CSC373— Algorithm Design, Analysis, and Complexity — Spring 2018

Solutions for Tutorial Exercise 6: Catch Up and Network Flow

---

1. **Reading Week.** Catch up on your studying for the course and bring your unresolved questions to the next tutorial, which is on Mon., Feb 26.

2. **Teaching Assignments.** You are chair of the Department of Computer Science, and are given the following information. There is a set of profs $P = \{p_n\}_{n=1}^N$, a set of courses $C = \{c_k\}_{k=1}^K$, and the number of course sections $S = \{s_k\}_{k=1}^K$ being offered next term. Here s_k is a positive integer equal to the number of different sections of course c_k that need to be taught next term. In addition, for each prof p_n, you are given the set of courses that he/she likes to teach, say $T_n \subset C$, along with her teaching load L_n. This teaching load L_n is the number of courses that prof p_n is assigned to teach next term (here $L_n \geq 0$ is an integer). Finally, suppose the sum of teaching loads, $L = \sum_{n=1}^N L_n$, is equal to the number of course sections that need to be taught, $S = \sum_{k=1}^K s_k$.

   The prof-to-course assignment problem is then to assign each prof p_n to exactly L_n course sections, with the courses themselves chosen from T_n, and ensure that each section of each course gets assigned a professor.

   (a) Represent the prof-to-course assignment problem as a $s - t$ network flow problem. Describe exactly the relationship between flows in your graph and prof-to-course assignments. That is, on one hand, given any flow (or perhaps a restricted family of flows) explain what assignment it represents. And, on the other hand, given an assignment explain how this assignment can be represented as a valid flow in your graph.

   (b) One question we might ask is whether an assignment is even possible? Can the professors cover all the sections of all the courses? Describe in detail how you can answer that question using your network flow. (You can use algorithms described in the course notes or text without writing them out again.)

   (c) In situations where a prof-to-course assignment is possible, explain how your network flow provides a suitable assignment. In particular, explain why this assignment must satisfy the constraints of the prof-to-course assignment problem.

   (d) In situations where it is not possible to cover all the courses, how can you clearly convince the faculty that no such assignment is possible? Your argument must be worded in terms of the sets of courses professors like to teach, their teaching loads, and the sections that need to be taught, and **not** in terms for your network flow problem. It must be verifiable from these given quantities using only simple arithmetic.

Soln 2a: Draw a bipartite graph, with vertices for professors $\{p_n\}_{n=1}^N$ on the left, and courses $\{c_k\}_{k=1}^k$ on the right.

- There is a directed edge from p_n to each c_k in T_n with capacity ∞ (or some integer $> S$).
- Add a source node s, with no incoming edges, and the only outgoing edges are those to each $p_n \in P$. The (s, p_n) edge has capacity equal to that professor's teaching load L_n.
- Add a sink node t, with no outgoing edges, and the only incoming edges are from each $c_k \in C$. The (c_k, t) edge has capacity equal to the number of sections s_k that need to be taught.

We assume that one prof will teach one section. No course sharing. So the flow needs to be integer valued. Given an integer-valued, feasible flow $f(e) \geq 0$, then $f((p_n, c_k))$ represents the number of sections of course c_k that prof p_n is assigned to teach. Also $f((s, p_n))$ is the total number of sections p_n is assigned (using flow conservation at vertex p_n), and this is bounded above by the capacity L_n of the edge. Finally, $f((c_k, t))$ is the total number of sections of course c_k that have an assigned professor (using flow conservation at vertex c_k), and this is bounded above by the total number of sections to be taught s_k, which is the capacity of the edge. The reverse is similar. Given an integer valued assignment of professors to courses they like to teach, the number of sections of each course, and so on, define a function $f(e)$ as described in the previous paragraph.

