Flow network application example: committee assignment problem

CSCI 382, Algorithms

November 1, 2019

The input to the **committee assignment problem**, consists of the following data:

- A set of people $P = \{p_1, p_2, \ldots, p_m\}$.
- A set of committees $C = \{c_1, c_2, \ldots, c_n\}$.
- Each person p_i has a set of committees W_i they are willing to be on.
- Each person p_i has a max number of committees, m_i they are willing to be on in total.
- Each committee c_i has a required number of people z_i.

The problem is to come up with an assignment of people to committees that satisfies all the constraints (*i.e.* each committee has exactly z_i different people assigned to it, and each person p_i is assigned only to committees they are willing to be on, and to no more than m_i committees in total), or report that this is not possible.

Solution via flow network

The solution is to turn the problem into an appropriate flow network; finding a max flow will then correspond to solving the problem. In particular, we build a flow network $G = (V, E)$ as follows:

- $V = P \cup C \cup \{s, t\}$, that is, there is one vertex for each person, one for each committee, and two extra vertices to serve as the source and sink.
- Connect the source s to each p_i with a directed edge of capacity m_i.
- Connect each c_i to the sink t with a directed edge of capacity z_i.
- Make a directed edge (p_i, c_j) with capacity 1 for each $c_j \in W_i$; that is, connect each person to those committees they are willing to be on.

When writing up a solution involving a flow network, you are also encouraged to draw it instead of tediously specifying all the vertices and edges formally; I was just too lazy to draw a nice picture for this writeup.

To solve the committee assignment problem, we can now find a max flow on G (using *e.g.* the Ford-Fulkerson algorithm, or any of a number of variant algorithms for finding max flows). If the max flow has value $z_1 + z_2 + \cdots + z_n$ (that is, if it completely saturates all the incoming edges to the sink), then a valid committee assignment exists, and we can read it off by assigning person p_i to committee c_j if and only if the edge (p_i, c_j) has a flow of 1. Otherwise, no valid committee assignment exists.

Runtime analysis

Ford-Fulkerson can be implemented to run in $O(m \cdot \min(C_s, C_t))$, where m is the number of edges in the network, C_s is the total capacity of edges leaving the source node s, and C_t is the total capacity of edges entering the target node t. In our case, there are $|P|$ edges from s to all the p_i, $|C|$ edges from all the c_j to t, and a maximum of $|P||C|$ internal edges (the worst case is if everyone is willing to be on every committee), for a total of $m = |P| + |C| + |P||C| = O(|P||C|)$ edges. Also, we can assume that $C_t \geq C_s$ (otherwise it is impossible to fill all the committees). The total capacity of edges entering t is $z_1 + z_2 + \cdots + z_n = Z$, the total number of committee slots to be filled. Overall, then, this algorithm will run in $O(|P| \cdot |C| \cdot Z)$ time, the product of the number of people, number of committees, and number of committee slots.

Actually we could be even more precise: the number of edges is exactly $|P| + |C| + \sum_i |W_i|$.

Proof

Why does this work? The essential idea is to prove that valid committee assignments correspond exactly to maximum flows on the network G we constructed. We start be explaining how to construct a committee assignment from a given flow, or vice versa.

- Given a flow f, construct a committee assignment by assigning person p_i to committee c_j if and only if there is a unit of flow going along the edge (p_i, c_j).
- Likewise, given a committee assignment, put one unit of flow along edge (p_i, c_j) if and only if person p_i has been assigned to committee c_j; then add flow to edges from s or edges to t in such a way that flow is preserved at each vertex.

We now need to show that valid flows correspond to valid committee assignments and vice versa.

It is probably possible to recast these two lemmas as a single "if and only if" statement with a single, combined proof; but I write them out separately here for simplicity and to emphasize the need for both directions. It is easily possible to imagine scenarios where one of these directions would be true but not the other. For example, if the capacity of the (p_i, c_j) edges was ∞ instead of 1, then not every valid flow would be a valid committee assignment; or if the capacity of the (s, p_i) edges was 1 instead of m_i, then not every valid committee assignment would yield a valid flow.

Lemma 0.1. *If we start with a valid flow f, then the corresponding committee assignment will respect everyone's preferences (though it may not fill all the committee seats). Moreover, the number of committee seats filled will be equal to the value of the flow.*

Proof. The fact that the number of committee seats filled is equal to the value of the flow is easy to see, since every unit of flow has to pass through exactly one of the edges from some p_i to p_j (since they all have capacity 1), and for each such edge we fill exactly one committee seat. For the same reason, we can never have the same person more than once on the same committee.

To see that everyone's preferences will be respected, we argue as follows.

- Suppose person p_i is assigned to x committees, where $x > m_i$ (*i.e.* they are assigned to more committees than they were willing to be on). Then by our construction that means there must have been x different edges from p_i to various c_j with one unit of flow along each. So there would be x units of flow leaving node p_i, and by preservation of flow there must therefore be x units entering node p_i along the edge from s. But this is a contradiction since the capacity of that edge is m_i.

- It is also clear that no one can ever be assigned to a committee they are not willing to be on, because there are no edges between any person and a committee they don't want to be on.

SDG

Lemma 0.2. *If we start with a committee assignment that respects everyone's preferences (possibly with some seats unfilled), then the corresponding flow will be valid and will have a value equal to the number of filled committee seats.*

Proof. To show that a flow network is valid, we have to show that flow is preserved at each vertex and that the flow never exceeds the capacity of any edge.

- Flow will be preserved at each vertex because we explicitly constructed the flow that way.

- Flow along any (p_i, c_j) edge will never exceed the capacity (namely, 1), since we explicitly put a flow of 1 (or 0) along these edges.

- Flow along any (s, p_i) edge will never exceed the capacity of m_i, since the flow along this edge will be equal to the number of (p_i, c_j) edges with 1 unit of flow, which is by construction equal to the number of committees person p_i is assigned to. Since the committee assignment respects p_i's preferences we know they aren't assigned to more than m_i committees.

- Flow along any (c_j, t) edge will never exceed the capacity of z_i, since by similar reasoning this flow will be equal to the number of people assigned to committee c_j, which can't be more than z_i for a valid committee assignment.

∎

We can now make short work of the main result:

Theorem 0.3. *The committee assignment problem has a solution if and only if the corresponding flow network has a max flow equal to* $z_1 + \cdots + z_n$.

Proof. If there exists a solution to the committee assignment problem, then by Lemma 0.2 there exists a corresponding valid flow which will have value $z_1 + \cdots + z_n$. This is a max flow since it is equal to the total capcity coming into vertex t.

Conversely, if there is a max flow equal to $z_1 + \cdots + z_n$, then by Lemma 0.1 there is a corresponding valid committee assignment which fills all the committee seats. ∎