Lecture 12

Lecturer: Aleksander Mądry *Scribe: Amy Ousterhout*

1 Overview

In this lecture we continue our exploration of the multiplicative weights update method–based approach to solving the maximum flow problem. In particular, we will improve upon our previous simple electrical flow–based oracle by removing from the graph the edges that cause the electrical flow to deviate significantly from the maximum flow. We will then argue that this leads to a faster maximum flow algorithm that constitutes an improvement both over the $\tilde{O}(m^{3/2}\varepsilon^{-5/2})$-time algorithm we obtained before using the standard regularized electrical flow–based oracle as well as over the best bound achieved by the algorithmic graph theory community for this problem in the last 40 years.

2 Recap of the Last Lecture

In the previous lecture we developed an electrical flow–based oracle for our multiplicative weights update method–based framework for solving the maximum flow problem. More precisely, applying binary search for the value F^* of the maximum flow enabled us to reduce maximum flow computations to solving the following feasibility problem $\bar{\mathcal{F}}(F)$ that corresponds to determining if $F \leq F^*$:

$$\begin{aligned}\bar{\mathcal{F}}(F) : Fy &\leq \vec{1}\\ y &\in \bar{\mathcal{F}}_{st}.\end{aligned}$$

Here, $\bar{\mathcal{F}}_{st} := \{\mathrm{abs}(f) \mid Bf = \chi_{st}\}$ is the set of edge flow vectors of unit s-t flows, and $\mathrm{abs}(f)$ is an m-dimensional vector defined as $\mathrm{abs}(f)_e = |f_e|$.

Our goal then was to design a (θ, ρ)-oracle for this feasibility problem $\bar{\mathcal{F}}(F)$. That is, we wanted to design an algorithm that, given an input $p \in \Delta_m$,

(i) if $F \leq F^*$, i.e., if $\bar{\mathcal{F}}(F)$ is feasible, returns a flow $\tilde{f}$ with $B\tilde{f} = \chi_{st}$ such that

$$|P\tilde{f}|_1 = \sum_e p_e |\tilde{f}_e| \leq \frac{1+\varepsilon}{F} \tag{1}$$

and

$$F|\tilde{f}_e| - 1 \in [-\theta, \rho], \tag{2}$$

for all edges e, where P is an m-by-m diagonal matrix with $P_{e,e} := p_e$, for each edge e.

(ii) otherwise, i.e. if $F > F^*$, returns $\perp$ or a flow $\tilde{f}$ as above.

Intuitively, condition (1) indicates that the flow $\tilde{f}$ is feasible on average, while the condition (2) ensures that each individual constraint neither has a too large slack ($\geq -\theta$) nor is violated by too much ($\leq \rho$). We also observed that the fact that $|\tilde{f}_e|$ is always non-negative implies that θ can always be taken to be equal to 1.

As we have shown before, once a (θ, ρ)-oracle such as above is designed we can use it – in a black-box manner – to get an $(1-\varepsilon)$-approximate maximum flow algorithm.

Theorem 1 *For any $\varepsilon > 0$, given an (θ, ρ)-oracle $\mathcal{O}$ we can compute a $(1-\varepsilon)$-approximation to the maximum flow in time $O(\frac{\rho \ln m}{\varepsilon^2}) \cdot \tau(\mathcal{O})$, where $\tau(\mathcal{O})$ is the running time of the oracle $\mathcal{O}$.*

Looking at the above theorem, we see that the parameter ρ – often called the *width* of the oracle $\mathcal{O}$ – is the key parameter that determines the number of oracle calls in the resulting algorithm. Thus our aim is to get this value to be as small as possible, while having the oracle be still fast.

2.1 An Electrical Flow-Based Oracle

Next, we considered constructing a (θ, ρ)-oracle that was based on electrical flow computations. Basically, it output an electrical flow wrt resistances r given by the input $p \in \Delta_m$. To our disappointment, this oracle turned out to have a width of $\rho = F - 1$, which is the maximal possible and far from satisfactory.

