Notes on connected cuts

Tomáš Turek

October 23, 2024

Contents

1 Basic cuts

In these notes we will define few problems which may be or not know to others. All of these problems have some properties in common. Mostly that we are going to talk about cuts which are supposed to be connected. To be precise one or all of the parts induces connected subgraphs.

Let us now define few of these problems.

Connected $s-t$ cut.

For a connected graph $G = (V, E)$ and vertices $s, t \in V$ such that $s \neq t$ we define *connected $s-t$ cut* as $S \subseteq V$ for which all following properties hold:

1. $s \in S$,
2. $t \notin S$ and
3. both $G[S]$ and $G[V \setminus S]$ are connected.

Note that the size of this cut is defined as $E(S, V \setminus S)$. (Where $E(X, Y)$ is for the number of edges between sets X and Y.)

Multi-way connected cut.

For a connected graph $G = (V, E)$ and vertices $s_1, s_2, \ldots, s_k \in V$ for $k \in \mathbb{N}$ ("sources") we define connected cut as partition $\mathcal{V} = \{V_1, V_2, \ldots, V_k\}$ of vertices (that is $\bigcup_{i=1,\ldots,k} V_i = V$ and for $i \neq j$ $V_i \cap V_j = \emptyset$) such that the following holds:

1. $\forall i \in [k] : s_i \in V_i$ and
2. $\forall i \in [k] : G[V_i]$ is connected.

Now for the sizes we define two versions.

- *Sum* size as $\sum_{i<j} E(V_i, V_j)$ or
- *Max* size as $\max_{i \in [k]} E(V_i, V \setminus V_i)$.

Observe that the sum size is already computed with multi-commodity cut. Also we may define **Flexible multi-way connected cut** as relaxing the previous problem. That is the partition will have l partitions where $0 < l \leq k$ and only l sources are representing their partition. So $\forall i \in [l], \exists k : s_k \in V_l$.

2 Connection to STC

Now for the connection to the STC problem. We will consider a graph where we have one vertex with high degree and all of its k neighbors will be the sources to our problem. Then we will obtain lower bound on STC and also present some algorithm to STC. If we would have flexible multi-way connected cut we could use this every time.

3 Other cuts

k-connected cut.

For a connected graph $G = (V, E)$ we say $S \subseteq V$ is k *connected cut* such that all properties hold:

1. $|S| = k$.
2. $G[S]$ is connected.
3. And we want to minimize $E(S, V \setminus S)$. Which is the size of such cut.

Note that choosing only two properties from all three can be computed. If we skip the very first one, we may use the result from Garg, which states a linear program having all vertices as such result. Excluding the second one can be also computed via some approximation algorithm for bisection. And Overlooking the last one we just use some search, because we don't care about the size of the result.

4 Absorptive flow

We will be now talking about an absorptive flow. Firstly we will state the problem in a common sense. For a graph and a source we get a flow which flows through the graph and every time it goes through a vertex some of the flow gets absorbed into it. After that we will define a cut which is induced by such flow and later on state an integer program and its linear approximation. Now we properly state the instance.

Definition (Absorptive flow)

For a graph $G = (V, E)$ and a vertex $s \in V$, also called as the *source*, and for $k \in \mathbb{N}$, such that $|V| \geq k$, we define **absorptive flow** as a tuple of functions denoted as $f = (f_V, f_E)$, where $f_V : V \to \mathbb{R}$ and $f_E : E \to \mathbb{R}$. Now these two function must have these properties.

1. $\sum_{v \in V} f_V(v) = k$, that is every part of the flow gets absorbed,
2. $f_V(s) = 1$,
3. $\forall v \in V : 0 \leq f_V(v) \leq 1$, so all vertices have some limits,
4. $\sum_{v \in V, (s,v) \in E} f_E((s, v)) = k - 1$, the flow starts from the source,
5. $\forall e \in E : 0 \leq f_E(e)$, the flow has to be non-negative, but can be unlimited,
6. $\forall v \in V, s \neq v : \sum_{u \in V, (u,v) \in E} f_E((u, v)) = \sum_{u \in V, (v,u) \in E} f_E((v, u)) + f_V(v)$, thus the whole flow continues unless part of it is absorbed.

One can already see that it resembles a linear program. One can expect we would define a size of the flow, but in this special instance we won't be defining it, since the main purpose is to look at the cut, which is defined by the flow. So now we will define the cut.

Firstly we will define $S \subseteq V$ as the vertices which have nonzero function, that is $\forall v \in S : f_V(s) > 0$. Then the **induced cut** defined by absorptive flow is defined as $E(S, V \setminus S)$ and its size as $e(S, V \setminus S) := |E(S, V \setminus S)|$. We will furthermore want to minimize the size of such cut.

So far the only property is that $s \in S$, which can be seen only from the definition. Next observations come from the linear program and its properties.

5 Integer program

In this section we will establish the linear program which works with this flow and its cut.

5.1 Variables

Firstly we declare the variables for edges and for vertices.

$$x_v = \begin{cases} 1 & \text{if it absorbs the flow} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{uv} \in [0, k] \text{ is for the amount of flow on the edge } uv.$$

$$z_{uv} = \begin{cases} 1 & \text{if } uv \in E(S, V \setminus S) \\ 0 & \text{otherwise} \end{cases}$$

See that these variables arise only from the definition of the problem.

5.2 Constraints

Now we need to state the constraints. Firstly set the connection between the absorbed vertices and the cut.

$$\begin{aligned} z_{uv} &\geq x_u - x_v \\ z_{uv} &\geq x_v - x_u \end{aligned}$$

Next we have to set the flow, so its properties hold.

$$\begin{aligned} \textstyle\sum_{v \in V, sv \in E} x_{sv} &= k - 1 \\ x_s &= 1 \\ \textstyle\sum_{u \in V, uv \in E} x_{uv} = \sum_{u \in V, vu \in E} x_{vu} + x_v & \quad \forall v \in V, s \neq v \\ \textstyle\sum_{u \in V} x_u &= k \end{aligned}$$

5.3 Optimization function

Lastly the optimization function will be to minimize the flow throughput and number of cut edges.

$$\min \sum_{e \in E} x_e + z_e$$

5.4 Properties

Lets talk about some crucial properties of this integer program.

Vertices always absorb.

Imagine that there is a vertex over which the flow is going and yet the vertex is not absorbing nothing from the flow. By seting the absorbption value of the vertex to one we will decrease the function value by (at least) one. But since it is the optimal value, then it must be increased by at least 2. And also we know that the x_e values did not increased in any way it must happend that z_e values got increased. Therefore there are two edges which became cut edges. Due to the fact, that we decreased the length of the flow and not increased we must say that both of them were occupied by a flow, but that is not possible, because the increse of values are in this state: $x_e < z_e$. Therefore every vertex always absorb the flow if that is possible.

Minimality of a cut.

Now what if there are multiple optimal solutions where some of x_e and z_e are exchanged?
Because we generaly want to minimize the cut.

TODO

Firstly observe that every time the flow goes through vertex some of its flow gets absorbed. This is always true if the function itself would be just $\min x_e$. In integer program we have only three kinds of edges. Either its inside the flow, on the cut or outside. Now see that if we would set $x_v = 0$ for some $V \in V$ even though there is a flow over it. Then we have to add 1 to some x_e.
S defined by the solution of this ILP has induced connected subgraph.