# The Algorithm

Define .

We reduce this to a max flow problem. We construct a graph. There is a source , which goes to nodes (first layer), each node of which then goes to nodes (second layer), then this second layer goes to nodes (third layer), which finally go to a sink . We label each node in the first layer as for candidate . We label each node in the second layer as , meaning candidate is available at timeslot . We label each node in the third layer as for recruiter . There is an edge from to every node in the first layer, and all of their edge capacities are . There is an edge from node in the first layer to in the second layer if candidate is available during timeslot ; all of their edge capacities are also . There is an edge from a node in the second layer to a node in the third layer if recruiter is qualified to interview candidate and recruiter is available at timeslot ; the capacities for all these edges are . There is an edge from every node in the third layer to , and its capacity is , the interviewing capacity of recruiter . Run Ford-Fulkerson on this graph. If the max flow is less than the number of candidates , then there is no interview schedule where every candidate gets interviewed; if the max flow is equal to , then there is.

Example: there are three candidates: and . There are two recruiters: and . could be interviewed by and and is available at 10 AM, 1 PM, and 5PM. could be interviewed by and and is available at 2 PM and 5 PM.

# Proof of Correctness

Call the maximum flow of this graph . Consider the set of paths on which the flow value is 1 and of the form where is the source, is a candidate and is from the first layer, is a node from the second layer, is a recruiter and is from the third layer, and is the sink. Let be but take away the and from every one of its elements. Here are three simple facts about the set .

**Claim 1**: contains paths.

**Proof**: Consider the cut in the graph with . The value of the flow is the total flow leaving minus the total flow entering . The first of these is simply the cardinality of . The second of these terms is since there are no edges entering . Thus, contains paths.

**Claim 2**: Each node in the 1st layer is the tail of at most one edge in , each node in the 2nd layer is the head and tail of at most one edge in , and each node in the 3rd layer is the head of at most edges (interviewing capacity of recruiter ) in .

**Proof**: For the first part, suppose that node were the tail of at least two edges in . Since our flow is integer-valued, this means that at least two units of flow leave from . By conservation of flow, at least two units of flow would have to come into – but this is not possible, since only a single edge of capacity enters . Thus is the tail of at most one edge in .

For the second part, each node in the 2nd layer is the head of at most one edge in because it only has edges entering from node , which is the tail of at most one edge in . And of course, it is the tail of at most one edge in because it is the head of at most one edge.

For the third part, each node in the 3rd layer is the head of at most edges in because if it has more than edges entering, then a flow of more than flows out from it and into the sink, which is impossible because the capacity of the edge is .

Combining these facts, we see that if we view (and thus, ) as a set of paths in the graph we initially constructed, we get an interviewee—recruiter matching of size .