We noticed though that this very bad width behavior happens only for very degenerate input convex combinations p. Consequently, we modified our oracle to make it regularize the input convex combination p before computing an electrical flow wrt it. This regularization corresponded to adding a mix-in of a uniform distribution. More precisely, our regularized electrical flow based oracle $\mathcal{O}_{ER}$ worked as described in Algorithm 1.

Algorithm 1 Regularized electrical flow–based oracle $\mathcal{O}_{ER}$

Input: A convex combination $p \in \Delta_m$
Set resistances $\hat{r}_e := p_e + \frac{\varepsilon}{2m}$, for each edge e
Compute a unit electrical s-t-flow $\hat{f}$ wrt r' ▷ Note that $B\hat{f} = \chi_{st}$ by definition.
if $|P\hat{f}|_1 \leq \frac{(1+\frac{\varepsilon}{2})}{F}$ **then**
  Return $\hat{f}$
else
  Return $\perp$
end if

We then proved that not only this algorithm is a $(1, \rho)$-oracle for the feasibility problem $\bar{\mathcal{F}}(F)$ but also that its width ρ is at most

$$\rho = O\left(\sqrt{\frac{m}{\varepsilon}}\right). \tag{3}$$

Roughly speaking, we established first a certain absolute upper bound on the total energy of the flow $\hat{f}$. Then, we argued that should a flow on some edge e violate the upper bound (3), the contribution of that edge e to the total energy of the flow $\hat{f}$ would be greater than our absolute bound on the total energy of $\hat{f}$, which is a contradiction.

In the light of this, as we can perform electrical flow computations in nearly-linear time, i.e., $\tau(\mathcal{O}_{ER}) = \widetilde{O}(m)$, Theorem 1 allows us to conclude that the resulting $(1-\varepsilon)$-approximate maximum flow algorithm runs in time

$$\widetilde{O}\left(m^{\frac{3}{2}}\varepsilon^{-\frac{5}{2}}\right).$$

For sparse graphs, i.e., when $m = O(n)$, this bound matches the best known and over forty years old running time bounds in algorithmic graph theory. (Although these bounds are for exact computations, no faster algorithms were known for $(1-\varepsilon)$-approximation regime either.)

3 Going Beyond the $\widetilde{O}\left(m^{\frac{3}{2}}\right)$ Running Time Bound

Our main goal for today is to improve over the $\widetilde{O}\left(m^{\frac{3}{2}}\varepsilon^{-\frac{5}{2}}\right)$ running time bound we obtained in the last lecture and thus to break the long-standing running time barrier of $O(n^{\frac{3}{2}})$ for sparse graphs.

Given the developments in the last lecture, a natural first step in this quest is to try to see if the bound (3) on the width of our regularized electrical flow–based oracle $\mathcal{O}_{ER}$ (see Algorithm 1) can be improved. Unfortunately, as we already discussed last time, this is not the case and, for example, in the graph presented in Figure 1 this bound is essentially tight.

Recall that in this example the problem was that the single "shortcutting" between s and t has its effective resistance be comparable to the effective resistance of the whole system of $\approx \sqrt{n}$ parallel $\approx \sqrt{n}$-length paths. (All of this is in the basic case of having all the resistances be uniform.) So, as $F^* \approx \sqrt{n}$ in this graph, this single edge is getting roughly the half of the total electrical flow between s and t making the width to be $\approx \frac{F^*}{2} \approx \Omega(\sqrt{n})$, which is $\Omega(\sqrt{m})$ if the graph is sparse. We therefore need to modify our implementation of the oracle $\mathcal{O}_{ER}$ from Algorithm 1 in some substantial manner.

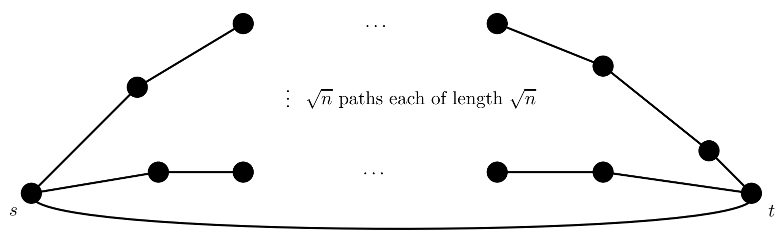

Figure 1: An example that shows our bound (3) on the width of the oracle $\mathcal{O}_{ER}$ (Algorithm 1) is essentially tight.

