1

1.1 Original integer program

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e\in E} w_e x_e \\
\text{subject to} \quad & \sum_{e\in E} x_e = |V| - 1 \\
& \sum_{e\in\delta(S)} x_e \geq 1 \forall \emptyset \subsetneq S \subsetneq V \\
& \sum_{e\in\delta(u)} x_e \leq k_u \forall u \in V \\
& x_e \in \{0,1\} \forall e \in E
\end{aligned}
\tag{IP}
$$

1.2 Relaxed integer program

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e\in E} w_e x_e + \sum_{u\in V} \lambda_u (k_u - \sum_{e\in\delta(u)} x_e) \\
\text{subject to} \quad & \sum_{e\in E} x_e = |V| - 1 \\
& \sum_{e\in\delta(S)} x_e \geq 1 \forall \emptyset \subsetneq S \subsetneq V \\
& x_e \in \{0,1\} \forall e \in E
\end{aligned}
\tag{LRIP}
$$

1.3 Solve $LRIP$ in poly-time

Denote the objective function of $LRIP$ as

$$t(\vec{\lambda}) := \sum_{e:=(u,v)\in T} (v_e - \lambda_u - \lambda_v) + \sum_{u\in V} \lambda_u k_u$$

where T is a spanning tree of G. The constraints of $LRIP$ are the same as the minimum spanning tree problem. Therefore, for a given $\vec{\lambda}$, we could use Kruskal's algorithm to solve the MST with edge weights $w'_{(u,v)} := w_{(u,v)} - \lambda_u - \lambda_v | \forall (u,v) \in E$ in polynomial time. Upon fine tuning of hyperparmeters of a steepest (sub) gradient ascent method, we can find $t(\vec{\lambda^*}) = \max(t(\vec{\lambda}))$ given a minimum spanning tree in a polynomial time. Such $t(\vec{\lambda^*})$ is the minimum of $LRIP$. Thus the overall algorithm for solving $LRIP$ is in poly-time.

2

2.1

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j\in[m]}\sum_{i\in[n]} x_{ij}v_{ij} \\
\text{subject to} \quad & \sum_{j\in[m]}\sum_{i\in[n]} x_{ij}c_{ij} \le b \\
& \sum_{i\in[n]} x_{ij} \le 1 \forall j \in [m] \\
& \sum_{j\in[m]} x_{ij} \le 1 \forall i \in [n] \\
& x_{ij} \in \{0,1\} \forall i \in [n], j \in [m]
\end{aligned} \tag{IP}
$$

2.2

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j\in[m]}\sum_{i\in[n]} x_{ij}v_{ij} \\
& + \sum_{j\in[m]} (\lambda_j(1 - \sum_{i\in[n]} x_{ij})) \\
& + \sum_{i\in[n]} (\mu_j(1 - \sum_{j\in[m]} x_{ij})) \\
\text{subject to} \quad & \sum_{j\in[m]}\sum_{i\in[n]} x_{ij}c_{ij} \le b \\
& x_{ij} \in \{0,1\} \forall i \in [n], j \in [m]
\end{aligned} \tag{LRIP1}
$$

$LRIP1$ can be regarded as a 0-1 knapsack problem where b is the weight capacity of the bag c_{ij} as item weights and $v_{ij} - \lambda_i - \mu_j$ as item values. This is a well known NP-hard problem thus there's no known polynomial time algorithm to solve it so far.

2.3

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j\in[m]}\sum_{i\in[n]} x_{ij}v_{ij} + \lambda(b - \sum_{j\in[m]}\sum_{i\in[n]} x_{ij}c_{ij}) \\
\text{subject to} \quad & \sum_{i\in[n]} x_{ij} \le 1 \forall j \in [m] \\
& \sum_{j\in[m]} x_{ij} \le 1 \forall i \in [n] \\
& x_{ij} \in \{0,1\} \forall i \in [n], j \in [m]
\end{aligned} \tag{LRIP2}
$$

$LRIP2$ is a max weighted matching problem for a bipartite graph where there are two partitions(one for states and the other for campaigns). Edges between two partitions are weighted as $v_{ij} - \lambda c_{ij}$. It is solvable in polynomial time by adding a source and a sink to the bipartite graph and connecting all states to the source and all campaigns to the sink. The flow capacities between two partitions are $v_{ij} - \lambda c_{ij}$. Now it becomes a max-flow/min-cut problem which is solvable in poly-time by

Ford-Fulkerson algorithm for a given λ. Then we call the steepest gradient descent oracle polynomial times to get the optimal value of the objective function of λ.

