Interior Point Methods for Maximum Flow

Rasmus Kyng *Lecture 14 — Wednesday, May 13th*

Notation

In this lecture, we're going to frequently need to refer to vectors arising from elementwise operations combining other vectors.

To that end, given two vector $\boldsymbol{a} \in \mathbb{R}^m$, and $\boldsymbol{b} \in \mathbb{R}^m$, we will use $\overrightarrow{(\boldsymbol{a}(i)\boldsymbol{b}(i))}$ to denote the vector $\boldsymbol{z}$ with $\boldsymbol{z}(i) = \boldsymbol{a}(i)\boldsymbol{b}(i)$ and so on.

Throughout this lecture, when we are working in the context of some given graph G with vertices V and edges E, we will let $m = |E|$ and $n = |V|$.

1 An Interior Point Method

The Maximum Flow problem in undirected graphs.

$$\begin{aligned} \max_{\boldsymbol{f} \in \mathbb{R}^E} \quad & F \qquad\qquad (1) \\ \text{s.t. } \boldsymbol{B}\boldsymbol{f} &= F\boldsymbol{b}_{st} \qquad \text{“The Undirected Maximum Flow Problem”} \\ -\boldsymbol{c} \leq \boldsymbol{f} &\leq \boldsymbol{c} \end{aligned}$$

We use $\mathrm{val}(\boldsymbol{f})$ to denote F when $B\boldsymbol{f} = F\boldsymbol{b}_{st}$.

As we develop algorithms for this problem, we will assume that we know the maximum flow value F^*. Let $\boldsymbol{f}^*$ denote some maximum flow, i.e. a flow with $-\boldsymbol{c} \leq \boldsymbol{f} \leq \boldsymbol{c}$ can $\mathrm{val}(\boldsymbol{f}^*) = F^*$.

In general, an a lower bound $F \leq F^*$ will allow us to find a flow with value F, and because of this, we can use a binary search to approximate F^*.

1.1 A barrier function and an algorithmic framework

$$V(\boldsymbol{f}) = \sum_e -\log(\boldsymbol{c}(e) - \boldsymbol{f}(e)) - \log(\boldsymbol{c}(e) + \boldsymbol{f}(e))$$

We assume the optimal value of Program (1) is F^*. Then for a given $0 \leq \alpha < 1$ we define a program

$$\begin{aligned} \min_{\boldsymbol{f} \in \mathbb{R}^E} \quad & V(\boldsymbol{f}) \qquad\qquad (2) \\ \text{s.t. } \boldsymbol{B}\boldsymbol{f} &= \alpha F^* \boldsymbol{b}_{st} \qquad \text{“The Barrier Problem”} \end{aligned}$$

This problem makes sense for any $0 \leq \alpha < 1$. When $\alpha = 0$, we are not routing any flow yet. This will be our starting point. For any $0 \leq \alpha < 1$, the scaled-down maximum flow $\alpha\boldsymbol{f}^*$ strictly satisfies the capacities $-\boldsymbol{c} < \alpha \boldsymbol{f}^* < \boldsymbol{c}$, and $\boldsymbol{B}\alpha\boldsymbol{f}^* = \alpha F^* \boldsymbol{b}_{st}$. Hence $\alpha\boldsymbol{f}^*$ is a feasible flow for this value of α and hence $V(\alpha\boldsymbol{f}^*) < \infty$ and so the optimal flow for the Barrier Problem at this α must also have objective value strictly below ∞, and hence in turn strictly satisfy the capacity constraints. Thus, if we can find the optimal flow for Program (2) for $\alpha = 1 - \epsilon$, we will have a feasible flow with Program (1), the Undirected Maximum Flow Problem, routing $(1-\epsilon)F^*$. This is how we will develop an algorithm for computing the maximum flow.

