15-451/651 Algorithm Design & Analysis, Fall 2024

Recitation #10

Objectives

- Understanding the analysis of Polytopes and their relevance to proofs of Integrality
- Understanding Approximation Algorithms and the technique of LP rounding

Recitation Problems

1. **(Integrality of flows)** In lecture we proved that given a graph G, the *matching polytope* MP_G defined by the constraints of the maximum matching problem always has integral vertices if G is bipartite.

   Let's prove something more general. Recall that bipartite matching is really just a special case of maximum flow, which is a special case of minimum-cost flows.

   Prove that for any given flow network G, the *flow polytope* FP_G defined by the LP constraints of the maximum flow / minimum-cost flow problem has integral vertices.

   Variables: $f_{u,v}$ for each edge (u,v)

   Objective: Maximize $\sum_{u\in V} f_{s,u} - \sum_{v\in V} f_{v,s}$

   Constraints:

   $$0 \le f_{u,v} \le c(u,v) \forall (u,v) \in E \quad \text{(capacity)}$$

   $$\sum_{u\in V} f_{u,v} = \sum_{w\in V} f_{v,w} \forall v \in V \setminus \{s,t\} \quad \text{(flow conservation)}$$

   (a) Consider a feasible solution $f = \{f_e\}$ that contains at least one fractional value. Here are some cases for the structure that the fractional values might appear in.

   Case 1: There exists two u-v paths for some vertices u, v, with non-integral flows on every edge.

   Case 2: There exists only one s-t path with non-integral flows on every edge.

   Case 3: There exists a cycle with non-integral flows on every edge.

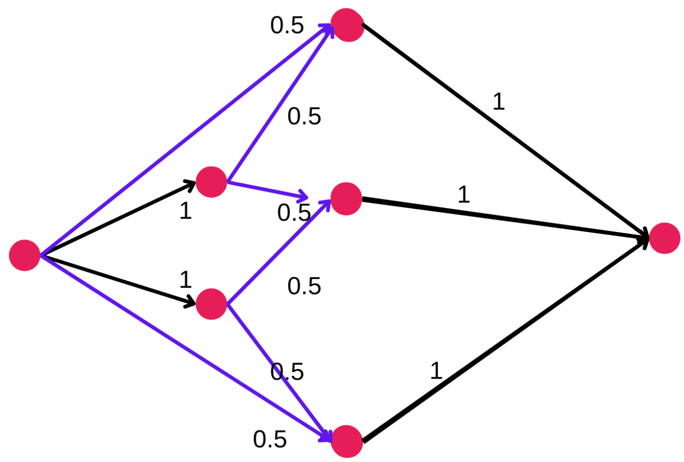
   Prove that in each of these cases, the solution is not a vertex of FP_G.

   Reminder: A vertex v of a polytope FP_G has the following properties.

   i. $v \in \mathrm{FP}_G$

   ii. For any $m \in \mathbb{R}^d$ with $m \neq 0$, then at least one of $v+m$ or $v-m$ is not in FP_G.

(b) There is a problem with this proof. In any casing proof there is the assumption or justification that the cases are exhaustive. Without that justification, a counterexample that fits in no case is good enough to disprove the proof. The previous proof has such a counterexample:



Modify the cases from the previous part to fix this counterexample (there does not have to be a one to one mapping from old cases to new cases).

(c) Provide correct proofs that the new cases do not correspond to vertex solutions.

2. **(CMU Mandated Algorithms Problem)** After Daniel leaves teaching, Jason cannot fathom continuing teaching 451 alone and becomes a telecommunications networks expert.

   He decides to tackle the problem of SONET ring loading, a classical problem in telecommunications networks and assigns you the following problem:

   We have a cycle with n vertices, numbered 0 through $n-1$ clockwise around the cycle. We are also given a set of requests. Each request is a pair (i, j) where i is the source vertex and j the target vertex. The call can be routed either clockwise or counterclockwise through the cycle. The objective is to route the calls so as to minimize the load (total number of uses) of the most loaded edge of the cycle.

   Write a linear program relaxation for the problem, and use it to give a 2-approximation algorithm using a rounding argument. Remember that a linear program relaxation is an LP such that if you could force some variables to be integers, you would solve the problem exactly.

(a) An intuitive start is have a variable L_k representing the load of an edge e_k, and to minimize the maximum load over all edges. However, this objective:

$$\textbf{minimize} \max_k L_k$$

as presented is not a linear function of the variables. Come up with a way to have an LP solver minimize the maximum load over all edges.

Hint: This might be more involved than just coming up with a new objective.

(b) Now that we have an objective function, let's build the rest of the LP relaxation.

What are the variables of our LP, and what do they represent in regards to the original problem?

(c) Finally, what are the constraints on these variables?

Hint: The hard constraint is the constraint for the load on an edge. Consider for a given edge e_k and a call (i, j) when that edge will actually have load on it from that call.

(d) Now, using this LP relaxation, give an algorithm for routing each call and prove why this gives a 2-approximation on the minimized maximum load.

3. **(Client Commissions)** Fed up with 451 students forgetting about how cool flow networks are as soon as they learn about LPs, Daniel decides to quit his job and become a freelance painter. For each commission i Daniel receives, it has an order time o_i, a painting time p_i and a finish time T_i. He wants to minimize $\sum_i T_i$.

   Note: this metric retains information about order, while $\max_i T_i$ does not.

   However, haunted by knowledge from his old life as an Algorithms professor, Daniel realizes that this problem is NP-hard, and asks you for help finding a 2-approximation in poly-time for the following relaxation:

   An "unrestricted" schedule is a schedule where Daniel can stop working on one commission in the middle of painting and resume from where he left off later. We can compute the optimal unrestricted schedule in polynomial time[1].

   Let T_i' be the time at which commission i is completed in the optimal unrestricted schedule.

[1] In fact, you can do it greedily. At any time, process the job with shortest remaining processing time. This is called the shortest remaining processing time (SRPT) rule. Try to show why this is optimal.

(a) Scheduling with "unrestricted" schedules is a relaxation of scheduling in general. In other words, any valid "restricted" schedule is also a valid "unrestricted" schedule. Note that this is only in one direction. It isn't true that a valid "unrestricted" schedule can be turned into a "restricted" schedule. What does this imply about the relationship (=, ≤, or ≥) between $\sum_i T_i'$ and OPT?

(b) For the following parts we will use a new indexing scheme. Let commission j be the j^{th} commission that the unrestricted schedule finishes. Prove the following:

$$T'_j \geq \max_{k=1}^{j} o_k$$

$$T'_j \geq \sum_{k=1}^{j} p_k$$

(c) Suppose that the restricted schedule finishes commissions in the exact same order as the unrestricted schedule. Show that $T_j \leq 2T_j'$, i.e. the j^{th} finished commission finishes at a time no greater than two times the finish time of the j^{th} finished commission in the unrestricted schedule. Conclude that the restricted schedule is a 2-approx of OPT.