Advanced Graph Algorithms and Optimization Graded Homework 2

Jonas Hübotter

June 13th, 2022

1 Maintaining an Expander Decomposition

Setting In this section we consider (dynamic) graphs on the vertex set *V* of size *n*.¹ We are given an *m*-edge, unweighted, undirected graph *G* with maximum degree $\Delta_{\max}(G) = \mathcal{O}(1)$. Let *H* be a 1/2expander² on vertices *V* such that $\Delta_{\max}(H) = \mathcal{O}(\log^2 n)$ with an embedding $\Pi_{H \mapsto G}$ with congestion at most 1/2 ϕ . By lemma 14.2.1 of the lecture notes, the existence of *H* implies that *G* is a ϕ -expander. We define the graph,

$$\Pi_{H\mapsto G}^{-1}(D) \doteq \{ e \in H \mid \Pi_{H\mapsto G}(e) \cap D \neq \emptyset \},\tag{1}$$

of all edges in *H* whose embedding in *G* uses an edge of *D*.

Let $D \subseteq G$ be any subset of edges of G. In the following, we describe and analyze the algorithm CERTIFYORCUT($G, \phi, H, \Pi_{H \mapsto G}, D$) that in time $\tilde{O}(|D|/\phi^2)$ either outputs

- (Certify): a graph *H'* being a Ω(1)-expander and an embedding Π_{H'→(G\D)} with congestion at most 1/2φ (certifying that G \ D is still a Ω(φ)-expander); or
- (Cut): a set $S \subseteq V$ such that (S, \overline{S}) is a ϕ -sparse cut.

W.l.o.g. we assume $|D| < \frac{\phi n}{8}$. The algorithm CERTIFYORCUT is described in alg. 1.

Flow Problem \mathcal{I} We let \overrightarrow{G} be the directed graph obtained by replacing each edge $e = \{u, v\} \in G \setminus D$ by two antiparallel edges $\overrightarrow{e} = (u, v)$ and $\overrightarrow{e} = (v, u)$. To these edges we assign capacities $c \doteq C \cdot \mathbf{1}$ where $C \doteq {}^{8\Delta_{\max}(H)}/{\phi}$, a *sink capacity* $\nabla \doteq \mathbf{1}$ (the amount of flow that one can route to the vertices), and a *supply* $\Delta \doteq 4 \cdot \deg_{\Pi_{H \to G}^{-1}(D)}$. We want to either find a feasible flow $f \in \mathbb{R}^{|\overrightarrow{G}|}$ routing all flow to sinks or certify that no such flow exists. Given the incidence matrix B of J, the problem is characterized as finding $Bf \leq d \doteq \nabla - \Delta$ such that $\mathbf{0} \leq f \leq c$. We denote by \overrightarrow{G}_f the residual graph with respect to a flow f.

1.1 Part A: Implementing the Flow Algorithm

We implement line 2 as follows. Denote by \overrightarrow{G}' the graph \overrightarrow{G} with two additional vertices *s* and *t*. We then add an edge (s, v) and (v, t)

¹ All considered graphs will be on this vertex set, so we generally refer to the edge set of a graph *G* simply by *G* and vice versa.

² with respect to sparsity

Notation

Given $S, T \subseteq V$, we denote by $E_G(S, T)$ the set of edges in G with one endpoint in S and one in T. Similarly, we denote by $\vec{E}_{\vec{G}}(S,T)$ the set of edges with tail in S and head in T.

Algorithm 1: CertifyOrCut($G, \phi, H, \Pi_{H \mapsto G}, D$)

¹ Compute a flow *f* by running Dinic's Blocking Flow algorithm for $h \doteq 16\Delta_{\max}(G) \log(4m) / \phi$ iterations on \mathcal{I} ² if $Bf \leq d$ then $H' \leftarrow H \setminus \Pi^{-1}_{H \mapsto G}(D)$ 3 Initialize $\Pi_{H'\mapsto(G\setminus D)}$ to $\Pi_{H\mapsto G}$ restricted to the edges in H'4 Let \mathcal{P}_f be a flow path decomposition of f5 foreach $u \cdot v$ path $\overrightarrow{\pi} \in \mathcal{P}_f$ in \overrightarrow{G} do 6 Add edge $e = \{u, v\}$ to H'7 Let π be the "undirected" version of $\overrightarrow{\pi}$ in $G \setminus D$ 8 $\Pi_{H'\mapsto(G\setminus D)}(e) \leftarrow \pi$ 9 return $(H', \Pi_{H' \mapsto (G \setminus D)})$ 10 11 else $S \leftarrow \{v \in V \mid (Bf - d)(v) > 0\}$ 12 while $|E_{G\setminus D}(S,\overline{S})| \ge \phi|S|$ do $S \leftarrow S \cup \{v \in V \mid (u,v) \in \overrightarrow{E}_{\overrightarrow{G}_f}(S,\overline{S})\}$ 13 14 return S 15

for all $v \in V$ with capacity $\Delta(v)$ and $\nabla(v)$, respectively. We denote the resulting edge capacities by c'. We then run Dinic's algorithm for h iterations to compute an *s*-*t* flow f'. We let f be the flow obtained by restricting f' to the edges of \overline{G} .

Lemma 2. We have for the residual graph \overrightarrow{G}_f that there is no path from any vertex $x \in V$ where (Bf - d)(x) > 0 to a vertex $y \in V$ with (Bf - d)(y) < 0 consisting of less than h edges.

Proof. First, observe that by construction, we have for any $v \in V$,

$$(Bf')(v) = (Bf)(v) + f'(s,v) - f'(v,t).$$
(2)

Also, observe that all blocking flows used by Dinic's algorithm to augment the initial flow **0**, are *s*-*t* path flows, and hence, we maintain for all $v \in V$ that (Bf')(v) = 0 (i.e., every internal vertex has as much incoming flow as outgoing flow).

Take any vertex $x \in V$ with (Bf - d)(x) > 0, that is, $(Bf)(x) > \nabla(x) - \Delta(x)$. We obtain,

$$0 = (Bf')(x) = (Bf)(x) + f'(s, x) - f'(x, t)$$

> $\nabla(x) - \Delta(x) + f'(s, x) - f'(x, t)$
 $\geq -\Delta(x) + f'(s, x),$ using $f'(x, t) \leq \nabla(x)$

and hence, $f'(s, x) < \Delta(x)$, implying that $(s, x) \in \overrightarrow{E}_{\overrightarrow{G}'_{f'}}$.

Similarly, take any vertex $y \in V$ with (Bf - d)(y) < 0, that is, $(Bf)(y) < \nabla(y) - \Delta(y)$. We obtain,

$$0 = (Bf')(y) = (Bf)(y) + f'(s,y) - f'(y,t) < \nabla(y) - \Delta(y) + f'(s,y) - f'(y,t) \leq \nabla(y) - f'(y,t),$$

and hence, $f'(y,t) < \nabla(y)$, implying that $(y,t) \in \overrightarrow{E}_{\overrightarrow{G}'_{f'}}$. A schematic illustration of the residual graph used for Dinic's algorithm is shown in fig. 1.

Next, consider the levels of the sink *t* in \vec{G}' during each of the iterations $0 < i \le h$. We write f'_i for the flow obtained in the *i*-th iteration of Dinic's algorithm. By construction of \vec{G}' , we have that before any iteration of Dinic's algorithm, the shortest *s*-*t* path has length 2, so, $\ell_{\vec{G}'_{f'_0}}(t) \ge 2$.

Using that the level of the sink vertex strictly increases during each iteration of Dinic's algorithm³ and a simple induction on the number of iterations i,

$$\ell_{\overrightarrow{G}'_{f'_{i}}}(t) \ge \ell_{\overrightarrow{G}'_{f'_{i-1}}}(t) + 1 \ge \ell_{\overrightarrow{G}'_{f'_{0}}}(t) + i \ge i + 2$$

Finally, consider the final flow $f' = f'_h$ and suppose for a contradiction that there exists an *x*-*y* path *P* in \overrightarrow{G}_f consisting of fewer than *h* edges. Then, by our previous argument, (s, x) + P + (y, t) is an *s*-*t* path in $\overrightarrow{G}'_{f'}$ consisting of fewer than h + 2 edges. But this contradicts $\ell_{\overrightarrow{G}'_{f'}}(t) \ge h + 2$.

Theorem 3. The expected running time of the flow procedure is at most $\tilde{O}(|D|/\phi^2)$.

Proof. We will prove the following two claims.

Claim 4. FINDBLOCKINGFLOW (and therefore each iteration of Dinic's algorithm) takes expected time $\tilde{O}(\|\Delta\|_1)$.

