Problem Set 8 Solutions

Problem 8-1. Inspirational fires

To foster a spirit of community and cut down on the cliqueishness of various houses, MIT has decided to sponsor community-building activities to bring together residents of different living groups. Specifically, they have started to sponsor official gatherings in which they will light copies of CLRS on fire.

Let G be the set of living groups at MIT, and for each $g \in G$, let $residents(g)$ denote the number of residents of living group g. President Hockfield has asked you to help her out with the beginning of her administration. She gives you a list of book-burning parties P that are scheduled for Friday night. For each party $p \in P$, you are given the number $size(p)$ of people who can fit into the site of party p.

The administration's goal is to issue party invitations to students so that no two students from the same living group receive invitations to the same book-burning party. Formally, they want to send invitations to as many students as possible while satisfying the following constraints:

- for all $g \in G$, no two residents of g are invited to the same party;
- for all $p \in P$, the number of people invited to p is at most $size(p)$.

(a) Formulate this problem as a linear-programming problem, much as we did for shortest paths. Any legal set of invitations should correspond to a feasible setting of the variables for your LP, and any feasible integer setting of the variables in your LP should correspond to a legal set of invitations. What objective function maximizes the number of students invited?

Solution: Let $x_{p,g}$ be a variable representing the number of invitations to party $p \in P$ sent to residents of group $g \in G$.

$$\begin{array}{llcl}
\max \sum_{p \in P, g \in G} x_{p,g} & & & \\
\text{s.t.} & & & \\
\quad \forall p \in P & \sum_{g \in G} x_{p,g} & \le & size(p) \\
\quad \forall g \in G & \sum_{p \in P} x_{p,g} & \le & residents(g) \\
\forall g \in G, p \in P & x_{p,g} & \le & 1 \\
\forall g \in G, p \in P & x_{p,g} & \ge & 0
\end{array}$$

If we have a legal set of invitations, then it is easy to see that all of these constraints are satisfied, since the set of invitations must satisfy the stated conditions. Similarly, any feasible integral setting of the variables yields $x_{p,g} \in \{0, 1\}$, and it is straightforward to verify that sending an invitation to party P to some resident of group g if and only if $x_{p,g} = 1$ will satisfy the requirements. (We do not permit the administration to send more than one invitation to a party p to the same student; thus we can send only one invitation to party p to group g.) Because the objective function $\sum_{p,g} x_{p,g}$ measures the number of invitations sent (and thus the number of students invited), an optimal setting of the variables for the LP therefore corresponds to a maximum number of invited students.

(b) Show how this problem can be solved using a maximum-flow algorithm. Your algorithm should return a set of legal invitations, if one exists, and return FAIL if none exists.

Solution: Define the graph $G' = (V, E)$, and capacities c on the edges, where

- $V = \{s, t\} \cup G \cup P$, where s and t are brand new source and sink nodes, respectively.
- $(g, p), (s, g), (p, t) \in E$ for every $g \in G, p \in P$.
- $c(g, p) = 1$, $c(s, g) = \mathit{residents}(g)$, and $c(p, t) = \mathit{size}(p)$.

Run the Edmonds/Karp max-flow algorithm on G' to get an (integral) max flow f on the graph. (There was a slight ambiguity in the phrasing of the question here: if you interpret "if one exists" to mean "if there exists any legal set of invitations", then you should never fail; if you interpret it to mean "if there exists a legal set of invitations such that every student gets an invitation", then you should return "fail" if the value of f is less than $\sum_{g \in G} \mathit{residents}(g)$.) For every party p, send an invitation to party p to an uninvited member of group g if and only if $f(g, p) = 1$.

The constraints on the graph guarantee that no two residents of the same house can get an invitation to the same party and that no party has more invitations sent to it than its size. Thus if the algorithm returns a set of invitations, then they are valid. Conversely, any legal set of i invitations can be expressed as a flow of value i.

(c) (*Optional.*) Can this problem can be solved more efficiently than with a maximum-flow algorithm?

Solution: It seems like some version of a greedy algorithm can be used to solve this problem more efficiently, but after an hour-long course-staff meeting, we weren't convinced either way.

Problem 8-2. Zippity-doo-dah day

On Interstate 93 south of Boston, an ingenious device for controlling traffic has been installed. A lane of traffic can be switched so that during morning rush hour, traffic flows northward to Boston, and during evening rush hour, if flows southward away from Boston. The clever engineering behind this design is that the reversible lane is surrounded by movable barriers that can be "zipped" into place in two different positions.

For some reason, gazillions of people have decided to drive from Gillette Stadium in Foxboro, MA to Fenway Park. (They seem to be cursing a lot, or, at the very least, you hear them shouting the word "curse" over and over.) Governor Mitt asks you for assistance in making use of the zipper-lane technology to increase the flow of traffic from Foxboro to Fenway.

