SARNET Update: Flow Filtering and Edge Deletion

Gleb Polevoy Stojan Trajanovski Paola Grosso Cees de Laat

SNE, The University of Amsterdam, The Netherlands

Problems

Consider problems like

- DoS
- Unimportant flows
- Malicious communication

Needs

While solving, we need to

- Minimize the effort
- Reasonable time

Filtering Undesired Flows

Any problem of filtering some "bad" flows to increase the "good" ones.

No theoretical approximations of such filtering.

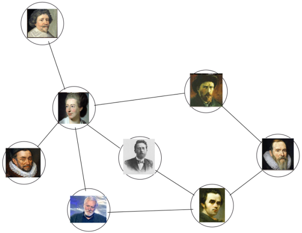
We

1. formally model
2. hardness
3. approximation

Model

1. The network is a directed capacitated graph $G = (N, E), c \colon E \to \mathbb{R}_+$.
2. A flow f from node o to d along a path, $f = (\underbrace{v(f)}_{\text{value}}, \underbrace{P(f)}_{\text{path}})$, such that

   for every edge e:
   $$\sum_{f : e \in P(f)} v(f) \leq c(e).$$



Model – BFF

Definition (Bad Flow Filtering (BFF))

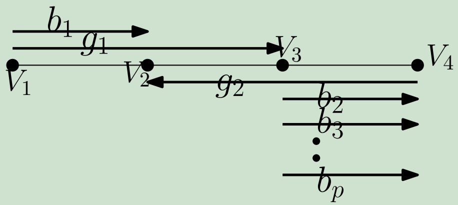
1. *Input:* $(G = (N, E), c \colon E \to \mathbb{R}_+, F, GF, BF, w \colon BF \to \mathbb{R})$.
2. *A solution S is a subset of bad flows to filter.*
3. *A feasible solution is a solution such that the good flows can be allocated values such that the total value of the good flows is the maximum possible.*
4. *Find a feasible solution with the minimum total weight* $w(S) \triangleq \sum_{b \in S} w(b)$.

Model – BFF – Example

The trivial feasible solution *BF* can be very far from the optimum.

Example

- Equal edge capacities c and equal weights.
- $v(b_1) = v(g_1) = v(g_2) = c/2$, $v(b_i) = \frac{c}{2(p-1)}$ $\forall i = 2, 3, \dots, p$.
- The optimal solution is $\emptyset$, ∞ times better than everything.



Model – BFF and UIBFF

Definition (Bad Flow Filtering (BFF))

Given $(G = (N, E), c\colon E \to \mathbb{R}_+, F, GF, BF, w\colon BF \to \mathbb{R})$, *minimize* $w(S)$ *such that the total good flow is maximum.*

Definition (Uniform Intersection Bad Flow Filtering (UIBFF))

BFF where every $g \in GF$ *has a set of edges on its path,* $E(g) \subseteq P(g)$, *such that every other good flow* g' *that intersects* g *fulfills:*
i.e. $P(g) \cap P(g') = E(g)$.

UIBFF is Hard

Hardness of approximation

UIBFF is not approximable within $c\left(\log\left(|\text{desirable flows}|\right)\right)$.

Proof.

Reduction from Set Cover. □

Our Algorithm – Measures

Definition

Given an instance of BFF, let k be the largest possible number of good flows that a given good flow intersects. Formally,

$$k \triangleq \max \left\{ \left| \left\{ g' \in GF \setminus \{g\} : P(g') \cap P(g) \neq \emptyset \right\} \right| : g \in G \right\}.$$

Definition

For a BFF instance, let b be the largest number of bad flows that intersect a good flow at any given edge. Formally,

$$b \triangleq \max \{ |\{ b \in BF : e \in P(b) \}| : g \in G, e \in P(g) \}.$$

Our Algorithm – Approximation Ratio

We provide a polynomial $b(k+1)$-approximation using local ratio.

Deleting Edges

Problems of removing "bad" flows by deleting edges on their paths, while trying not to remove too many "good" flows.

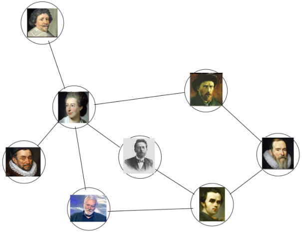
Similar network design problems exist, but not these ones.

We

1. formally model
2. hardness and approximation
3. for trees:
    1. Still hard
    2. Two better approximations
    3. If the flows are on a path from a root to the leaves, exact solution

Model

1. The network is a directed graph $G = (N, E)$.
2. A flow f from node o to d along a path, $f = (\underbrace{P(f)}_{\text{path}})$.
3. Capacities and flow values are irrelevant here.



Model – BFR and BBFR

Definition (Bad Flow Removing (BFR))

1. *Input:* $(G = (N, E), F, GF, BF, w \colon GF \to \mathbb{R}_+)$.
2. *A solution S is a subset of edges to delete.*
3. *A feasible solution is a solution removing all the bad flows.*
4. *Find a feasible solution with the minimum total weight of the removed good flows.*

Definition (Balanced Bad Flow Removing (BBFR))

1. *Input:* $(G = (N, E), F, GF, BF, w \colon F \to \mathbb{R}_+)$.
2. *A solution S is a subset of edges to delete.*
3. *Any solution is feasible.*
4. *Find a feasible solution with the minimum total weight of the remaining bad flows plus the weight of the removed good flows.*

BFR and BBFR are Hard

Hardness of approximation

It is impossible to approximate BFR or BBFR within $O(2^{\log^{1-\delta}|E|})$, for any $\delta > 0$.
For BBFR, it is also impossible within $O(2^{\log^{1-\delta}|BF|})$, for any $\delta > 0$.

BFR can be approximated withing $2\sqrt{|E| \log |BF|}$.
BBFR is approximable within $2\sqrt{(|E| + |BF|)\log(|BF|)}$.

Our Algorithms on Trees – Hardness and Approximation

Both problems are still MAX SNP-hard.

Definition

Let l be the maximum length of a good flow, i.e. $\max_{g \in GF} |P(g)|$.

We provide a polynomial l-approximation using primal-dual and a polynomial $2\sqrt{|E|}$-approximation for BFR.
This implies an l- and a $2\sqrt{|E| + |BF|}$-approximation for BBFR.

Moreover, if there exists a root $r \in N$ such that every flow flows on a path from r to a leaf, then a DP solves it exactly.

These gradually improving results suggest a gradual approach.

Conclusions

1. Modeling undesired flow problems (e.g., DoS, dispensable flows, malicious communication)
2. Important, but extremely hard to approximate
3. Good approximations for important subproblems

Future Work

- Arbitrary intersections (BFF)
- A given allocation algorithm, like max-min fairness (BFF)
- Other rankings
- Rerouting (BFR and BBFR)
- Strategic analysis (price of voting, improving NE)

Thank You!