CS 261 Problem Set 3

Luis A. Perez

Problem 19

Solution:

(a) We show that the algorithm that assigns each x_i the value TRUE with probability $\frac{1}{2}$ and the value FALSE with probability $\frac{1}{2}$ is a $\frac{1}{2}$-approximation algorithm.

Let S_j be an indicator variable representing whether or not clause C_j is satisfied. Let $|C_j|$ be the size of clause C_j, which is defined as the number of unique literals in the clause. Then the sum of the weights of the satisfied clauses is simply $S = \sum_{j=1}^m w_j S_j$. Taking the expected value of this, we have:

$$\begin{aligned}
E[S] &= \sum_{j=1}^m w_j E[S_j] && \text{(Definition of } S\text{)} \\
&= \sum_{j=1}^m w_j \Pr[S_j = 1] && \text{(Expected value of indicator variable)} \\
&= \sum_{j=1}^m w_j \left(1 - \frac{1}{2^{|C_j|}}\right) \\
& && \text{(}C_j\text{ is satisfied iff at least one of the variables is satisfied)} \\
&\geq \frac{1}{2}\sum_{i=1} w_j && (|C_j| \geq 1 \implies 1 - \tfrac{1}{2^{|C_j|}} \geq \tfrac{1}{2}) \\
&\geq \frac{1}{2} OPT && \text{(All clauses satisfied is at least as good as OPT)}
\end{aligned}$$

(b) The LP relaxation is straight-forward. We have the program L as per the below:

$$\max \quad \sum_{j=1}^m w_j z_j \tag{1}$$

subject to

$$\sum_{i \text{ st. } x_i \in C_j} y_i + \sum_{i \text{ st. } \bar{x}_i \in C_j} (1 - y_i) \geq z_j \quad \text{for all } C_j \tag{2}$$

$$0 \leq y_i \leq 1 \quad \text{for all } i = 1, \cdots, n \tag{3}$$

$$0 \leq z_j \leq 1 \quad \text{for all } j = 1, \cdots, m \tag{4}$$

Lemma 0.1. *0-1 solutions to L are in one-to-one correspondance with assignments to variables in the MAX SAT problem.*

Proof. For the first direction, suppose we have a 0-1 solution to L. Then for every y_i, set the corresponding x_i to TRUE if y_i is 1, else set to FALSE. By constraint 2, each non-zero z_j corresponds to a satisfied clause C_j. The weight of this assignment is therefore given by $\sum_{j=1}^{m} w_j z_j$.

For the second direction, suppose we have an assignment of the literals x_i. Then for each x_i that is TRUE, set y_i to 1, and 0 otherwise. For each satisfied disjunction C_j, set z_j to 1 and 0 otherwise. Note that this satisfies 2 since each z_j equal to 1, we must have that at least one of the $x_i, \bar{x}_i \in C_j$ is satsified, implying the sum above is non-empty. If x_i is satisfied, we sum y_i which is 1, so 2 holds. If $\bar{x}_i$ is satisfied, we sum $(1 - y_i)$ which is 1, so 2 holds.

Note that above transoformations above are objective function preserving. □

In conclusion, the 0-1 version of the LP presented above exactly solves the MAX SAT problem. As such, the above LP is a relaxation of the MAX SAT problem.

(c) We use the same variable definitions as in (a). We can show this directly.

$$\begin{aligned}
E[S] &= \sum_{j=1}^{m} w_j E[S_j] && \text{(Definition of } S \text{ and linearity of expectation)} \\
&= \sum_{j=1}^{m} w_j \Pr[S_j = 1] && \text{(Expected value of indicator variable)} \\
&= \sum_{j=1}^{m} w_j (1 - \Pr[S_j = 0]) && \text{(Complement)} \\
&= \sum_{j=1}^{m} w_j \left(1 - \prod_{x_i \in C_j} \Pr[x_i = 0] \prod_{\bar{x}_i \in C_j} \Pr[x_i = 1]\right) \\
& && (C_j \text{ is not satisfied iff all terms evaluate to false})
\end{aligned}$$

For a brief explanation for the above, note that C_j is not satisfied only if every variable is set such that each term evaluates to false. That means that for every non-negated term $x_i \in C_j$, we must have $x_i = 0$, and for every negated term $\bar{x}_i \in C_j$, we must have $x_i = 1$. Continuing, we have:

$$\begin{aligned}
E[S] &= \sum_{j=1}^{m} w_j \left(1 - \prod_{x_i \in C_j} \Pr[x_i = 0] \prod_{\bar{x}_i \in C_j} \Pr[x_i = 1]\right) && \text{(From above)} \\
&= \sum_{j=1}^{m} w_j \left(1 - \prod_{x_i \in C_j} (1 - y_i) \prod_{\bar{x}_i \in C_j} y_i\right) && (x_i \text{ is 1 with probability } y_i) \\
&\geq \sum_{j=1}^{m} w_j \left(1 - \left[\frac{\sum_{x_i \in C_j}(1 - y_i) + \sum_{\bar{x}_j \in C_j} y_i}{|C_j|}\right]^{|C_j|}\right)
\end{aligned}$$

Theh last line follows from the arithmetic mean-geometric mean inquality, which states that for any sequence of non-negative values $\{a_i\}$, the arithmetic mean is at least as large as the geometric mean. In other words, we know that:

$$\frac{1}{n}\sum_{i=1}^{n} a_i \geq \sqrt[n]{\prod_{i=1}^{n} a_i} \implies \left(\frac{1}{n}\sum_{i=1}^{n} a_i\right)^n \geq \prod_{i=1}^{n} a_i$$

By using precisely this fact, we can arrive at the lower-bound expressed above. Continuing:

$$\begin{aligned}
E[S] &\geq \sum_{j=1}^{m} w_j \left(1 - \left[\frac{\sum_{x_i \in C_j}(1 - y_i) + \sum_{\bar{x}_j \in C_j} y_i}{|C_j|}\right]^{|C_j|}\right) \\
&= \sum_{j=1}^{m} w_j \left(1 - \left[\frac{\sum_{x_i \in C_j}(1 - y_i) + \sum_{\bar{x}_j \in C_j} (1 - y_i')}{|C_j|}\right]^{|C_j|}\right) \\
& && \text{(Define } y_i' = 1 - y_i \text{ for negated variables)} \\
&= \sum_{j=1}^{m} w_j \left(1 - \left[1 - \frac{\sum_{x_i \in C_j} y_i + \sum_{\bar{x}_i \in C_j} y_i'}{|C_j|}\right]^{|C_j|}\right) \\
& && \text{(Simplifying inner summation)} \\
&= \sum_{j=1}^{m} w_j \left(1 - \left[1 - \frac{\sum_{x_i \in C_j} y_i + \sum_{\bar{x}_i \in C_j} (1 - y_i)}{|C_j|}\right]^{|C_j|}\right) && (y_i' = 1 - y_i) \\
&\geq \sum_{j=1}^{m} w_j \left(1 - \left[1 - \frac{z_j}{|C_j|}\right]^{|C_j|}\right) && \text{(Using constraint 2 from the LP)} \\
&\geq \sum_{j=1}^{m} w_j \left(1 - \left(1 - \frac{1}{|C_j|}\right)^{|C_j|} z_j\right)
\end{aligned}$$