We can model this road network as directed graph $G = (V, E)$ with source s (Foxboro), sink t (Fenway), and integer capacities $c : E \to \mathbb{Z}^+$ on the edges. You are given a maximum flow f in the graph G representing the rate at which traffic can move between these two locations. In this question, you will explore how to increase the maximum flow using "zippered" edges in the graph.

Let $(u, v) \in E$ be a particular edge in G such that $f(u, v) > 0$ and $c(v, u) \geq 1$. That is, there is positive flow on this edge already, and there is positive capacity in the reverse direction. Suppose that zipper technology increases the capacity of the edge (u, v) by 1 while decreasing the capacity of its ***transpose*** edge (v, u) by 1. That is, the zipper moves 1 unit of capacity from (v, u) to (u, v).

(a) Give an $O(V + E)$-time algorithm to update the maximum flow in the modified graph.

Solution: The algorithm is simple: increment the capacity of (u, v), and search for a single augmenting path in the residual graph in $O(E + V)$ time. Augment along that path if one is found. Decrement the capacity of (v, u), and return the resulting flow.

If there is a minimum cut in G that (u, v) does not cross, then the same flow f is a maximum flow in the modified graph: it is feasible because no capacities have decreased, and it is maximum because the same unmodified minimum cut still gives an upper bound on the size of the flow. If (u, v) does cross all minimum cuts, then the maximum flow will increase by 1. Thus, the new maximum flow has value either $|f|$ or $|f| + 1$, and a single augmenting path will suffice to find the updated maximum flow (since the residual capacities are all integral).

Finally, we observe that decrementing the capacity of (v, u) does not affect the maximum flow: because there is positive flow on (u, v) in f, there is net negative flow on (v, u), and the soon-to-be-deleted unit of capacity on (v, u) is unused. Eliminating any capacity unused by the max flow cannot make the flow infeasible, and it therefore must remain a maximum flow in the modified graph.

Zap 86 years into the future! Zipper lanes are commonplace on many more roads in the Boston area, allowing one lane of traffic to be moved from one direction to the other. You are once again given the directed graph $G = (V, E)$ and integer capacities $c : E \to \mathbb{Z}^+$ on the edges. You also have a zipper function $z : E \to \{0, 1\}$ that tells whether an additional unit of capacity can be moved from (v, u) to (u, v). For each $(u, v) \in E$, if $z(u, v) = 1$, then you may now choose to move 1 unit of capacity from the transpose edge (v, u) to (u, v). (You may assume that if $z(u, v) = 1$, then the edge (v, u) exists and has capacity $c(v, u) \geq 1$. Again, you are given a source node $s \in V$, a sink node $t \in V$, and a maximum flow f. Governor Mitt IV asks you to configure all the zippered lanes so that the maximum flow from s to t in the configured graph is maximized.

(b) Describe an algorithm that employs a maximum-flow computation to determine the following:

1. the maximum amount that the flow can be increased in this graph after your chosen zippered lanes are opened; and
2. a configuration of zippered lanes that allows this flow to be achieved.

Solution: One can solve this problem by extending the definition of an augmenting path to include the possibility of zipping an edge to increase the capacity of a min cut, but there's an easier way.

Note that in a net flow, there is never positive flow in both directions (u, v) and (v, u). The idea of the algorithm is this: since we can compute a maximum flow that only uses an edge in one direction, we'll add the zippered capacity *in both directions*, and compute max flow. The above note guarantees that the resulting flow satisfies the zippered capacity constraints—i.e., does not use the extra zipped capacity along both (u, v) and (v, u).

Given $G = (V, E)$, nodes s, t, capacity function c, and zipper function z:

1. For every pair $\{u, v\} \in E$, set $c'(u, v) := z(u, v) + c(u, v)$.
2. Run the Edmonds/Karp max flow algorithm on G with capacities c' to get a max flow f.
3. For every pair (u, v) with $z(u, v) = 1$, if $f(u, v) > 0$, then set the direction of the zippered lane on (u, v) so that the unit of capacity is moved from the transpose edge (v, u) to the edge (u, v). If $f(u, v) = f(v, u) = 0$, set the direction of the zipper lane for (u, v) arbitrarily.
4. Return f and the direction settings above.

The running time of this algorithm is then $O(VE^2)$.

To prove correctness, we'll show that (i) the value of f is at least the max flow f^* for the best zippered setting, and (ii) the value of f is at most f^*. For condition (i), observe the following: the capacity of each edge in the best zippered setting is upper bounded by the capacities c', which immediately implies that the value of f is at most f^*. For condition (ii), note that f is a feasible flow in some zippered graph (in particular, in the graph with edges oriented as per the returned directions).

Because the graph G is actually a network of roads, it is nearly planar, and thus $|E| = O(V)$.

(c) Give an algorithm that runs in time $O(V^2)$ to solve the graph configuration problem under the assumption that $|E| = O(V)$. You should assume that the original flow f

has already been computed and you are simply determining how best to increase the flow.

Solution: Run the algorithm from part (a) on each edge in E, in an arbitrary order. This takes $O(E^2) = O(V^2)$ time under the stated assumptions.