Claim 5. $\|\mathbf{\Delta}\|_1 \leq \frac{4|D|}{\phi}$.

Remark 6. It is important to note that at no point we fully construct the flow instance \mathcal{I} . That is, we do never construct the entire residual graph nor its level graph. Instead, we construct the parts of these graphs that are needed as we need them.

Using the two claims, running Dinic's algorithm for h iterations takes expected time,

$$\tilde{\mathcal{O}}(h \| \mathbf{\Delta} \|_{1}) = \tilde{\mathcal{O}}\left(\frac{\Delta_{\max}(G)}{\phi} \cdot \frac{|D|}{\phi}\right) = \tilde{\mathcal{O}}\left(\frac{|D|}{\phi^{2}}\right). \qquad \Box \qquad \text{using } \Delta_{\max}(G) = \mathcal{O}(1)$$





Figure 1: Schematic illustration of the residual flow graph with vertex $x \in V$ such that (Bf - d)(x) > 0 and vertex $y \in V$ such that (Bf - d)(y) < 0. ³ Lemma 12.3.1 of the lecture notes.

Corollary 7. $\|\Delta\|_1 < \frac{n}{2} < \|\nabla\|_1 = n$

Proof. Using the assumption $|D| < \frac{\phi n}{8}$, this is an immediate corollary of claim 5.

Proof of claim 4. In the following, we denote the level graph with respect to which we want to find a blocking flow by *L*. We implement FINDBLOCKINGFLOW using link-cut trees almost identically to the lecture notes. The only small adjustment we make is that we do not initialize the link-cut tree on all edges of the level graph, but add edges only when they are "reached".

Recall that we used the procedure TRANSFORM(*L*) to transform the level graph into a graph where each edge e = (u, v) is replaced by a vertex *m* incident to its two endpoints; the endpoints *u* and *v* receive cost ∞ , and the vertex *m* receives cost c'(e). We write $L' \doteq$ TRANSFORM(*L*). As mentioned in the previous remark, we do not construct *L'* explicitly, but will refer to its parts (i.e., vertices and their neighborhood) as we need them. Clearly, each particular vertex and neighborhood can be constructed in time $1 + \Delta_{max}(G) = O(1)$.

We define the operation LC-TREE.ADDEDGE receiving an already existing vertex u and a vertex v to be added with cost cost(v) as follows.

Algorithm 8: LC-TREE.ADDEDGE(<i>u</i> , <i>v</i>)	
1 LC-TREE.ADDVERTEX (v)	
² LC-TREE.AddCost(v , cost(v))	

 $_{3}$ LC-TREE.LINK(u, v)

Note that as v is initially isolated, the cost-operation will not affect any other vertices. FINDBLOCKINGFLOW is described in alg. 9. There, H tracks all parts of the level graph that have been explored,⁴ thus if the neighborhood of s in H is identical to the neighborhood of s in L' the algorithm terminates. If this is not true, then we find an edge in the level graph that is not (and was not previously) in the link-cut tree and add it.

Observe that FINDBLOCKINGFLOW behaves analogously to the procedure defined in the lecture notes, except that vertices and edges are added lazily using the operation LC-TREE.ADDEDGE. It therefore follows immediately that the algorithm is correct and \hat{f}' is indeed a blocking flow of the level graph *L*.

It remains to show that the runtime of the procedure is at most $\tilde{O}(\|\Delta\|_1)$. We distinguish two cases. First, consider the case where the shortest *s*-*t* path in *L* has length two, that is, there exists some vertex $v \in V$ with edges (s, v) and (v, t) in *L*. Note that $\Delta \in \mathbb{N}_0^n$, implying that there are at most $\|\Delta\|_1$ vertices v with edge (s, v) in

⁴ In a way (though not quite), *H* corresponds to the complement of *H* as used in the definition of FINDBLOCK-INGFLOW in the lecture notes.

```
Algorithm 9: FINDBLOCKINGFLOW(s, t, L)
1 LC-Tree \leftarrow Initialize(\emptyset)
<sup>2</sup> LC-TREE.ADDVERTEX(s)
<sup>3</sup> Let H be an empty graph on vertices V \cup \{s, t\}
4 while \deg_H(s) < \deg_{L'}(s) do
      u \leftarrow \text{LC-TREE}.\text{FindRoot}(s)
 5
      if u = t then
 6
          (w,c) \leftarrow \text{LC-TREE}.\text{FINDMIN}(s)
 7
          LC-TREE.AddCost(s, -c)
 8
          Add to H and LC-TREE (via Cut(\cdot)) all edges incident
 9
            to w
      else if there is an edge (u, v) \in L' such that (u, v) is
10
        not in LC-Tree and (u, v) \notin H then
        LC-TREE.ADDEDGE(u, v)
11
       else
12
          Add to H and LC-TREE (via Cut(\cdot)) all edges incident
13
            to u
<sup>14</sup> Construct the blocking flow \hat{f}' by setting for each edge (u, v) of
```

¹⁴ Construct the blocking flow f' by setting for each edge (u, v) of L, with mid-point m in L', the flow to equal cost(m) minus the cost on m just before it was added to H

L. The only outgoing edge of such a vertex *v* in the level graph is the edge (v, t). Therefore, we add $\mathcal{O}(\|\Delta\|_1)$ many vertices to the link-cut tree.

Now, consider the case where the shortest *s*-*t* path in *L* has a length greater than two. We know that for any vertex $v \in V$ that does not have edge (v, t) in the level graph, it must send $\nabla(v) = 1$ units of flow to *t*. As the flow is upper bounded by $\|\Delta\|_1$, there can be at most $\|\Delta\|_1$ such "internal" vertices. Moreover, we know that the out-degree in *L* of any vertex $v \in V$ is at most $\Delta_{\max}(G) = \mathcal{O}(1)$. Therefore, we add $\mathcal{O}(\|\Delta\|_1)$ many vertices to the link-cut tree.

Using that each operation on the link-cut tree takes amortized time $\mathcal{O}(\log^2 n)$, the statement follows.

Proof of claim 5. We have,

$$\begin{split} \|\mathbf{\Delta}\|_{1} &= 4 \left\| \operatorname{deg}_{\Pi_{H \mapsto G}^{-1}(D)} \right\|_{1} \\ &= 8 \cdot |\Pi_{H \mapsto G}^{-1}(D)| \\ &\leq 8 \cdot \operatorname{cong}(\Pi_{H \mapsto G}) \cdot |D|, \end{split}$$

using the handshaking lemma, $\|\mathbf{deg}_E\|_1 = 2|E|$

as in the worst case, every edge in *D* has congestion $cong(\Pi_{H\mapsto G})$ and each is contributing path $\Pi_{H\mapsto G}(e)$ is the embedding of a different edge $e \in H$. Using $cong(\Pi_{H\mapsto G}) \leq 1/2\phi$,

$$\leq \frac{4|D|}{\phi}.$$

1.2 Part B: Implementing the Certify Outcome

We now consider the case where the flow f satisfies the if-condition, $Bf \leq d$.

Theorem 10. H' as returned by the algorithm is a 1/8-expander.

Proof. Consider any cut (S, \overline{S}) with $|S| \leq |\overline{S}|$. As H is a 1/2-expander, we have $|E_H(S, \overline{S})| \geq |S|/2$. We will consider two cases.

First, if $|E_{H'}(S,\overline{S})| \ge |E_H(S,\overline{S})|/4$ (that is, "few edges are removed from the cut"), we immediately obtain that H' is a 1/8-expander, as we have,

$$|E_{H'}(S,\overline{S})| \ge \frac{|E_H(S,\overline{S})|}{4} \ge \frac{|S|}{8},$$

using that *H* is a 1/2-expander.

Now, suppose that $|E_{H'}(S,\overline{S})| < |E_H(S,\overline{S})|/4$, that is, "many edges are removed from the cut". We will show that this implies that a sufficient amount of flow "escapes" the set *S*, yielding a sufficient number of edges to be added to *H'* based on the flow path decomposition \mathcal{P}_f . First, observe that if *F* units of flow are transported from *S*



Figure 2: Schematic illustration of the two cases. If not too many edges were removed, the graph is still an expander. If many edges were removed, the path decomposition leads to the addition of sufficiently many new edges.

to \overline{S} ("across the cut"), then $|E_{\mathcal{P}_f}(S,\overline{S})| \ge F$, as each vertex $v \in V$ can absorb at most $\nabla(v) = 1$ units of flow. For the same reason, we know that,

$$F \geq \mathbf{1}_{S}^{\top} \boldsymbol{\Delta} - \mathbf{1}_{S}^{\top} \boldsymbol{\nabla} = \mathbf{1}_{S}^{\top} \boldsymbol{\Delta} - |S|.$$

Hence, it suffices to show that $\mathbf{1}_{S}^{\top} \Delta \geq \frac{9}{8}|S|$, as then at least $\mathbf{1}_{S}^{\top} \Delta - |S| \geq |S|/8$ edges across the cut are added to H' due to the flow path decomposition.