This is a feasible flow, i.e. $0 \le f(e) \le c(e)$ for each edge e, and flow conservation holds at each vertex $v \in V \backslash \{s,t\}$

Soln 2b-2c: Compute the max flow $f(e)$. Since all the capacities are integer valued, we can take this flow to be integer valued. Then we claim $v(f) = S$ iff an assignment is possible. We explain this next.

First, note the computed max flow is necessarily feasible. Consider the value of the flow $v(f)$, which is defined as the sum of the flow on edges leaving s. We are interested in the flow into t. In the notation used in the lecture notes, the flow into t is the flow across the $s-t$ cut $(V\backslash\{t\}, \{t\})$, which is the net flow $N_f(V\backslash\{t\})$. By the Net Flow Lemma, $N_f(V\backslash\{t\}) = v(f)$. Therefore, $v(f) = S$ iff the flow into t is S.

But since $f(e) \le c(e)$, and $\sum_{e=(c_k,t)} c(e) = S$, we know $v(f) = S$ iff every edge into t is saturated. Therefore $v(f) = S$ iff every course c_k has been assigned $c((c_k,t)) = s_k$ professors.

Secondly, in the case $v(f) = S$, this assignment must satisfy all the constraints given. In particular the constraints of flow conservation at vertices other than s, t, and that the assignments must be non-negative and integer valued are all satisified. The only remaining constraint is that no prof is teaching more than her assigned load, and this is ensured by the capacity $c(e) = L_n$ on the edge $e = (s, p_n)$.

Soln 2d: Use the max flow to compute a min cut (A,B) using the residual graph, as described in the lecture notes. In the case an assignment is not possible, we saw from (b-c) above that it must be the case that $v(f) < S$. By the min-flow, max-cut theorem, $cap(A,B) = v(f)$, and therefore $cap(A,B) < S$. This cut therefore provides a suitable certificate that there is no flow with $v(f) = S$, and no suitable assignment is possible.

We next translate this to a simple message for the chair to give to the department.

Let A be the set of vertices reachable from s in the residual graph G_f for a maximum flow f. Then

$$A = \{s, P_A, C_A\},\ B = \{t, P_B, C_B\} \tag{1}$$

where the set of profs P is the disjoint union of P_A and P_B and the set of courses C is the disjoint union of C_A and C_B.

The courses in C_B are unreachable from s in G_f. Note C_B cannot be empty, otherwise the cut (A,B) includes all course-to-t edges (and the capacity $cap(A,B) \ge S$, a contradiction).

Chair's Simple Claim: The sum of the course loads for all the profs who like to teach any of the courses in C_B is less than the total number of sections required for these courses in C_B. Clearly we cannot make a feasible assignment in this case. C'mon man!

Proof: Since there are only s to prof, prof to course and course to t edges:

$$\begin{aligned} cap(A,B) &= cap(\{s, P_A, C_A\}, \{P_B, C_B, t\}) \\ &= cap(\{s\}, P_B) + cap(P_A, C_B) + cap(C_A, \{t\}) \\ &< L = S. \end{aligned}$$

Note that, by the definition of capacity, any edges from P_B to C_A do not contribute to the cut capacity $cap(A,B)$.

With the construction of the graph the prof-to-course edges (where they exist) have huge capacities, $(\ge S)$. So none of them can be cut in the minimum cut (A,B). That is $cap(P_A, C_B) = 0$. This means that no prof in P_A likes to teach any course in C_B. In other words, all the profs who like to teach any courses in C_B are in P_B.

Since the (c,t) edge for any course c has the capacity s_c, it follows that

$$cap(C_A, \{t\}) = \sum_{c \in C_A} s_c = S - \sum_{c \in C_B} s_c.$$

So we have:

$$S > cap(A,B) = cap(\{s\}, P_B) + S - \sum_{c \in C_B} s_c$$

Or equivalently:

$$\sum_{c \in C_B} s_c > cap(\{s\}, P_B) = \sum_{p \in P_B} L_p,$$

which proves the chair's claim.