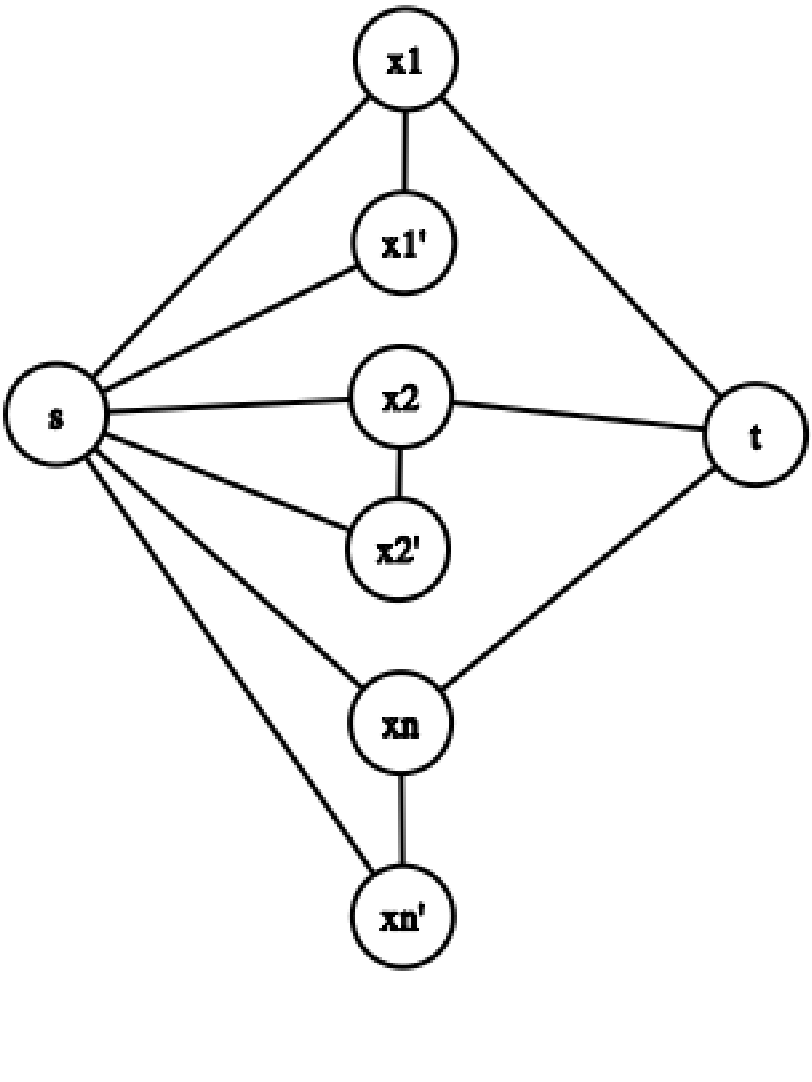
COS 521 - Homework 1

Nameless author :)

September 23, 2022

Problem 1

Proof. Consider the graph $G = (V, E)$ depicted below. Let $n = \frac{|V|-2}{2}$ be the number of pairs of intermediate nodes.



Observe that there are many possible $s - t$ cuts. For example, the cut $(X, \bar{X})$ given by $X = \{s, x_1, x_1'\}$ has a value of $2n - 1$. There is, however, only one $s - t$ cut that has a value of n, which is precisely the cut given by $X = V \setminus \{t\}$ and $\bar{X} = \{t\}$. The below Lemma proves this.

Lemma 1. *The minimum $s - t$ cut is uniquely given by $X = V \setminus \{t\}$ and $\bar{X} = \{t\}$.*

Proof. We will show that any other $s - t$ cut will always have more edges crossed. If we were to add any intermediate nodes to $\bar{X}$, the number of edges crossed will necessarily increase. To see this, note that adding x_i but not x_i' for some i will add two edges ((s, x_i) and (x_i, x_i')) and subtract only one ((x_i, t)), making the cut worse. Similarly, adding x_i' but not x_i for some i will add two edges ((x_i, x_i') and (s, x_i)) to the cut and not subtract any, making the cut worse. Adding a pair of x_i and x_i' will add two edges ((s, x_i) and (s, x_i')) and subtract one (x_i, t), making the cut worse. So, we can see that the cut $X = V \setminus \{t\}$ and $\bar{X} = \{t\}$ must be the minimum $s - t$ cut since inclusion of any intermediate nodes only makes things worse. ■