Because for every removed cut-edge $\{u, v\}$ where $u \in S, v \in \overline{S}$, $\deg_{\prod_{u,v \in C}^{-1}(D)}(u)$ increases by one,

$$\begin{split} \mathbf{1}_{S}^{\top} \mathbf{\Delta} &= 4 \cdot \mathbf{1}_{S}^{\top} \mathbf{deg}_{\Pi_{H \mapsto G}^{-1}(D)} \geq 4 \Big(|E_{H}(S,\overline{S})| - |E_{H \setminus \Pi_{H \mapsto G}^{-1}(D)}(S,\overline{S})| \Big) \\ &\geq 4 \big(|E_{H}(S,\overline{S})| - |E_{H'}(S,\overline{S})| \big) \\ &> 3 |E_{H}(S,\overline{S})| \\ &\geq \frac{3}{2} |S| \end{split}$$

Theorem 11. $G \setminus D$ is a $\tilde{\Omega}(\phi)$ -expander.

Proof. We will prove the following claim.

Claim 12. $\Pi_{H'\mapsto(G\setminus D)}$ has congestion at most $\tilde{\mathcal{O}}(1/\phi)$.

Now, consider any cut (S,\overline{S}) with $|S| \leq |\overline{S}|$. Since H' is a $\Omega(1)$ expander, we have that $|E_{H'}(S,\overline{S})| = \Omega(|S|)$. From $\Pi_{H'\mapsto(G\setminus D)}$,
we know that for each $\{u,v\} \in E_{H'}(S,\overline{S})$, we can find a u-v path
in $G \setminus D$ that has to cross the cut (S,\overline{S}) at least once. By the given
claim, each edge in $G \setminus D$ is on at most $\tilde{O}(1/\phi)$ such paths, so at least $\tilde{\Omega}(\phi)|E_{H'}(S,\overline{S})| = \tilde{\Omega}(\phi|S|)$ edges in $G \setminus D$ cross the cut (S,\overline{S}) .

Proof of claim 12. We write $H'_1 \doteq H \setminus \Pi^{-1}_{H \mapsto G}(D)$ and H'_2 for the edges $\{u, v\}$ in H' resulting from a u-v path $\overrightarrow{\pi}$ from the flow path decomposition \mathcal{P}_f . We write $\Pi_{H'_1 \mapsto (G \setminus D)}$ and $\Pi_{H'_2 \mapsto (G \setminus D)}$ for the embedding $\Pi_{H' \mapsto (G \setminus D)}$ restricted to H'_1 and H'_2 , respectively. Note that $H' = H'_1 \cup H'_2$ and $\Pi_{H' \mapsto (G \setminus D)} = \Pi_{H'_1 \mapsto (G \setminus D)} \cup \Pi_{H'_2 \mapsto (G \setminus D)}$. Thus, it is sufficient to show that both $\Pi_{H'_1 \mapsto (G \setminus D)}$ and $\Pi_{H'_2 \mapsto (G \setminus D)}$ have congestion at most $\tilde{\mathcal{O}}(1/\phi)$.

Recall that, by assumption, the congestion of $\Pi_{H\mapsto G}$ (and therefore also $\Pi_{H'_{1}\mapsto (G\setminus D)}$) is at most $1/2\phi = \mathcal{O}(1/\phi)$.

Finally, due to our choice of the edge capacities of the flow problem \mathcal{I} , each edge can be used by at most *C* many paths of \mathcal{P}_f , as every *u*-*v* path transports at least,

$$\min\{\boldsymbol{\Delta}(u), C, \boldsymbol{\nabla}(v)\} \geq \min\{4, C, 1\},\$$

units of flow.⁶ Hence, the congestion of $\Pi_{H'_2 \mapsto (G \setminus D)}$ is at most,

$$\begin{split} & \text{using} \; |E_{H \setminus \Pi_{H \to G}^{-1}(D)}(S,\overline{S})| \leq |E_{H'}(S,\overline{S})| \\ & \text{using} \; |E_{H'}(S,\overline{S})| < \frac{|E_{H}(S,\overline{S})|}{4} \end{split}$$

using $|E_H(S,\overline{S})| \geq \frac{|S|}{2}$

⁵ This proof is analogous to the proof of lemma 14.2.1 of the lecture notes.

using that $\Delta(u) > 0$, as "some" flow is transported away from u

⁶ Since the edge capacity *C* is integral (otherwise we cannot apply Dinic's algorithm); and any path flow must saturate at least one of its constituting edges.

$$C = \frac{8\Delta_{\max}(H)}{\phi} = \mathcal{O}\left(\frac{\log^2 n}{\phi}\right) = \tilde{\mathcal{O}}\left(\frac{1}{\phi}\right)$$

and the statement follows.

1.3 Part C: Implementing the Cut Outcome

Finally, we consider the case where the flow f does not satisfy the if-condition, i.e., $Bf \leq d$. We define the residual graph \vec{G}_f such that an edge (u, v) is in the residual graph iff (u, v) is not saturated by f. This ensures that the residual graph is simple.

Let $S_0 \doteq \{v \in V \mid (Bf - d)(v) > 0\}$ and for i > 0,7

$$S_i \doteq \{ v \in V \mid \exists s \in S_0 \text{ with } \operatorname{dist}_{\overrightarrow{G}_{\epsilon}}(s, v) \le i \}.$$
(3)

Observe that because the if-condition is not satisfied, $S_0 \neq \emptyset$. Letting k be the number of iteration of the while-loop, observe that for $0 < i \le k$, the sets S_i coincide with the set S after the *i*-th iteration of the while-loop. In particular, the set S_k is returned by the algorithm.

Lemma 13. *For each* $0 \le i < k$ *,*

$$\operatorname{vol}_{\overrightarrow{G}_{f}}(S_{i+1}) \ge \left(1 + \frac{\phi}{8\Delta_{\max}(G)}\right) \operatorname{vol}_{\overrightarrow{G}_{f}}(S_{i}).$$
(4)

Proof. Due to the definition of the set S_{i+1} , we have,

$$\operatorname{vol}_{\overrightarrow{G}_{f}}(S_{i+1}) \ge \operatorname{vol}_{\overrightarrow{G}_{f}}(S_{i}) + |\overrightarrow{E}_{\overrightarrow{G}_{f}}(S_{i}, \overline{S_{i}})|.$$
(5)

This is because in each iteration of the while-loop we add all vertices to S_{i+1} that have distance one to the set S_i , and each of these new vertices will contribute at least one to the volume.

Recall that due to our definition of the residual graph, we have that $\overrightarrow{e} \in \overrightarrow{E}_{\overrightarrow{G}_{f}}(S_{i}, \overline{S_{i}})$ iff $f(\overrightarrow{e}) < C$. Thus,

$$\begin{aligned} |\overrightarrow{E}_{\overrightarrow{G}_{f}}(S_{i},\overline{S_{i}})| &= |\{\overrightarrow{e} \in \overrightarrow{E}_{\overrightarrow{G}}(S_{i},\overline{S_{i}}) | f(\overrightarrow{e}) < C\}| \\ &= |E_{\overrightarrow{G}}(S_{i},\overline{S_{i}})| - |\{\overrightarrow{e} \in \overrightarrow{E}_{\overrightarrow{G}}(S_{i},\overline{S_{i}}) | f(\overrightarrow{e}) = C\}| \\ &\geq |E_{\overrightarrow{G}}(S_{i},\overline{S_{i}})| - \frac{\mathbf{1}_{S_{i}}^{\top} \mathbf{\Delta}}{C} \\ &= |E_{G \setminus D}(S_{i},\overline{S_{i}})| - \frac{\mathbf{41}_{S_{i}}^{\top} \mathbf{deg}_{\Pi_{H \mapsto G}^{-1}(D)}}{C} \\ &\geq \phi|S_{i}| - \frac{4|S_{i}|\Delta_{\max}(H)}{C} \\ &= \phi|S_{i}| - \frac{1}{2}\phi|S_{i}| \\ &= \frac{1}{2}\phi|S_{i}|. \end{aligned}$$
(6)

using that the number of saturated cut-edges is at most $\mathbf{1}_{S_i}^{\mathsf{T}} \boldsymbol{\Delta} / c$ and the amount of flow leaving S_i is at most its supply, $\mathbf{1}_{S_i}^{\mathsf{T}} \boldsymbol{\Delta}$

using the while-condition, $|E_{G\setminus D}(S_i, \overline{S_i})| \ge \phi |S_i|$, and $\Pi_{H \mapsto G}^{-1}(D) \subseteq H$

⁷ Note that the graphs are unweighted, so dist(u, v) corresponds to the length of the shortest u-v path.