The last line follows from the provided hint. We now that for any integer $\ell \geq 1$ and for $0 \leq x \leq$ the following holds:

$$1 - \left(1 - \frac{x}{l}\right)^l \geq 1 - \left(1 - \left(1 - \frac{1}{l}\right)\right) x$$

In our case, we have $\ell = |C_j| \geq 1$ and we have $x = z_j \in [0, 1]$ as enforced by the LP constraints. Continuing with the analysis we have:

$$\begin{aligned}
E[S] &\geq \sum_{j=1}^{m} w_j \left(1 - \left(1 - \frac{1}{|C_j|}\right)^{|C_j|} z_j\right) \\
&\geq \sum_{j=1}^{m} w_j \left(1 - \frac{1}{e}\right) z_j \\
&\qquad \text{(Mathematical fact (3) as provided, using } k = |C_j| \geq 1\text{)} \\
&= \left(1 - \frac{1}{e}\right) \sum_{j=1}^{m} w_j z_j && \text{(Simplifying)} \\
&\geq \left(1 - \frac{1}{e}\right) OPT
\end{aligned}$$

where the last line follows from the fact that the relaxed LP maximizes $\sum_{j=1}^{m} w_j z_j$ which includes the weight of the optimal possible assignment (solution) to MAX SAT.

Putting it all in one line, the above shows that:

$$E[S] \geq \left(1 - \frac{1}{e}\right) OPT$$

which means that our randomized algorithm is a $(1 - \frac{1}{e})$-approximation.

(d) We claim that the provided algorithm is an α-approximation given that the randomized algorithm is an α-approximation. The value of our algorithm is given by:

$$E[W \mid x_1 \leftarrow b_1, \cdots, x_n \leftarrow b_n]$$

Note that this not really an expected value, since all of our variables are fixed, but it's useful to think about it in this way. As such, what we want to show is that:

$$E[W \mid x_1 \leftarrow b_1, \cdots, x_n \leftarrow b_n] \geq \alpha OPT$$

We will do this by proving a slightly stronger statement.

Lemma 0.2. *For all $i = 1, \cdots, n$, we have the following:*

$$E[W \mid x_1 \leftarrow b_1, \cdots x_i \leftarrow b_i] \geq \alpha OPT$$

We proof the above by induction. Let us begin with $i = 1$. We have:

$$\begin{aligned} E[W] &= E[W \mid x_1 \leftarrow \textsc{True}] \Pr[x_1 \leftarrow \textsc{True}] + E[W \mid x_1 \leftarrow \textsc{False}] \Pr[x_1 \leftarrow \textsc{False}] \\ &\qquad \text{(By Law of Total Expectation)} \\ &= E[W \mid x_1 \leftarrow \textsc{True}] y_1 + E[W \mid x_1 \leftarrow \textsc{False}](1 - y_1) \\ &= W_{1,1} y_1 + W_{1,0}(1 - y_1) \end{aligned}$$

Now recall that $E[W] \geq \alpha OPT$, as such, it must be the case that:

$$E[W \mid x_1 \leftarrow \textsc{True}] y_1 + E[W \mid x_1 \leftarrow \textsc{False}](1 - y_1) \geq \alpha OPT$$

Given that $y_1 \in [0, 1]$, the above implies that one of $E[W \mid x_1 \leftarrow \textsc{True}] \geq \alpha OPT$ or $E[W \mid x_1 \leftarrow \textsc{False}] \geq \alpha OPT$. As such, we have:

$$\begin{aligned} E[W \mid x_1 \leftarrow b_1] &= \max\{W_{1,1}, W_{1,0}\} && \text{(By our algorithm's choice of } b_1) \\ &= \max\{E[W \mid x_1 \leftarrow \textsc{True}], E[W \mid x_1 \leftarrow \textsc{False}]\} && \text{(Definitions)} \\ &\geq \alpha OPT && \text{(By previous argument)} \end{aligned}$$

For our inductive hypothesis, let us assume that $E[W \mid x_1 \leftarrow b_1, \cdots x_i \leftarrow b_i] \geq \alpha OPT$. We now show that $E[W \mid x_1 \leftarrow b_1, \cdots x_{i+1} \leftarrow b_{i+1}] \geq \alpha OPT$. The argument is similar to our base case. We lay it out here for explicitness. First note that:

$$\begin{aligned} E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i] &= E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{True}] \Pr[x_{i+1} \leftarrow \textsc{True}] \\ &\quad + E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{False}] \Pr[x_{i+1} \leftarrow \textsc{False}] \\ &\qquad \text{(By Law of Total Expectation)} \\ &= E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{True}] y_i \\ &\quad + E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{False}](1 - y_i) \\ &= W_{i,1} y_i + W_{i,0}(1 - y_i) \end{aligned}$$

Now recall that $E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i] \geq \alpha OPT$ by our inductive hypothesis. As such, we must have that:

$$\begin{aligned} &E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{True}] y_i \\ &+ E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{False}](1 - y_i) \geq \alpha OPT \end{aligned}$$

Given that $y_i \in [0, 1]$, the above implies that one of $E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{True}] \geq \alpha OPT$ or $E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{False}] \geq \alpha OPT$. As such, we have:

$$\begin{aligned} E[W \mid x_1 \leftarrow b_1, \cdots, x_{i+1} \leftarrow b_{i+1}] &= \max\{W_{i+1,1}, W_{i+1,0}\} \\ &\qquad \text{(By our algorithm's choice of } b_1) \\ &= \max \begin{cases} E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{True}] \\ E[W \mid x_1 \leftarrow b_1, \cdots, x_i \leftarrow b_i, x_{i+1} \leftarrow \textsc{False}] \end{cases} \\ &\qquad \text{(Definitions)} \\ &\geq \alpha OPT \qquad \text{(By previous argument)} \end{aligned}$$

This concludes our proof.

Taking the special case of $i = n$, the above shows that:

$$E[W \mid x_1 \leftarrow b_1, \cdots x_n \leftarrow b_n] \geq \alpha OPT$$

As such, our deterministic algorithm is an α-approximation.