We now know by Lemma 1 that in order for Karger's algorithm to output a minimum $s - t$ cut, it must output precisely the cut $(X, \bar{X}) = (V \setminus \{t\}, \{t\})$. Note that in order for this to happen, the algorithm must contract at least $2n$ of the edges that aren't of the form (x_i, t), since that is the only way to form a supernode X out of the $2n + 1$ vertices that aren't t, which is the only way to achieve the min $s - t$ cut. (We say at least $2n$ because it is possible that contracting creates parallel edges, which may require more contractions before converging on an $s - t$ cut). We can see that each edge has a $\frac{1}{2n-i+1}$ probability of being contracted during the ith iteration of Karger's algorithm. So, in order for the algorithm to still be able to potentially output the min $s - t$ cut, the probability that the n edges we want to maintain are not selected in the first $2n$ selections is

$$\prod_{j=1}^{2n} \frac{1}{2n - i + 1}$$

We can note that for $j < n$, each element of the product is clearly less than $\frac{1}{2}$. So, we can perform a terrible, but still valid, bound on the probability to see that the probability of outputting the min $s - t$ cut is at most

$$\left(\frac{1}{2}\right)^n = 2^{-\Omega(n)}$$

■

Problem 2

We begin with two lemmas, in the style of the course notes :)

Lemma 2. *Let G' be an undirected graph, and let $(X, \bar{X})$ be any B-approximate min cut of G'. The probability that $(X, \bar{X})$ survives a random edge contraction is at least $\left(1 - \frac{2B}{n}\right)$.*

Proof. Let $(X, \bar{X})$ be a B-approximate min cut in $G' = (V, E)$. This cut is killed by contracting an edge (u, v) if and only if $|X \cap \{u, v\}| = 1$. In other words, the cut is killed if and only if we contract an edge that goes across the cut. Using Lemma 1 from Lecture 1, there are at least $\frac{cn}{2}$ possible random edges to contract. Also, if x is the value of the B-approximate cut $(X, \bar{X})$, we know that there are precisely $x \leq Bc$ edges going across the cut. So, the probability of the cut being killed by randomly selecting an edge is at most

$$\frac{x}{|E|} \leq \frac{Bc}{\frac{cn}{2}} = \frac{2B}{n}$$

Therefore, the probability of the cut surviving is at least $\left(1 - \frac{2B}{n}\right)$. ■

Lemma 3. *After $n - 2B$ iterations of Karger's algorithm, there will be $2B$ supernodes remaining. Furthermore, the probability of any original B-approximate minimum cut, say $(X, \bar{X})$, surviving up to this point is $\binom{n}{2B}^{-1}$.*

Proof. Clearly, since each iteration of the algorithm contracts two nodes that are connected by an edge to a single node, the number of nodes decreases by one each iteration; then, after $n - 2B$ iterations we are left with $2B$ supernodes.
Let $(X, \bar{X})$ be a B-approximate min cut in the original graph, which has n nodes. By Lemma 2, we know that the probability of this cut surviving the first iteration is $\left(1 - \frac{2B}{n}\right)$. Similarly, the probability that it survives the second iteration, conditioned on the cut surviving the first iteration, is $\left(1 - \frac{2B}{n-1}\right)$. We can continue this to see that the probability of the cut $(X, \bar{X})$ surviving the first $n - 2B$ iterations is

$$\begin{aligned}
&\prod_{i=1}^{n-2B} 1 - \frac{2B}{n-i+1} = \prod_{i=1}^{n-2B} \frac{n-i+(1-2B)}{n-1+1} \\
&= \frac{n-2B}{n} \cdot \frac{n-1-2B}{n-1} \cdot \ldots \cdot \frac{n-(n-2B-1)+(1-2B)}{n-(n-2B-1)+1} \cdot \frac{n-(n-2B)+(1-2B)}{n-(n-2B)+1} \\
&= \frac{n-2B}{n} \cdot \frac{n-1-2B}{n-1} \cdot \ldots \cdot \frac{2}{2B+2} \cdot \frac{1}{2B+1} \\
&= \frac{(n-2B)!}{n \cdot \ldots \cdot (2B+1)} = \frac{(n-2B)!(2B)!}{n!} = \binom{n}{2B}^{-1}
\end{aligned}$$

■

Proof. We can now prove the result. If we run Karger's algorithm until we have $2B$ supernodes, there will be $2^{2B} - 2$ possible remaining cuts that can be made (each supernode can be placed in one of two nonempty subsets). For any original B-approximate min cut $(X, \bar{X})$, by Lemma 3 there is a $\binom{n}{2B}^{-1}$ probability it has survived up to this point. If from this point we select a random remaining cut, the probability that it materializes into our original $(X, \bar{X})$ is exactly $\frac{1}{2^{2B}-2}$. So, the probability that Karger's algorithm outputs $(X, \bar{X})$ is lower bounded by $\left(\binom{n}{2B}(2^{2B} - 2)\right)^{-1}$. Since this holds for each initial $(X, \bar{X})$ cut and the survival of each $(X, \bar{X})$ is a disjoint event, we find that the number of B-approximate min cuts is upper bounded by $\binom{n}{2B}(2^{2B} - 2) \leq (2n)^{2B}$. ■

