# Due March 21, 11:59pm

**1. (10 + 10 pts.) Max-Flow Variants**

Show how to reduce the following variants of Max-Flow to the regular Max-Flow problem, i.e. do the following steps for each variant: Given a graph $G$ and the additional variant constraints, show how to construct a graph $G'$ such that

(1) If $F$ is a flow in $G$ satisfying the additional constraints, there is a flow $F'$ in $G'$ of the same size,

(2) If $F'$ is a flow in $G'$, then there is a flow $F$ in $G$ satisfying the additional constraints with the same size.

Prove that properties (1) and (2) hold for your graph $G'$.

(a) **Max-Flow with Vertex Capacities:** In addition to edge capacities, every vertex $v \in G$ has a capacity $c_v$, and the flow must satisfy $\forall v : \sum_{u:(u,v)\in E} f_{uv} \leq c_v$.

(b) **Max-Flow with Multiple Sources:** There are multiple source nodes $s_1, \ldots, s_k$, and the goal is to maximize the total flow coming out of all of these sources.

(c) **Feasibility with Capacity Lower Bounds (extra credit) :** In addition to edge capacities, every edge $(u,v)$ has a demand $d_{uv}$, and the flow along that edge must be at least $d_{uv}$. Instead of proving (1) and (2), design a graph $G'$ and a number $D$ such that if the maximum flow in $G'$ is at least $D$, then there exists a flow in $G$ satisfying $\forall (u,v) : d_{uv} \leq f_{uv} \leq c_{uv}$.

**Solution:**

(a) It suffices to split every vertex into an 'incoming' and 'outgoing' vertex, and connect the two via an edge whose capacity is the capacity of the original vertex. This "internal" edge enforces the vertex capacity constraint, because all flow that goes into and out of the original vertex will be funneled through this edge. Details follow.

Split every vertex $v$ into two vertices, $v_{in}$ and $v_{out}$, and direct all incoming edges for $v$ into $v_{in}$ and all outgoing edges out of $v_{out}$ as follows: for each edge $(u,v)$ with capacity $c_{uv}$ in the original graph, create an edge $(u_{out}, v_{in})$ with capacity $c_{uv}$. Finally, if $v$ has capacity $c_v$, then create an edge $(v_{in}, v_{out})$ with capacity $c_v$. If $F'$ is a flow in this graph, then setting $F(u,v) = F'(u_{out}, v_{in})$ gives a flow in the original graph. Moreover, since the only outgoing edge from $v_{in}$ is $(v_{in}, v_{out})$, and incoming flow must be equal to outgoing flow, there can be at most $c_v$ flow passing through $v$. Likewise, if $F$ is a flow in the original graph, setting $F'(u_{out}, v_{in}) = F(u,v)$, and $F'(v_{in}, v_{out}) = \sum_u F(u,v)$ gives a flow in $G'$. One can easily see that these flows have the same size.

(b) A source can put out as much flow as can be carried by later edges, so there is no harm to add a meta-source that connects to all of these sources via edges having infinite capacity. Details follow.

Create one "supersource" $S$ with edges $(S, s_i)$ for each $s_i$, and set the capacity of these edges to be infinite. Then if $F$ is a flow in $G$, set $F'(S, s_i) = \sum_u F(s_i, u)$. Conversely, if $F'$ is a flow in $G'$, just set $F(u,v) = F'(u,v)$ for $u \neq S$, and just forget about the edges from $S$. One can easily see that these flows have the same size.

(c) The intuitive solution is to simply subtract off the demands from the capacities, and run max flow. However, this is not enough because a valid flow on $G'$ does not correspond to a valid flow on $G$, because when you add back the demands you may violate some flow conservation constraints (e.g., try to take the zero flow from $G'$ to $G$). However, this simple idea is in the right direction, because there is a way to modify $G'$ so that its valid flows do correspond to valid flows on $G$. Roughly, the idea is to add, for each vertex $v$, one incoming edge and one outgoing edge that totals the demands of the incoming and outgoing edges in the original graph, then subtract the demands from the original capacities of the edges. The new incoming and outgoing edge would carry the flow that corresponds to the demand; because this flow must be conserved, when moving back to $G$, we do get a valid flow after adding back the demands. Details follow.

