EECS 127/227AT Optimization Models in Engineering
Spring 2020 Discussion 10

1. The max-flow min-cut theorem

In this question, we explore how strong duality can be used to solve a canonical problem in network theory. Specifically, we will prove the *max-flow min-cut theorem*, which can be used to calculate the maximum flow through a network of interest. We will first introduce this theorem by means of an example, then prove it for a fairly general case.

Problem definition. Consider a directed graph ("digraph") $G = (V, E)$, where V denotes a set of vertices of size n and $E \subseteq V \times V$ denotes a set of edges of size m. Note that the elements of E are ordered pairs of vertices, and we will refer to an edge $e = (u, v) \in E$ as an *incoming* edge of v and an *outgoing* edge of u. We define two vertices with special properties: a "source" s, out of which data (or water, or current ...) is flowing, and a "sink" t, into which data is flowing. An example graph with these properties is shown in Fig. 1.

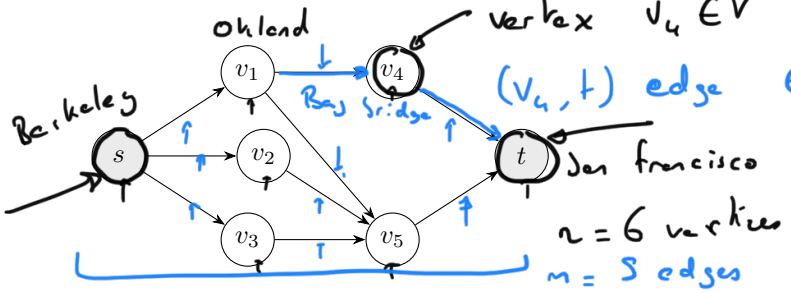
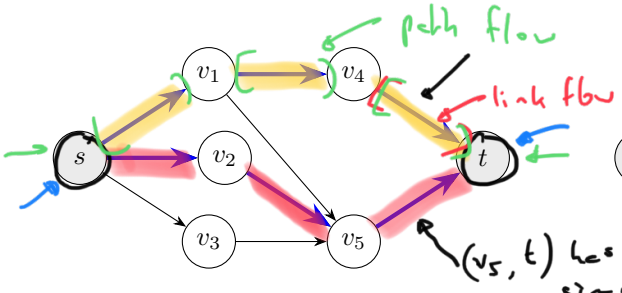
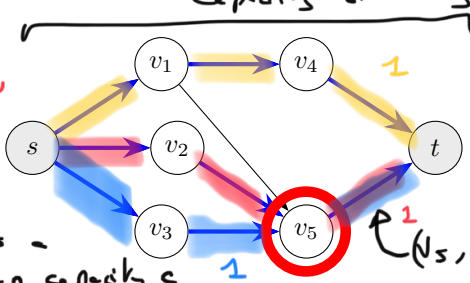


Figure 1: An example digraph with source s and sink t.

We define the *max-flow problem* as the problem of transporting the maximum amount of data from source s to destination t, assuming that s can generate infinite data but we are subject to capacity constraints on each edge of the digraph, and no data can leave the network through any node but t (i.e., for all nodes except s and t, flow in is equal to flow out). For this problem, we will assume that each edge e can support at most one unit of data flow, though the theorem holds in the more general case where edges have different capacities. Example valid and invalid "flows" for the graph above are shown below in Fig. 2, including the maximum flow.



(a) Valid maximum flow from source to sink of 2 units.



(b) Invalid flow of 3 units from source — flow into v_5 is not equal to flow out of v_5.

Figure 2: Valid maximum *(left)* and invalid *(right)* flows through the example digraph.

Lastly, we define the *min-cut* problem as the problem of partitioning the vertices of the digraph into two pieces, with s on one side and t on the other, while slicing along edges with the minimum total flow capacity. (Note that in our problem, where all edges have an equal capacity of one, this is equivalent to finding the cut that intersects the minimum number of edges.) Several possible "cuts" for the graph above are shown below in Fig. 3, including the minimum cut.