It turns out that, if we have a solution $\boldsymbol{f}_1$ to this problem for some $\alpha_1 < 1$, then we can find a solution $\boldsymbol{f}_2$ for some $\alpha_2 > \alpha_1$. And, we can compute $\boldsymbol{f}_1$ using a small number of Newton steps, each of which will only require a Laplacian linear equation solve, and hence is computable in $\widetilde{O}(m)$ time. Concretely, for any $0 \leq \alpha_1 < 1$, given the optimal flow at this α_1, we will be able to compute the optimal flow at $\alpha_2 = \alpha_1 + (1-\alpha_1)\frac{1}{70\sqrt{m}}$. This means that after $T = 70\sqrt{m}\log(1/\epsilon)$ updates, we have a solution for $\alpha_T \geq 1 - \epsilon$.

Program (2) has the Lagrangian

$$\mathcal{L}(\boldsymbol{f}, \boldsymbol{x}) = V(\boldsymbol{f}) + \boldsymbol{x}^\top(\alpha F^* \boldsymbol{b}_{st} - \boldsymbol{B}\boldsymbol{f})$$

And we have optimality when

$$\boldsymbol{B}\boldsymbol{f} = \alpha F^* \boldsymbol{b}_{st} \text{ and } -\boldsymbol{c} \leq \boldsymbol{f} \leq \boldsymbol{c} \tag{3}$$

“Barrier feasibility”

and $\boldsymbol{\nabla}_{\boldsymbol{f}} \mathcal{L}(\boldsymbol{f}, \boldsymbol{x}) = \boldsymbol{0}$, i.e.

$$\boldsymbol{\nabla} V(\boldsymbol{f}) = \boldsymbol{B}^\top \boldsymbol{x} \tag{4}$$

”Lagrangian barrier gradient optimality”

1.2 Updates using Divergence

$$\begin{aligned} \max_{\boldsymbol{\delta} \in \mathbb{R}^E} \quad & V(\boldsymbol{\delta} + \boldsymbol{f}) - (V(\boldsymbol{f}) + \langle \boldsymbol{\nabla} V(\boldsymbol{f}), \boldsymbol{\delta} \rangle) \\ \text{s.t. } \boldsymbol{B}\boldsymbol{\delta} = \alpha' F^* \boldsymbol{b}_{st} \end{aligned} \tag{5}$$

“The Update Problem”

This problem has Lagrangian

$$\mathcal{M}(\boldsymbol{\delta}, \boldsymbol{z}) = V(\boldsymbol{f}) + \boldsymbol{z}^\top((\alpha + \alpha')F\boldsymbol{b}_{st} - \boldsymbol{B}(\boldsymbol{f} + \boldsymbol{\delta}))$$

And we have optimality when

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{f} + \boldsymbol{\delta}) = (\alpha + \alpha')F^* \boldsymbol{b}_{st} \\ \text{and } -\boldsymbol{c} \leq \boldsymbol{f} + \boldsymbol{\delta} \leq \boldsymbol{c} \end{aligned} \tag{6}$$

“Update feasibility”

and $\boldsymbol{\nabla}_{\boldsymbol{\delta}} \mathcal{M}(\boldsymbol{\delta}, \boldsymbol{z}) = \boldsymbol{0}$, i.e.

$$\boldsymbol{\nabla} V(\boldsymbol{f} + \boldsymbol{\delta}) - \boldsymbol{\nabla} V(\boldsymbol{f}) = \boldsymbol{B}^\top \boldsymbol{z} \tag{7}$$

Note that if we simultaneously have (3), the "barrier feasibility" condition, and (6), the"update feasibility" condition, satisfied along with Equations (4) and (7), then $\boldsymbol{f}+\boldsymbol{\delta}$ satisfies the "barrier feasibility" with α replaced by $\alpha+\alpha'$ and we have

$$\boldsymbol{\nabla} V(\boldsymbol{f}+\boldsymbol{\delta}) = \boldsymbol{B}^\top(\boldsymbol{x}+\boldsymbol{z}).$$

