Q1 Teaching Assignment

Suppose there are \(m\) courses: \(c_1, \ldots, c_m\). For each \(j \in \{1, \ldots, m\}\), course \(c_j\) has \(s_j\) sections. There are \(n\) professors: \(p_1, \ldots, p_n\). For each \(i \in \{1, \ldots, n\}\), professor \(p_i\) has a teaching load of \(\ell_i\), and has a preference for a subset of courses \(A_i \subseteq \{c_1, \ldots, c_m\}\).

Your goal is to find a feasible assignment of professors to courses such that:

- each professor \(p_i\) is assigned exactly \(\ell_i\) courses,
- each course \(c_j\) assigned to exactly \(s_j\) professors,
- no professor is assigned a course that they do not prefer,
- no professor teaches multiple sections of the same course.

Reduce this problem to the network flow problem. Justify that your reduction is correct. What is the worst-case running time of the naïve Ford-Fulkerson algorithm on the network you created?

Solution to Q1

Create a flow network with the set of vertices \(\{s, t, p_1, \ldots, p_n, c_1, \ldots, c_m\}\) and the following edges:

- \((s, p_i)\) with capacity \(\ell_i\), for each \(p_i\);
- \((c_j, t)\) with capacity \(s_j\), for each \(c_j\);
- \((p_i, c_j)\) with capacity 1, for each \(p_i\) and \(c_j\) such that \(p_i\) prefers to teach \(c_j\) (i.e. \(c_j \in A_i\)).

The reduction now proceeds as follows.

1. Compute a maximum flow \(f\) in this network via the Ford-Fulkerson algorithm. Note that it will return an integral flow.

2. If \(f(s, p_i) = \ell_i\) for each professor \(p_i\) and \(f(c_j, t) = s_j\) for each course \(c_j\), then a feasible assignment exists. Assign \(p_i\) to teach \(c_j\) whenever \(f(p_i, c_j) = 1\).

3. Otherwise, a feasible assignment does not exist.

To see that this algorithm is correct, we notice that there is a 1-1 correspondence between an integral flow \(f\) and an assignment of professors to courses in the following way:

- \(f(p_i, c_j) = 1\) if and only if \(p_i\) is assigned to teach \(c_j\), and 0 otherwise.
- \(f(s, p_i)\) is the number of courses that \(p_i\) is assigned to teach.
- \(f(c_j, t)\) is the number of professors assigned to course \(c_j\).
Due to the way the edges are created in the flow network, the third and fourth constraints in the problem statement are satisfied in the assignment corresponding to any valid flow. Therefore, a feasible assignment exists if and only if the maximum flow $f$ satisfies $f(s, p_i) = \ell_i$ for each professor $p_i$ and $f(c_j, t) = s_j$ for each course $c_j$, as we are checking in the second step.\footnote{Note that the capacity constraints would only ensure $f(s, p_i) \leq \ell_i$ and $f(c_j, t) \leq s_j$; they would not ensure equality.}

This flow network has $m + n + 2$ vertices and $m + n + \sum_{i=1}^n |A_i| \leq m + n + m \cdot n$ edges. The sum of capacities of edges leaving $s$ is $\sum_{i=1}^n \ell_i \leq \sum_{i=1}^n m = n \cdot m$.\footnote{Clearly, for a feasible solution to exist, we must have $\ell_i \leq m$ for each $i$. If you prefer, you can add a preprocessing step where you check this and return “No feasible solution exists” if it is violated.} Hence, the worst-case running time of the naïve Ford-Fulkerson algorithm is $O(n^2 \cdot m^2)$, which is polynomial in the input length.

### Q2 Mobile Computing

Suppose there are $n$ mobile computing clients in a certain town. Each client $i \in \{1, \ldots, n\}$ is located at a given point $(x_{ci}, y_{ci})$, and must be connected to exactly one of $m$ “base stations”. We are also given the coordinates $(x_{bj}, y_{bj})$ of each base station $j \in \{1, \ldots, m\}$.

There is a “range parameter” $r$: a client can only be connected to a base station within a Euclidean distance of $r$. There is also a “load parameter” $L$: no more than $L$ clients can be connected to any single base station.

Reduce this problem to the network flow problem. Justify that your reduction is correct. What is the worst-case running time of the naïve Ford-Fulkerson algorithm on the network you created?

#### Solution to Q2

Create a flow network with the set of vertices $\{s, t, c_1, \ldots, c_n, b_1, \ldots, b_m\}$ (where $c_i$ represents client $i$ and $b_j$ represents base station $j$) and the following edges:

- $(s, c_i)$ with capacity 1, for each $c_i$;
- $(b_j, t)$ with capacity $L$, for each $b_j$;
- $(c_i, b_j)$ with capacity 1, for all $c_i$ and $b_j$ such that $\sqrt{(x_{ci} - x_{bj})^2 + (y_{ci} - y_{bj})^2} \leq r$.\footnote{Putting any capacity on this edge that is at least 1 will yield the same maximum flow because, due to $(s, c_i)$ having capacity 1, only a unit flow can enter $c_i$, so only a unit flow can leave $c_i$.}

The reduction now proceeds as follows.

1. Compute a maximum flow $f$ in this network via the Ford-Fulkerson algorithm. Note that it will return an integral flow.

2. If $f(s, c_i) = 1$ for each client $i$, then a valid assignment exists. In this case, for each client $i$, because $c_i$ is receiving a unit flow from $s$ and $f$ is integral, there will be a unique $j$ such that $f(c_i, b_j) = 1$. Assign each client $i$ to base station $j$ such that $f(c_i, b_j) = 1$.

3. Otherwise, a valid assignment does not exist.

---

1. Note that the capacity constraints would only ensure $f(s, p_i) \leq \ell_i$ and $f(c_j, t) \leq s_j$; they would not ensure equality.
2. Clearly, for a feasible solution to exist, we must have $\ell_i \leq m$ for each $i$. If you prefer, you can add a preprocessing step where you check this and return “No feasible solution exists” if it is violated.
3. Putting any capacity on this edge that is at least 1 will yield the same maximum flow because, due to $(s, c_i)$ having capacity 1, only a unit flow can enter $c_i$, so only a unit flow can leave $c_i$. 

---

2
The proof of correctness is similar to that in Q1. The 1-1 correspondence between an integral flow $f$ and an assignment of clients to base stations is given as: $f(c_i, b_j) = 1$ if and only if client $i$ is assigned to base station $j$. The way the edges are created ensures that the range constraints are respected. The capacity of $L$ on each $(b_j, t)$ edge ensures that the load constraints are also respected. Thus, a valid assignment exists if and only if each client is actually assigned, i.e., if each $c_i$ receives a flow of 1 from $s$, as we check in the second step.

The flow network has $m + n + 2$ vertices and at most $m + n + m \cdot n$ edges. The total capacity of edges leaving $s$ is $n$. Hence, the worst-case running time of the naïve Ford-Fulkerson is $O(m \cdot n^2)$. 