Finally, observe that,

$$\operatorname{vol}_{\overrightarrow{G}_{f}}(S_{i}) \leq 2\operatorname{vol}_{G \setminus D}(S_{i})$$
$$= 2 \sum_{v \in S_{i}} \operatorname{deg}_{G \setminus D}(v)$$
$$\leq 2|S_{i}|\Delta_{\max}(G). \tag{7}$$

Combining inequalities eq. (5), eq. (6), and eq. (7) yields the desired bound.8

Lemma 14. $k \leq h$.

Proof. We prove the statement by contradiction. Let $h' \ge h$. By the previous lemma and a simple induction,

$$\operatorname{vol}_{\overrightarrow{G}_{f}}(S_{h'}) \geq \left(1 + \frac{\phi}{8\Delta_{\max}(G)}\right)^{h'} \operatorname{vol}_{\overrightarrow{G}_{f}}(S_{0})$$
$$\geq \left(1 + \frac{\phi}{8\Delta_{\max}(G)}\right)^{h}$$
$$\geq \left(1 + \frac{\phi}{16\Delta_{\max}(G)} + \left(\frac{\phi}{16\Delta_{\max}(G)}\right)^{2}\right)^{h}$$
$$\geq e^{\frac{h\phi}{16\Delta_{\max}(G)}}$$
$$= e^{\log(4m)} = 4m_{t}$$

contradicting,

$$\operatorname{vol}_{\overrightarrow{G}_f}(S_{h'}) \leq 2\operatorname{vol}_G(S_{h'}) = 2\sum_{v \in S_{h'}} \deg_G(v) \leq 4m.$$

Theorem 15. $|S_k| \leq |\overline{S_k}|$ and $|E_{G \setminus D}(S_k, \overline{S_k})| < \phi |S_k|.9$

Proof. Observe that $|E_{G \setminus D}(S_k, \overline{S_k})| < \phi |S_k|$ follows immediately because the set S_k corresponds to the set after the final iteration of the while-loop, implying that the wile-condition is dissatisfied for S_k .

Claim 16. $|S_h| \leq \|\Delta\|_1$.

Using this claim, we obtain,

$$|S_k| \le |S_h| \le \|\mathbf{\Delta}\|_1 \le \frac{n}{2}.$$

Proof of claim 16. By lemma 2, there is no vertex *y* with (Bf - d)(y) < dy0 in S_h . Therefore, $|S_h|$ "consumes" at least $\mathbf{1}_{S_h}^{\top} \nabla = |S_h|$ units of flow. However, note that there is at most $\|\mathbf{\Delta}\|_1$ units of flow in total, and hence, we must have that $|S_h| = \mathbf{1}_{S_h}^\top \nabla \le \|\mathbf{\Delta}\|_1$. using $S_k \subseteq S_h$ as $k \le h$ and $\|\mathbf{\Delta}\|_1 \le \frac{n}{2}$

by corollary 7

using that \overrightarrow{G}_f is simple but may contain antiparallel edges

8 Even slightly better by a small constant factor.

using that $S_0 \neq \emptyset$, $\operatorname{vol}_{\overrightarrow{G}_f}(S_0) \ge 1$

using $\phi \leq \Delta_{\max}(G)$ from the definition of sparsity

using $e^x < 1 + x + x^2$ for $0 < x \le 1$

⁹ Hence, $(S_k, \overline{S_k})$ is a ϕ -sparse cut.

2 An ℓ_1 -Interior Point Method for Maximum Flow

Setting We consider an undirected graph G = (V, E) with capacities $c \in \mathbb{N}^{|E|}$. We write $n \doteq |V|, m \doteq |E|$, and assume m > 10 and $\|c\|_1 \le m^{10}$. Let $s, t \in V$ be some source and sink vertex. We assume that we are given the maximum *s*-*t* flow value $0 < F \le m^{10}$, which we assume to be integral.

In the following, we will consider the graph $\tilde{G} = (V, \tilde{E})$ where $\tilde{E} \doteq E \cup \{\tilde{e}\}$ and $\tilde{e} = \{s, t\}$ is an edge with unlimited capacity. We assign an arbitrary orientation to edges *E* and orient the edge \tilde{e} such that its tail is *s* and its head is *t*. Let **B** be the incidence matrix of \tilde{G} .

We are interested in finding an *s*-*t* flow $f \in \mathbb{R}^{|E|}$ with value *F*. We will do so by considering *s*-*t* flows $f \in \mathbb{R}^{|\tilde{E}|}$ that do not send any flow from *s* to *t* on edge \tilde{e} , i.e., $f(\tilde{e}) \leq 0$. We obtain the program,

$$\min_{\substack{f \in \mathbb{R}^{|\tilde{E}|} \\ Bf = F(\mathbf{1}_t - \mathbf{1}_s) \\ \forall e \in E: -c(e) \le f(e) \le c(e)}} f(\tilde{e}).$$
(8)

Barrier Program We now consider a variant of the above program using barrier functions. Let,

$$\mathcal{I} \doteq \{ f \in \mathbb{R}^{|\tilde{E}|} \mid f(\tilde{e}) > 0 \text{ and } \forall e \in E \colon -c(e) < f(e) < c(e) \},$$

be a capacity-constrained flow set. We define a barrier $B : \mathcal{I} \to \mathbb{R}$ by,

$$B(f) \doteq \sum_{e \in E} -\log\left(1 - \frac{f(e)}{c(e)}\right) - \log\left(1 + \frac{f(e)}{c(e)}\right),$$

and a potential function $\Phi : \mathcal{I} \to \mathbb{R}$, $\Phi(f) \doteq 10m \log(f(\tilde{e})) + B(f)$. The *barrier program* is described by,

$$\min_{\substack{f \in \mathcal{I} \\ Bf = F(\mathbf{1}_t - \mathbf{1}_s)}} \Phi(f).$$
(9)

2.1 Part A: The Potential Function: Initialization and End-Goal

Lemma 17. Program (8) is convex.

Proof. This immediately follows because the objective is linear, the equality constraint is linear, and the inequality constraints are linear (hence, convex). \Box

Lemma 18. The potential function Φ is non-convex.

Proof. Observe that Φ is twice continuously differentiable. We have for its gradient,

$$\nabla \Phi(f)(e) = \begin{cases} \frac{10m}{f(\tilde{e})} & e = \tilde{e} \\ \frac{1}{c(e) - f(e)} - \frac{1}{c(e) + f(e)} & e \in E, \end{cases}$$
(10)

and for its (diagonal!) Hessian,

$$H_{\Phi}(f)(e,e) = \begin{cases} -\frac{10m}{f(\tilde{e})^2} & e = \tilde{e} \\ \frac{1}{(c(e) - f(e))^2} + \frac{1}{(c(e) + f(e))^2} & e \in E. \end{cases}$$
(11)

Clearly, $H_{\Phi}(f)$ is not positive semi-definite, and hence, by the second-order characterization of convexity, Φ is non-convex.

Corollary 19. Program (9) is non-convex.

Lemma 20. Any optimal solution f^* for the program (8) routes F units of flow from s to t on the edges of E and has $f^*(\tilde{e}) = 0$.

Proof. Due to the definition of *F*, there exists a flow $f \in \mathbb{R}^{|E|}$ routing *F* units of flow from *s* to *t*. By definition, *f* only uses the edges of *E*. This implies that $f^*(\tilde{e}) \leq 0$.

Now suppose that $f^*(\tilde{e}) < 0$. But then, the flow f defined by $f(e) \doteq f^*(e)$ for $e \in E$ and $f(\tilde{e}) = 0$ routes $F - f^*(\tilde{e}) > F$ units of flow from s to t, contradicting the definition of F.