Problem 3

Part A

Proof. The LP listed below is indeed a valid formulation for the value of the max flow.

$$\begin{array}{ll} \text{maximize} & \sum_u f((u,t)) \\ \text{subject to} & \forall e=(u,v)\in E, f((u,v))\leq c_e \\ & \forall v\notin\{s,t\}, \sum_{(u,v)\in E} f((u,v)) \geq \sum_{(v,w)\in E} f((v,w)) \\ & \forall e\in E, f(e)\geq 0 \end{array}$$

The objective we are maximizing is exactly the value of the flow: conservation of flow yields that the value of flow going out of the source must equal the value of flow going into the target node, which is objective the value we are maximizing. The first constraint ensures that the value of flow assigned across every edge is $\leq$ the capacity of that edge. The third constraint ensures that every flow value assigned is nonnegative. These constraints and objectives together precisely formulate the max flow problem. The second constraint asserts conservation of flow: that the sum of flow values along edges entering a node equals the sum of flow values along edges exiting the node. A more obvious translation of this would require equality in the constraint, but we show below that we do not lose generality by making this a $\geq$.

Lemma 4. *The constraint $\forall v \notin \{s,t\}, \sum_{(u,v)\in E} f((u,v)) \geq \sum_{(v,w)\in E} f((v,w))$ is sufficient to require conservation of flow.*

Proof. Let LP_I be the above LP written with the inequality, and let LP_E be the LP written with equality in the second constraint. Suppose by way of contradiction that LP_I is not a valid formulation of the max flow problem. Then there must be some solution that is feasible to LP_I that achieves a strictly higher value. So, there must be some $(u,t)\in E$ such that $f((u,t)) > f'((u,t))$, where f' is the optimal solution to LP_E in the second constraint. However, the second constraint in LP_I confirms that the outflow of u is less than or equal to the inflow of u. So, we should be able to increase the outflow of u in the optimal solution of LP_E. This would be a feasible solution of LP_E that would have a higher value, contradicting the assumption of optimality of the solution to LP_E. So, we see that the solution to LP_I cannot do better than the solution to LP_E. Since every feasible solution to LP_E is also a solution to LP_I, we can conclude that LP_I is also a valid formulation of the max flow problem. ■

■

Part B

Proof. We can compute the dual in the usual way: by writing an upper bound for the objective of the primal LP, and tightening the bound as much as we can. We find that the dual is the following problem, where each vertex $v\in V\setminus\{s,t\}$ has a variable y_v and each edge $(u,v)=e\in E$ has a variable $z_{u,v} = z_e$:

$$\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e \cdot z_e \\
\text{subject to} \quad & z_{s,v} - y_v \geq 0 \quad \forall (s,v) \in E \\
& z_{u,t} + y_u \geq 1 \quad \forall (u,t) \in E \\
& y_u - y_v + z_{u,v} \geq 0 \quad \forall (u,v) \in E \text{ with } u,v \notin \{s,t\} \\
& 0 \leq z_e \leq 1 \quad \forall e \in E \\
& 0 \leq y_v \leq 1 \quad \forall v \notin \{s,t\}
\end{aligned}$$

Since y_s and y_t are not variables of the dual, we can freely set them to 0 and 1, respectively. The first constraint simplifies to $z_{s,v} \geq y_v - y_s$ and the second constraint simplifies to $z_{u,t} \geq y_t - y_u$. Also, the third constraint simplifies to $z_{u,v} \geq y_v - y_u$. All of these constraints are immediately satisfied if we relate the variables $z_{u,v} = max\{0, y_v - y_u\}$ to simplify the dual LP. With this in mind, we arrive at a simplified dual over the variables $\{y_u \quad | \quad u \in V \setminus \{s,t\}\}$

$$\begin{aligned}
\text{minimize} \quad & \sum_{(u,v) \in E} c_{u,v} \cdot max\{0, y_v - y_u\} \\
\text{subject to} \quad & 0 \leq y_u \leq 1 \quad \forall u \in V \setminus \{s,t\}
\end{aligned}$$

Observe that this is precisely the formulation of the min fractional $s - t$ cut problem, since the constraint (along with the definitions $y_s = 0, y_t = 1$) define a fractional cut, and the objective function is exactly the value of a fractional cut. By Strong LP Duality, if the max flow achieves a finite value of C (which means it is both bounded and feasible), then the dual min fractional $s - t$ cut is also bounded and feasible, and achieves an equal optimum of C. Therefore, the min fractional $s - t$ cut problem cannot achieve any value ¡ C, since its dual has an optimum of C. ■

Part C

Proof. We want a rounding scheme that produces a cut $(X, \bar{X})$ deterministically from a minimum fractional $s - t$ cut with value C, and such that the expected value of the cut is

$$C = \mathbb{E}\left[\sum_{(u,v) \in E} c_{u,v} \cdot \mathbb{1}[u \in X \cap v \in \bar{X}]\right] = \sum_{(u,v) \in E} c_{u,v} \mathbb{P}[u \in X \cap v \in \bar{X}]$$

In order for this to be true, we would want that the probability of an edge $(u,v) = e \in E$ lying across the cut is $\mathbb{P}[u \in X \cap v \in \bar{X}] = max\{0, y_v - y_u\}$, since this would match the cost of the min fractional $s - t$ cut.