Thus, we can conclude, by the Karush-Kuhn-Tucker theorem from the previous lecture, that $\boldsymbol{f}+\boldsymbol{\delta}$ is optimal for the optimization problem

$$\begin{aligned} \min_{\boldsymbol{f}\in\mathbb{R}^E} \quad & V(\boldsymbol{f}) \\ \text{s.t. } \boldsymbol{B}\boldsymbol{f} &= (\alpha+\alpha')F^* \boldsymbol{b}_{st} \end{aligned} \tag{8}$$

Lemma 1.1. *Suppose $\boldsymbol{f}$ is the minimizer of Problem* (2) *(the Barrier Problem with parameter α) and $\boldsymbol{\delta}$ is the minimizer of Problem* (5) *(the Update Problem with parameters $\boldsymbol{f}$ and α'), then $\boldsymbol{f}+\boldsymbol{\delta}$ is optimal for Problem* (2) *with parameter $\alpha+\alpha'$ (i.e. a new instance of the Barrier problem).*

I.e. $\boldsymbol{f}+\boldsymbol{\delta}$ is the argmin for Problem (8)*.*

Algorithm 1: Interior Point Method

$\boldsymbol{f} \leftarrow \boldsymbol{0}$;
$\alpha \leftarrow 0$;
while $\alpha < 1-\epsilon$ **do**
  $a' \leftarrow \frac{1-\alpha}{20\sqrt{m}}$;
  Compute $\boldsymbol{\delta}$, the minimizer of Problem (5);
  Let $\boldsymbol{f} \leftarrow \boldsymbol{f}+\boldsymbol{\delta}$ and $\alpha \leftarrow \alpha+\alpha'$;
end
return $\boldsymbol{f}$

Pseudotheorem 1.1. *Let $\boldsymbol{f}$ be the minimizer of Problem* (2)*. Then, when $a' \leq \frac{1-\alpha}{20\sqrt{m}}$, the minimizer $\boldsymbol{\delta}$ of Problem* (8) *can be computed in $\widetilde{O}(m)$ time.*

The key insight in this type of interior point method is that when the update α' is small enough,

Theorem 1.2. *Algorithm 1 returns a flow $\boldsymbol{f}$ that is feasible for Problem* (1) *in time $\widetilde{O}(m^{1.5}\log(1/\epsilon))$.*

Proof Sketch. First note that for $\alpha = 0$, the minimizer of Problem (2) is $\boldsymbol{f} = \boldsymbol{0}$. The proof now essentially follows by Lemma (1.1), and Pseudotheorem 1.1. Note that $1-\alpha$ shrinks by a factor $(1-\frac{1}{20\sqrt{m}})$ in each iteration of the while-loop, and so after $20\sqrt{m}\log(1/\epsilon)$ iterations, we have $1-\alpha \leq \epsilon$, at which point the loop terminates. To turn this into a formal proof, we need to take care of the fact the proper theorem corresponding to Pseudotheorem 1.1 only gives a highly accurate but not exact solution δ to the "Update Problem". But it's possible to show that this is good enough (even though both $\boldsymbol{f}$ and $\boldsymbol{\delta}$ end up not being exactly optimal in each iteration). $\square$

Remark 1.3. For the maximum flow problem, when capacities are integral and polynomially bounded, if we choose $\epsilon = m^{-c}$ for some large enough constant c, given a feasible flow with $\mathrm{val}(\boldsymbol{f}) = 1-\epsilon$, is it possible to compute an exact maximum flow in nearly linear time. Thus

Theorem 1.2 can also be used to compute an exact maximum flow in $\widetilde{O}(m)$ time, but we omit the proof. The idea is to first round to an almost optimal, feasible integral flow (which requires a non-trivial combinatorial algorithm), and then to recover the exact flow using Ford-Fulkerson.