Lemma 21 (Termination). If $f \in \mathcal{I}$ and $\Phi(f) \leq -10m \log m$,¹⁰ then $f(\tilde{e}) \leq 1/m$.¹¹

Proof. As $f \in \mathcal{I}$, we know that $f^{(e)}/c^{(e)} \in (-1, 1)$. We write $\Delta(e) \doteq |f^{(e)}/c^{(e)}| \in [0, 1)$. Because the logarithm is concave, we have that $|\log(1 - \Delta(e))| > |\log(1 + \Delta(e))|$. We also have that $\log(1 - \Delta(e)) < 0$ and $\log(1 + \Delta(e)) > 0$. This shows that $B(f) \ge 0$.

As we assumed $\Phi(f) \leq -10m \log m$, this implies,

Lemma 22 (Initialization). For $f_0 \doteq F\mathbf{1}_{\tilde{e}}$, we have $Bf_0 = F(\mathbf{1}_t - \mathbf{1}_s)$, $f_0 \in \mathcal{I}$, and $\Phi(f_0) \leq 100m \log m$.

Proof. (1) We have $Bf_0 = FB\mathbf{1}_{\tilde{e}} = F(\mathbf{1}_t - \mathbf{1}_s)$ per the definition of the orientation of \tilde{e} .

(2) We have $f_0(\tilde{e}) = F > 0$ and we have that for any $e \in E$,

$$f_0(e) = 0 \in (-c(e), c(e))$$

so $f_0 \in \mathcal{I}$.

(3) Observe that $B(f_0) = \sum_{e \in E} -2\log(1) = 0$. We have,

$$\begin{split} \Phi(f_0) &= 10m \log(f_0(\tilde{e})) + B(f_0) \\ &= 10m \log(F) \\ &\leq 10m \log(m^{10}) \\ &= 100m \log m. \end{split}$$

using that the sum of capacities is at most m^{10}

¹⁰ The assumption $Bf = F(\mathbf{1}_t - \mathbf{1}_s)$ is not needed here. ¹¹ So *f* is an approximate solution to program (8).

2.2 Part B: IPM Progress using Updates

Given $f \in \mathcal{I}$, we define $l \in \mathbb{R}^{|\tilde{E}|}$ by $l(\tilde{e}) \doteq 1/f(\tilde{e})$ and

$$l(e) \doteq \frac{1}{\min\{c(e) - f(e), c(e) + f(e)\}}$$

for all $e \in E$. Observe that $l_f > 0$ (in each coordinate) as $f(\tilde{e}) > 0$ and $f(e) \in (-c(e), c(e))$ for all $e \in E$. We define a corresponding diagonal matrix $L_f \doteq \operatorname{diag}_{e \in \tilde{E}}(l_f(e))$.

In the following we will consider "updates" (that is, circulations¹²) $\delta \in \mathbb{R}^{|\tilde{E}|}$.

¹² A circulation is a flow δ satisfying $B\delta = 0$.

Lemma 23. If $\|L_f \delta\|_{\infty} \leq 1/2$, then $f + \delta \in \mathcal{I}$.

Proof. (1) By assumption, $|l_f(\tilde{e}) \cdot \delta(\tilde{e})| = l_f(\tilde{e}) \cdot |\delta(\tilde{e})| \le 1/2$. This implies $|\delta(\tilde{e})| \le 1/2l_f(\tilde{e}) = f(\tilde{e})/2$. From this, we get,

$$f(ilde{e})+\delta(ilde{e})\geq f(ilde{e})-rac{f(ilde{e})}{2}=rac{f(ilde{e})}{2}>0.$$

- (2) Fix any $e \in E$. By assumption, $|l_f(e) \cdot \delta(e)| = l_f(e) \cdot |\delta(e)| \le 1/2$. Consider two cases:
 - (i) If $f(e) \ge 0$, then $l_f(e) = \frac{1}{c(e) f(e)}$, implying, $|\delta(e)| \le \frac{c(e) f(e)}{2}$. Thus,

$$\begin{split} f(e) + \delta(e) &\leq f(e) + \frac{c(e) - f(e)}{2} = \frac{f(e) + c(e)}{2} < c(e) \quad \text{and} \qquad \text{using } f(e) < c(e) \\ f(e) + \delta(e) &\geq f(e) - \frac{c(e) - f(e)}{2} \\ &= \frac{3f(e) - c(e)}{2} \geq -\frac{c(e)}{2} > -c(e). \qquad \text{using } f(e) \geq 0 \end{split}$$

(ii) If f(e) < 0, then $l_f(e) = \frac{1}{c(e)+f(e)}$, implying, $|\delta(e)| \le \frac{c(e)+f(e)}{2}$. Thus,

$$\begin{split} f(e) + \delta(e) &\geq f(e) - \frac{c(e) + f(e)}{2} = \frac{f(e) - c(e)}{2} > -c(e) \quad \text{and} \quad \text{using } f(e) > -c(e) \\ f(e) + \delta(e) &\leq f(e) + \frac{c(e) + f(e)}{2} \\ &= \frac{3f(e) + c(e)}{2} < \frac{c(e)}{2} < c(e). \quad \text{using } f(e) < 0 \end{split}$$

Altogether, we have $(f + \delta)(\tilde{e}) > 0$ and $(f + \delta)(e) \in (-c(e), c(e))$ for all $e \in E$, so $f + \delta \in \mathcal{I}$.

Lemma 24. If $\|L_f \delta\|_{\infty} \leq 1/2$, then

$$\Phi(f+\delta) \le \Phi(f) + \nabla \Phi(f)^{\top} \delta + 4 \left\| L_f \delta \right\|_1^2.$$
(12)

Proof. We have seen that $f + \delta \in \mathcal{I}$, so $\Phi(f + \delta)$ is well-defined. By Taylor's theorem (second-order form),

$$\Phi(f+\delta) = \Phi(f) + \nabla \Phi(f)^{\top} \delta + \frac{1}{2} \delta^{\top} H_{\Phi}(\tilde{f}) \delta, \qquad (13)$$

where $\tilde{f} \doteq (1 - \theta)f + \theta(f + \delta) = f + \theta\delta$ for some $\theta \in [0, 1]$. We have,

$$egin{aligned} \delta^{ op} H_{\Phi}(f+ heta\delta)\delta \ &= -10mrac{\delta(ilde{e})^2}{(f(ilde{e})+ heta\delta(ilde{e}))^2} \ &+ \sum_{e\in E}igg(rac{1}{(c(e)-f(e)- heta\delta(e))^2}+rac{1}{(c(e)+f(e)+ heta\delta(e))^2}igg)\delta(e)^2 \ &\leq \sum_{e\in E}igg(rac{1}{(c(e)-f(e)+ heta|\delta(e)|)^2}+rac{1}{(c(e)+f(e)- heta|\delta(e)|)^2}igg)\delta(e)^2. \end{aligned}$$

using our characterization of H_{Φ} from eq. (11)

From $\|L_f \delta\|_{\infty} \leq 1/2$, we conclude that

$$|\delta(e)| \le rac{1}{2l_f(e)} = rac{\min\{m{c}(e) - f(e), m{c}(e) + f(e)\}}{2}$$

for all $e \in E$. This implies that $\frac{1}{(c(e)-f(e)+\theta|\delta(e)|)^2} \leq \frac{4}{(c(e)-f(e))^2}$ and similarly in the other case, $\frac{1}{(c(e)+f(e)-\theta|\delta(e)|)^2} \leq \frac{4}{(c(e)+f(e))^2}$, yielding,

$$\begin{split} &\leq 4\sum_{e\in E} \left(\frac{1}{(\boldsymbol{c}(e)-f(e))^2} + \frac{1}{(\boldsymbol{c}(e)+f(e))^2}\right) \delta(e)^2 \\ &\leq 8\sum_{e\in E} l_f(e)^2 \delta(e)^2 \qquad \qquad \text{using } \overline{l_f(e)} \\ &\leq 8 \left(\sum_{e\in E} l_f(e)|\delta(e)|\right)^2 \qquad \qquad \qquad \text{using } \Sigma_t^{\prime} \\ &= 8 \left\|\boldsymbol{L}_f \delta\right\|_1^2. \end{split}$$

Plugging this inequality into eq. (13), we obtain the desired bound.