We can round in the following way: fix some threshold $T \sim U[0,1]$ (sampled from uniform distribution), and assign vertices $v \notin \{s,t\}$ to X if $y_v < T$ and to $\bar{X}$ if $y_v \geq T$ (and assign s to X, t to $\bar{X}$). Then, for any edge (u,v), we have that

$$\mathbb{P}[u \in X \cap v \in \bar{X}] = \mathbb{P}[y_u < T, y_v \geq T] = max\{0, y_v - y_u\}$$

using the joint CDF of the uniform distribution. So, we see that this random rounding scheme of sampling a threshold uniformly from $[0,1]$ performs just as well in expectation as the fractional solution. To derandomize it, compute a long list of pre-computed random threshold values, apply the rounding procedure separately with each one, and select the solution with the smallest cost. This will once again, in expectation, have a cost that is as good as the min fractional $s - t$ cut, but will be deterministic (solving the relaxed LP and rounding it for the same graph will produce the same cut). We can conclude the max flow - min cut theorem: because the rounding algorithm from a min fractional $s - t$ cut to a min cut is a 1-approximation, we can conclude that the min cut and max flow have the same value. ■

Problem 4

Part A

Proof. We generalize the $MAXSAT$ problem as follows: given n literals and m clauses of length $\leq n$, each with weight w_l, we wish to set the literals to maximize the weighted sums of the satisfied clauses. Indexing clauses by l and literals by j (so $x_j \in \{0,1\}$ is the jth literal), we define the following symbol:

$$y_l^{(j)} = \begin{cases} t_j & \text{literal } j \text{ in clause } l \text{ is } x_j \\ 1 - t_j & \text{literal } j \text{ in clause } l \text{ is } \neg x_j \\ 0 & \text{literal } j \text{ does not appear in clause } l \end{cases}$$

So, if the first clause was $x_1 \vee \neg x_2$, then $y_1^{(1)} = t_1$, $y_1^{(2)} = 1 - t_2$, and $y_1^{(j)} = 0$ for all $j > 2$. Then, each $y_l^{(j)}$ takes value of 1 if literal x_j satisfies clause l, and 0 otherwise. Let us also define a variable z_l for each clause l that takes a value of 1 if the assignment satisfies the clause, and 0 if not. So, we can write an integer program over the variables of $\{t_1, ..., t_n\} \cup \{z_1, ..., z_m\}$ as

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{l=1}^{m} w_l \cdot z_l \\ \text{subject to} & t_j \in \{0,1\} \quad \forall j \in \{1, ..., n\} \\ & 0 \leq z_l \leq 1 \quad \forall l \in \{1, ..., m\} \\ & z_l \leq \displaystyle\sum_{j=1}^{n} y_l^{(j)} \quad \forall l \in \{1, ..., m\} \end{array}$$

Note that the last two constraints ensure that each z_l indicates that clause l is satisfied exactly as we want: if any of the literals satisfy the clause, the sum is nonzero and z_l is allowed to take value of 1 (which it will want to to maximize the objective). If, however, none of the literals satisfy clause l, the sum in the last constraint is 0 and z_l is forced to take value of 0. This is exactly what we want, and so we can relax the integrality condition to get an LP over variables of $\{t_1, ..., t_n\} \cup \{z_1, ..., z_m\}$

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{l=1}^{m} w_l \cdot z_l \\ \text{subject to} & 0 \leq t_j \leq 1 \quad \forall j \in \{1, ..., n\} \\ & 0 \leq z_l \leq 1 \quad \forall l \in \{1, ..., m\} \\ & z_l \leq \displaystyle\sum_{j=1}^{n} y_l^{(j)} \quad \forall l \in \{1, ..., m\} \end{array}$$

■

Part B

We will use the same rounding scheme from class: namely, given fractional solutions for each t_j, we set each literal x_j to true independently with probability t_j.