Problem 20

Solution:

(a) We prove that the maximum coverage problem is a special case of the budget compute problem proposed. We do this by taking an instance of a maximum coverage problem and reducing it into an instace of the budget compute problem whose solution is also a solution to the maximum coverage problem.

Suppose we are given a number k and a collection of sets $S = \{S_1, S_2, \cdots, S_m\}$. We wish to use pick k sets such that the total number of elements covered is maximized. Then to convert this to maximization of submudular functions with cardinality constraints, simply take $k = k$ (the cardinatity constraint), $U = S$ (the universe) and f (the submodular function) given by:

$$f(S) = \left| \bigcup_{S_i \in S} S_i \right|$$

All we need to show is that the optimal solution to the above gives a solution the the maximum coverage problem, and that f is monotone and submodular.

Lemma 0.3. *An optimum solution to the proposed maximization of submodular functions with cardinality constraints gives an optimal solution to the maximum coverage problem.*

Proof. To see this, simply take the optimal solution $S^* \subset U = S$. This is the optimal solution to our original maximum coverage problem. Note that that $|S^*| = k$, by the imposed cardinality constraint. Furthermore, note that $f(S^*)$ is maximal, which means that S^* contains exactly the k sets (out of all possible sets collections of k sets) that maximize the size of their union. This is exactly what it means to maximize the number of covered elements. □

Lemma 0.4. *The proposed function f is monotone and submodular.*

Proof. Suppose we have $S \subseteq T \subseteq U$. Then:

$$\begin{aligned}
f(S) &= \left| \bigcup_{S_i \in S} S_i \right| && \text{(Definition)} \\
&\leq \left| \bigcup_{S_i \in S} S_i \cup \bigcup_{S_i \in T \setminus S} S_i \right| && \text{(Adding element can only grow the set)} \\
&= \left| \bigcup_{S_i \in T} S_i \right| && \text{(Definition)}
\end{aligned}$$

As such f is monotone. Now let us take some $S_* \notin T$.

$$\begin{aligned}
f(T \cup \{S_*\}) - f(T) &= f(T) + |\{s \in S_* \mid s \notin A, \forall A \in T\}| - f(T) \\
&= |\{s \in S_* \mid s \notin A, \forall A \in T\}| \\
&\leq |\{s \in S_* \mid s \notin A, \forall A \in S\}| \\
&\text{($S \subseteq T$, so there can only be more elements it does not contain)} \\
&= f(S) + |\{s \in S_* \mid s \notin A, \forall A \in S\}| - f(S) \\
&= f(S \cup \{S_*\}) - f(S)
\end{aligned}$$

As such, we conclude that:

$$f(T \cup \{S_*\}) - f(T)) \leq f(S \cup \{S_*\}) - f(S)$$

which means that f is submodular. □

(b) Let $f(S)$ denote the expected number of active vertices at the conclusion of the cascade, given that the vertices of S are active at the beginning. To assist with the proofs below, we can think of our randomized algorithm A as being defined by deterministic algorithms A_r where we compute the sequence of coin-flips before-hand (once these are fixed, the algorithm is deterministic). As such, our algorithm is really just a distribution p_r over the deterministic A_r.

Lemma 0.5. *f is monotone.*

Proof. Consider any arbitrary f_r for some fixed quence of coin-flips r. Then suppose we have $S \subseteq T \subseteq V$. We claim that $f_r(S) \leq f_r(T)$. This follows almost immediately, since $S \subseteq T$ any vertices which are activated by S would also be activated by T (under the r sequence of coinflips).

From the above, we have:

$$\begin{aligned}
f(S) &= \sum_r f_r(S) p_r && \text{(Definition of expectation)} \\
&\leq \sum_r f_r(T) p_r && \text{(Using results argued previously)} \\
f(T) & && \text{(Definition of expectation)}
\end{aligned}$$

As such, we have that $f(S) \leq f(T)$.

□

Lemma 0.6. *f is submodular.*

Proof. Let us take some arbitrary f_r. Suppose we have $S \subseteq T \subseteq V$ with $i \not{i}nT$. We claim that $f_r(T \cup \{i\}) - f_r(T) \leq f_r(S \cup \{i\}) - f_r(S)$. To see this, note that for our fixed sequence of coin-flips r, the set of vertices activated by S will be a subset of those activated by T ($A_S \subseteq A_T$). Now consider the set of vertices activated by i, A_i. Then we have:

$$\begin{aligned} f_r(T \cup \{i\}) - f_r(T) &= |A_T \cup A_i| - |A_T| \\ &= |A_T \setminus A_S \cup A_i \cup A_S| - |A_T| \\ &\leq |A_T \setminus A_S| + |A_i \cup A_S| - |A_T| \\ &= |A_T| - |A_S| + |A_i \cup A_S| - |A_T| && (S \subseteq T) \\ &= f_r(S \cup \{i\}) - f_r(S) \end{aligned}$$

As such, we have:

$$\begin{aligned} f(T \cup \{i\}) - f(T) &= \sum_r p_r[f_r(T \cup \{i\}) - f_r(T)] && \text{(Expectation over coin flips)} \\ &\leq \sum_r p_r[f_r(S \cup \{i\}) - f_r(S)] && \text{(Results from above)} \\ &= f(S \cup \{i\}) - f(S) \end{aligned}$$

As such, we've shown that f is sub-modular. □

(c)

Lemma 0.7. *Let f be monotone submodular. Let i be the element that increases f the most. We claim that:*

$$f(S \cup \{i\}) - f(S) \geq \frac{1}{k}(OPT - f(S))$$

Proof. Following the hint, suppose that at the i-th iteration, our algorithm was allowed to pick k elements. Certainly then, we would be able to increase our objectve by exactly $OPT - f(S)$ - we could simply pick the k elements in the optimal set, S_{OPT}. As such, we know that:

$$f(S \cup S_{OPT}) - f(S) = OPT - f(S)$$

Define $S^m_{OPT} = \{x_1, \cdots, x_m\}$ where $S^k_{OPT} = S_{OPT}$ be the set of elements we added. Then since we only add k elements but increase in total by $OPT - f(S)$, we must have that at some point adding the m-th element led to a sufficiently large increase. That is, we must have there exists m such that:

$$f(S \cup S^{m-1}_{OPT} \cup \{x_m\}) - f(S \cup S^{m-1}_{OPT}) \geq \frac{1}{k}(OPT - f(S))$$

Then we have the following logic:

$$\begin{aligned} f(S \cup \{i\}) - f(S) &\geq f(S \cup \{x_m\}) - f(S) && (i \text{ leads to maximal marginal increase}) \\ &\geq f(S \cup S^{m-1}_{OPT} \cup \{x_m\}) - f(S \cup S^{m-1}_{OPT}) \\ & && \text{(By Submodularity of } f \text{ using } T = S \cup S^{m-1}_{OPT}) \\ &\geq \frac{1}{k}(OPT - f(S)) && \text{(By argument above)} \end{aligned}$$

This concludes our proof. □

(d) We proof that for every monotone submodular function, the greedy algorithm is a $(1-\frac{1}{e})$-approximation.