Add two vertices to $G$, call them $s'$ and $t'$, and add edges $(s',v)$ with capacity $\sum_u d(u,v)$ and $(v,t')$ with capacity $\sum_u d(v,u)$. Add an edge $(t,s)$ with capacity $\infty$, and change the capacities of all the edges in $G$ to $c(u,v)-d(u,v)$. Let $D=\sum_{(u,v)\in E} d(u,v)$. We consider $s'$ and $t'$ to be the new source and sink. Note that the cuts that consist of only $\{s'\}$ or only $\{t'\}$ have value $D$, so $D$ is an upper bound on the size of the max-flow.

If $G$ has a feasible flow $F$, we construct a flow $F'$ on $G'$ by fully saturating the edges leaving $s'$ and coming into $t'$, setting $F'(u,v)=F(u,v)-d(u,v)$, and $F'(t,s)=\text{size}(F)$. Since $F(u,v)\le c(u,v)$, all the capacity constraints are satisfied, and adding up the incoming flow at a single vertex, we get

$$\sum_{u\in G'} F'(u,v) = F'(s',v) + \sum_{u\in G}(F(u,v)-d(u,v)) = \sum_{u\in G} F(u,v)$$

which is the incoming flow to $v$ in $F$. Likewise, the outgoing flow in $F'$ is also equal to the outgoing flow in $F$, and since $F$ is a flow these must be equal, so indeed $F'$ is a valid flow. Since all the $(s',v)$ edges are fully saturated, this flow has size exactly $D$.

To show the other direction, let $G'$ have a flow of size exactly $D$. Note this implies that all the $(s',v)$ and $(v,t')$ edges must be fully saturated, since the corresponding cuts have value $D$. Set $F(u,v)=F'(u,v)+d(u,v)$. Since $F'(u,v)\le c(u,v)-d(u,v)$, these values satisfy the capacity and demand constraints. Now adding up the incoming flow at a single (non-source or -sink) vertex,

$$\sum_{u\in G} F(u,v) = \sum_{u\in G', u\neq s'} (F'(u,v)+d(u,v)) = \sum_{u\in G', u\neq s'} F(u,v) + F(s',v) = \sum_{u\in G'} F'(u,v)$$

where the last equality holds because each $F(s',v)$ must be fully saturated. Thus the flow incoming into $v$ in $F$ is equal to the flow incoming into $v$ in $F'$. A similar argument shows the outgoing flow from $v$ in $F$ is equal to the outgoing flow from $v$ in $F'$. Since $F'$ is a flow, these must be equal, and so $F$ is a flow.

## 2. (15 pts.) Zero-Sum Primary

Bernie Sanders and Hillary Clinton are competing to win the votes of the ”Fans of Florida” constituency in the Democratic primary. This set of voters is guaranteed to vote for either Clinton or Sanders, and currently 10 million plan to vote for Sanders.

The table below indicates the number of voters Clinton would gain (and Sanders would lose), in millions, if the candidates choose the indicated strategy. For example, if Sanders takes the low road (makes personal attacks on Clinton, etc.) and Clinton takes the high road (focuses on the issues), Clinton will lose 2 million voters.

Find the optimal mixed strategy for the candidates and the expected outcome of that strategy. Show the linear equations you created. Feel free to use an online LP solver to solve the equations.

| | | Sanders: | | |
|---|---|---|---|---|
| | | high road | low road | drop out |
| Clinton: | high road | 4 | -2 | 10 |
| | low road | 1 | 2 | 10 |

**Solution:** Clinton wants to pick a probability $(h_c, l_c)$ of (high road, low road) such that if Sanders knew these probabilities and plays his best counter-move, Clinton still does as well as possible under the circumstances. If Sanders chooses to take the high road, Clinton would get a payoff of $4h_c + l_c$, and if he chooses to take the low road, Clinton would get a payoff of $-2h_c + 2l_c$. He could also drop out, resulting in $10h_c + 10l_c = 10$, which is the worst option for him. Clearly Sanders would make the choice that results in the lower of the previous two numbers, so Clinton's payoff is $z = min\{4h_c + l_c, -2h_c + 2l_c\}$. Of course, $h_c + l_c = 1$ and $h_c, l_c \geq 0$. Clinton wants to choose $h_c$ and $l_c$ to maximize her payoff $z$, and this is achieved by choosing $h_c = 1/7$; $l_c = 6/7$; leading to a payoff $z = 10/7$.