It turns out that, even though the example presented in Figure 1 was the worst-case example for our current oracle, it also provides us with an idea on how to remedy the corresponding shortcomings of that oracle. Namely, observe that if we removed the shortcutting s-t edge from that graph then the behavior of the electrical s-t-flow would improve dramatically. In fact, the electrical s-t-flow would exactly match the maximum s-t flow in that graph.

Also, this removal of an edge would not impact the value of the maximum flow by too much. After all, the total maximum flow value F^* is approximately $\sqrt{n}$ there, so removing one edge would amount to a reduction by a factor of the order of $(1 - \frac{1}{\sqrt{n}})$, which is completely acceptable for us.

So, we see that a simple removal of the offending edge led to a much better behavior of our oracle in the case of this particular graph. This gives rise to a very tempting idea: maybe this kind of strategy can work for general graphs too?

As we will show today, this is indeed the case! Specifically, let us consider the following improved oracle $\widehat{\mathcal{O}}$ that appears as Algorithm 2.

Algorithm 2 Regularized electrical flow–based oracle $\widehat{\mathcal{O}}$

Input: A convex combination $p \in \Delta_m$
Set resistances $\hat{r}_e := p_e + \frac{\varepsilon}{2m}$, for each edge e
flow (re-)computation: Compute a unit electrical s-t-flow $\hat{f}$ wrt r' in (the current version of) G
if $|P\hat{f}|_1 \leq \frac{(1+\varepsilon)}{F}$ **then**
  if Exists $\bar{e}$ such that $F \cdot |\hat{f}_{\bar{e}}| - 1 > \rho'$ **then** ▷ We intend $\rho' \ll O\left(\sqrt{\frac{m}{\varepsilon}}\right)$
    Remove the edge $\bar{e}$ from G (permanently) and go back to *flow (re-)computation* step
  else
    Return $\hat{f}$
  end if
else
  Return $\perp$
end if

Note that, in contrast to the Algorithm 1, we have a slightly less stringent check on the value of $|P\hat{f}|_1$ here. That is, we check if $|P\hat{f}|_1 \leq \frac{(1+\varepsilon)}{F}$ instead of checking if $|P\hat{f}|_1 \leq \frac{(1+\frac{\varepsilon}{2})}{F}$ as we did before. (This less stringent check is still valid from the point of view of the requirements we have in our definition of the (θ, ρ)-oracle above.)

Intuitively, this oracle tries to repeatedly remove edges from the graph G that cause the electrical s-t flow behavior to be very different to the maximum s-t flow behavior. In particular, as we will see later, the parameter ρ' corresponds to the width of the oracle that we intend to "force". (Note that, in a sense, width is a measure of discrepancy in electrical flow and maximum flow behavior.) Naturally, we want $\rho' \ll O\left(\sqrt{\frac{m}{\varepsilon}}\right)$, i.e., we want this width to be much smaller than the width bound 3 we established before.

Observe that if we knew that the above oracle always ends up returning a flow $\hat{f}$ then, by its construction, its width would indeed be ρ'. So, the key question is: how small ρ' can we set to ensure that not only it always returns $\hat{f}$ when it should, i.e., when $F \leq F^*$, but also that the number of flow recomputations is not too high either?

Clearly, there is a tension here. If we set ρ' to be very small then most likely the oracle will keep removing edges from G and eventually fail. On the other hand, setting ρ' to be too large might result in an unacceptably large width bound. In particular, we know that setting $\rho' = \rho$, as in bound (3), would make this oracle be correct and never need to recompute the flow – but by doing so we would give up on obtaining any improvement.

We thus proceed to analysis of this problem. This analysis divides into two parts: arguing the correctness of that oracle, and bounding its execution time. We will address these points in Sections 3.1 and 3.2, respectively.