Lemma 25 (Update). If for some $\kappa > 1$, we have $\|L_f \delta\|_1 \leq \kappa$ and $\nabla \Phi(f)^\top \delta = -1$, then

$$\Phi\left(f + \frac{1}{8\kappa^2}\delta\right) \le \Phi(f) - \frac{1}{16\kappa^2}.$$
(14)

Proof. Let $\delta' \doteq \frac{1}{8\kappa^2} \delta$. We have,

$$\left\|L_f\delta'
ight\|_{\infty} = rac{1}{8\kappa^2} \left\|L_f\delta
ight\|_{\infty} \leq rac{1}{8\kappa^2} \left\|L_f\delta
ight\|_1 \leq rac{1}{8\kappa} \leq rac{1}{2}.$$
 using $\left\|\cdot
ight\|_{\infty} \leq \left\|\cdot
ight\|_1$ and $\left\|L_f\delta
ight\|_1 \leq \kappa$

Therefore, by lemma 24,

$$\Phi(f+\delta') \leq \Phi(f) + \nabla \Phi(f)^{\top} \delta' + 4 \left\| L_f \delta' \right\|_1^2$$

using $\frac{1}{(c(e)-f(e))^2} + \frac{1}{(c(e)+f(e))^2} \le 2l_f(e)^2$

using
$$\sum_{i=1}^{n} a_i^2 \le (\sum_{i=1}^{n} a_i)^2$$
 if $a_i \ge 0$

$$\begin{split} &= \Phi(f) + \frac{1}{8\kappa^2} \underbrace{\nabla \Phi(f)^\top \delta}_{=-1} + \frac{1}{16\kappa^4} \underbrace{\left\| L_f \delta \right\|_1^2}_{\leq \kappa^2} \\ &\leq \Phi(f) - \frac{1}{8\kappa^2} + \frac{1}{16\kappa^2} \\ &= \Phi(f) - \frac{1}{16\kappa^2}. \end{split}$$

2.3 Part C: The Update

Let us consider the update program,

$$\min_{\substack{\delta \in \mathbb{R}^{|\tilde{E}|} \\ B\delta = \mathbf{0} \\ \nabla \Phi(f)^{\top} \delta = -1}} \left\| L_f \delta \right\|_1,$$
(15)

given the current flow $f \in \mathcal{I}$.

Lemma 26. The update program (15) is convex.

Proof. Observe that the two equality constraints are linear in δ . It is therefore sufficient to show that the objective is convex.

Fix any $\theta \in [0,1]$ and $x, y \in \mathbb{R}^{|\tilde{E}|}$. We have,

$$\begin{split} \left\| L_f(\theta x + (1-\theta)y) \right\|_1 &= \left\| \theta L_f x + (1-\theta) L_f y \right\|_1 \\ &\leq \left\| \theta L_f x \right\|_1 + \left\| (1-\theta) L_f y \right\|_1 \\ &= \theta \left\| L_f x \right\|_1 + (1-\theta) \left\| L_f y \right\|_1. \end{split}$$

Lemma 27. Let γ^* be the value of the update program (15). We have, $\gamma^* \leq 1.^{13}$

Proof. Let f^* be an optimal solution and f be a feasible solution (that is not optimal) to program (8) and let $\bar{\delta} \doteq \alpha(f^* - f)$ where

$$lpha \doteq -rac{1}{oldsymbol{
abla} \Phi(f)^{ op}(f^*-f)}$$

and *f* is chosen such that $\nabla \Phi(f)^{\top}(f^* - f) \neq 0.^{14}$ We will first show that $\bar{\delta}$ is a feasible solution for program (15) and then see that $\|L_f \bar{\delta}\|_1 \leq 1$, implying $\gamma^* \leq 1$.

We have,

$$\boldsymbol{B}\bar{\boldsymbol{\delta}} = \boldsymbol{\alpha}(\boldsymbol{B}\boldsymbol{f}^* - \boldsymbol{B}\boldsymbol{f}) = \boldsymbol{\alpha}(F(\boldsymbol{1}_t - \boldsymbol{1}_s) - F(\boldsymbol{1}_t - \boldsymbol{1}_s)) = \boldsymbol{0},$$

so $\overline{\delta}$ is indeed a cycle flow. Moreover,

$$\boldsymbol{\nabla}\Phi(f)^{\top}\bar{\boldsymbol{\delta}} = \boldsymbol{\alpha}\boldsymbol{\nabla}\Phi(f)^{\top}(f^*-f) = -\frac{\boldsymbol{\nabla}\Phi(f)^{\top}(f^*-f)}{\boldsymbol{\nabla}\Phi(f)^{\top}(f^*-f)} = -1.$$

This shows that $\overline{\delta}$ is feasible.

It remains to show that $\|L_f \bar{\delta}\|_1 \leq 1$. We will use the following two claims, which we will prove later.

using the triangle inequality

¹³ This is a slightly better constant than what we were asked for.

¹⁴ Such an f clearly exists, take f_0 for example.

using feasibility of f^* and f w.r.t. program (8)

Claim 28. For all $e \in E$, $\frac{f^*(e) - f(e)}{c(e) - f(e)} \le 1 - \frac{|f^*(e) - f(e)|}{c(e) - f(e)}$. **Claim 29.** For all $e \in E$, $-\frac{f^*(e)-f(e)}{c(e)+f(e)} \le 1 - \frac{|f^*(e)-f(e)|}{c(e)+f(e)}$.

Using our characterization of $\nabla\Phi$ from eq. (10), we obtain,

$$\begin{split} \nabla \Phi(f)^{\top}(f^* - f) \\ &= 10m \frac{f^*(\tilde{e}) - f(\tilde{e})}{f(\tilde{e})} + \sum_{e \in E} \left(\frac{1}{c(e) - f(e)} - \frac{1}{c(e) + f(e)} \right) (f^*(e) - f(e)) \\ &= -10m + \sum_{e \in E} \left(\frac{1}{c(e) - f(e)} - \frac{1}{c(e) + f(e)} \right) (f^*(e) - f(e)) \\ &\leq -8m - \sum_{e \in E} \left(\frac{1}{c(e) - f(e)} + \frac{1}{c(e) + f(e)} \right) |f^*(e) - f(e)|. \end{split}$$
 using claim 28 a

ng claim 28 and claim 29

Next, observe that,

$$\begin{split} &\sum_{e \in E} \left(\frac{1}{c(e) - f(e)} + \frac{1}{c(e) + f(e)} \right) |f^*(e) - f(e)| \\ &\geq \sum_{e \in E} l_f(e) \cdot |f^*(e) - f(e)| \\ &= \left\| L_f(f^* - f) \right\|_1 - l_f(\tilde{e}) \cdot |f^*(\tilde{e}) - f(\tilde{e})| \\ &= \left\| L_f(f^* - f) \right\|_1 - \frac{|f(\tilde{e})|}{f(\tilde{e})} \\ &\geq \left\| L_f(f^* - f) \right\|_1 - 1. \end{split}$$

Combining the two inequalities, we obtain,

$$\boldsymbol{\nabla} \Phi(f)^{\top}(f^* - f) \leq -8m - \left\| \boldsymbol{L}_f(f^* - f) \right\|_1 + 1 \leq - \left\| \boldsymbol{L}_f(f^* - f) \right\|_1.$$
 In particular, $\boldsymbol{\nabla} \Phi(f)^{\top}(f^* - f) < 0$, using that $\boldsymbol{\nabla} \Phi(f)^{\top}(f^* - f) \neq 0.$

Altogether, we have,

$$\begin{split} \left\| L_{f} \bar{\delta} \right\|_{1} &= |\alpha| \left\| L_{f}(f^{*} - f) \right\|_{1} \\ &= \frac{\left\| L_{f}(f^{*} - f) \right\|_{1}}{\left| \nabla \Phi(f)^{\top}(f^{*} - f) \right|_{1}} \\ &= -\frac{\left\| L_{f}(f^{*} - f) \right\|_{1}}{\nabla \Phi(f)^{\top}(f^{*} - f)} \\ &\leq \frac{\left\| L_{f}(f^{*} - f) \right\|_{1}}{\left\| L_{f}(f^{*} - f) \right\|_{1}} \\ &= 1. \end{split} \qquad using \nabla \Phi(f)^{\top}(f^{*} - f) < 0$$

Proof of claim 28. Fix any $e \in E$. We have,

$$\begin{split} \frac{f^*(e) - f(e)}{c(e) - f(e)} &= \frac{f^*(e) - c(e) + c(e) - f(e)}{c(e) - f(e)} \\ &= 1 - \frac{|f^*(e) - c(e)|}{c(e) - f(e)} \\ &\leq 1 - \frac{|f^*(e) - f(e)|}{c(e) - f(e)}. \end{split} \qquad \text{using } |f(e)| \leq c(e) \end{split}$$

Proof of claim 29. Fix any $e \in E$. We have,

$$\begin{aligned} -\frac{f^*(e) - f(e)}{c(e) + f(e)} &= -\frac{f^*(e) + c(e) - (c(e) + f(e))}{c(e) + f(e)} \\ &= 1 - \frac{|f^*(e) + c(e)|}{c(e) + f(e)} \\ &\leq 1 - \frac{|f^*(e) - f(e)|}{c(e) + f(e)}. \end{aligned} \qquad \text{using } |f(e)| \leq c(e) \end{aligned}$$

Lemma 30. There exists an optimal solution δ^* for the update program (15), which is supported on a single cycle.