Lemma 5. *The probability that any clause l is satisfied equals $1 - \prod_{j=1}^{n}(1 - y_l^{(j)})$. This probability is at least $1 - \left(1 - \frac{z_l}{n}\right)^n$.*

Proof. The probability that all of the literals don't satisfy the clause is $\prod_{j=1}^{n}(1-y_l^{(j)})$. Therefore, the probability that clause l is satisfied is $1-\prod_{j=1}^{n}(1-y_l^{(j)})$. We can apply the AM/GM inequality on the sequence of $\{1-y_l^{(1)},...,1-y_l^{(n)}\}$ to see that

$$\prod_{j=1}^{n}(1-y_l^{(j)}) \leq \left(\frac{1}{n}\sum_{j=1}^{n}(1-y_l^{(j)})\right)^n = \left(1-\frac{\sum_j y_l^{(j)}}{n}\right)^n \leq \left(1-\frac{z_l}{n}\right)^n$$

where the last step comes from the last constraint of our LP. So, we find that the probability that clause l is satisfied is

$$1-\prod_{j=1}^{n}(1-y_l^{(j)}) \geq 1-\left(1-\frac{z_l}{n}\right)^n$$

■

Lemma 6. *For all $z \in [0,1]$ and all $n \in \mathbb{N}$, the following inequality holds:*

$$1-\left(1-\frac{z}{n}\right)^n \geq \left(1-\frac{1}{e}\right)z$$

Proof. This result holds trivially for $n = 1$; so, suppose that $n \geq 2$. We begin by noting that the LHS and RHS are equal when $z = 0$, and that the inequality holds when $z = 1$ by the fact that $\frac{1}{e} = lim_{k\to\infty}\left(1-\frac{1}{k}\right)^k \implies \frac{1}{e} \geq \left(1-\frac{1}{n}\right)^n$. It remains to show that over the interval $z \in (0,1)$, the LHS is concave down; this would imply that if the inequality is not violated at the endpoints, it cannot be violated over $z \in [0,1]$ since the RHS is linear. We can find that the second derivative of the LHS is

$$\frac{d^2}{dz^2}\left[1-\left(1-\frac{z}{n}\right)^n\right] = \frac{d}{dz}\left[\frac{1}{n}\cdot n\cdot\left(1-\frac{z}{n}\right)^{n-1}\right] = -\frac{n-1}{n}\left(1-\frac{z}{n}\right)^{n-2} < 0$$

So, the LHS is concave down over $z \in (0,1)$ and lies above the RHS line at its endpoints: therefore, it lies above this line over the entire interval. ■

Proof. Using Lemmas 5 and 6, we arrive at the fact that the probability that any arbitrary clause l is satisfied is $\mathbb{P}\{l \text{ satisfied}\} \geq \left(1-\frac{1}{e}\right)z_l$. So, the expected cost of a rounded LP solution is

$$\sum_{l=1}^{m} w_l\cdot\mathbb{P}\{l \text{ satisfied}\} \geq \left(1-\frac{1}{e}\right)\sum_{l=1}^{m} w_l\cdot z_l = \left(1-\frac{1}{e}\right)OPT_f$$

So, the rounded solution to the LP is a $\left(1-\frac{1}{e}\right)$-approximation to the optimal solution for *MAXSAT*.
■

Part C

Proof. We will analyze the case where we randomly choose one of the two algorithms to run, each with probability $\frac{1}{2}$. If we can show that this setup is a $\frac{3}{4}$-approximation of *MAXSAT*, then it clearly follows that selecting the best result of these two algorithms is at least as good. Now, if we were to randomly select an algorithm, we can find the expected value of the objective function by plugging in the probabilities of satisfying each clause. The expected value, where k_l is the number of literals in clause l, is

$$\frac{1}{2}\sum_{l=1}^{m} w_l\cdot\left(1-\prod_{j=1}^{n}(1-y_l^{(j)})\right)+\frac{1}{2}\sum_{l=1}^{m} w_l\cdot\left(1-\prod_{j=1}^{k_l}\left(\frac{1}{2}\right)\right) = \sum_{l=1}^{m} w_l\cdot\left(1-\frac{2^{-k_l}+\prod_{j=1}^{n}(1-y_l^{(j)})}{2}\right)$$

Lemma 7. *For every clause l, we have the inequality*

$$1 - \frac{2^{-k_l} + \prod_{j=1}^{n}(1 - y_l^{(j)})}{2} \geq \frac{3}{4} z_l$$

Proof. We can use the result of Lemma 5 to see: (for notation let $k := k_l$)

$$1 - \frac{2^{-k} + \prod_{j=1}^{n}(1 - y_l^{(j)})}{2} \geq 1 - \frac{2^{-k} + \left(1 - \frac{z_l}{n}\right)^n}{2} \geq 1 - \frac{2^{-k} + \left(1 - \frac{z_l}{k}\right)^k}{2}$$