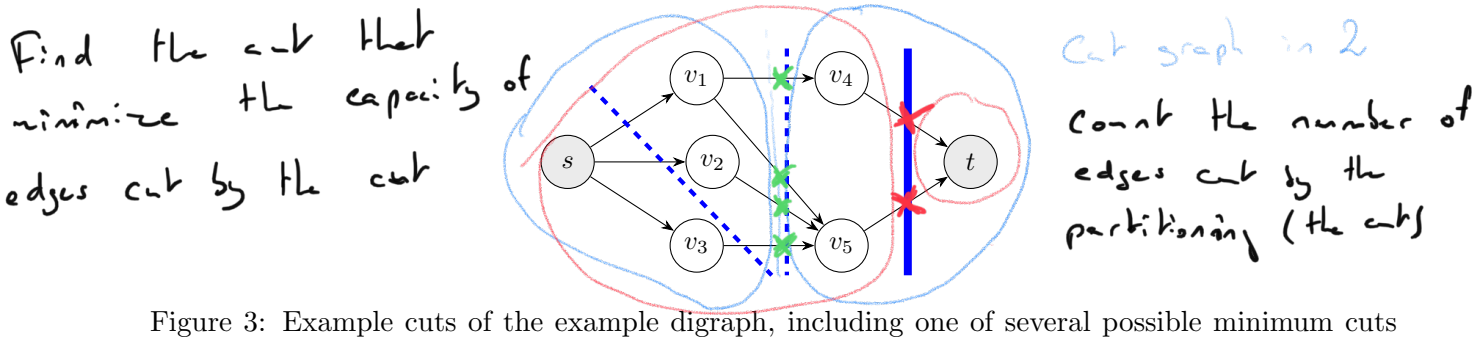


Figure 3: Example cuts of the example digraph, including one of several possible minimum cuts *(solid)* that slices through a capacity of 2 units.

The *max-flow min-cut* theorem states that the solutions to the *max-flow* and *min-cut* problems are equal — i.e., the maximum flow through a digraph of this form is exactly equal to the total capacity of all edges sliced in the minimum cut. This may be intuitive from the figures above: conceptually, the *max-flow* problem is computing flow directly, and the *min-cut* problem is finding the "bottleneck" that is preventing more data from flowing. We will now prove this theorem mathematically using duality.

(a) ***Formulating the max-flow problem (primal).*** To calculate the maximum flow from s to t through a general network, we first define $\vec{f} \in \mathbb{R}^m$ as the m-dimensional vector whose entries $f_{(u,v)} \geq 0$ are the flow through each edge (u, v). The *max-flow* problem is then the problem of maximizing the sum of these flows $f_{(s,v)}$ out of the source s (or, equivalently, the sum of the flows $f_{(v,t)}$ into t, the sink[1]) while obeying the constraints of the network, i.e.,

$$\begin{aligned} \max_{\vec{f} \geq \vec{0}} \quad & \sum_{v:(s,v)\in E} f_{(s,v)} = \sum_{v:(v,t)\in E} f_{(v,t)} \\ \text{s.t.} \quad & \sum_{u:(u,v)\in E} f_{(u,v)} = \sum_{u:(v,u)\in E} f_{(v,u)} \quad \forall v \notin \{s,t\} \\ & f_{(u,v)} \leq 1, \quad \forall (u,v) \in E. \end{aligned} \tag{Max-Flow-LP}$$

Note that the first set of constraints enforces that flow into and out of each non-source/sink node is equal, and the second that flow on each edge does not exceed capacity. We also restrict each entry of $\vec{f}$ to be nonnegative, since negative flow would be physically nonsensical.

[1] These values are equivalent for any digraph in which s has no incoming edges and t has no outgoing edges.

i. Compute the Lagrangian $\mathcal{L}(\vec{f}, \vec{\lambda}, \vec{\mu})$ of the above formulation by introducing auxiliary variables μ_v for each of the equality constraints and $\lambda_{(u,v)}$ for each of the inequality constraints. Note that $\vec{\lambda} \in \mathbb{R}^m$ and $\vec{\mu} \in \mathbb{R}^n$ for our graph of m edges and n vertices.

$$\mathcal{L}(\vec{f}, \vec{\lambda}, \vec{\mu}) = \sum_{v:(s,v)\in E} f_{(s,v)} + \sum_{v \notin \{s,t\}} \mu_v \left[\sum_{u:(u,v)\in E} f_{(u,v)} - \sum_{u':(v,u')\in E} f_{(v,u')} \right] + \sum_{(u,v)\in E} \lambda_{(u,v)} \left(1 - f_{(u,v)} \right)$$

$$\max_{\vec{f}\in X} T(\vec{f}) = \max_{\vec{f}\geq 0} \min_{\substack{\vec{\lambda}\geq 0 \\ \vec{\mu}}} \mathcal{L}(\vec{f}, \vec{\lambda}, \vec{\mu})$$

Do not dualize the constraints $\vec{f} \geq 0$

ii. Formulate the dual of the linear program Max-Flow-LP. For simplicity of formulation, assume that there exists no edge between s and t, i.e., $(s,t) \notin E$.

$$g(\vec{\lambda}, \vec{\mu}) = \max_{\vec{f}\geq 0} \mathcal{L}(\vec{f}, \vec{\lambda}, \vec{\mu})$$

