This assignment is **due on Oct 5** and should be submitted on Gradescope. All submitted work must be *done individually* without consulting someone else's solutions in accordance with the University's "Academic Dishonesty and Plagiarism" policies. You may like to refer to the attached "Advice on how to do the assignment".

Problem 1. (10 points) We consider a variant of the bipartite vertex cover problem. In this problem we are given a **bipartite** graph $G(V, E)$. We must choose a set of vertices $V^* \subseteq V$ and a set of edges $E^* \subseteq E$ such that for every edge $(u, v) \in E$ we have at least one of: $u \in V^*$, $v \in V^*$ or $(u, v) \in E^*$. The cost of our solution is $|V^*| + \rho|E^*|$ for some fixed constant $\rho \in (0, 1)$, we aim to minimize this cost.

a) Formulate this problem as an integer program

b) Show the constraint matrix of your problem is totally unimodular.

c) Now let's generalise the problem. For each edge e we now have two parameters $k_e^+, k_e^- \in \{0, 1, 2\}$ with $k_e^- \leq k_e^+$. Instead of at least one of: $u \in V^*$, $v \in V^*$ or $(u, v) \in E^*$ for each $(u, v) \in E$ we will require at least $k_{(u,v)}^-$ and at most $k_{(u,v)}^+$ of the options hold. Write down this generalised problem as an integer program.

d) Show the constraint matrix of the generalised problem is totally unimodular.

Solution 1.

a) This IP is very close to the vertex cover IP.

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{u\in V} x_u + \rho \sum_{(u,v)\in E} y_{(u,v)} & \\ \text{subject to} & x_u + x_v + y_{(u,v)} \geq 1 & \forall (u,v) \in E \\ & x_u \in \{0,1\} & \forall u \in V \\ & y_{(u,v)} \in \{0,1\} & \forall (u,v) \in E \end{array}$$

b) Let A be a $|E| \times |V|$ matrix, with rows indexed by E and columns indexed by V, where $a_{e,u} = 1$ if edge e is incident on vertex u and $a_{e,u} = 0$ otherwise. Then input graph G is bipartite; we color each column of A blue if its vertex belongs to one shore of the bipartition and red if its vertex belongs to the other shore. Notice that each row of A corresponding to (u,v) has exactly two non-zero elements, a 1 at the columns for vertex u and vertex v. Because each edge goes from one shore to the other, each row has exactly one red 1 and one blue 1. Therefore no matter what column-induced submatrix we take no row can have more than one 1 of either colour, so this bi-coloring is equitable for all column-induced submatrices. Hence A is totally unimodular.
From Section 6.2 of the lecture notes if A is totally unimodular then so is $B := \begin{bmatrix} A & I \end{bmatrix}$. But B is exactly the constraint matrix of this problem. Hence this problem has a totally unimodular constraint matrix.

c)

$$\begin{array}{lll} \text{minimize} & \sum_{u\in V} x_u + \rho \sum_{(u,v)\in E} y_{(u,v)} & \\ \text{subject to} & x_u + x_v + y_{(u,v)} \geq k^-_{(u,v)} & \forall (u,v) \in E \\ & -x_u - x_v - y_{(u,v)} \geq -k^+_{(u,v)} & \forall (u,v) \in E \\ & x_u \in \{0,1\} & \forall u \in V \\ & y_{(u,v)} \in \{0,1\} & \forall (u,v) \in E \end{array}$$

d) Defining A as in the previous section, we observe that the constraint matrix of this problem is $C := \begin{bmatrix} A & I \\ -A & -I \end{bmatrix}$. We already know A is totally unimodular. From Section 6.2 we therefore know that $\begin{bmatrix} A & I \end{bmatrix}$ is totally unimodular, and therefore, applying another result from Section 6.2, we know $\begin{bmatrix} A & I \\ -A & -I \end{bmatrix}$ is totally unimodular.

Problem 2. (10 points) The Advanced Consortium of Multinational Enterprises (ACME) must select a representative committee to attend its annual conference. There are n people eligible to attend the conference; each person $i \in [n]$ has a *quality* q_i that measures how suitable they are to attend the conference, a higher quality is better. However, the conference centre only has accommodation for k people.

ACME is a consortium of m_1 businesses $\{B_1, B_2, \ldots, B_{m_1}\}$ that operate across m_2 countries $\{C_1, C_2, \ldots, C_{m_2}\}$. Each business could operate in many countries. Each person works for exactly one business and lives in exactly one country. We will overload notation slightly and call the set of people who work for business i B_i and the set of people who live in country j C_j. In the interests of fairness ACME leadership have allocated a budget $b_B(i)$ to every business B_i and a budget $b_C(j)$ to every country C_j.