Note that an expression such as $a \leq \min\{b, c\}$ can be written as $a \leq b$ and $a \leq c$. As in lecture or the book, we can cast the above as a linear program, by letting $z$ denote Clinton's payoff, and expressing the above as inequalities:

$$
\begin{aligned}
\max \quad & z & \\
-4h_c - l_c + z &\leq 0 & \text{(Sanders chooses high road)}\\
2h_c - 2l_c + z &\leq 0 & \text{(Sanders chooses low road)}\\
-10h_c - 10l_c + z &\leq 0 & \text{(Sanders chooses drop out)}\\
h_c + l_c &= 1 & \\
h_c, l_c &\geq 0 &
\end{aligned}
$$

This yields $h_c = 1/7$; $l_c = 6/7$; and $z = 10/7$: the expected outcome is a gain of 1.4 million votes for Clinton. As a sanity check, what if Clinton always took the low road (not just 6/7 of the time)? Then Sanders could always take the high road, and Clinton would gain only 1 million votes.

Let's look at the situation from Sanders's point of view: he wants to pick $(h_s, l_s, d)$ to minimize the maximum expected value of any of Clinton's responses. This yields

$$
\begin{aligned}
\min \quad & z & \\
-4h_s + 2l_s - 10d + z &\geq 0 & \text{(Clinton chooses high road)}\\
-h_s - 2l_s - 10d + z &\geq 0 & \text{(Clinton chooses low road)}\\
h_s + l_s + d &= 1 & \\
h_s, l_s, d &\geq 0 &
\end{aligned}
$$

Sanders's optimal strategy is $h_s = 4/7$; $l_s = 3/7$; and $d = 0$. As you may have expected, Sanders should never drop out. Again, $z = 10/7$: an important characteristic of zero-sum games is that no matter which player must reveal their strategy first (the first LP corresponds to Clinton "revealing" her strategy first and thus having to act defensively), the expected outcome is identical.

**3. (20 pts.) Modeling - Linear Regression**

One of the most important problems in the field of *statistics* is the *linear regression problem*. Roughly speaking, this problem involves fitting a straight line to statistical data represented by points – $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

– on a graph. Denoting the line by $y = a + bx$, the objective is to choose the constants $a$ and $b$ to provide the "best" fit according to some criterion. The criterion usually used is the *method of least squares*, but there are other interesting criteria where linear programming can be used to solve for the optimal values of $a$ and $b$. For each of the following criteria, formulate the linear programming model for this problem:

(a) Minimize the sum of the absolute deviations of the data from the line; that is,

$$\text{Minimize } \sum_{i=1}^{n} |y_i - (a + bx_i)|$$

(*Hint:* define new variables $z_i = y_i - (a + bx_i)$, as well as non-negative variables $z_i^+$ and $z_i^-$ such that $z_i = z_i^+ - z_i^-$. How to minimize $|z_i|$ by optimizing $z_i^+$ and $z_i^-$?)

(b) Minimize the maximum absolute deviation of the data from the line; that is,

$$\text{Minimize } \max_{i=1\ldots n} |y_i - (a + bx_i)|.$$

(*Hint:* use the above hint.)

**Solution:**

(i) We first notice that each $z_i$ is independent of each other, so we first solve the simpler problem with a single $|z|$. Since $|z|$ is a non-linear function, we must modify the formulation for linear programming to work. We write $z == z^+ - z^-$, with $z^+ \geq 0$ and $z^- \geq 0$. Now we would like to think of $|z|$ as being equal to $z^+ + z^-$. Of course, this is not true. e.g. if $z = -9$, we can set $z^+ = 0$ and $z^- = 9$, and then $z^+ + z^- = 9 = |z|$. But we could equally well have set $z^+ = 1$ and $z^- = 10$, and then $z^+ + z^- = 11 \neq |z|$. We now note that when our objective function is to minimize $|z|$, then this trick does work. i.e. we set our objective function to *minimize*$\{z^+ + z^-\}$ subject to the constraints $z = z^+ - z^-$ and $z^+ \geq 0$ and $z^- \geq 0$.