3.1 Correctness of Our Oracle

Observe that to prove the correctness of our oracle we need to argue that whenever $F \leq F^*$ then

$$|P\hat{f}|_1 \leq \frac{(1+\varepsilon)}{F}. \tag{4}$$

To this end, recall that in our analysis of the previous electrical flow–based oracle $\mathcal{O}_{ER}$ (see Algorithm 1) we actually proved an analogous – in fact, even slightly stronger – statement. That is, we showed that whenever $F \leq F^*$ it is the case that

$$|P\hat{f}|_1 \leq \frac{(1+\frac{\varepsilon}{2})}{F}.$$

The proof of that latter statement relied on the following lemma that ties the ℓ_1-norm $|P\hat{f}|_1$ of the electrical flow $\hat{f}$ to the ℓ_2-norm $\|\widehat{R}^{\frac{1}{2}} f\|_2^2$ – or, equivalently, the energy – of any unit s-t flow f. (Here, $\widehat{R}$ is an m-by-m diagonal matrix with each diagonal entry $\widehat{R}_{ee}$ equal to $\hat{r}_e = p_e + \frac{\varepsilon}{2m}$.)

Lemma 2 *For any unit s-t flow f, i.e., any f such that $Bf = \chi_{st}$, we have that*

$$|P\hat{f}|_1 \leq \|\widehat{R}^{\frac{1}{2}} f\|_2.$$

Proof Note that by the Cauchy-Schwarz inequality we have that

$$|P\hat{f}|_1 \leq \sqrt{\|p^{\frac{1}{2}}\|_2^2 \cdot \|P^{\frac{1}{2}}\hat{f}\|_2^2} \leq \|P^{\frac{1}{2}}\hat{f}\|_2 \leq \|\widehat{R}^{\frac{1}{2}}\hat{f}\|_2,$$

where we used the fact that $\|p^{\frac{1}{2}}\|_2^2 = \sum_e p_e = 1$ and that $p_e \leq \hat{r}_e$, for all e.

Also, by definition, $\hat{f}$ is the minimizer of the energy $\|\widehat{R}^{\frac{1}{2}} f\|_2^2$ over all the unit s-t flows f. We thus have that

$$|P\hat{f}|_1^2 \leq \|\widehat{R}^{\frac{1}{2}}\hat{f}\|_2^2 \leq \|\widehat{R}^{\frac{1}{2}} f\|_2^2$$

for any such unit s-t flow f. Taking a square root of both sides gives us the statement of the lemma. ■

Armed with this lemma, we took f to be simply the maximum s-t flow f^* in G and concluded that, if $F \leq F^*$,

$$|P\hat{f}|_1 \leq \|\widehat{R}^{\frac{1}{2}} f^*\|_2 \leq \sqrt{\sum_e \hat{r}_e (f_e^*)^2} \leq \sqrt{\sum_e \frac{\hat{r}_e}{(F^*)^2}} \leq \frac{\sqrt{(1+\frac{\varepsilon}{2})}}{F^*} \leq \frac{\sqrt{(1+\frac{\varepsilon}{2})}}{F} \leq \frac{(1+\frac{\varepsilon}{2})}{F},$$

where we also used the fact that, by definition, $|f_e^*| \leq \|f^*\|_\infty \leq \frac{1}{F^*}$, for each edge e, and that $\sum_e \hat{r}_e = 1 + \frac{\varepsilon}{2}$. This established the correctness of the oracle $\mathcal{O}_{ER}$.

Unfortunately, the above proof does not directly apply to our new oracle. The reason is that now, whenever we compute the electrical flow $\hat{f}$, we do that in a version of G that might be missing some of the original edges. In particular, the max flow f^* might *not* be a valid unit s-t flow in that new version of G. In fact, it is not hard to see that, for example, if we removed from G all the edges of some s-t cut then the correctness bound (4) would not hold.

To address this problem we will impose in our analysis an invariant that ensures that we never remove too many edges.

Invariant 1 *The set H of permanently removed edges is never larger than $\frac{\varepsilon}{3}F$, i.e., $|H| \leq \frac{\varepsilon}{3}F$.*

Now, observe that even though the above invariant does not necessarily prevent f^* from becoming an invalid s-t flow, it ensures that, whenever $F \leq F^*$, the "damage" made to f^* by edge removal is not too large. Further, this damage can be "fixed" while not impacting the properties of f^* that we care about here.