Proof. Let S_i be the set proposed by the algorithm after iteration i, with e_i as the element added at iteration i. We have $S_0 = \emptyset, S_1 = \{e_1\}, S_2 = \{e_1, e_2\}, \cdots$ and S_k as the greedy algorithm's output. Then let us consider $f(S_k)$. For simplicity, we assume that $f(S_0) = 0$ (if this is not the case, note that you can always shift f by a constant factor without affecting the solution).

$$\begin{aligned}
f(S_k) &= (f(S_k) - f(S_{k-1})) + f(S_{k-1}) && \text{(Adding 0)} \\
&= [f(S_{k-1} \cup \{e_k\}) - f(S_{k-1})] + f(S_{k-1}) && \text{(Using } S_k = S_{k-1} \cup \{e_k\}\text{)} \\
&\geq \frac{1}{k}(OPT - f(S_{k-1})) + f(S_{k-1}) && \text{(Results from (c))} \\
&= \frac{OPT}{k} + (1 - \frac{1}{k})f(S_{k-1}) && \text{(Simplifying)}
\end{aligned}$$

If we apply the same logic once again to expand $f(S_{k-1})$, we have:

$$\begin{aligned}
f(S_k) &\geq \frac{OPT}{k} + (1-\frac{1}{k})\left[\frac{OPT}{k} + (1-\frac{1}{k})f(S_{k-2})\right] \\
&= \frac{OPT}{k} + \frac{OPT}{k}(1-\frac{1}{k}) + f(S_{k-2})(1-\frac{1}{k})^2
\end{aligned}$$

From the above, we can immediately identify a pattern. Unrolling it out until we get to $f(S_0)$, we have:

$$f(S_k) \geq \frac{OPT}{k}\left[1 + (1-\frac{1}{k}) + (1-\frac{1}{k})^2 + \cdots + (1-\frac{1}{k})^{k-1}\right]$$

Recalling the fact that $\sum_{j=0}^{k-1}(1-\frac{1}{k})^j = \frac{1-(1-\frac{1}{k})^k}{\frac{1}{k}}$, we have:

$$\begin{aligned}
f(S_k) &\geq OPT\left[1 - (1-\frac{1}{k})^k\right] \\
&\geq OPT(1-\frac{1}{e}) && \text{(From Problem 19)}
\end{aligned}$$

As such, we can conclude that the greedy algorithm is a $(1-\frac{1}{e})$-approximation algorithm. □

Problem 21

Solution:

(a) We proof a slightly stronger statement.

Lemma 0.8. *Let S_i be the set of i points at iteration i of the algorithm ($S_1 = \{z\}$). Let $r_{i+1} = \max_{v \in V}\{d(v, S_i)\}$ and $x_{i+1} = \arg\max_{v \in V}\{d(v, S_i\}$. We claim that the set of $i+1$ points $S_{i+1} = S_i \cup \{x_{i+1}\}$ have minimum pairwise distance r_{i+1}.*

Note that our original statement is simply the above when $i = k$.

Proof. We proof this by induction on i. For our base case, consider $i = 1$. We have $S_1 = \{z\}$, $r_2 = \max_{v \in V}\{d(v, \{z\})\}$ and $x_2 = \arg\max_{v \in V}\{d(v, \{z\}\}$. As such, our set of only has two points $z \in V$ and $x_2 \in V$. The only (and as such minimum) pairwise distance between these two points is $d(x_2, z) = r_2$ by construction.

Our inductive hypothesis is as follows. Define S_i and r_{i+1}, x_{i+1} as above. Then the set of $i+1$ points $S_{i+1} = S_i \cup \{x_{i+1}\}$ has minimum pairwise distance r_{i+1}.

We wish to show that the above holds for $i = i+1$ as well. Suppose we have the set S_{i+1} as constructed by our algorithm. Now let us consider the set $S_{i+2} = S_{i+1} \cup \{x_{i+2}\}$. We claim that $r_{i+1} \geq r_{i+2}$.

$$\begin{aligned}
r_{i+2} &= \max_{v \in V}\{d(v, S_{i+1})\} && \text{(Definition of } r_{i+2}\text{)} \\
&= \max_{v \in V} \min_{s \in S_{i+1}} \{c_{sv}\} && \text{(Definition of } d\text{)} \\
&= \max_{v \in V} \min_{s \in S_i \cup \{x_{i+1}\}} \{c_{sv}\}\} && \\
&= \max_{v \in V} \min\{\min_{s \in S_i} c_{sv}, c_{x_{i+1}v}\} && \\
&\leq \max_{v \in V} \min_{s \in S_i c_{sv}} && \text{(Taking min over fewer values)} \\
&= \max_{v \in V} d(v, S_i) && \text{(Definition of } d\text{)} \\
&= r_{i+1}
\end{aligned}$$

What we wish to show is that the minimum pairwise distance of $S_{i+2} = S_{i+1} \cup \{x_{i+2}\}$ is r_{i+2}. For all cases $u, v \in S_{i+1}$, we know that the minimum pairwise distance is r_{i+1} (by our inductive hypothesis). As such, let us consider the distances $c_{ux_{i+2}}$ where $u \in S_{i+1}$.

$$\begin{aligned}
c_{ux_{i+2}} &\geq \min_{s \in S_{i+1}} c_{sx_{i+2}} && \text{(A specific instance must be at least the minimum value)} \\
&= d(x_{i+2}, S_{i+1}) && \text{(Definition of } d\text{)} \\
&= \max_{v \in V} d(v, S_{i+1}) && \text{(By definition of } x_{i+2}\text{)}
\end{aligned}$$

As such, we conclude that $c_{ux_{i+2}} \geq r_{i+2}$ for all $u \in S_{i+1}$. Furthermore, we know that equality is achieved for *some* $u \in S_{i+1}$ by definition of x_{i+2}. To summarize, we have:

- r_{i+1} is the minimum pairwise distances for S_{i+1} (by induction)
- The minimum distance between x_{i+2} and any vertex in S_{i+1} is r_{i+2} (by arguments immediately above), and this distance is achieved by construction of x_{i+2} for some $u \in S_{i+1}$.
- $r_{i+2} \leq r_{i+1}$ (by argument preceeding the one above)

From the above three statements, we have that the minimum pairwise distance for $S_{i+2} = S_{i+1} \cup \{x_{i+2}\}$ is precisely r_{i+2}. □

Setting $i = k$, our lemma reads:

Let S be the set of k returned by our algorithm. Let $r = \max_{v \in V}\{d(v, S)\}$ and $x = \arg\max_{v \in V}\{d(v, S\}$. Then the set of $k+1$ points $S \cup \{x\}$ have minimum pairwise distance r.