Remark 1.4. It is possible to reduce an instance of directed maximum flow to an instance of undirected maximum flow in nearly-linear time, in such a way that if we can *exactly* solve the undirected instance, then in nearly-linear time we can recover an exact solution to the directed maximum flow problem. Thus Theorem (1.2) can also be used to solve directed maximum flow.

Remark 1.5. For sparse graphs with $m = \widetilde{O}(n)$ and large capacities, this running time is the best known, and improving it is major open problem.

1.3 Understanding the Divergence Objective

$$D(x) = -\log(1-x) - x$$

We let

$$\boldsymbol{c}_+(e) = \boldsymbol{c}(e) - \boldsymbol{f}(e) \text{ and } \boldsymbol{c}_-(e) = \boldsymbol{c}(e) + \boldsymbol{f}(e)$$

So then

$$\begin{aligned}
D_V(\boldsymbol{\delta}) &= V(\boldsymbol{\delta}+\boldsymbol{f}) - (V(\boldsymbol{f}) + \langle \boldsymbol{\nabla} V(\boldsymbol{f}), \boldsymbol{\delta}\rangle) \\
&= \sum_e -\log\left(\frac{\boldsymbol{c}(e) - (\boldsymbol{\delta}(e)+\boldsymbol{f}(e))}{\boldsymbol{c}(e)-\boldsymbol{f}(e)}\right) - \frac{\boldsymbol{\delta}(e)}{\boldsymbol{c}(e)-\boldsymbol{f}(e)} \\
&\qquad -\log\left(\frac{\boldsymbol{c}(e) + (\boldsymbol{\delta}(e)+\boldsymbol{f}(e))}{\boldsymbol{c}(e)+\boldsymbol{f}(e)}\right) + \frac{\boldsymbol{\delta}(e)}{\boldsymbol{c}(e)+\boldsymbol{f}(e)} \\
&= \sum_e D\left(\frac{\boldsymbol{\delta}(e)}{\boldsymbol{c}(e)-\boldsymbol{f}(e)}\right) + D\left(-\frac{\boldsymbol{\delta}(e)}{\boldsymbol{c}(e)+\boldsymbol{f}(e)}\right) \\
&= \sum_e D\left(\frac{\boldsymbol{\delta}(e)}{\boldsymbol{c}_+(e)}\right) + D\left(-\frac{\boldsymbol{\delta}(e)}{\boldsymbol{c}_-(e)}\right)
\end{aligned}$$

Note that we can express Problem (5) as

$$\begin{aligned}
\min_{\boldsymbol{\delta}\in\mathbb{R}^E} &\quad D_V(\boldsymbol{\delta}) \qquad (9)\\
\text{s.t. } \boldsymbol{B}\boldsymbol{\delta} &= \alpha' F^* \boldsymbol{b}_{st} \qquad \text{The Update Problem, restated}
\end{aligned}$$

Note that $D_V(\boldsymbol{\delta})$ is strictly convex of over the feasible set, so the argmin is unique.

1.4 Quadratically Smoothing Divergence and Local Agreement

$$\tilde{D}_\epsilon(x) = \begin{cases} -\log(1-x) - x & \text{if } |x| \le \epsilon \\ D(\epsilon) + D'(\epsilon)(x-\epsilon) + \frac{D''(\epsilon)}{2}(x-\epsilon)^2 & \text{if } x \ge \epsilon \\ D(-\epsilon) + D'(-\epsilon)(x+\epsilon) + \frac{D''(-\epsilon)}{2}(x+\epsilon)^2 & \text{if } x \le -\epsilon \end{cases}$$

For brevity, we define

$$\tilde{D}(x) = \tilde{D}_{0.1}(x)$$

Lemma 1.6.