Proof. Let δ be a feasible solution for the update program. We will later prove the following claim.

Claim 31. Any circulation flow δ (i.e., $\mathbf{B}\delta = \mathbf{0}$) can be decomposed into k cycle flows $\delta_1, \ldots, \delta_k$ (i.e., $\delta = \sum_{i=1}^k \delta_i$) such that the cycle flows¹⁵ send flow into the same direction on every edge. More formally, for all $e \in \tilde{E}$ and $i, j \in [k]$, we have that either $\delta_i(e), \delta_i(e) \leq 0$ or $\delta_i(e), \delta_i(e) \geq 0$.

Visually, this property corresponds to the fact that a drawing of $\delta_1, \ldots, \delta_k$ does not have antiparallel edges. In the following, we will therefore call cycle decompositions obeying by this property *parallel* and cycle flows mutually violating this property on some edge *antiparallel*.

Let $\delta_1, \ldots, \delta_k$ be an parallel cycle decomposition of δ . We know that $\nabla \Phi(f)^{\top} \delta = -1$. Because this is a linear function in δ , there must exist some cycle flow δ_j such that $\nabla \Phi(f)^{\top} \delta_j \leq -1$. We define the cycle flow,

$$ar{\delta} \doteq lpha \delta_j, \quad ext{where} \quad lpha = -rac{1}{oldsymbol{
abla} \Phi(f)^ op \delta_j} = rac{1}{|oldsymbol{
abla} \Phi(f)^ op \delta_j|} \leq 1.$$

In particular, $\|L_f \bar{\delta}\|_1 \leq \|L_f \delta_j\|_1$. We will show that $\bar{\delta}$ is at least as "good" as δ w.r.t. the update program.

First, observe that $\bar{\delta}$ is a feasible solution as $B\bar{\delta} = \alpha B\delta_i = 0$ and

$$\boldsymbol{\nabla} \Phi(f)^{\top} \bar{\boldsymbol{\delta}} = \boldsymbol{\alpha} \boldsymbol{\nabla} \Phi(f)^{\top} \boldsymbol{\delta}_{i} = -1.$$

We have,

$$\begin{aligned} \left\| \boldsymbol{L}_{f} \boldsymbol{\delta} \right\|_{1} &= \sum_{e \in \tilde{E}} \boldsymbol{l}_{f}(e) \cdot \left| \boldsymbol{\delta}(e) \right| \\ &= \sum_{e \in \tilde{E}} \boldsymbol{l}_{f}(e) \cdot \left| \sum_{i=1}^{k} \boldsymbol{\delta}_{i}(e) \right| \\ &= \sum_{e \in \tilde{E}} \boldsymbol{l}_{f}(e) \cdot \left(\sum_{i=1}^{k} \left| \boldsymbol{\delta}_{i}(e) \right| \right) \end{aligned}$$

¹⁵ A cycle flow is a circulation flow that sends the same amount of flow on every edge in its support, that is, it is supported on a single cycle. In particular, any cycle flow δ satisfies $B\delta = 0$

using that the cycle decomposition is parallel

$$= \sum_{i=1}^{k} \sum_{e \in \tilde{E}} l_f(e) \cdot |\delta_i(e)|$$
$$= \sum_{i=1}^{k} ||L_f \delta_i||_1$$
$$\geq ||L_f \delta_j||_1$$
$$\geq ||L_f \bar{\delta}||_1.$$

In particular, given the optimal solution δ^* , $\overline{\delta}^*$ is also optimal.

Proof of claim 31. Let $\delta_1, \ldots, \delta_k$ be any cycle decomposition of the circulation δ with l > 0 antiparallel pairs of mutually antiparallel cycle flows. We will give a procedure that yields a cycle decomposition $\delta'_1, \ldots, \delta'_{k'}$ with at most l - 1 mutually antiparallel cycle flows. Repeating this procedure at most l times yields the desired cycle decomposition. We denote by,

$$C_i \doteq \{e \in \tilde{E} \mid |\delta_i(e)| > 0\},\$$

the support of the *i*-th cycle flow of the cycle decomposition.

The procedure works as follows. Let δ_i , δ_j be any pair of mutually antiparallel cycle flows on edges $\overline{E} \subseteq \widetilde{E}$. W.l.o.g. we assume $\delta_i(\overline{e}) > 0$, $\delta_j(\overline{e}) < 0$ for $\overline{e} \in \overline{E}$ and that δ_i and δ_j route α and β units of flow, respectively. To simplify the presentation, we also assume $\alpha \ge \beta$, the other case is symmetric.

First, assume that δ_i and δ_j do not share any parallel edges. A schematic illustration is given in fig. 3. We define the cycle flows,

$$\delta_i'(e) \doteq egin{cases} eta & e \in C_i \cup C_j \setminus ar{E} \ 0 & ext{otherwise,} \ \end{array} \ \delta_j'(e) \doteq egin{cases} lpha - eta & e \in C_i \ 0 & ext{otherwise.} \ \end{cases}$$

Note that $\delta'_i = \mathbf{0}$ if $\alpha = \beta$.

Clearly, $\delta'_i + \delta'_j = \delta_i + \delta_j$. Moreover, δ'_i and δ'_j are not mutually antiparallel, as any edge $\bar{e} \in \bar{E}$ where flow was sent into opposite directions is only in the support of δ'_j and on all edges in the shared support $C_i \setminus \bar{E}$ flow is sent into the same direction. Hence, the cycle decomposition,

$$\delta_1,\ldots,\delta_{i-1},\delta'_i,\delta_{i+1},\ldots,\delta_{j-1},\delta'_j,\delta_{j+1},\ldots,\delta_k,$$

has the desired properties.

Finally, if δ_i and δ_j share some parallel edges $\hat{E} \subseteq \tilde{E}$, then we can find a new cycle decomposition consisting of more than two

using $\left\| L_f \bar{\delta} \right\|_1 \leq \left\| L_f \delta_j \right\|_1$



Figure 3: Schematic illustration of the cycle flow update. δ_i sends α units of flow and δ_j sends β units of flow. The net flow on the edge \bar{e} is $\alpha - \beta$.



Figure 4: Schematic illustration of the cycle flow update when parallel edges are present. δ_i sends α units of flow and δ_j sends β units of flow. The net flow on the edge \bar{e} is $\alpha - \beta$ and the net flow on the edge \hat{e} is $\alpha + \beta$.

cycles. A schematic illustration is given in fig. 4. δ'_i and δ'_j are defined similarly to the previous case but accordingly to the edges as shown in the schematic illustration. We define the new cycle flow $\delta'_{k'+1}$ analogously routing β units of flow. By the same arguments as in the previous case, $\delta'_i + \delta'_j + \delta'_{k'+1} = \delta_i + \delta_j$ and the new flows are not mutually antiparallel.

It is easy to see that the schematic illustration also covers the case where $|\hat{E}| > 1$ and $|\bar{E}| > 1$, though we might need to add more cycles.

We now describe an algorithm solving the undirected maximum flow problem. Suppose we are given the following subroutines:

- (1) COMPUTESTEP(\tilde{G}, f, τ) that given the graph, $f \in \mathcal{I}$, and a parameter $\tau > 1$ returns δ that is feasible for the update program (15) and has $\|L_f \delta\|_1 \leq 10 \cdot 2^{\tau}$. This algorithm takes time $T_{\text{COMPUTESTEP}} \doteq 2^{\log n/\tau} m$.
- (2) ROUNDFLOW(f) that given $f \in \mathcal{I}$ with $Bf = F(\mathbf{1}_t \mathbf{1}_s)$ returns an integral flow \hat{f} with $\hat{f}(\tilde{e}) = 0$, $-c(e) \leq \hat{f}(e) \leq c(e)$ for all $e \in E$, and $B\hat{f} = \hat{F}(\mathbf{1}_t - \mathbf{1}_s)$ where $\hat{F} \geq F - f(\tilde{e}) - 10$. This algorithm takes time $T_{\text{ROUNDFLOW}} \doteq \tilde{O}(m)$.

```
Algorithm 32: COMPUTEMAXFLOW(G, s, t)
```

 $f \leftarrow \mathbf{0}$ $f(\tilde{e}) \leftarrow F$ $f(\tilde{e}) \leftarrow F$ $f(\tilde{e}) \leftarrow F$ $f(\tilde{e}) \leftarrow COMPUTESTEP(\tilde{G}, f, \tau)$ $f \leftarrow f + \frac{1}{8\kappa^2}\delta$ $f \leftarrow ROUNDFLOW(f)$ $f \leftarrow ROUNDFLOW(f)$ $f \leftarrow FINDAUGMENTINGPATH(G_{\hat{f}}, s, t)$ $f \leftarrow \hat{f} + \tilde{f}$ $f \leftarrow return \hat{f}$

Theorem 33. Algorithm 32 returns the maximum s-t flow in the undirected graph G in time $m^{2+o(1)}$.