3

3.1

$$f(C) := \sum_{i\in[n]} \min{(k_i, \sum_{E_j\in C} x_{E_j})} \text{ where } x_{E_j} = 1 \text{ if } i \in E_j, x_{E_j} = 0 \text{ otherwise}$$

3.2

Suppose that $A \subseteq B \subseteq \mathcal{U}$ and we define $D = B \setminus A$. Then we have

$$\begin{aligned}
f(B) &= \sum_{i\in[n]} \min{(k_i, \sum_{E_j\in B} x_{E_j})} \\
&= \sum_{\substack{i\in[n] \\ k_i \le \sum_{E_j\in A} x_{E_j}}} k_i + \sum_{\substack{i\in[n] \\ k_i > \sum_{E_j\in A} x_{E_j} \\ E_j \notin D}} \sum_{E_j\in A} x_{E_j} + \sum_{\substack{i\in[n] \\ k_i > \sum_{E_j\in A} x_{E_j} \\ E_j \in D}} (\sum_{E_j\in A} x_{E_j} + \sum_{E_j\in D} x_{E_j}) \\
&= \sum_{i\in[n]} \min{(k_i, \sum_{E_j\in A} x_{E_j})} + \sum_{\substack{i\in[n] \\ k_i > \sum_{E_j\in A} x_{E_j} \\ E_j \in D}} (\sum_{E_j\in A} x_{E_j} + \sum_{E_j\in D} x_{E_j}) \\
&= f(A) + \sum_{\substack{i\in[n] \\ k_i > \sum_{E_j\in A} x_{E_j} \\ E_j \in D}} (\sum_{E_j\in A} x_{E_j} + \sum_{E_j\in D} x_{E_j}) \\
&\ge f(A) \text{ since } x_{E_j} \in \{0, 1\}
\end{aligned}$$

This can be understood in such a way: if there's some E_j not in A but in B, those students i attending in E_j do not contribute to the total count if the count is already cap'ed at k_i. But they increase those counts are not saturated. Therefore the overall count increases or at least the same.
So f is monotone non-decreasing.

3.3

Suppose that $A \subseteq B$ and $x \in \mathcal{U} \setminus B$.

$$f(A + x) - f(A) = \sum_{\substack{i\in[n] \\ k_i > \sum_{E_j\in A} x_{E_j} \\ E_j \in x}} (\sum_{E_j\in A} x_{E_j} + \sum_{E_j\in x} x_{E_j})$$

$$f(B + x) - f(B) = \sum_{\substack{i\in[n] \\ k_i > \sum_{E_j\in B} x_{E_j} \\ E_j \in x}} (\sum_{E_j\in B} x_{E_j} + \sum_{E_j\in x} x_{E_j})$$

Since $A \subseteq B$,

$$f(A+x) - f(A) - f(B+x) + f(B) = \sum_{\substack{i\in[n] \\ k_i > \sum_{E_j \in A} x_{E_j} \\ E_j \notin B}} \left(\sum_{E_j \in A} x_{E_j} \right)$$
$$\geq 0 \text{ since } x_{E_j} \in \{0,1\}$$

The overall count increases more as there's less saturated k_i in A than B before we add x(attending some event E_j).
So f is submodular.

4

4.1

$\mathcal{U} = V =$ vertices = the set of students

4.2

$f(C) :=$ the cardinality (size) of the set of reachable vertices from $C = |\{v \in V \setminus C | \exists (u,v) \in E : u \in C\}|$

4.3

Suppose that $A \subseteq B \subseteq \mathcal{U} = V$.
$f(B) - f(A) = |\{v \in V \setminus B | \exists (u,v) \in E : u \in B \setminus A \text{ and } \nexists (u,v) \in E : u \in A\}|$. In plain english, the size of the set containing those nodes reachable from nodes in $B \setminus A$ but not reachable from nodes in A. As a property of a norm, this size is non negative. $\Rightarrow f(B) - f(A) \geq 0$ so f is monotone non decreasing.

4.4

Suppose that $A \subseteq B$ and $x \in \mathcal{U} \setminus B$.

$$\begin{aligned} & f(A+x) - f(A) - f(B+x) + f(B) = \\ & \quad |\{v \in V \setminus (A+x) | \exists (u,v) \in E : u \in x \text{ and } \nexists (u,v) \in E : u \in A\} := \mathbf{A}'| \\ & \quad - \\ & \quad |\{v \in V \setminus (B+x) | \exists (u,v) \in E : u \in x \text{ and } \nexists (u,v) \in E : u \in B\} := \mathbf{B}'| \end{aligned}$$

Such $\mathbf{B}' \subseteq \mathbf{A}'$ because $A \subseteq B \Rightarrow |A'| - |B'| \geq 0$ (you can also think of the complement sets of A and B. And $f(\bar{B}) - f(\bar{A}) = f(A) - f(B)$ Otherwise, the graph is directional.)
$\Rightarrow f(A+x) - f(A) - f(B+x) + f(B) \geq 0 \Rightarrow f$ is submodular.