1. *$1/2 \leq \tilde{D}''(\boldsymbol{x}) \leq 2$.*

2. *For $x \geq 0$, we have $x/2 \leq \tilde{D}'(\boldsymbol{x}) \leq 2x$ and $-2x \leq \tilde{D}'(-\boldsymbol{x}) \leq -x/2$.*

3. *$x^2/4 \leq \tilde{D}(\boldsymbol{x}) \leq x^2$.*

We also define

$$\tilde{D}_V(\boldsymbol{\delta}) = \sum_e \tilde{D}\left(\frac{\boldsymbol{\delta}(e)}{\boldsymbol{c}_+(e)}\right) + \tilde{D}\left(-\frac{\boldsymbol{\delta}(e)}{\boldsymbol{c}_-(e)}\right)$$

We can now introduce the smoothed optimization problem

$$\begin{aligned} \min_{\boldsymbol{\delta} \in \mathbb{R}^E} \quad & \tilde{D}_V(\boldsymbol{\delta}) && (10) \\ \text{s.t. } \boldsymbol{B}\boldsymbol{\delta} = \alpha' F^* \boldsymbol{b}_{st} & && \text{“The Smoothed Update Problem”} \end{aligned}$$

Note that $\tilde{D}_V(\boldsymbol{\delta})$ is strictly convex of over the feasible set, so the argmin is unique.

Pseudoclaim 1.7. *We can compute the argmin $\boldsymbol{\delta}^*$ of Problem* (10)*, the Smoothed Update Problem, using the Newton-Steps for K-stable Hessian convex functions that we saw in last lecture, in $\widetilde{O}(m)$ time.*

Sketch of proof. Problem (10) fits the class of problems for which we showed in Lecture 13 that (appropriately scaled) Newton steps converge. This is true because the Hessian is always a 2-spectral approximation of the Hessian at $\tilde{D}_V(\boldsymbol{\delta}^*)$, as can be shown from Lemma 1.6. Because the Hessian of $\tilde{D}_V(\boldsymbol{\delta})$ is diagonal, and the constraints are flow constraints, each Newton step boils down to solving a Laplacian linear system, which can be done to high accuracy $\widetilde{O}(m)$ time. □

Remark 1.8. There are three things we need to modify to turn the pseudoclaim into a true claim, addressing the errors arising from both Laplacian solvers and Newton steps:

1. We need to rephrase the claim to so that we only claim $\boldsymbol{\delta}^*$ has been computed to high accuracy, rather than exactly.

2. We need to show that we can construct an initial guess to start off Newton's method $\boldsymbol{\delta}_0$ for which the value $\tilde{D}_V(\boldsymbol{\delta}_0)$ is not too large. (This is easy).

3. We need show that Newton steps converge despite using a Laplacian solver that doesn't give exact solutions, only high accuracy solutions. (Takes a bit of work, but is ultimately not too difficult).

Importantly, to ensure our overall interior point method still works, we also need to show that it converges, even if we're using approximate solutions everywhere. This also takes some work to show, again is not too difficult.

Local Agreement Implies Same Optimum.

Lemma 1.9. *Suppose $S \subseteq \mathbb{R}^n$ are a convex sets with $T \subseteq S$, and let $f, g : S \to R$ be convex functions. Let $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in T} f(\boldsymbol{x})$. Suppose f, g agree on a neighborhood of $\boldsymbol{x}^*$ in S (i.e. an open set containing $\boldsymbol{x}^*$). Then $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in T} f(\boldsymbol{x})$.*

We define

$$\widehat{\boldsymbol{c}}(e) = \min(\boldsymbol{c}_+(e), \boldsymbol{c}_-(e)) \tag{11}$$

Lemma 1.10. *Suppose $\boldsymbol{\delta}^*$ is the argmin of Problem* (10)*, the Smoothed Update Problem, and* $\left\| \overrightarrow{(\boldsymbol{\delta}^*(e)/\widehat{\boldsymbol{c}}(e))} \right\|_\infty < 0.1$. *Then $\boldsymbol{\delta}^*$ is the argmin of Problem* (9)*.*