This is exactly what we wished to proof.

(b) We can proof this by contradiction. Our claim is as follows:

Lemma 0.9. *The algorithm from part (a) is a 2-approximation. That is to say, for the S returned by our algorithm:*

$$\max_{v \in V} d(v, S) \leq 2OPT$$

where:

$$OPT = \max_{v \in V} d(v, S_{OPT})$$

where S_{OPT} are the k optimal centers

Proof. For the sake of contradiction, suppose the above statement were not true. Letting $x = \arg\max_{v \in V} d(v, S)$ and $r = \max_{v \in V} d(v, S)$, we must have $r > 2OPT$. However, by the results from (a), this implies that the minimum pairwise distance for the set $S \cup \{x\}$ is $r > 2OPT$. Our set has $k+1$ points, which means at least two points (by the pigeon whol principle) must share a center in S_{OPT}. Takes these two points $u, v \in S \cup \{x\}$ and let $o \in S_{OPT}$ be the shared center. However, this leads to the following:

$$\begin{aligned} r &\leq c_{uv} && \text{(The minimum pairwise distance for } S \cup \{x\} \text{ is } r) \\ &\leq c_{uo} + c_{ov} && \text{(Triangle inequality)} \\ &\leq OPT + OPT = 2OPT && (o \text{ is the shared center in the optimal solution}) \end{aligned}$$

However, this contradicts our assumption that $r > 2OPT$. As such, we must have that:

$$\max_{v \in V} d(v, S) \leq 2OPT$$

□

(c) First, note that if we have a dominating set of size k, then we have dominating sets of size $k \geq k$ (simply add any node to these sets). As such, finding a dominating set of size at most k is equivalent to finding a dominating set of size exactly k.

Next, note that $\epsilon \in (0,1)$, since $\epsilon \geq 1$ would give us a solution to the metric k-center problem.

For this reduction, we need to show how to use a good metric k-center approximation algorithm to solve the Dominating Set problem. Given an instance $G = (V, E)$ and k of the latter problem, we transform it into an instance $G' = (V', E', k')$ of the metric k-center problem, where:

- V' = V
- E' is all edges (so (V', E') is the complete graph)
- For each $e \in E'$, set:
$$c_e = \begin{cases} 1 & e \in E \\ > (2-\epsilon) & e \notin E \end{cases}$$
- $k' = k$ where $2 - \epsilon > 1$ is the approximation factor we cant to rule out. Note that with the above definition, the triangle inequality holds since:
$$c_{uv} + c_{vw} \geq 1 + 1 \geq 2 - \epsilon \geq c_{uw}$$
Using the fact that $\epsilon \in (0,1)$.

The key point is that there is a one-to-one correspondence between the Dominating Sets of size k and the k-center sets with furthests distance to the nearest center of 1. To see this, note that:

- If G has dominating set of size k (S), this means that for all $u \in V$, either $u \in S$ or u has a neighbor $v \in S$. In the former cases, this means that in G' we have $d(u,S) = 0$. In the latter case, we must have $(u,v) \in E$ which implies $d(u,S) = c_{uv} = 1$. As such, if we let S instead be our k-centers, then the furthest distance to the nearest center is 1.
- If G has no dominating set of size k, then for all sets $S \subseteq V$ such that $|S| = k$, we have that $\exists u \notin S$ that is not connected to any vertex in S (as per G). This means that in our graph G', we must have $d(u,S) > 2 - \epsilon$ (as per the construction above). Note that this holds for all $S \subseteq V$ such that $S = |k|$. As such, we know that every k-center set has some point which is at least $2 - \epsilon$ from its closest center.

Now, suppose there were an $(2-\epsilon)$-approximation algorithm $\mathcal{A}$ for the metric k-center problem. We can use $\mathcal{A}$ to solve the Dominating set problem: given an instance (G, k) of the problem, run the reduction above and then invoke $\mathcal{A}$ on the produced metric k-center problem. Since there is more than an $(2 - \epsilon)$ factor gap between (i) and (ii) and $\mathcal{A}$ is an $(2-\epsilon)$-approximation, the output of $\mathcal{A}$ indicates whether or not there exists a dominating set of size k in G. If yes, then our approximation algorithm will return a set S whose values it at most $(2-\epsilon)$ (the optimal is 1). If not, then our approximation algorithm will return a set S whose value is strictly greater than $(2-\epsilon)$.

Problem 22

Solution: We follow the hint and prove the stronger statement.

Lemma 0.10. *The algorithm's final partition has objective function value at least $\frac{k-1}{k}$ times the sum of all the edge weights.*

Proof. We proof directly for arbitrary k. First, we claim that for all $v \in V$, the weight contributed by v to the k-cut is at least $\frac{k-1}{k}$ the weight of all incident edges of v. More precisely, we claim that $\forall v \in V$ where $v \in S_i$:

$$\underbrace{\sum_{u \notin S_i} w_{uv}}_{\text{weight of edges of } v \text{ in cut}} \geq \frac{k-1}{k} \underbrace{\sum_{u \in V} w_{uv}}_{\text{all edges containing } v}$$

To see why, suppose for contradiction there $\exists v \in V, v \in S_i$ such that the above inequality does not hold. That is to say, we have $v \in V, v \in S_i$ such that $\sum_{u \notin S_i} w_{uv} < \frac{k-1}{k} \sum_{u \in V} w_{uv}$. Since there are $k-1$ other partitions and all the weights are non-negative, there must exists some partition $S_j \neq S_i$ such that:

$$\begin{aligned}
\sum_{u \in S_j} w_{uv} &\leq \frac{1}{k-1} \sum_{u \notin S_i} w_{uv} && \text{(Some partition must be at most the average weight)} \\
&< \frac{1}{k-1} \frac{k-1}{k} \sum_{u \in V} w_{uv} && \text{(By our assumption)} \\
&= \frac{1}{k} \sum_{u \in V} w_{uv} \\
&< \sum_{u \in S_i} w_{uv} && \text{(By our assumption)}
\end{aligned}$$

The above shows that $\sum_{u \in S_j} w_{uv} < \sum_{u \in S_i} w_{uv}$. However, this implies that moving $u \in S_i$ to the partition S_j would strictly increase our objective function, since we lose the weight $\sum_{u \in S_j} w_{uv}$ but gain the weight $\sum_{u \in S_i} w_{uv}$.

