Summary of solutions to problem set 8

Problem 1: Flow decomposition

Algorithm:

1. Start with $i = 1$. Find an s-t path P_i in $\operatorname{supp}(f)$. Set $\gamma_i = \min\{u_a : a \in P_i\}$. Reduce f by $\gamma_i P_i$. Increase i. Repeat.
2. Start with $j = 1$. Find a cycle C_j in $\operatorname{supp}(f)$. Set $\eta_j = \min\{u_a : a \in C_j\}$. Reduce f by $\eta_j C_j$. Increase j. Repeat.

Each iteration reduces f_a to 0 for at least one arc, so we make at most $|A|$ iterations, thus the total number of paths and cycles in the decomposition is at most $|A|$. Each iteration takes $O(|V| + |A|)$ time (BFS to find an s-t path or a cycle), so the overall complexity is $O(|A| \cdot (|V| + |A|))$.

Problem 2: Improving over Edmonds-Karp: Blocking flows and Dinic's algorithm

(a) Check capacity and balance constraints. Plug $v = s$ into $f'\left(\delta_G^+(v)\right) - f'\left(\delta_G^-(v)\right)$.

(b) If a maximum flow were not a blocking flow, we would be able to augment it by $\varepsilon \cdot P$ where P is an s-t path that has no saturated arcs and $\varepsilon > 0$ is sufficiently small.

(c) See solutions. Note that a similar counterexample works for layered subgraphs.

(d) On the one hand, every shortest s-t path is included in the s-t layered subgraph $G_{L,f}$ by construction. On the other hand, by construction, $\forall a = (u, v) \in G_{L,f}$ we have $d(s, v) = d(s, u) + 1$, which means that every s-t path in $G_{L,f}$ is a shortest s-t path in (V, U_f).

(e.i) Use Problem 1, note that there are no cycles in the decomposition, then apply (d).

(e.ii) Induct on the number k of terms in the decomposition $f_0 = \sum_{i=1}^{k} \gamma_i \chi^{P_i}$ of the blocking flow f_0. The base case, $k = 1$, is simple. For the induction step, carefully show that $\gamma_k \chi^{P_k}$ is an augmenting flow in $G_{f^{(1)}}$ where $f^{(1)} = f + \sum_{i=1}^{k-1} \gamma_i \chi^{P_i}$.

(f) Proof by contradiction. Assume there exists a path P in $(V, U_{f'})$ of length $d(s, t)$. Note that P was not present in (V, U_f), because otherwise it would contain an arc a that is saturated by the blocking flow f, but then a could not appear in $(V, U_{f'})$. Thus there exists an arc $b \in P$ that is present in $(V, U_{f'})$ and not in (V, U_f), which means that the back arc b^R was used in (V, U_f). By point (e), there is some path P_i in the decomposition of the blocking flow that uses b^R. Since P and P_i have the same length, they would appear in the same phase of the Edmonds–Karp algorithm, but they are using b and b^R respectively, which cannot happen in the same phase of the Edmonds–Karp algorithm. This contradiction proves the desired result.

(g) Combine points (f) and (e) and use correctness of the Edmonds–Karp algorithm.

(h) Run BFS twice: starting from s and from t. Include an arc (u, v) into the s-t layered subgraph if and only if $d(s, v) = d(s, u) + 1$.

(i) Initialization takes time $O(m)$. Since the initial length of a shortest s-t path is at most $n - 1$ and the length of a shortest s-t path strictly decreases after each augmentation by a blocking flow, we make $O(n)$ augmentations.

(j) Run DFS to find an s-t path in the layered subgraph, augment the blocking flow along this path (push flow equal to minimal residual capacity on this path and update residual capacities), repeat. Whenever a dead end is encountered (i.e. if there are no arcs from the current vertex), remove the last arc we took as we exit the recursion. In other words, we always keep the incidence lists updated so that the first arc in every list does not lead to a discovered dead end. Note that each path augmentation saturates (and thus removes from further consideration) at least one arc and that if an arc leads to a dead end at some point in the algorithm, then it will be considered at most once. Therefore, we perform $O(m)$ path augmentations each taking $O(n)$ time and we spend $O(m)$ ammortized time (i.e. over the course of the whole procedure) removing arcs leading to dead ends. Thus, the overall complexity of this algorithm for finding blocking flows is $O(mn + m) = O(mn)$.