Proof. We observe that if $\left\| \overrightarrow{(\boldsymbol{\delta}^*(e)/\widehat{\boldsymbol{c}}(e))} \right\|_\infty < 0.1$, then $\tilde{D}_V(\boldsymbol{\delta}^*) = D_V(\boldsymbol{\delta}^*)$, and, for all $\boldsymbol{\tau} \in \mathbb{R}^m$ with norm

$$\left\| \overrightarrow{(\boldsymbol{\tau}(e)/\widehat{\boldsymbol{c}}(e))} \right\|_\infty < 0.1 - \left\| \overrightarrow{(\boldsymbol{\delta}^*(e)/\widehat{\boldsymbol{c}}(e))} \right\|_\infty$$

we have that $\tilde{D}_V(\boldsymbol{\delta}^* + \boldsymbol{\tau}) = D_V(\boldsymbol{\delta}^* + \boldsymbol{\tau})$. Thus $\tilde{D}_V$ and D_V agree on a neighborhood around $\boldsymbol{\delta}^*$ and hence by Lemma 1.9, we have that $\boldsymbol{\delta}^*$ is the argmin of Problem (9). □

1.5 Step size for divergence update

Definition 1.11 (*s-t well-conditioned* graph)**.** An undirected, capacitated multi-graph $G = (V, E, \boldsymbol{c})$ with source s and sink t is *s-t well-conditioned* if, letting U denote the maximum edge capacity $U = \|\boldsymbol{c}\|_\infty$, we have at least $\frac{2}{3}m$ multi-edges of capacity U going directly from s to t.

Remark 1.12. It is straightforward to make a graph s-t well-conditioned. We just add $2m$ new edges of capacity U directly between s and t. Given an exact maximum flow in the new graph, it is trivial to get one in the original graph: Just remove the flow on the new edges.

Definition 1.13. Given a *directed* graph $G = (V, E, \boldsymbol{c})$, the *symmetrization* of G is the undirected $\widehat{G} = (V, \widehat{E}, \widehat{\boldsymbol{c}})$ is the undirected graph given by

$$\{a, b\} \in \widehat{E} \text{ if } (a, b) \in E \text{ AND } (b, a) \in E$$

and

$$\widehat{\boldsymbol{c}}(\{a, b\}) = \min(\boldsymbol{c}(a, b), \boldsymbol{c}(b, a)).$$

Note that when $\widehat{G}_{\boldsymbol{f}}$ is the symmetrization of the residual graph $G_{\boldsymbol{f}}$ (which we define in Lecture 10), then $\widehat{\boldsymbol{c}}$ matches exactly the definition of $\widehat{\boldsymbol{c}}$ in Equation (11).

Lemma 1.14. *Let G be an undirected, capacitated multi-graph $G = (V, E, \boldsymbol{c})$ which is s-t well-conditioned. Let $\boldsymbol{f}$ be the minimizer of Program* (2)*. Let $\widehat{G}_{\boldsymbol{f}}$ be the* symmetrization *of the residual graph $G_{\boldsymbol{f}}$ (in the sense of Lecture 10). Then there exists a flow $\hat{\boldsymbol{\delta}}$ which satisfies $\boldsymbol{B}\hat{\boldsymbol{\delta}} = \frac{1-\alpha}{5} F^* \boldsymbol{b}_{st}$ and is feasible in $\widehat{G}_{\boldsymbol{f}}$. Note that we can also state the feasibility in $\widehat{G}_{\boldsymbol{f}}$ as*

$$\left\| \overrightarrow{\left(\hat{\boldsymbol{\delta}}(e)/\widehat{\boldsymbol{c}}(e)\right)} \right\|_\infty \leq 1$$

Proof. We recall since $\boldsymbol{f}$ is the minimizer of Program (2), there exists dual-optimal voltages $\boldsymbol{x}$ such that

$$\boldsymbol{B}^{\top}\boldsymbol{x} = \boldsymbol{\nabla} V(\boldsymbol{f}) = \overrightarrow{\left(\frac{1}{\boldsymbol{c}(e) - \boldsymbol{f}(e)} - \frac{1}{\boldsymbol{c}(e) + \boldsymbol{f}(e)}\right)}$$