This contradicts the fact that our algorithm terminated. As such, we conclude that $\forall v \in V$ where $v \in S_i$:

$$\sum_{u \notin S_i} w_{uv} \geq \frac{k-1}{k} \sum_{u \in V} w_{uv}$$

Since the above holds for every vertex $v \in V$, it implies that in the final partition, at least $\frac{k-1}{k}$ of the total edge-weight is part of the produced cut. Letting C be the set of cut edges, we have:

$$\sum_{e \in C} w_e \geq \frac{k-1}{k} \sum_{e \in E} w_e$$

This concludes our proof. □

Problem 23

Solution:

(a) As instructed, we construct the dual of the LP. We will have $n+m-2$ variables and m constraints where $|E| = m$ and $|V| = n$. We have variables z_v for $v \neq s, t$ and variables ℓ_e for all $e \in E$.

$$\min \sum_{e \in E} u_e \ell_e \tag{5}$$

subject to

$$\begin{aligned}
\ell_{uv} + z_u - z_v &\geq 0 \quad \text{for all } (u,v) \in E, u \neq s, v \neq t &(6)\\
\ell_{ut} + z_u &\geq 1 \quad \text{for all } (u,t) \in E &(7)\\
\ell_{sv} - z_v &\geq 0 \quad \text{for all } (s,v) \in E &(8)\\
z_v &\geq 0 \quad \text{for all } v \in V, v \neq s, t &(9)\\
\ell_e &\geq 0 \quad \text{for all } e \in E &(10)
\end{aligned}$$

We now argue that there exists an optimal solution to the primal where the flow conservation inequalities are tight. Take some optimal solution f^*. If the flow conservation inequalities are tight, we are done. Otherwise, there must exist at least one $v \in V$ such that the total outflow is less than the inflow.

$$\underbrace{\sum_{e \in \delta^+(v)} f_e^*}_{\text{outflow}} < \underbrace{\sum_{e \in \delta^-(v)} f_e^*}_{\text{inflow}}$$

Consider all s-v-t paths (paths from s to t going through v). Since the total outflow from v along these paths is smaller than the inflow, we can reduce the flow along the s-v paths without affecting the flow reaching t (eg, maintatining the total outflow of v constant). We can do this to give us a new flow f' such that the inflow matches the outflow at v (the inequality is tight) without affecting the total flow from s to t.

As such, this new flow f' has the same objective value as our flow f^*. Repeating this process for any $v \in V$ such that the total outflow is less than the total inflow, we can construct a flow f with the same objective value as f^* but for which all of the inequality constraints are tight.

(b) We let (z^*, ℓ^*) be an optimal dual solution, and define $z_s^* = 0$ and $z_t^* = 1$. Note that by 6, for all $(u,v) \in E$ where $u \neq s$ and $v \neq t$ we have:

$$\begin{aligned}
\ell_{uv}^* + z_u^* - z_v^* &\geq 0 &\text{(Constraint 6)}\\
\implies \ell_{uv}^* &\geq z_v^* - z_u^*
\end{aligned}$$

Similarly, by 7 and 8, we have:

$$\ell^*_{ut} \geq 1 - z^*_u = z^*_t - z^*_u \qquad \text{(By Constraint 7 and } z^*_t = 1)$$
$$\ell^*_{sv} \geq z^*_v = z^*_v - z^*_s \qquad \text{(By Constraint 8 and } z^*_s = 0)$$

Combining with the fact that $\ell \geq 0$, the three inqualities above yield:

$$\ell^*_{uv} \geq \max\{z^*_v - z^*_u, 0\} \qquad (\text{ for all } (u,v) \in E)$$

By strong duality and the results from (a), all inequalities above are tight. As such, we have:

$$\ell^*_{uv} = \max\{z^*_v - z^*_u, 0\}$$

(c) We show that $S(r), V \setminus S(r)$ is an s-t cut. That is to say, it is a partion of V where $s \in S(r)$ and $t \in V \setminus S(r)$.

Lemma 0.11. *$s \in S(r)$*

Proof. Note that by construction, $z^*_s = 0$. As such, no matter our choice of $r \in [0,1)$, we always have $z^*_s = 0 \leq r$ which means $s \in S(r)$. □

Lemma 0.12. *$t \notin S(r)$*

Proof. Note that by construction, $z^*_t = 1$. As such, no matter our choice of $r \in [0,1)$, we always have $z^*_s = 1 > r$ which means $t \notin S(r)$. □

It is clear that $A = S(r), B = V \setminus S(r)$ is a partition. As argued above, $s \in A$ and $t \in B$. As such, this define an s-t cut.

(d)

Lemma 0.13. *For any edge $e \in E$, the probability that e contributes to the capacity of the cut $(S(r), V \setminus S(r))$ is exaclty l^*_e.*

Proof. Consider the edge $e = (u,v)$. It will contribute to the capacity of the corresponding s-t cut if and only if $u \in S(r)$ and $v \notin S(r)$. This will occur if and only if $z^*_u \leq r$ and $z^*_v > r$. Since $r \in [0,1)$ uniformly at random, this will occur if and only if we happen to chose r such that $z^*_u \leq r < z^*_v$, which means we happen to choose $r \in [z^*_u, z^*_v)$. The probability of this occurring is precisely:

$$\max\left\{\frac{z^*_v - z^*_u}{1-0}, 0\right\} = \max\{z^*_v - z^*_u, 0\} = \ell^*_e$$

where the last equality follows from the results in (b). □

(e)

Lemma 0.14. *The expected capacity of the s-t cut $(S(r), V \setminus S(r))$ is $\sum_{e \in E} \ell_e^* u_e$.*

Proof. Consider each edge. If it is included in the cut, it will contribute u_e to the capacity of the cut. If it is not, it will contribute 0. As such, the expected capacity of the cut is given simply by:

$$\begin{aligned} E[\text{capacity of cut}] &= \sum_{e \in E} \Pr[e \text{ contributes to cut}] u_e \\ &= \sum_{e \in E} \ell_e^* u_e && \text{(By results from (d))} \end{aligned}$$

This is precisely the objective of the dual LP. □

(f) We argue that for any fixed $r \in [0, 1)$, the corresponding 0-1 solution to our dual is optimal by arguing that it satisfies the complementary slackness conditions. As such, it achieves the optimal value for of $\sum_{e \in E} \ell_e^* u_e$, meaning it is a min-cut.

Proof. Fix some $r \in [0, 1)$ and define the cut $(S(r), V \setminus S(r))$. We proof that this is a min-cut by showing that the complementary slack conditions hold for L and P.