Specifically, let us consider $\hat{f}^*$ to be the maximum flow f^* after we remove from it all the flow-paths that contain edges from H. Clearly, $\hat{f}^*$ is a valid s-t flow in our new version of the graph. Also, by Invariant 1, we know that the value $\hat{v}$ of $\hat{f}^*$ is at least

$$\hat{v} \geq 1 - \sum_{e \in H} |f_e^*| \geq 1 - \frac{|H|}{F^*} \geq 1 - \frac{|H|}{F} \geq 1 - \frac{\varepsilon}{3}.$$

Thus, if we apply Lemma 2 with f equal to $\frac{\hat{f}^*}{\hat{v}}$, i.e., the flow $\hat{f}^*$ rescaled to make it be a unit flow, we obtain that

$$|P\hat{f}|_1 \leq \frac{1}{\hat{v}}\|\widehat{R}^{\frac{1}{2}}\hat{f}^*\|_2 \leq \frac{1}{\hat{v}}\sqrt{\sum_e \hat{r}_e(\hat{f}_e^*)^2} \leq \frac{\sqrt{(1+\frac{\varepsilon}{2})}}{\hat{v}F^*} \leq \frac{\sqrt{(1+\frac{\varepsilon}{2})}}{(1-\frac{\varepsilon}{3})F} \leq \frac{(1+\varepsilon)}{F}, \tag{5}$$

which is exactly the desired condition (4). So, our new oracle is indeed correct, provided the Invariant 1 is never violated.

3.2 Bounding the Running Time of Our Oracle

Once we established that our oracle is correct (as long as the Invariant 1 holds), we can focus on finding the setting of our parameter ρ' so that this invariant is indeed never violated and the total oracle execution time, which is dominated by the time needed for all the electrical flow recomputations, is minimized. More precisely, we want to establish the following bound.

Lemma 3 *Provided Invariant 1 holds, during a single oracle call we have at most*

$$T' = O\left(\frac{m}{(\rho')^2\varepsilon}\log\frac{m}{\varepsilon}\right)$$

electrical flow recomputations.

Observe that in our oracle each removal of an edge from G triggers an electrical flow recomputation. So, once we prove the above lemma then it suffices to choose ρ' large enough so that

$$\frac{\varepsilon}{3}F \geq \frac{\varepsilon}{3}\rho' \geq T' = O\left(\frac{m}{(\rho')^2\varepsilon}\log\frac{m}{\varepsilon}\right) \tag{6}$$

to ensure that Invariant 1 is always preserved. (Here, we used the fact that $\rho' \leq F$, as otherwise we always have $F \cdot |\hat{f}_e| - 1 < \rho'$ and thus there is no need for electrical flow recomputations in the first place.)

Consequently, rearranging the terms in (6) tells us that we should set ρ' to be

$$\rho' = \Theta\left(\left(\frac{m\log\frac{m}{\varepsilon}}{\varepsilon^2}\right)^{\frac{1}{3}}\right) \tag{7}$$

and the resulting running time of the oracle becomes, by Lemma 3,

$$\widetilde{O}\left(mT'\right) = \widetilde{O}\left(m^{\frac{4}{3}}\varepsilon^{-\frac{2}{3}}\right). \tag{8}$$

3.3 Proof of Lemma 3

The proof of Lemma 3 is a potential-based argument. Specifically, we define $\mathcal{E}(H)$ to be the energy of a unit s-t flow wrt resistances $\hat{r}$ in the graph G with the edge set H removed. It is not hard to see that the following upper and lower bounds on $\mathcal{E}(\cdot)$ hold.

