The Algorithm

Define $T_p = number\ of\ timeslots\ recruiter\ p\ is\ available\ for$

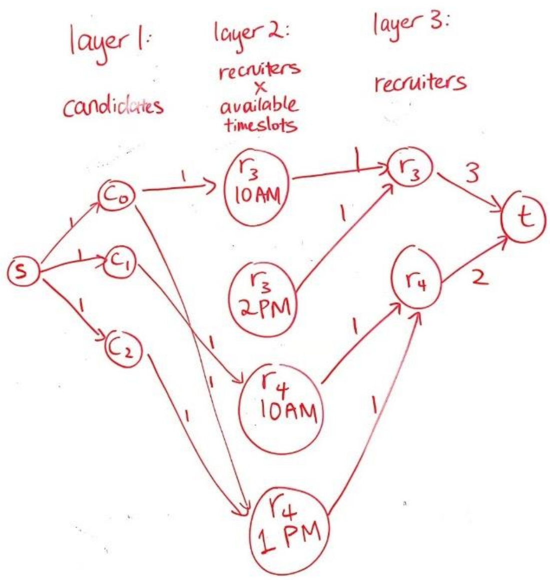
We reduce this to a max flow problem, where one unit of flow means one interview. We construct a graph. There is a source s, sink t, and three layers between.

1. The first layer has n nodes for the n candidates
2. the second layer has $k * T_p$ nodes for the k recruiters and their available timeslots
3. the third layer has k nodes for the k recruiters

1. There is a directed edge (s, c_i) of capacity 1 from s to every node c_i in the first layer
2. There is a directed edge $(c_i, r_p t_j)$ of capacity 1 from the first to second layer if recruiter p is qualified to interview candidate i and candidate i is available at timeslot j
3. There is a directed edge $(r_p t_j, r_p)$ of capacity 1 from the second to third layer for all recruiters p.
4. There is a directed edge (r_p, t) of capacity c_p (recruiter p's interviewing capacity) for all recruiters p.

Run Ford-Fulkerson on this graph. If the max flow is less than the number of candidates n, then there is no interview schedule where every candidate gets interviewed; if the max flow is equal to n, then there is.

Example: there are three candidates: c_0, c_1, and c_2. There are two recruiters: r_3 and r_4. c_0 could be interviewed by r_3 and r_4 and is available at 10 AM and 1 PM. c_1 could be interviewed by r_4 and is available at 10 AM and 2 PM. c_2 could be interviewed by r_4 and is available at 1 PM. r_3 has capacity 3 and is available at 10 AM and 2 PM. r_4 has capacity 2 and is available at 10 AM and 1 PM. The constructed graph for this problem would look as such:



Proof of Correctness

This problem is under several constraints:

1. Each candidate gets at most 1 interview
   ⇒This is satisfied by the edges from layer s to layer 1, which all have capacity 1.
2. Each candidate can only be interviewed at a timeslot he is available for
   ⇒This is satisfied by the edges from layer 1 to layer 2, which only exist if the candidate is available at the time slot the edge is connecting him to
3. Each candidate can only be interviewed by qualified recruiters
   ⇒This is also satisfied by the edges from layer 1 to layer 2, which only exist if the recruiter is qualified to interview the candidate the edge is connecting her to
4. Each recruiter r_p conducts at most c_p interviews
   ⇒This is satisfied by the edges from layer 3 to t, which have capacities c_p
5. Each recruiter can only conduct interviews at timeslots she is available for
   ⇒This is satisfied by the layer 2 itself, where each node $r_p t_j$ only exists if recruiter p is available at timeslot j.
6. Each timeslot can have multiple interviews, but they must be of different recruiters *and* different candidates, i.e. a timeslot can't have one recruiter interviewing more than one candidate and a timeslot can't have a candidate interviewed by more than one recruiter.
   ⇒This is satisfied by the edges between layers 1 and 3 all having capacities 1, meaning if a flow of value 1 goes into a node in layer 2, then only 1 can flow out (thus one candidate only has one recruiter), and if a flow of value 1 goes out from layer 2, that means a flow of 1 went in (thus one recruiter only interviews one candidate)

Runtime Analysis

The runtime of Ford-Fulkerson is $O(mC)$, where m is the number of edges in the graph and C is the sum of the capacities of all edges leaving the source. m here is $n * (n * kT) * (kT * k) * k$. C here is n, This algorithm is thus $O(k^3T^2n^3)$, which is polynomial is all the variables involved.