Define the solutions to our dual as the below, for some fixed $r \in [0, 1)$:

$$\begin{aligned} z_v &= \begin{cases} 0 & z_v^* \leq r \text{ or } v = s \\ 1 & z_v^* > r \text{ or } v = t \end{cases} \\ \ell_{uv} &= \max\{z_v - z_u, 0\} \\ f_e &= \ell_e^* u_e \end{aligned}$$

Note that the solutions represented by the above are feasible. For our primal program, setting $f_e = \ell_e^* u_e$ trivial satisfies the fact that $f_e \geq 0$ and $f_e \leq u_e$ for all $e \in E$. The conservation constraints are also satisifed by noting that:

$$\begin{aligned} OPT_D &= \sum_{e \in E} \ell_e^* u_e && \text{(Definition of dual)} \\ &= \sum_{e \in E} f_e && \text{(Definition of } f_e\text{)} \\ &= OPT_P && \text{(Strong duality)} \end{aligned}$$

As such, the f_e as defined are also an optimal solution to the primal, which by the results in (a), imply that the flow conservation constraints are tight (and therefore satisfied).

Similarly. note that as defined for $u \neq s$ and $v \neq t$ we trivially have:

$$\begin{aligned} \ell_{uv} + z_u - z_v &= \max\{z_v - z_u, 0\} - (z_v - z_u) \geq 0 \\ \ell_{ut} + z_u &= \max\{1 - z_u, 0\} + z_u = 1 \geq 1 \\ \ell_{sv} - z_v &= \max\{z_v, 0\} - z_v \geq 0 \end{aligned}$$

The positivity constraints are trivially satsified.

As such, we have solutions defined by our s-t cut which are feasible. We now show that both complementary slackness conditions are satisfied.

- Whenever $f_e \neq 0$, z, ℓ satisfies the e-th constraint with equality. Suppose f_e is non-zero. Then $e = (u, v)$ is not in our s-t cut. If $u = s$, then this means that $z_v = 0$ as well, meaning that $l_{uv} - z_v = \max\{0 - 0, 0\} - 0 = 0$. If $v = t$, then this means that $z_u = 1$ as well, meaning that $l_{uv} + z_u = \max\{1 - 1, 0\} + 1 = 1$. Otherwise, if (u, v) is entirely within one set, we have $z_u = z_v$ which means $\ell_{uv} + z_u - z_v = \max z_u - z_u, 0 + z_u - z_u = 0$. The last possibility is that (u, v) is a backwards edge, in which case we have $z_v = 0$ and and $z_u = 1$. This gives us $\ell_{uv} + z_u - z_v = \max\{0 - 1, 0\} + 1 = 0$. As such, we satisfie the e-th contraint with equality.
- Whenever $z_v \neq 0$, the corresponding constraint is satisfied with equality. This is trivially satisfied by what we showed in (a). As such, we focus instead in the case when $\ell_e \neq 0$. By our definition of f_e, this corresponding constraint is also trivially satisfied.

As such, our new solution to the dual constructed from the s-t derived from the fixed value of r satisfies the complementary slackness conditions. This implies that it is an optimal solution, which means that the s-t cut is a min cut. □

(g) The idea here is to take advantange of the fact that we can slowly rotate the vectors v_i. To get an intuition as to why this works, let's think about what each of the restrictions implies.

The fact that for all i we have $v_i \cdot v_i = 1$ means that all feasible solutions must be on the surface of the unit hyper sphere (they are all unit length vectors). Recalling the general defintion of the dot product:

$$v_i \cdot v_j = ||v_i||_2||v_j||_2 \cos\theta = \cos\theta$$

where θ is the angle formed between the two vectors v_i and v_j, and we use the fact that they are unit-length.

The constraint that $v_s \cdot v_t = -1 = \cos\theta$ means that the angle formed between v_s and v_t is π (they are at antipodal points on the unit hypersphere).

So taking the above into account, the SDP relaxation is trying to find v_i such that any nodes which are connected (and edge exist) have as small an angle between them (eg, $v_i \cdot v_j = 1 = \cos\theta \implies \theta = 0$) with the restriction that two of our nodes, s and t, have an angle of π between them.

Intuitively, one method to minimize this would be to construct a sequence of edges corresponding to connected vertices v_i where each v_i is rotated by $\frac{pi}{n}$. More precisely, let us construct a graph $G = (V, E)$ where we have $|V| = n$ and $s, t \in V$. Letting

$v_1 = s$ and $v_n = t$, we have edges of the form $(v_i, v_{i+1}) \in E$. That is to say, our graph is simply a single, simple undirected path between s and t.

$$s \to v_2 \to v_3 \to \cdots \to v_{n-2} \to v_{n-1} \to t$$

It is clear that for G, the min cut is 1 (cut any edge). Note that this value of the min cut is so, no matter the value of n.

Now, let us construct $v_i \in \mathbb{R}^n$. Let $v_1 = v_s$ be any arbitrary vector in our space. Then we must have $v_n = v_t = -v_s$. For the remaining vectors, define them as follows:

- Define some hyperplane H containing v_s and v_t (any such hyper-plane will do).
- Given v_i, let v_{i+1} be a vector in H which is a rotation of v_i by $\frac{\pi}{n-1}$ radians, towards v_t (the angle between v_t and v_{i+1} decreases) and away from v_s (the angle between v_s and v_{i+1} increases).
- Continue the above process until we've constructed every v_i.

Lemma 0.15. *We claim that this $\{v_i\}$ as constructed above are a feasible solution to the given SDP.*

Proof. To see this, note that they satisfy the contraints. Since every v_i is just a rotation of v_s, and v_s is a unit vector, we have that $v_i \cdot v_i = 1$ for all i. Similarly, by construction, we have that $v_s \cdot v_t = -v_s \cdot v_s = -1$.

The last thing to note is that the construction above is valid. We have exactly n vertices, each rotated by $\frac{\pi}{n-1}$ radians. As such, the last rotation leads to an angle between $v_1 = v_s$ and $v_n = v_t$ of π radians. □

As such, the $\{v_i\}$ constructed above are a feasible solution to the SDP.

Lemma 0.16. *Next, we claim that as $n \to \infty$, the $\{v_i\}$ constructed above achieve an objective value of* 0.

