15-451/651 Algorithm Design & Analysis, Fall 2023

Recitation #10

Objectives

- Finding approximations with relaxations to known algorithms
- Finding approximations with LP relaxations

Recitation Problems

1. **(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.

   As a freelance painter, he receives commissions from clients, and wants to find a schedule to complete them in a timely manner so that more people want to commission art from him. As an artist dedicated to his craft, Daniel has the self-imposed rule that whenever he starts working on a commission he doesn't stop until it's done. Daniel also does not need to eat or sleep and can spend every single second painting.

   Namely, for each commission i with an order time o_i (when the commission is ordered by a client), and a painting time p_i (how long he must work on painting that commission), Daniel has a finish time T_i. He wants to minimize $\sum_i T_i$.

   Why Daniel uses this metric and not something such as $\max_i T_i$ is because if given one big and one small commission at the same time, that metric can't differentiate between doing the small first and the large second, or vice versa, while the sum metric can.

   However, haunted by knowledge from his old life as an Algorithms professor, Daniel realizes that this problem is NP-hard. So, he wants to find a way to approximate it using a poly-time algorithm. Since he is done with his old life of algorithms, Daniel asks you for help finding a 2-approximation.

   Consider the following relaxation of the problem: 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 finish time of job i in the optimal unrestricted schedule.

   (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 ($=$, $\leq$, or $\geq$) between $\sum_i T_i'$ and OPT?

[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.

(b) For the following parts we will use a specific 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 for the j^{th} finished commission , $T_j \leq 2T'_j$. Conclude that the restricted schedule is a 2-approx of OPT.

2. **(CMU Mandated Algorithms Problem)** After Daniel leaves teaching, Danny cannot fathom continuing teaching 451 alone. More than that, he cannot imagine even being in the field of algorithms any more. So he decides to switch fields and become a telecommunications networks expert.

   As his first foray into the field, he decides to tackle the problem of SONET ring loading, a classical problem in telecommunications networks.

   Unfortunately, CMU is not happy with the fact that a core class just had one of its professors quit and the other change fields. So they force Danny to continue teaching 451, and tell him that he must continue assigning work that requires students to do algorithms.

   However disappointed, Danny realizes that he can at least give algorithms problems on telecommunications networks topics. So, he assigns you a basic version of his new research area, the SONET ring loading problem. The problem goes as such:

   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 an 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:

   $$\textbf{minimize} \max_i L_k$$

   as presented is not a linear combination.

   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.