We want to show that the RHS lies above the line $\frac{3}{4} z_l$. If $k = 1$, the RHS is equal to $1 - \frac{3}{4} + \frac{z_l}{2} = \frac{1}{4} + \frac{z_l}{2} \geq \frac{3z_l}{4}$. So, suppose that $k \geq 2$. We can check that the RHS lies above the line $\frac{3}{4} z_l$ at the endpoints of the interval $z_l \in [0, 1]$ and is concave down over the interior of this interval; this is equivalent to proving that the RHS always lies above the line. We check that at the endpoints, $z_l = 0 \implies 1 - \frac{2^{-k}+1}{2} \geq 0$ and that $z_l = 1 \implies 1 - \frac{2^{-k}}{2} \geq 1 - \frac{2^{-1}}{2} = \frac{3}{4}$; in both cases, the inequality holds. Now, over the interval $z_l \in (0, 1)$, we check for concave down:

$$\frac{d^2}{dz_l{}^2}\left[1 - \frac{2^{-k} + \left(1 - \frac{z_l}{k}\right)^k}{2}\right] = \frac{d}{dz_l}\left[\frac{1}{2k} \cdot k \cdot \left(1 - \frac{z_l}{k}\right)^{k-1}\right] = -\frac{k-1}{2k}\left(1 - \frac{z_l}{k}\right)^{k-2} < 0$$

So, we can conclude that the RHS lies above the line, and from this we can conclude the Lemma. ■

Using Lemma 7, we find that the expected value of choosing one of the two algorithms randomly is lower bounded by $\frac{3}{4}\sum_{l=1}^{m} w_l \cdot z_l$, which is exactly $\frac{3}{4}$ times the value of the optimal solution to the unrounded LP OPT_f. So, we conclude that randomly selecting one of the two algorithms is a $\frac{3}{4}$-approximation of *MAXSAT*, and therefore so is selecting the best algorithm. ■

Part D

Proof. Consider the function that maps outputs $t_j \in [0, 1]$ of the fractional LP to probabilities $f(t_j)$ with which to assign true to literal x_j via

$$f(t_j) = 4^{(t_j - 1)}$$

Firstly, we see that these outputs are indeed between 0 and 1, and so are valid probabilities with which to assign true to x_j. Next, we can compute the probability of each clause l being satisfied. Let $I_l^+ \subset \{1, ..., n\}$ be the set of indices of literals in clause l that are of the form $... \vee x_j \vee ...$. Similarly, let $I_l^- \subset \{1, ..., n\}$ be the set of indices of literals in clause l that are of the form $... \vee \neg x_j \vee ...$. Then, we can write clause l as $\left(\vee_{j \in I_l^+} x_j\right) \vee \left(\vee_{j \in I_l^-} \neg x_j\right)$. We can say that by Lemma 5

$$\mathbb{P}[\text{clause } l \text{ is satisfied}] = 1 - \left(\prod_{j \in I_l^+}(1 - f(t_j)) \prod_{j \in I_l^-} f(t_j)\right) = 1 - \left(\prod_{j \in I_l^+}\left(1 - 4^{t_j - 1}\right) \prod_{j \in I_l^-} 4^{t_j - 1}\right)$$

Since $-4^{2x-1} \leq 0 \implies 1 - 4^{2x-1} \leq 1 \implies 1 - 4^{x-1} \leq 4^{-x}$, we get

$$\begin{aligned}\mathbb{P}[\text{clause } l \text{ is satisfied}] &\geq 1 - \left(\prod_{j \in I_l^+} 4^{-t_j} \prod_{j \in I_l^-} 4^{t_j - 1}\right) \\ &= 1 - \frac{1}{4}^{\left(\sum_{j \in I_l^+} t_j + \sum_{j \in I_l^-}(1 - t_j)\right)}\end{aligned}$$

Our LP guarantees that $z_j \leq \sum_{j \in I_l^+} t_j + \sum_{j \in I_l^-}(1 - t_j)$, and so we see that since $0 \leq z_j \leq 1$

$$\mathbb{P}[\text{clause } l \text{ is satisfied}] \geq 1 - \left(\frac{1}{4}\right)^{z_j} \geq 1 - \frac{1}{4} = \frac{3}{4}$$

So, we can see that the expected objective function value given by this rounding technique is

$$\sum_{l=1}^{m} w_l \cdot \mathbb{P}[\text{clause } l \text{ is satisfied}] \geq \frac{3}{4}\sum_{l=1}^{m} w_l \geq \frac{3}{4}\sum_{l=1}^{m} w_l \cdot z_l = \frac{3}{4}OPT_f$$

So, this rounded solution is a $\frac{3}{4}$-approximation for the optimal solution to *MAXSAT*. ■