Proof. To see this, we simple expand the objective value function.

$$\begin{aligned}
\sum_{(i,j)\in E} \frac{1 - v_i \cdot v_j}{2} &= \sum_{i=1}^{n-1} \frac{1 - v_i \cdot v_{i+1}}{2} && \text{(By construction of } G\text{)} \\
&= \sum_{i=1}^{n-1} \frac{1 - \cos\left(\frac{\pi}{n-1}\right)}{2} && \text{(By construction of } v_i\text{)} \\
&= \frac{1}{2}(n-1)(1 - \cos\left(\frac{\pi}{n-1}\right)) && \text{(Simplifying)}
\end{aligned}$$

Taking the limit of the above as $n \to \infty$, we have:

$$\begin{aligned}
\frac{1}{2}\lim_{n\to\infty}(n-1)(1-\cos\left(\frac{\pi}{n-1}\right)) &= \frac{1}{2}\left[\lim_{n\to\infty} n-1\right]\left[1-\cos\left(\lim_{n\to\infty}\frac{\pi}{n-1}\right)\right] \\
&= \frac{1}{2}\lim_{n\to\infty} n[1-\cos 0] \\
&= \frac{1}{2}\lim_{n\to\infty} n[1-1] & = 0
\end{aligned}$$

Putting everything together, we have, therefore have the following, letting OPT_{SDP} be the optimal value of the SDP relaxation and OPT be the optimal value of the min-cut.

$$\begin{aligned}
\lim_{n\to\infty}\frac{OPT_{SDP}}{OPT} &\leq \lim_{n\to\infty}\frac{cost(\{v_i\})}{OPT} \\
&\qquad (OPT_{SDP} \text{ is at most the cost of our feasible solution}) \\
&= \frac{0}{OPT} && \text{(Results from above)} \\
&= \frac{0}{1} && \text{(OPT is always 1)} \\
&= 0
\end{aligned}$$

As such, the ration can be made arbitrarily small as we increase the value of n. □

Problem 24

Solution:

(a) We first proof some statements.

Lemma 0.17. *Let $G = (V, E)$. Consider a set of edges $E' \subset E$ such that for all $v \in V$, there is exactly one incoming edges and one outgoing edge in E'. Then E' is a cycle cover.*

Proof. Since all v have one incoming and one outgoing edge, each v must be part of a cycle (if not, we would have some vertex with no outgoing edge and some vertex with no incoming edge forming a path). Since they have *exactly* one incoming and one outgoing edge, each v must be part of just one cycle. Since the cycles are directed, each cycle must have at least two edges. This is exactly the definition of a cycle cover. □

Lemma 0.18. *Let $E' \subseteq E$ be the set of edges forming a cycle cover of $G = (V, E)$. Then for all $v \in V$, there is exactly one incoming and one outgoing edge.*

Proof. For any $v \in V$, by definition of a cycle cover, v must belong to exactly one simple, directed cycle C_i with at least two edges. As such, v has exactly one incoming and one outgoing edge from the edges forming the cycle cover. □

With the above, we can now show how to reduce the problem of finding a cycle cover of $G = (V, E)$ to that of finding a minimum weight perfect matching in a bipartite graph (which from lecture we know we can do in polynomial time).

To do this, contruct a new graph $H = (V', E')$ where $V' = V_{in} \cup V_{out}$ where each of $V_{in} = V$ and $V_{out} = V$ (we duplicate the nodes but label them as in and out). For E', for every edge $(u, v) \in E$ with cost c_{uv}, add the edge (u_{out}, v_{in}) to E' and set the weight of the edge to c_{uv}.

Lemma 0.19. *The cycle covers of G are in one-to-one correspondance with the perfect matchings of H.*

Proof. To see this, suppose we have a perfect matching, M, in H. Take any edge $(u_{out}, v_{in}) \in M$, and add the corresponding edge (u, v) to our cycle cover. Then note that by definition of a perfect matching, every node has exactly one outgoing edge (as per V_{out}) and one incoming edge (as per V_{in}). As argued previosly, these set of edges form a cycle cover.

For the other direction, suppose we have a cycle cover C. By our arguments above, this means that we have a set of edges $(u, v) \in M$ such that each node has exactly one incoming edge and one outgoing edge. Then the edges (u_{out}, v_{in}) correspond exactly to a perfect matching in H. □

From the above, we conclude that solving the minimum perfect matching problem in H' will yield the corresponding cycle cover with minimum total cost.

(b) We use the subrouting to give a $\frac{3}{2}$-approximation algorithm for the $\{1, 2\}$ special case of the ATSP problem. The algorithm works as follows, ginve a complete directed graph $G = (V, E)$ with all $n(n-1)$ directed edges.

- Use the algorithm from (a) to compute the cycle cover of G, $C_1, \cdots, C_k$.
- Delete all but one vertex from every C_i, and let this be $G' = (V', E')$.
- Find any arbitrary tour that visits all vertices in G', A.
- Combine the edges from $C_1, \cdots, C_k$ with the edges from A to form a complete tour of G called C, with possibly repeated vertices. Shortcut any repeated vertices to form our ATSP tour and output it, call it T.

We proof that the above algorithm produces an ATSP tour that is at most $\frac{3}{2}OPT$.

Proof. To see this, note that the cost of the cycle cover $C_1, \cdots, C_k$ is at most $\frac{OPT}{2}$. First, consider our augmented graph constructed in part (a), $G' = (V_{in} \cup V_{out}, E')$. Then note that the optimal ATSP tour essentially consists of two perfect matchings on this graph G' (every vertex has an in-edge and an out edge). As such, OPT must be the sum of these two perfect matchings. Given that the cost of the cycle-cover is the minimum perfect matching, we must have that the cost of $C_1, \cdots, C_k$ is at most $\frac{OPT}{2}$.

Next, let us consider the patching. In the worst case, since each cycle C_i must contain at least 2 nodes, when we delete all but one vertex we're left with $\frac{n}{2}$ vertices. As such, in the worst case, our arbitrary tour T has cost at most n (there are $\frac{n}{2}$ edges and each edge has at most cost of 2). We also know that $OPT \geq n$, since we need to traverse at least n edges and the smallest possible cost is 1. Putting this together, we have that $\text{cost}(T) \leq n \leq OPT$.

As such, putting all these arguments together we arrive at:

$$\begin{aligned}
\underbrace{\text{cost}(T)}_{\text{our ATSP tour}} &\leq \underbrace{\text{cost}(C)}_{\text{tour of } G \text{ formed from patched cycle cover}} && \text{(By triangle inequality due to shortcutting)} \\
&= \underbrace{\text{cost}(C_1 \cup \cdots \cup C_k)}_{\text{cost of cycle cover}} + \underbrace{\text{cost(A)}}_{\text{cost of patching}} && \text{(By construction)} \\
&\leq \frac{OPT}{2} + OPT && \text{(As argued above)} \\
&= \frac{3}{2}OPT
\end{aligned}$$

As such, we conclude our arugment that this is a $\frac{3}{2}$-approximation algorithm for the $\{1, 2\}$ special case of the ATSP problem. □