From Lecture 10, we know that there is flow $\bar{\boldsymbol{\delta}}$ that is feasible with respect to the residual graph capacities of the graph $G_{\boldsymbol{f}}$ such that $\boldsymbol{B}\bar{\boldsymbol{\delta}} = (1-\alpha)F^{*}\boldsymbol{b}_{st}$. Note when treating $\bar{\boldsymbol{\delta}}$ as an undirected flow, feasibility in the residual graph means that $\bar{\boldsymbol{\delta}}(e) < \boldsymbol{c}(e) - \boldsymbol{f}(e)$ and $-\bar{\boldsymbol{\delta}}(e) < \boldsymbol{c}(e) + \boldsymbol{f}(e)$. Thus,

$$(1-\alpha)F^{*}\boldsymbol{b}_{st}^{\top}\boldsymbol{x} = \bar{\boldsymbol{\delta}}\boldsymbol{B}^{\top}\boldsymbol{x} = \sum_{e} \frac{\bar{\boldsymbol{\delta}}}{\boldsymbol{c}(e) - \boldsymbol{f}(e)} - \frac{\bar{\boldsymbol{\delta}}}{\boldsymbol{c}(e) + \boldsymbol{f}(e)} \leq m$$

Now, because the graph is s-t well-conditioned, there are at $\frac{2}{3}m$ edges directly from s to t with capacity U and each of these e satisfy by the Lagrangian gradient optimality condition (4)

$$\boldsymbol{b}_{st}^{\top}\boldsymbol{x} = \frac{1}{U - \boldsymbol{f}(e)} - \frac{1}{U + \boldsymbol{f}(e)}$$

Note that $\frac{2}{3}mU \leq F^{*} \leq mU$ because the graph is s-t well-conditioned. To complete the analysis, we consider three cases.

Case 1: $|\boldsymbol{f}(e)| \leq \frac{2}{3}U$. Then the capacity on each of these edges in the symmetrized residual graph $\widehat{G}_{\boldsymbol{f}}$ is at least $U/3$. As there are $\frac{2}{3}m$ of them, we get that there is a feasible flow in $\widehat{G}_{\boldsymbol{f}}$ of value at least $\frac{2}{9}mU \geq \frac{1}{10}F^{*}$.

Case 2: $\boldsymbol{f}(e) < -\frac{2}{3}U$. By the gradient condition, we have the same flow on all of the $\frac{2}{3}m$ s-t edges, adding up to at least $\frac{2}{3}mU$ going from t to s. This means that we must have at least $\frac{2}{3}mU$ flow going from s to t via the remaining edges. But, their combined capacity is at most $\frac{1}{3}mU$, so that cannot happen. Thus we can rule out this case entirely.

Case 3: $\boldsymbol{f}(e) > \frac{2}{3}U$. Then

$$\frac{m}{(1-\alpha)F^{*}} \geq \boldsymbol{b}_{st}^{\top}\boldsymbol{x} \geq \frac{1}{U - \boldsymbol{f}(e)} - \frac{1}{U + \boldsymbol{f}(e)} \geq \frac{4/5}{U - \boldsymbol{f}(e)}$$

So

$$U - \boldsymbol{f}(e) \geq \frac{4}{5}\frac{(1-\alpha)F^{*}}{m} \geq \frac{1}{2}(1-\alpha)U$$

In this case, the capacity on each of the $\frac{2}{3}m$ s-t edges with capacity U in G will have capacity $(1-\alpha)U/2$ in $\widehat{G}_{\boldsymbol{f}}$. This guarantees that there is feasible flow in $\widehat{G}_{\boldsymbol{f}}$ of value at least $\frac{1}{3}(1-\alpha)mU \geq \frac{1}{3}(1-\alpha)F^{*}$. □

Lemma 1.15. *Let* $0 < \alpha' \leq \frac{1-\alpha}{150\sqrt{m}}$. *Then the minimizer* $\boldsymbol{\delta}^{*}$ *of Problem* (10) *satisfies* $\left\|\overrightarrow{(\boldsymbol{\delta}^{*}(e)/\widehat{\boldsymbol{c}}(e))}\right\|_{\infty} < 0.1$.