Fact 4 *Provided Invariant 1 holds, i.e., provided $|H| \leq \frac{\varepsilon}{3}F$, we have that*

$$\begin{aligned} \mathcal{E}(\emptyset) &\geq \frac{\varepsilon}{2m^2} \\ \mathcal{E}(H) &\leq \frac{(1+\varepsilon)^2}{F^2} \leq 2. \end{aligned}$$

Roughly speaking, the lower bound above follows from the fact that $\hat{r}_e \geq \frac{\varepsilon}{2m}$ and noticing that in the "worst" case all (at most m) edges of G are parallel edges joining s and t. The upper bound follows directly from the calculations in (5). (Note that, in principle, these calculations assumed that $F \leq F^*$. However, if the resulting upper bound on energy becomes violated at any point then we know that $F > F^*$ and can just stop and return $\perp$.)

Now, the key driver of our analysis will be a direct connection between removal of edges and the resulting change in the energy $\mathcal{E}(\cdot)$. It is not hard to see that such removal always results in the energy increase. However, one can make this statement quantitative and in the next problem set we will prove the following lemma.

Lemma 5 *Let $\hat{f}$ be a unit electrical s-t flow $\hat{f}$ wrt resistances $\hat{r}$ in the graph G with the edges in H removed. If $\bar{e}$ is an edge in G that contributes at least a δ-fraction of energy of $\hat{f}$, i.e., $\hat{r}_{\bar{e}}\hat{f}_{\bar{e}}^2 \geq \delta \sum_e \hat{r}_e \hat{f}_e^2 = \delta\mathcal{E}(H)$ then*

$$\mathcal{E}(H \cup \{\bar{e}\}) \geq (1+\delta)\cdot\mathcal{E}(H).$$

So, the larger contribution of an edge to the total energy the greater is the increase in the energy after this edge's removal. The rough intuition for this phenomena stems from the fact that electrical flow computations correspond to energy minimization and that this is essentially ℓ_2-minimization. As ℓ_2-norm tends to penalize any non-uniformity in contributions of the individual coordinates – and this penalty increases non-linearly with the extent of that non-uniformity – having a single edge with a significant contribution to the minimized ℓ_2-norm is pretty costly. This, in turn, means that if our routing still relies in large part on that single edge, this edge must be very crucial for making the whole flow be "energy efficient" and without it the only routings that remain are much more expensive energy-wise.

Finally, it turns out that the fact that our oracle removes only edges that have a large flow in $\hat{f}$ implies that each one of these edges has a sizable contribution to the energy. This is made precise in the following lemma.

Lemma 6 *As long as Invariant 1 holds, if an edge $\bar{e}$ is such that $F|\hat{f}_{\bar{e}}| - 1 > \rho'$ then it contributes at least Δ-fraction of the total energy of the flow $\hat{f}$, where*

$$\Delta \geq \Omega\left(\frac{\varepsilon\cdot(\rho')^2}{m}\right).$$

Proof Observe that the contribution of the edge $\bar{e}$ to the energy can be lower bounded as

$$\hat{r}_{\bar{e}}\hat{f}_{\bar{e}}^2 \geq \frac{\varepsilon}{2m}\hat{f}_{\bar{e}}^2 > \frac{\varepsilon}{2m}\left(\frac{1+\rho'}{F}\right)^2 \geq \frac{\varepsilon}{2m}\left(\frac{\rho'}{F}\right)^2. \tag{9}$$

On the other hand, from Fact 4 we know that the energy of the flow $\hat{f}$ is at most $\frac{(1+\varepsilon)^2}{F^2}$. Combining this with the above bound (9) on the contribution of $\bar{e}$, we get that

$$\Delta \geq \frac{\frac{\varepsilon}{2m}\left(\frac{\rho'}{F}\right)^2}{\frac{(1+\varepsilon)^2}{F^2}} = \frac{\varepsilon\,(\rho')^2}{2(1+\varepsilon)^2 m} = \Omega\left(\frac{\varepsilon(\rho')^2}{m}\right),$$

as desired. ■

We can now put Fact 4 and Lemmas 5 and 6 together and note that if H is the set of edges removed after T' electrical flow (re-)computations then

$$\mathcal{E}(H) \geq (1+\Delta)^{|H|}\,\mathcal{E}(\emptyset) \geq (1+\Delta)^{T'}\,\frac{\varepsilon}{2m^2} \geq \exp\left(\frac{\Delta T'}{2}\right)\frac{\varepsilon}{2m^2},$$

where we used the fact that $(1+x) \geq \exp(\frac{x}{2})$, for $x \leq \frac{1}{2}$.