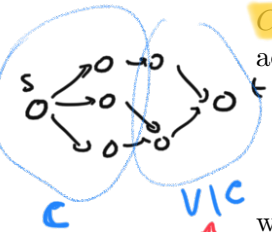
Recall that $\vec{\lambda} \geq 0$, $\vec{\mu}$

$$\mathcal{L}(\vec{f}, \vec{\lambda}, \vec{\mu}) = \sum_{v:(s,v)\in E} f_{(s,v)} \underbrace{\left(1 + \mu_v - \lambda_{(s,v)}\right)}_{\leq 0} + \sum_{u:(u,t)\in E} f_{(u,t)} \underbrace{\left(-\mu_u - \lambda_{(u,t)}\right)}_{\leq 0} + \sum_{\substack{(u,v)\in E \\ u,v \neq s,t}} f_{(u,v)} \left(\mu_v - \mu_u - \lambda_{(u,v)}\right) + \sum_{(u,v)\in E} \lambda_{(u,v)}$$

Dual problem:

$$\min_{\substack{\vec{\lambda}\geq 0 \\ \vec{\mu}}} \sum_{(u,v)\in E} \lambda_{(u,v)} \quad \text{s.t.} \quad 1 + \mu_v - \lambda_{(s,v)} \leq 0 \;\; \forall v, \qquad \mu_u + \lambda_{(u,t)} \geq 0 \;\; \forall u, \qquad \mu_v - \mu_u - \lambda_{(u,v)} \leq 0 \;\; \forall u,v \neq s,t$$

(b) ***Formulating the min-cut problem (dual).*** Recall that the *min-cut* problem is the problem of partitioning our digraph $G = (V, E)$ into two sides while slicing through the minimum number of edges. To formalize this, we define a *cut* C in G as a partition of V into two sets C and $V \setminus C$ such that $s \in C$ and $t \in V \setminus C$. The *min-cut* solution is thus the total capacity across the cut that crosses the minimum number of edges, i.e., the minimum value of



$$q(C) \doteq \sum_{(u,v)\in E} \mathbb{I}_{\{u\in C \cap v \notin C\}}$$

$$\sum_{(u,v)\in E} h_C(u,v) = \begin{cases} 1 : \text{if } u \in C \text{ and } v \notin C \\ 0 : \text{otherwise} \end{cases}$$

where the indicator function $\mathbb{I}_{\{\cdot\}}$ is equal to 1 for values in the subscript set and 0 otherwise. Note that when the capacity of each edge is one, as we assume here, $q(C)$ is exactly the number of edges crossed by a partition into C and $V \setminus C$. For clarity, we denote the minimum cut value $q^* \doteq q(C^*)$ for optimal cut C^*.

set of vertices not in C

$$\min_{C} \; q(C) \quad \text{s.t. } C \text{ a valid cut}$$

We will now show that the ?? problem we formulated above is equivalent to computing this minimum cut. We first rewrite this dual as

$$d^* = \min_{\lambda \geq 0,\ \mu_v} \sum_{(u,v)\in E} \lambda_{(u,v)} \quad \text{s.t. } \mu_v - \mu_u - \lambda_{(u,v)} \leq 0 \quad \forall (u,v) \in E, \quad \mu_s = -1,\ \mu_t = 0. \tag{Min-Cut-LP}$$

Note that this problem is exactly equivalent to ??; we can rewrite all 3 sets of ?? constraints as one by enforcing the values of μ_s and μ_t as indicated. We refer to this as the Min-Cut-LP, though we have not yet shown that it is equivalent to calculating the minimum cut. We will show this equivalence by proving that the solution of Min-Cut-LP both upper and lower bounds the minimum value of $q(C)$.

i. Show that the optimal value of Min-Cut-LP is at most $q^* = q(C^*)$, the value of the minimum cut of G.
Hint: Show that for any cut C, the optimal value of Min-Cut-LP is less than $q(C)$.

ii. (Optional) Show that the optimal value of Min-Cut-LP is at least $q^* = q(C^*)$, the value of the minimum cut of G.
Hint: For an arbitrary feasible $(\vec{\lambda}, \vec{\mu})$, first sort the distinct values of μ_v in increasing order, then consider cuts of the form $C_\alpha \doteq \{v : \mu_v \leq \alpha\}$ for different values $\alpha \in \mathbb{R}$.
Note: The proof of this inequality involves a combinatorial argument and is beyond the scope of this class. Do not feel obligated to understand it in full; we present it for completeness.

(c) ***Concluding.*** Conclude that the *max-flow min-cut* theorem holds for the examined set of digraphs.

Because of strong duality

$$p^* = d^* = q(C^*)$$

max flow

min cut

Look at solutions, Look at Yeshwanth recording

Hard is:
- Understand problem formulation
- Understand how assumptions are used in the problem formulation and solutions
- Understand a cut
- Show that $d^* \geq q(C^*)$
- Show that $d^* \leq q(C)$ → Very hard (optional)
- Showing that min-cut LP = max-flow Dual