010.

[34] I. Koutis, G. L. Miller, and R. Peng. A nearly $m \log n$-time solver for SDD linear systems. In *FOCS'11: Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, pages 590–598, 2011.

[35] R. Kyng, Y. T. Lee, R. Peng, S. Sachdeva, and D. A. Spielman. Sparsified cholesky and multigrid solvers for connection Laplacians. In *STOC'16: Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, 2016.

[36] R. Kyng and S. Sachdeva. Approximate Gaussian elimination for Laplacians: Fast, sparse, and simple. In *FOCS'16: Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, 2016.

[37] Y. T. Lee, S. Rao, and N. Srivastava. A new approach to computing maximum flows using electrical flows. In *STOC'13: Proceedings of the 45th Annual ACM Symposium on the Theory of Computing*, pages 755–764, 2013.

[38] Y. T. Lee and A. Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{rank})$ iterations and faster algorithms for maximum flows. In *FOCS'14: Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 424–433, 2014.

[39] L. Lovász. On determinants, matchings and random algorithms. *Fundamentals of Computation Theory*, 565–574, 1979.

[40] L. Lovász and D. M. Plummer. *Matching Theory*. Elsevier Science, 1986.

[41] A. Mądry. Fast approximation algorithms for cut-based problems in undirected graphs. In *FOCS'10: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 245–254, 2010.

[42] A. Mądry. *From Graphs to Matrices, and Back: New Techniques for Graph Algorithms*. PhD thesis, Massachusetts Institute of Technology, 2011.

[43] A. Mądry. Navigating central path with electrical flows: from flows to matchings, and back. In *FOCS'13: Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 253–262, 2013.

[44] S. Micali and V. V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algoithm for finding maximum matching in general graphs. In *FOCS'80: Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science*, pages 17–27, 1980.

[45] M. Mucha. *Finding maximum matchings via Gaussian elimination*. PhD thesis, University of Warsaw, 2005.

[46] M. Mucha and P. Sankowski. Maximum matchings via Gaussian elimination. In *FOCS'04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 248–255, 2004.

[47] J. B. Orlin. Max flows in O(nm) time, or better. In *STOC'13: Proceedings of the 45th Annual ACM Symposium on the Theory of Computing*, pages 765–774, 2013.

[48] R. Peng. Approximate undirected maximum flows in $O(mpolylog(n))$ time. In *SODA'16: Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1862–1867, 2016.

[49] M. O. Rabin and V. V. Vazirani. Maximum matchings in general graphs through randomization. *J. Algorithms*, 10(4):557–567, Dec. 1989.

[50] A. Schrijver. On the history of the transportation and maximum flow problems. *Mathematical Programming*, 91:437–445, 2002.

[51] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[52] A. Schrijver. On the history of combinatorial optimization (till 1960). In R. W. K. Aardal, G.L. Nemhauser, editor, *Handbook of Discrete Optimization*, pages 1–68. Elsevier, 2005.

[53] J. Sherman. Breaking the multicommodity flow barrier for $O(\sqrt{\log n})$-approximations to sparsest cuts. In *FOCS'09: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 363–372, 2009.

[54] J. Sherman. Nearly maximum flows in nearly linear time. In *FOCS'13: Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 263–269, 2013.

[55] D. A. Spielman and S.-H. Teng. Solving sparse, symmetric, diagonally-dominant linear systems in time $O(m^{1.31})$. In *FOCS'03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 416–427, 2003.

[56] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC'04: Proceedings of the 36th Annual ACM Symposium on the Theory of Computing*, pages 81–90, 2004.

[57] W. T. Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 22:107–111, 1947.

[58] P. M. Vaidya. Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners. Unpublished manuscript, UIUC 1990. A talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation, October 1991, Mineapolis.

[59] V. Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *STOC'12: Proceedings of the 44th Annual ACM Symposium on the Theory of Computing*, pages 887–898, 2012.

[60] V. V. Vazirani. A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{|V|}|E|)$ general graph matching algorithms. *Combinatorica*, 14 (1):71–109, 1994.