CS 256: Algorithm Design and Analysis

Problem Set 7 (due 04/23/25)

Instructor: Sam McCauley

Problem 1. (KT 7.5) Are the following statements true or false? If true, you must give a justification.; if false, you must give a counterexample.

(a) Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e. If f is a maximum $s - t$ flow in G, then f saturates every edge out of s with flow (i.e., for all edges e out of s, we have $f(e) = c_e$).

(b) Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e. Let (A, B) be a minimum s-t cut with respect to the capacities $\{c_e : e \in E\}$. Now suppose we add 1 to every capacity; then (A, B) is still a minimum s-t cut with respect to the new capacities $\{1 + c_e : e \in e\}$.

Solution.

□

Problem 2. (Modified KT 7.23 and 7.24) Suppose you're looking at a flow network G with source s and sink t, and you want to be able to express something like the following intuitive notion: *Some nodes are clearly on the "source side" of the main bottlenecks; some nodes are clearly on the "sink side" of the main bottlenecks; and some nodes are in the middle.* However, G can have many minimum cuts, so we have to be careful in how we try making this idea precise. Here's one way to divide the nodes of G into three categories of this sort.

- We say a node v is *upstream* if, for all minimum s-t cuts (A, B), we have $v \in A$—that is, v lies on the source side of every minimum cut.
- We say a node v is *downstream* if, for all minimum s-t cuts (A, B), we have $v \in B$—that is, v lies on the sink side of every minimum cut.
- We say a node v is *central* if it is neither upstream nor downstream; there is at least one minimum s-t cut (A, B) for every $v \in A$, and at least one minimum s-t cut (A', B') for which $v \in B'$.

In this question, we design an algorithm to classify vertices of G into these categories and use the classification to characterize graphs that have a unique minimum cut. Let f be a maximum flow in G. Consider the cut (A^*, B^*), where $A^* = \{u \mid u \text{ is reachable from } s \text{ in } G_f\}$ and let $B^* = V - A^*$. Thus, $v(f) = cap(A^*, B^*)$ and (A^*, B^*) is a minimum cut of G.

(a) Show that the set A^* is the set of upstream vertices of G, that is, v is upstream if and only if $v \in A^*$.

(b) Using part (a), describe an efficient algorithm to find the downstream vertices in G. (*Hint.* Consider the graph G^R, with all direction of edges in G reversed. G^R now has source t and sink s.)

(c) Show that G has a unique minimum cut if and only if G has no central vertices, that is, the union of upstream and downstream vertices is the set V.

Solution.

□

Note. This question is about network flow reductions. We will begin those on Monday.

Problem 3. Consider a survey company that wants to ask consumers a question about products that they own. In particular, we are given a set of x consumers and y products. Each consumer i owns a subset of the products; we can only ask a consumer a question about a product they own.

We want to "evenly spread" out our questions: we want to (roughly) spread our questions evenly between the customers, and we want to ask roughly the same number of questions about each product. In particular, consumer i must be asked between a_i and b_i questions. Each product j must be surveyed by between p_j and q_j consumers. We will assume that $\sum_i a_i > \sum_j p_j$ (this will be important in our reduction).

Our goal: Determine whether it is possible to design a survey satisfying these constraints. For this problem, **I will give you the reduction**. Your job is to prove that it is correct: that this flow network has a given value if and only if there is an assignment as described above.

