CMPUT 675 - Assignment #3
Due Oct 28 by 1pm

Fall 2016, University of Alberta

Pages: 4

Note: For this assignment you may assume linear programs can be solved in polynomial time.

Exercise 1)

Marks: 5

- Let $\mathbf{A} \in \{0,1\}^{m \times n}$ be such that in every column, all of the 1s in that column appear consecutively. For example:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

  In 2 different ways, show any such matrix $\mathbf{A}$ is totally unimodular. In particular,

  1. Show it is a network matrix
  2. Use the Ghouila-Houri theorem.

- In the Interval Covering problem, we have n intervals of the form $[s_i, t_i]$ where $s_i \leq t_i$ for each $1 \leq i \leq n$ and each interval has a coverage requirement $r_i \in \mathbb{Z}_{\geq 0}$.

  Also suppose we have finite set of points $T \subseteq \mathbb{R}$ with, say, $|T| = m$. For each $t \in T$, there is a cost $c_t \in \mathbb{R}_{\geq 0}$.

  We want to find a cheapest subset S of T that ensures $|S \cap [s_i, t_i]| \geq r_i$ for each $1 \leq i \leq n$ (or determine no such subset exists). Show how to do this in polynomial time.

- Suppose $r_i = 1$ for all $1 \leq i \leq n$ and $c_t = 1$ for all $t \in T$. Show how to solve such an Interval Covering instance in $O(n \log n + m \log m)$ time.
  Hint: apart from some sorting the rest can be done in linear time.

Exercise 2)

Marks: 5

Let X be a finite set of n elements. A **laminar family over** X is a collection $\mathcal{L} \subseteq 2^X$ of nonempty subsets of X satisfying the following property: for any two $A, B \in \mathcal{L}$ either $A \subseteq B, B \subseteq A$ or $A \cap B = \emptyset$.

- Show $|\mathcal{L}| \leq 2|X| - 1$ for any laminar family $\mathcal{L}$ over X.
  Note: this is not needed in the remaining parts of this exercise, but we will need it later in the course so might as well state and prove it now.
- Consider the MAXIMUM-VALUE LAMINAR-CONSTRAINED b-MATCHING problem.

  We are given a bipartite graph $\mathcal{G} = (L \cup R; E)$ and laminar families $\mathcal{L}_L, \mathcal{L}_R$ over L, R (respectively). For each $X \in \mathcal{L}_L$ we are given $b_X \in \mathbb{Z}_{\geq 0}$ and for each $Y \in \mathcal{L}_R$ we are given $b_Y \in \mathbb{Z}_{\geq 0}$. Finally, for each $e \in E$ we have a value $v_e \in \mathbb{R}_0$.

  Consider the problem of finding the set $F \subseteq E$ of maximum value subject to the constraint for any $X \in \mathcal{L}_L$ we have $|\delta(X) \cap F| \leq b_X$ and for any $Y \in \mathcal{L}_R$ we have $|\delta(Y) \cap F| \leq b_Y$.

  Briefly mention how this generalizes the problem of finding a maximum-value matching in a bipartite graph.
- Show how to express the MAXIMUM-VALUE LAMINAR-CONSTRAINED b-MATCHING problem as a linear program with a totally unimodular constraint matrix.

Exercise 3)

Marks: 8

For this exercise, fix $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}\}$ and suppose $\mathcal{P} \subseteq [0,1]^n$ (i.e. every coordinate is between 0 and 1 in any feasible solution).

- Show every point in $\mathcal{P} \cap \mathbb{Z}^n$ is an extreme point of $\mathcal{P}$.
- Call a set $\mathcal{C} \subseteq \mathbb{R}^n$ **convex** if for any $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ and any $0 \leq \lambda \leq 1$ we have $\lambda \cdot \mathbf{x} + (1 - \lambda) \cdot \mathbf{y} \in \mathcal{C}$.

  Show $\mathcal{P}$ is convex. Note, this does not need the assumption $\mathcal{P} \subseteq [0,1]^n$.
- Let $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k} \in \mathbb{R}^n$. A **convex combination** of these points is a point $\mathbf{p}$ of the form

  $$\mathbf{p} = \sum_{i=1}^{n} \lambda_i \cdot \mathbf{x_i}$$

  where the λ_i are nonnegative values such that $\sum_{i=1}^{n} \lambda_i = 1$.