Proof. By Lemma 1.14, there exists a flow $\hat{\boldsymbol{\delta}}$ which satisfies $\boldsymbol{B}\hat{\boldsymbol{\delta}} = \frac{1-\alpha}{5}F^{*}\boldsymbol{b}_{st}$ and $\left\|\overrightarrow{\left(\hat{\boldsymbol{\delta}}(e)/\widehat{\boldsymbol{c}}(e)\right)}\right\|_{\infty} \leq 1$. Hence for any $0 < \alpha' \leq \frac{1-\alpha}{150\sqrt{m}}$, the flow $\tilde{\boldsymbol{\delta}} = \alpha'\frac{5}{1-\alpha}\hat{\boldsymbol{\delta}}$ satisfies $\boldsymbol{B}\tilde{\boldsymbol{\delta}} = \alpha' F^{*}\boldsymbol{b}_{st}$

and $\left\|\overrightarrow{\left(\tilde{\boldsymbol{\delta}}(e)/\widehat{\boldsymbol{c}}(e)\right)}\right\|_\infty \leq \frac{1}{30\sqrt{m}}$. This means that

$$\begin{aligned}
\tilde{D}_V(\tilde{\boldsymbol{\delta}}) &= \sum_e \tilde{D}\left(\frac{\tilde{\boldsymbol{\delta}}(e)}{\boldsymbol{c}_+(e)}\right) + \tilde{D}\left(-\frac{\tilde{\boldsymbol{\delta}}(e)}{\boldsymbol{c}_-(e)}\right) \\
&\leq \sum_e 4\left(\frac{\tilde{\boldsymbol{\delta}}(e)}{\boldsymbol{c}_+(e)}\right)^2 + 4\left(-\frac{\tilde{\boldsymbol{\delta}}(e)}{\boldsymbol{c}_-(e)}\right)^2 \\
&\leq \sum_e 8\left(\frac{\tilde{\boldsymbol{\delta}}(e)}{\widehat{\boldsymbol{c}}(e)}\right)^2 \\
&\leq 8/900 < 1/100.
\end{aligned}$$

This then means that the minimizer $\boldsymbol{\delta}^*$ of Problem (10) also satisfies $\tilde{D}_V(\tilde{\boldsymbol{\delta}}) < 1/100$.

$$\begin{aligned}
\left\|\overrightarrow{(\boldsymbol{\delta}^*/\widehat{\boldsymbol{c}}(e))}\right\|_\infty^2 &\leq \sum_e \left(\frac{\boldsymbol{\delta}^*(e)}{\boldsymbol{c}_+(e)}\right)^2 + \left(-\frac{\boldsymbol{\delta}^*(e)}{\boldsymbol{c}_-(e)}\right)^2 \\
&\leq \sum_e \tilde{D}\left(\frac{\boldsymbol{\delta}^*(e)}{\boldsymbol{c}_+(e)}\right) + \tilde{D}\left(-\frac{\boldsymbol{\delta}^*(e)}{\boldsymbol{c}_-(e)}\right) && \text{By Lemma 1.6.} \\
&= \tilde{D}_V(\tilde{\boldsymbol{\delta}}) < 1/100.
\end{aligned}$$

Hence $\left\|\overrightarrow{(\boldsymbol{\delta}^*/\widehat{\boldsymbol{c}}(e))}\right\|_\infty < 0.1$. $\square$