Problem 5

Proof. Let m_j denote the jth point in our finite metric space. Let i index houses. We will construct a program over a matrix $X \in \mathbb{R}^{n \times m}$ such that each element $X_{i,j}$ is 1 if there is a firehouse in location j that is the closest firehouse to house i, and 0 elsewhere. Then, the distance from each house i to their closest firehouse is simply

$$d(v_i, u_i) = \sum_{j=1}^{m} X_{i,j} d(v_i, m_j)$$

Also, in order for each house to have exactly one closest firehouse (we don't care how tiebreaks are assigned), we wish for

$$\sum_{j=1}^{m} X_{i,j} = 1 \quad \forall i \in \{1, ..., n\}$$

In order to constrain the number of firehouses, we add an additional row (row $n+1$) that indicates whether there is a firehouse at that position. In other words, we now consider matrices $X \in \mathbb{R}^{(n+1)\times m}$ such that

$$X_{i,j} = \begin{cases} 1 & \text{the firehouse closest to house } i \leq n \text{ is located at location } j \\ 0 & \text{the firehouse closest to house } i \leq n \text{ is not located at location } j \\ 1 & \text{there is a firehouse at location } j \text{ and } i = n+1 \\ 0 & \text{there is not a firehouse at location } j \text{ and } i = n+1 \end{cases}$$

Then, we constrain the number of firehouses by ensuring that $\sum_{j=1}^{m} X_{n+1,j} \leq k$. Lastly, to ensure that the indicators in the last row actually indicate presence of a firehouse, we can write the following constraints:

$$\begin{aligned} X_{n+1,j} \geq X_{i,j} \quad & \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \\ X_{i,j} \in \{0, 1\} \quad & \forall i \in \{1, ..., n+1\}, j \in \{1, ..., m\} \end{aligned}$$

Note that if the indicator in $X_{n+1,j}$ is 0, then the above constraints force all of the entries in column j to also be 0; alternatively, if the indicator is 1 they allow the entries above it to take whichever values they want. This corresponds to the fact that if we do not place a firehouse at position j, there can be no house whose closest firehouse is at position j. Otherwise, anything goes. Putting all of the above together, we get the integer program of

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} X_{i,j} d(v_i, m_j) \\ \text{subject to} \quad & \sum_{j=1}^{m} X_{i,j} = 1 \quad \forall i \in \{1, ..., n\} \\ & X_{n+1,j} \geq X_{i,j} \quad \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \\ & \sum_{j=1}^{m} X_{n+1,j} \leq k \\ & X_{i,j} \in \{0, 1\} \quad \forall i \in \{1, ..., n+1\}, j \in \{1, ..., m\} \end{aligned}$$

The optimal solution to this program exactly solves the firehouse problem, and so has value OPT. We can relax it into an LP by removing the integer constraint to get a fractional solution:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} X_{i,j} d(v_i, m_j) \\
\text{subject to} \quad & \sum_{j=1}^{m} X_{i,j} = 1 \quad \forall i \in \{1, ..., n\} \\
& X_{n+1,j} \geq X_{i,j} \quad \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \\
& \sum_{j=1}^{m} X_{n+1,j} \leq k \\
& 0 \leq X_{i,j} \leq 1 \quad \forall i \in \{1, ..., n+1\}, j \in \{1, ..., m\}
\end{aligned}$$

Observe that this fractional LP (which can be solved in *poly(m)* time) has *OPT* as a feasible solution, and so the fractional optimal solution OPT_f must be at least as good. We wish to devise a way to place $O(klog(n))$ firehouses in expectation such that, in expectation, the resulting rounded allocation has cost $OPT_r \leq OPT_f \leq OPT$. We start by noticing that, if we define $\mathbb{E}[d_i]$ as the expected distance from each house v_i to the nearest firehouse in the fractional LP solution, we get that $\mathbb{E}[d_i] = \sum_{j=1}^{m} X_{i,j} d(v_i, m_j)$. This is because the fractional optimal $X_{i,j}$'s form a probability distribution for each house due to the first LP constraint. We can write the cost of the fractional LP solution as $OPT_f = \sum_{i=1}^{n} \mathbb{E}[d_i]$. To ensure our rounded solution does better than this, and therefore better than OPT, we now want to place our $O(klog(n))$ firehouses in such a way that each house has in expectation a distance of less than $\mathbb{E}[d_i]$ to its nearest firehouse.

To formalize this, let $\delta > 0$ be arbitrary and define a neighborhood around house v_i as the location indices that are within distance of $(1+\delta)\mathbb{E}[d_i]$ of v_i:

$$N_i = \{j \in \{1, ..., m\} \quad | \quad d(v_i, m_j) < (1+\delta)\mathbb{E}[d_i]\}$$

Then, we want our rounding algorithm to almost surely place a firehouse at a location in N_i for each i. Rigorously, let $\epsilon > 0$ be arbitrary. We wish for the probability that our allocation ends up with at least 1 firehouse in each N_i to be at least $1 - \epsilon$.