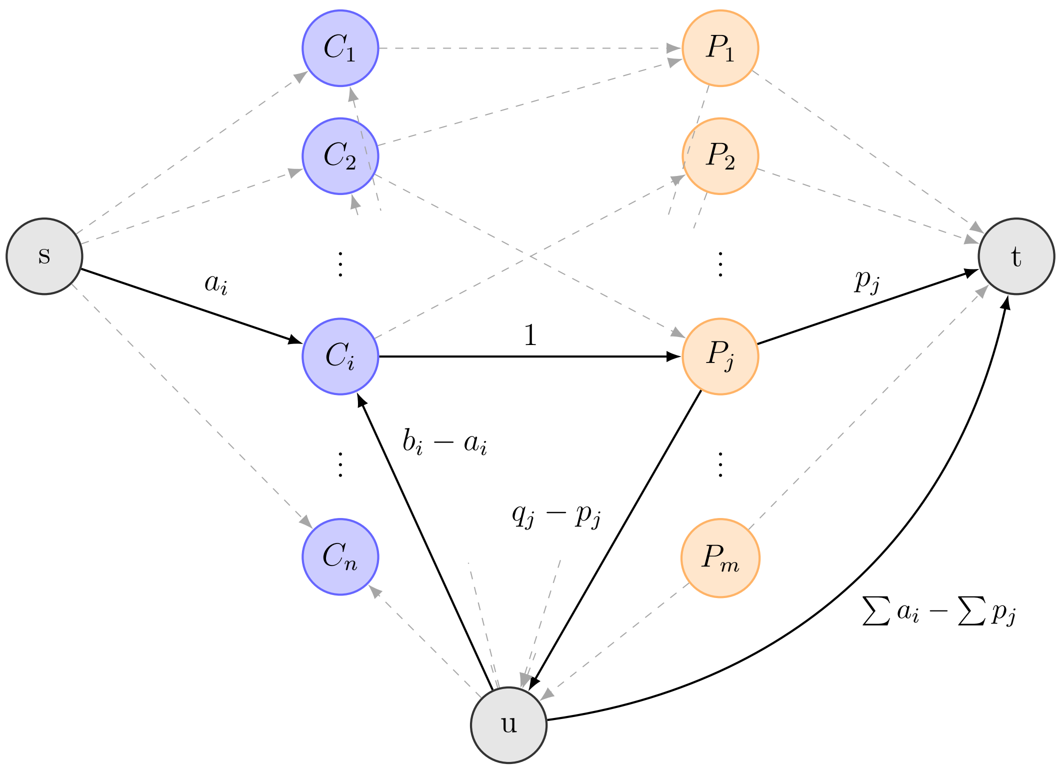

Figure 1: A diagram of the flow network given above. To keep things simple, the focus is on the edges involving a customer C_i and a product P_j (only the capacities of these edges are included). Some edges (like between u and C_1) are truncated to avoid muddying the diagram. Note that there is only an edge between a consumer and a product if the consumer owns the product: so in the above, C_1 owns P_1 but not P_2; C_2 owns P_1 and P_j, and so on.

Flow network: We construct a flow network G with the following nodes:

- A source node s and a sink node t.
- A node for each consumer i and each product j.
- An auxiliary node u to manage upper and lower bound constraints.

And the following edges (see the diagram on the next page):

- For each (i, j) where consumer i owns product j, add an edge from consumer node i to product node j with capacity 1.
- For each consumer i, add an edge from s to i with capacity a_i.
- For each product j, add an edge from j to t with capacity p_j.
- For each consumer i, add an edge from u to i with capacity $b_i - a_i$.
- For each product j, add an edge from j to u with capacity $q_j - p_j$.
- Add an edge from u to t with capacity $\sum_i a_i - \sum_j p_j$.

(a) Prove that if there exists a way to ask customers questions about the products they own satisfying the above, then there exists a flow through this network with value $\sum_i a_i$. Your proof should state what the flow is through each edge in the network, and briefly explain why this is a valid flow.

> **Hint.** Start with the bipartite matching reduction we saw in class. How can we extend it to this problem (and how can we extend it to the extra vertex and extra edges in the network)?

Solution.

□

(b) Show that if the flow through this network is $\sum_i a_i$, then there exists a way to ask customers questions about the products they own satisfying the above. Your answer should state what customers should be asked about what products, and should explain why these questions satisfy the constraints above.

> **Hint.** Using what we learned in class, if the total flow through the network is $\sum_i a_i$, can we immediately figure out the assignment of flow to the edges adjacent to s and t? Combine this observation with your answer above.

Solution.

□