Applying this reasoning to each $z_i$, and considering the objective function that considers the addition of all these contributions gives the linear program:

$$\text{Minimize } \sum_{i=1}^{n} z_i^+ + z_i^-$$
$$\text{subject to} \begin{cases} y_i - (a + bx_i) = z_i^+ - z_i^- & \text{for } 1 \leq i \leq n \\ z_i^+ \geq 0 & \text{for } 1 \leq i \leq n \\ z_i^- \geq 0 & \text{for } 1 \leq i \leq n \end{cases}.$$

(ii) The solution involves two ingredients. First, we use the same idea as in the previous part, where instead of minimizing $|z|$ we minimize: $z^+ - z^-$ subject to $z = z^+ - z^-$ and $z^+, z^- \geq 0$. However, now instead of having to minimize the sum of contributions, we need to minimize their maximum. But this is similar to the way we have dealt with taking maxima/minima in LPs for zero-sum games. Namely, if we want to minimize $\max\{z_1, \ldots, z_n\}$ it suffices to minimize $t$ subject to $t \geq z_1, \ldots, t \geq z_n$. Combining the two ideas yields the solution.

Hence

$$\text{Minimize } \max_{i=1\ldots n} |y_i - (a + bx_i)|$$

is equivalent to

$$\text{Minimize } t$$
$$\text{subject to} \begin{cases} y_i - (a + bx_i) = z_i^+ - z_i^- & \text{for } 1 \le i \le n \\ z_i^+ \ge 0 & \text{for } 1 \le i \le n \\ z_i^- \ge 0 & \text{for } 1 \le i \le n \\ z_i^+ \le t & \text{for } 1 \le i \le n \\ z_i^- \le t & \text{for } 1 \le i \le n \end{cases}.$$

*Remark:* The following is an alternative solution to (i):

$$\text{Minimize } \sum_{i=1}^{n} t_i$$
$$\text{subject to} \begin{cases} y_i - (a + bx_i) \le t_i & \text{for } 1 \le i \le n \\ y_i - (a + bx_i) \ge -t_i & \text{for } 1 \le i \le n \end{cases}.$$

The following is an alternative solution to (ii):

$$\text{Minimize } t$$
$$\text{subject to} \begin{cases} y_i - (a + bx_i) \le t & \text{for } 1 \le i \le n \\ y_i - (a + bx_i) \ge -t & \text{for } 1 \le i \le n \end{cases}.$$

**4. (20 pts.) Boundedness and Feasibility of Linear Programs**

Find necessary and sufficient conditions on the reals $a$ and $b$ under which the linear program

$$\max_{x,y} x + y$$
$$ax + by \le 1$$
$$x, y \ge 0$$

(a) Is infeasible

(b) Is unbounded

(c) Has a unique optimal solution

**Solution:**

(a) There is no way to set $a$ and $b$ to make this LP infeasible. In particular, setting $x = y = 0$ satisfies the constraints, regardless of the choices of $a, b$.

(b) If $a$ or $b$ are non-positive, the LP becomes unbounded. This occurs because we wish to maximize $x + y$; intuitively, we wish to maximize $x, y$ subject to the constraints.
If both $a$ and $b$ are positive, then the constraint $ax + by \le 1$ prevents us from increasing $x, y$ to infinity. More precisely, $0 \le x \le 1/a$ and $0 \le y \le 1/b$, forcing $x + y \le 1/a + 1/b$ which is bounded.

However, setting $a$ or $b$ to a non-positive value removes this limit on $x$ or $y$, allowing us to increase one of them to $\infty$. Geometrically, non-values of $a, b$ "open up" the bottom of the polytope of the feasible region (try to draw this in order to see it).

(c) This LP has multiple optimal solutions when it is feasible if and only if $a = b$ and it is bounded ($a > 0, b > 0$).