Proof. We denote by *k* the number of iterations of the first while-loop and by f_i and δ_i the flow and update after/during the *i*-th iteration of said while-loop, respectively. We fix $\tau \doteq \log \log m$.¹⁶ We have, $\log \log m > \log \log 8 = \log 3 > 1$, where we used our assumption m > 10.

¹⁶ Throughout our analysis, we assume that the logarithm is with respect to base 2.

Observe that f_0 (the flow before the first iteration of the whileloop) coincides with our characterization of f_0 in lemma 22, and hence, $Bf_0 = F(\mathbf{1}_t - \mathbf{1}_s), f_0 \in \mathcal{I}$, and $\Phi(f_0) \leq 100m \log m$.

Let us assume $f_i \in \mathcal{I}$. Per the definition of COMPUTESTEP, $B\delta_{i+1} = 0, \nabla \Phi_{f_i}^{\top}\delta_{i+1} = -1$, and $\|L_{f_i}\delta_{i+1}\|_1 \leq 10 \cdot 2^{\tau} \doteq \kappa$.¹⁷ By lemma 25, it follows that $Bf_{i+1} = F(\mathbf{1}_t - \mathbf{1}_s), f_{i+1} \in \mathcal{I}$, and

$$\Phi(f_{i+1}) = \Phi\left(f_i + rac{1}{8\kappa^2}\delta_{i+1}
ight) \le \Phi(f_i) - rac{1}{16\kappa^2}.$$

We have,

$$\Phi(f_i) = \Phi(f_0) + \sum_{j=1}^i \Phi(f_j) - \Phi(f_{j-1})$$
$$\leq 100m \log m - \frac{i}{16\kappa^2}$$

For any $0 \le i < k$, the while-condition is satisfied, implying,

$$\begin{array}{l} -10m\log m < \Phi(f_i) \leq 100m\log m - \frac{i}{16\kappa^2} \\ \Rightarrow & \frac{i}{16\kappa^2} < 110m\log m \\ \Rightarrow & i = \mathcal{O}\Big(\kappa^2 m\log m\Big). \end{array}$$

In particular,

_

=

$$k = \mathcal{O}\left(\kappa^2 m \log m\right) = \mathcal{O}\left(2^{2\tau} m \log m\right) = \mathcal{O}\left(m \log^3 m\right) = m^{1+o(1)}.$$

Evaluating ROUNDFLOW(f_k) yields a feasible integral flow \hat{f} supported on E routing \hat{F} units of flow from s to t, where

$$\hat{F} \ge F - f_k(\tilde{e}) - 10$$

 $\ge F - \frac{1}{m} - 10$ using lemma 21
 $\ge F - 11.$

Finally, we find an augmenting path for \hat{f} in the residual graph $G_{\hat{f}}$ until the flow is optimal (and no such augmenting path exists). As \hat{f} and *c* are integral, each augmenting path increases the value of the flow by at least one. As the initial \hat{f} is almost optimal, we only have to find a constant number of augmenting paths. Each augmenting path can be found in $\mathcal{O}(m)$ time using breadth-first search.

The total runtime of COMPUTEMAXFLOW is,

$$\begin{aligned} k \cdot T_{\text{COMPUTESTEP}} + T_{\text{ROUNDFLOW}} + \mathcal{O}(m) \\ &= m^{2+o(1)} \cdot 2^{\frac{\log n}{\tau}} + \tilde{\mathcal{O}}(m) + \mathcal{O}(m) \\ &= m^{2+o(1)} \cdot 2^{\frac{\log m}{\log \log m}} \\ &= m^{2+o(1)+\frac{1}{\log \log m}} \\ &= m^{2+o(1)}. \end{aligned}$$
 using $n = \mathcal{O}(m)$

¹⁷ Note that $\kappa > 1$.

using $\Phi(f_{i}) - \Phi(f_{i-1}) \le -1/16\kappa^{2}$

2.4 Part D: Stability

Finally, we will see that an update δ with small norm $||L_f \delta||_1$ can only cause $l_{f+\delta}$ to change significantly in very few entries. This can be used to compute each individual update in time $m^{o(1)}$.

We define $s \doteq L_f^{-1} l_{f+\delta}$ and $\mathcal{U} \doteq \{e \in \tilde{E} \mid |s(e) - 1| > 1/2\}$.

Lemma 34. If $\|L_f \delta\|_1 \leq 1/2$, then $|\mathcal{U}| = \mathcal{O}(1)$.

Proof. In the following, we will disregard whether $\tilde{e} \in U$, as this only changes the size of U by a constant. By assumption,

$$\sum_{e \in E} l_f(e) \cdot |\delta(e)| \le \left\| L_f \delta \right\|_1 \le \frac{1}{2}.$$

We have for all $e \in E$,

$$s(e) = rac{l_{f+\delta}(e)}{l_f(e)}.$$

We will prove the following two claims.

Claim 35. For any $e \in E$, if $f(e) \ge 0$ and |s(e) - 1| > 1/2, then $|\delta(e)| > \frac{c(e) - f(e)}{3}$.

Claim 36. For any $e \in E$, if f(e) < 0 and |s(e) - 1| > 1/2, then $|\delta(e)| > \frac{c(e) + f(e)}{3}$.

Using the two claims, we obtain,

$$\begin{split} \sum_{e \in E} l_f(e) \cdot |\delta(e)| &= \sum_{\substack{e \in E \\ f(e) \ge 0}} \frac{|\delta(e)|}{c(e) - f(e)} + \sum_{\substack{e \in E \\ f(e) < 0}} \frac{|\delta(e)|}{c(e) + f(e)} \\ &> \frac{|\mathcal{U}|}{3}, \end{split}$$

leading to a contradiction if $|\mathcal{U}| = \omega(1)$.

First, observe that |s(e) - 1| > 1/2 iff either s(e) < 1/2 or s(e) > 3/2.

Proof of claim 35. Fix any $e \in E$ with $f(e) \ge 0$. We have, $l_f(e) = c(e) - f(e)$. We consider two cases: (1) If $s(e) > \frac{3}{2}$,

$$\frac{3}{2} < s(e) = \frac{c(e) - f(e)}{\min\{c(e) - f(e) - \delta(e), c(e) + f(e) + \delta(e)\}} \\ \leq \frac{c(e) - f(e)}{c(e) - f(e) - |\delta(e)|}$$

So, $|\delta(e)| > \frac{c(e) - f(e)}{3}$.

(2) If $s(e) < \frac{1}{2}$, $\frac{1}{2} > s(e) = \frac{c(e) - f(e)}{\min\{c(e) - f(e) - \delta(e), c(e) + f(e) + \delta(e)\}}$ $\ge \frac{c(e) - f(e)}{c(e) - f(e) - \delta(e)}$

So, $\delta(e) < -c(e) + f(e) < 0$, and hence, $|\delta(e)| > c(e) - f(e)$. \Box

Proof of claim 36. Fix any $e \in E$ with f(e) < 0. We have, $l_f(e) = c(e) + f(e)$. We consider two cases: (1) If $s(e) > \frac{3}{2}$,

$$\begin{aligned} \frac{3}{2} < s(e) &= \frac{c(e) + f(e)}{\min\{c(e) - f(e) - \delta(e), c(e) + f(e) + \delta(e)\}} \\ &\leq \frac{c(e) + f(e)}{c(e) + f(e) - |\delta(e)|} \end{aligned}$$

So, $|\delta(e)| > \frac{c(e) + f(e)}{3}$. (2) If $s(e) < \frac{1}{2}$,

$$\frac{1}{2} > s(e) = \frac{c(e) + f(e)}{\min\{c(e) - f(e) - \delta(e), c(e) + f(e) + \delta(e)\}} \\ \ge \frac{c(e) + f(e)}{c(e) + f(e) + \delta(e)}$$

So, $\delta(e) > c(e) + f(e)$, and hence, $|\delta(e)| > c(e) + f(e)$.