We can relate δ to our fractional LP solution by noting that

$$\begin{aligned}
\mathbb{E}[d_i] = \sum_{j=1}^{m} X_{i,j} d(v_i, m_j) \geq \sum_{j \notin N_i} X_{i,j} d(v_i, m_j) \geq (1+\delta)\mathbb{E}[d_i] \sum_{j \notin N_i} X_{i,j} \\
\implies \sum_{j \notin N_i} X_{i,j} \leq \frac{1}{1+\delta} \implies \sum_{j \in N_i} X_{i,j} = 1 - \sum_{j \notin N_i} X_{i,j} \geq 1 - \frac{1}{1+\delta} \\
\implies \sum_{j \in N_i} X_{n+1,j} \geq 1 - \frac{1}{1+\delta}
\end{aligned}$$

where the last line comes from the second constraint of our LP.

Let us determine the probabilities of sampling a location for a firehouse. For the first firehouse we sample, we choose from locations $j \in \{1, ..., m\}$ with probability $\frac{X_{n+1,j}}{k}$ (we must normalize by k because of the third constraint in the fractional LP). After each step of this sampling without replacement, we have to increase the probabilities of the remaining firehouse locations in order to keep the distribution normalized; so, in the later steps we choose from locations j with probability $\geq \frac{X_{n+1,j}}{k}$. In any case, since sampling different positions to place firehouses in are disjoint events, we can sum the probabilities to see that for each neighborhood N_i around a house, the probability a firehouse gets placed there in a particular step is

$$\mathbb{P}[\text{a firehouse in } N_i] = \sum_{j \in N_i} \mathbb{P}[\text{firehouse at } j] \geq \sum_{j \in N_i} \frac{X_{n+1,j}}{k} \geq \frac{1 - \frac{1}{1+\delta}}{k}$$

Let $t := t(k, n, \delta)$ be a function of $k, n,$ and δ that determines how many firehouses we sample (we would like to find a form of t to make things nice). Then, we can bound the probability that no firehouse is placed in a neighborhood N_i during our sampling procedure with

$$\mathbb{P}[\text{no firehouse in } N_i] = (1 - \mathbb{P}[\text{a firehouse in } N_i])^t \leq \left(1 - \frac{1 - \frac{1}{1+\delta}}{k}\right)^t = \left(1 - \frac{1 + \delta - 1}{k(1+\delta)}\right)^t$$
$$= \left(1 - \frac{\delta}{k(1+\delta)}\right)^t \leq \left(1 - \frac{\delta}{k \cdot ln(1+\delta)}\right)^t$$

Where the last step holds because $ln(1 + \delta) \leq 1 + \delta$ for all δ. Suppose now, by way of fancy analysis magic, that we choose to sample a number of points given by $t(k, n, \delta) = L \cdot ln(\frac{n}{\epsilon})$ for $L = \frac{k \cdot ln(1+\delta)}{\delta}$. Then, we find that

$$\mathbb{P}[\text{no firehouse in } N_i] \leq \left(1 - \frac{1}{L}\right)^{L \cdot ln(\frac{n}{\epsilon})} \leq \left(\frac{1}{e}\right)^{ln(\frac{n}{\epsilon})} = \frac{\epsilon}{n}$$

So, since placing a firehouse in each N_i happens disjointly between neighborhoods, the probability that our random sampling method will place a firehouse in every neighborhood is at least

$$1 - \sum_{i=1}^{n} \mathbb{P}[\text{no firehouse in } N_i] \geq 1 - n \cdot \left(\frac{\epsilon}{n}\right) = 1 - \epsilon$$

There are a lot of moving parts here. To recap, we devised a relaxed LP that returns non-integer numbers, but whose optimal cost OPT_f will certainly be better than the integral optimal cost OPT. We showed that OPT_f is the sum of the expected distances from each house to their nearest firehouse, and reasoned that if we could almost surely place a firehouse closer than this expected distance for every house, we would almost surely get a better cost $OPT_r \leq OPT_f \leq OPT$. Lastly, we showed that by placing $t = O(klog(n))$ firehouses randomly without replacement using probabilities returned by the fractional LP solution, we get the result that with probability $\geq 1 - \epsilon$ we will place a firehouse within a distance of $(1 + \delta)\mathbb{E}[d_i]$ of every house, for arbitrary $\epsilon > 0, \delta > 0$. This proves that, in expectation, we will use $O(klog(n))$ firehouses and will, in expectation, place a firehouse within the neighborhood of radius $\mathbb{E}[d_i]$ of every house, therefore (in expectation) achieving a cost $\leq OPT$. This whole thing happens in $poly(m)$ time. ■