Symmetry often causes multiple solutions, so the first observation is that $x, y$ equally contribute to the objective function. If $a = b$, then we can freely substitute $x, y$ for each other, meaning that there will not be a single unique optimal solution. This occurs because what matters are not the individual values of $x$ and $y$, but their sum. Also, if the LP is unbounded, then there are no optimal solutions, so in particular there is no unique one. Figuring out the rest of the proof involves showing that these conditions are enough to characterize.

If $a > b$, then $x = 0$ in the optimal solution because otherwise shifting mass from $x$ to $y$ only increases the value of the objective. If $a < b$, then $y = 0$ in the optimal solution. In both of these situations, the value of the unset variable is uniquely specified in the optimal solution. If $a = b$ and the LP is bounded (i.e. $a > 0, b > 0$), then the line segment $ax + by = 1$ for $x, y \geq 0$ is optimal. The feasible region contains a nontrivial line segment, so there are multiple optimal solutions.

Geometrically, the optimization objective function describes a set of contour lines, each of which corresponds to a particular value of the objective function. Optimizing this function corresponds to increasing the "value" of this contour line, until we reach the limits of our constraints. The intersection between this optimally-valued contour line and the constraints describes the optimal solution. When $a = b$, the contour line and the constraint $ax + by \leq 1$ have the same slope; therefore, their intersection is an entire line, resulting in infinitely many optimal solutions.

**5. (20 pts.) Salt**

Salt is an extremely valuable commodity. There are $m$ producers and $n$ consumers, each with their own supply $[a_1 ... a_m]$ and demand $[b_1 ... b_n]$ of salt.

Note: Solve parts $(b), (c)$ independently of each other.

(a) Each producer can supply to any consumer they choose. Find an efficient algorithm to determine whether it is feasible for all demand to be met.

(b) Each producer is willing to deliver to consumers at most $c_i$ distance away. Each producer $i$ has a distance $d_{i,j}$ from consumer $j$. Solve part $(a)$ with this additional constraint.

(c) Each producer and consumer now belongs to one of $p$ different countries. Each country has a maximum limits on the amount of salt that can be imported $(e_k)$ or exported $(f_k)$. Deliveries within the same country don't contribute towards this limit. Solve part $(a)$ with this additional constraint.

**Solution:**

(a) Formulate this as a max flow problem. Set up a bipartite graph, with one node for each of the $m$ producers on the left, and one for each of the $n$ consumers on the right.

For every producer $i$, create an edge between the source and producer of capacity $a_i$, to represent the maximum supply. Similarly, for every consumer $j$ create an edge between the destination and consumer of capacity $b_j$ to represent the maximum demand. Finally, create an edge between every producer and consumer of capacity $\infty$, since each producer can deliver to any consumers.

Running a max-flow algorithm on this graph, and checking whether the flow is greater than $\sum_j b_j$

will determine whether it is feasible for all demand to be met.

Alternate (Trivial) Solution: Simply check whether $\sum_i a_i \geq \sum_j b_j$. This answer was accepted for part $(a)$, but doesn't provide a framework to solve parts $(b)$, $(c)$.

(b) We modify the max-flow solution from part $a$. Now only create the edge between a producer $i$ and consumer $j$ if the distance is within the limit; $d_{i,j} \leq c_i$.

Creating the intermediate edges in this manner ensures that producers can deliver only to consumers within their distance limit.

(c) We modify the max-flow solution from part $a$. For each country, create two dummy nodes $X_k, X_k'$ and an edge between them of capacity $e_k$; this represents the limit on exports. Create dummy nodes $Y_k, Y_k'$ and an edge between them of capacity $f_k$, to represent the limit on imports.

Now create an edge of capacity $\infty$ between each producer $i$ and its respective export node $X_k$, as well as each consumer and its respective import node $Y_k'$, to ensure that the import/exports are accounted for in the graph. Also, create an edge of capacity $\infty$ between any consumer and producer that are located in the same country, to represent deliveries within the same country.

Funneling all exports/imports through country nodes $X_k, Y_k'$ allows us to account for the total imports/exports for a country. The intermediate edges between the dummy nodes, $X_k \to X_k', Y_k \to Y_k'$ enforce the export/import limits $e_k, f_k$.