  Show that any $\overline{\mathbf{x}} \in \mathcal{P}$ is a convex combination of the extreme points of $\mathcal{P}$.
  Hint: use induction on $\text{rank}(\mathbf{A}^{\overline{\mathbf{x}}})$. This also does not require $\mathcal{P} \subseteq [0,1]^n$.
- Show if any $\overline{\mathbf{x}} \in \mathcal{P}$ can be written as a convex combination of points in $\mathcal{P} \cap \mathbb{Z}^n$, then every extreme point of $\mathcal{P}$ is integral.

Bonus (0.5 mark)
Show that in part 3, we can write any $\overline{\mathbf{x}} \in \mathcal{P}$ as a convex combination of only $n - \text{rank}(\mathbf{A}^{\overline{\mathbf{x}}}) + 1$ extreme points of $\mathcal{P}$.

Exercise 4)

Marks: 5

In the DEMAND-ROUTING IN A TREE problem, we are given a tree $\mathcal{T} = (V; E)$ and a collection of paths $D = \{P_1, P_2, \ldots, P_k\}$ in $\mathcal{T}$. A subset $D' \subseteq D$ is said to be **routable** if no edge $e \in E$ lies on more than one path in D'.

Give a polynomial-time algorithm for finding a routable subset $D' \subseteq D$ of maximum size. For simplicity, you may assume each path P_i has both endpoints being leaves in $\mathcal{T}$ and that each leaf is the endpoint of precisely one path (this is mostly without loss of generality, there is a simple reduction from the general problem to this more structured one that increases the problem size by a small amount).

I will allow you to use the fact that MAXIMUM-WEIGHT MATCHING in a general graph can be solved in polynomial time. We will eventually see this.

Hint: Imagine the tree is rooted at some arbitrarily chosen vertex r. Use dynamic programming over subtrees.

Bonus (0.5 marks)
Devise an algorithm that does not rely on MAXIMUM-WEIGHT MATCHING. It can still use the algorithm for unweighted matching we saw in the lectures.

Exercise 5)

Marks: 3

Let $\mathcal{G} = (V; E)$ be a graph. Consider the following polyhedron.

$$\mathcal{P} = \{\mathbf{x}_e \in \mathbb{R}^E_{\geq 0} : \mathbf{x}(\delta(v)) \leq 1 \text{ for each } v \in V\}.$$

We saw that if $\mathcal{G}$ is bipartite then all extreme points of $\mathcal{P}$ are integral. You must show the converse. Show that if $\mathcal{G}$ is not bipartite then some extreme point $\overline{\mathbf{x}} \in \mathcal{P}$ has $\overline{\mathbf{x}} \notin \mathbb{Z}^E$.

Exercise 6)

Marks: 5

Find a maximum matching in the following graph $\mathcal{G} = (V; E)$. A matching M is already given with bold edges. You should follow the algorithm from class to find an alternating path. Moreover, you must follow these steps. In each iteration when you are trying to find an M-alternating path:

- Find a maximal M-alternating forest.
- If there is an edge between even height nodes in the same component, contract that blossom. If there are many, choose one that creates the smallest blossom (I'm forcing this order so you see the effects of contracting blossoms very clearly). Restart with the new contracted graph.
- If there is an edge between even height nodes in different components, alternate the matching along the corresponding path. Do this only if no blossoms can be found in the previous step.

If you find an alternating path in the graph that has some blossoms contracted, show every step of expanding the blossoms one-by-one in reverse order they were contracted. After each expansion, show how to adjust the matching from the graph with the contracted blossom to the graph with the graph with this blossom expanded.

Finally, after you increase the size of the matching in the graph you should restart the whole procedure with the larger matching. Stop when you finally have reached an M-alternating forest for some graph with contracted blossoms that has no edge between even-height vertices.

Once you have a maximum matching M^*, demonstrate a set of vertices $X \subseteq V$ such that $|M^*| = \frac{|V| - q_{\mathcal{G}}(X) + |X|}{2}$.