On the other hand, by Fact 4, we know that $\mathcal{E}(H) \leq 2$, which allows us to conclude that

$$\exp\left(\frac{\Delta T'}{2}\right)\frac{\varepsilon}{2m^2} \leq \mathcal{E}(H) \leq 2$$

and thus, by taking the natural logarithms of both sides and rearranging the terms, we can conclude that the number of edge removals T' is at most

$$T' \leq \frac{2}{\Delta}\ln\left(\frac{4m^2}{\varepsilon}\right) = O\left(\frac{m}{\varepsilon(\rho')^2}\ln\frac{m}{\varepsilon}\right),$$

which is exactly the bound we wanted to obtained. Lemma 3 follows.

3.4 Running Time of the Resulting Maximum Flow Algorithm

The above analysis allows us to conclude that our new oracle is correct and has a width of at most ρ', where, by (7),

$$\rho' = \Theta\left(\left(\frac{m\log\frac{m}{\varepsilon}}{\varepsilon^2}\right)^{\frac{1}{3}}\right).$$

Also, the running time τ of a single call to that oracle – which is dominated by the time needed to perform all the electrical flow recomputations triggered – is, by (8), at most

$$\tau \leq \widetilde{O}\left(m^{\frac{4}{3}}\varepsilon^{-\frac{2}{3}}\right).$$

By Theorem 1, this immediately implies an $(1-\varepsilon)$-approximate maximum flow algorithm that runs in time

$$\widetilde{O}\left(\tau\rho'\varepsilon^{-2}\right) = \widetilde{O}\left(m^{\frac{5}{3}}\varepsilon^{-\frac{10}{3}}\right).$$

Observe that this running time bound is much *worse* than the $\widetilde{O}(m^{\frac{3}{2}}\varepsilon^{-5/2})$ bound we got previously, with a lot less effort, by simply using the basic electrical flow–based oracle $\mathcal{O}_{ER}$! What's going on?

Fortunately, the above startling conclusion is not really accurate. Namely, it turns out that our bound on the running time of a single oracle call, which essentially was a bound on T', i.e., the number of electrical flow recomputations, can be made much stronger. It actually can be used to bound the *total* number of electrical flow recomputations across *all* oracle calls.

We will not provide the formal reasoning here. However, the intuition is that the evolution of the convex combinations $p \in \Delta_m$ that are fed to the oracle in different rounds of our multiplicative weight update method–based framework is fairly steady. As a result, our potential function, i.e., the energy $\mathcal{E}(H)$, that we based our analysis of a single oracle call on does not change too much between oracle calls either. (Note that we carry on the "memory" of our "improvements" of the graph's electrical flow vs. maximum flow behavior, i.e., the set of already removed edges H, from one oracle call to another.) Consequently, our overall bound that relied on this potential function holds throughout the whole execution.

All of this implies that the total running time overhead caused by the need to sometime recompute electrical flow inside the oracle is at most

$$\widetilde{O}\left(m^{\frac{4}{3}}\varepsilon^{-\frac{2}{3}}\right),$$

and, once this overhead was accounted for, we can think of each oracle call being executed without any electrical flow recomputations and thus in time that is only nearly-linear in m. Therefore, by Theorem 1, the total running time of our algorithm will be at most

$$\widetilde{O}\left(m\rho'\varepsilon^{-2}\right) + \widetilde{O}\left(m^{\frac{4}{3}}\varepsilon^{-\frac{2}{3}}\right) \leq \widetilde{O}\left(m^{\frac{4}{3}}\varepsilon^{-\frac{8}{3}}\right).$$

Note that for sparse graph this running time bound becomes $\widetilde{O}(n^{\frac{4}{3}}\varepsilon^{-\frac{8}{3}})$ and thus constitutes the first in 40 years improvement over the notorious $O(n^{\frac{3}{2}})$ running time barrier for (approximate) maximum flow algorithms.