You are in charge of selecting delegates to attend the ACME annual conference. You must select a committee of no more than k people which maximizes the total quality of the committee. However you may not select more than $b_B(i)$ people who work for business i and you may not select more than $b_C(j)$ people who live in country j, for all i and j.

a) Write down an integer program formulation of this problem.

b) Prove that this problem *is not* a matroid.

c) Prove that the constraint matrix for this problem is totally unimodular.

d) You now need to solve the same problem for the annual conference of the Basic Consortium of Mononational Enterprises (BCME). The only difference between ACME and BCME is each company that participates in the BCME annual conference operates in only one country. That is for each $i \in [m_1]$ there is a $j \in [m_2]$ such that $B_i \subseteq C_j$. Prove that this problem *is* a matroid.

Solution 2.

a)

$$\begin{aligned}
\text{maximize} \quad & \sum_{p=1}^{n} x_p v_p \\
\text{subject to} \quad & \sum_{p=1}^{n} x_p \le k \\
& \sum_{p \in B_i} x_p \le b_B(i) && \forall i \in [m_1] \\
& \sum_{p \in C_i} x_p \le b_C(j) && \forall j \in [m_2] \\
& x_p \in \{0,1\} && \forall p \in [n]
\end{aligned}$$

b) The following counter-example shows the problem is not a matroid. We have three candidates: Amélie, Bernhard and Clara. Amélie lives in France, while Bernhard and Clara live in Germany. Amélie and Bernhard work for Alpha Corporation, while Clara works for Beta Group. The budgets for France, Germany, Alpha Corporation and Beta Group are all 1. Let $S = \{\text{Bernhard}\}$ and $T = \{\text{Amélie}, \text{Clara}\}$. Both S and T are feasible and $|S| < |T|$. However neither Amélie nor Clara can be added to S without making it infeasible. Hence this problem is not a matroid. See Figure 1 for an illustration.

c) Observe the constraint matrix has one row for the committee size, a row for each business and each country, and a column for each person. Each column contains all zeroes except for exactly three ones: in the first row, the row corresponding to that person's business and the row corresponding to that person's country. We will color the rows rather than the columns, i.e. prove that the transpose of the constraint matrix is totally unimodular; which proves that the constraint matrix itself is totally unimodular. We will need to handle two separate cases, if the first (committee size) row is in the row-induced submatrix or not.
If it is not, we color all the business rows red, and all the country rows blue. Hence every column has one or zero red ones, one or zero blue ones and the rest 0. Hence the number of red ones and the number of blue ones in any column of this submatrix matrix differ by at most one.
If the first row is included in the submatrix we color it red, and color all other rows blue. Then each column contains one red one, and between zero and two blue ones with the other entries zero. Hence the number of red ones and the number of blue ones in any column of this submatrix matrix differ by at most one.
Hence every row-induced submatrix is equitably bi-colorable and thus the constraint matrix is totally unimodular.

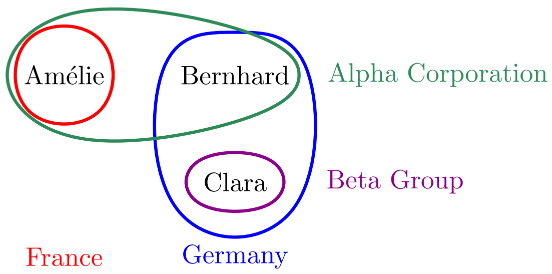
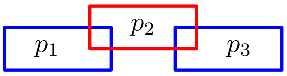
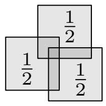


Figure 1: Problem 2 Part b Counterexample

d) Let S and T be arbitrary feasible subsets in the BCME problem such that $|S| < |T|$. We will prove that there exists $e \in T \setminus S$ such that $S + e$ is feasible.
For each country j we consider $S \cap C_j$ and $T \cap C_j$. Because $|S| < |T|$ and every person belongs to a country, there must be some country $C_{j'}$ such that $|S \cap C_{j'}| < |T \cap C_{j'}|$.
We then repeat this process, for every business i with $B_i \subseteq C_{j'}$ we compare $S \cap C_{j'} \cap B_i = S \cap B_i$ and $T \cap C_{j'} \cap B_i = T \cap B_i$. Because $|S \cap C_{j'}| < |T \cap C_{j'}|$ and every person belongs to a business there must be some business $B_{i'} \subseteq C_{j'}$ such that $|S \cap B_{i'}| < |T \cap B_{i'}|$.
Therefore there is a person $e \in B_{i'}$ with $e \in T$ but $e \notin S$. Since $B_{i'} \subseteq C_{j'}$ we know $e \in C_{j'}$. Then $|(S + e) \cap B_{i'}| = |S \cap B_{i'}| + 1 \leq |T \cap B_{i'}| \leq b_B(i')$, where the first inequality follows from $|S \cap B_{i'}| < |T \cap B_{i'}|$ and the fact that set cardinalities are integers, while the second inequality follows because T is feasible. Similarly $|(S + e) \cap C_{j'}| = |S \cap C_{j'}| + 1 \leq |T \cap C_{j'}| \leq b_C(j')$. Hence $S + e$ is feasible and we have proved the matroid exchange axiom.



a) Counterexample



b) Counterexample

Figure 2: Problem 3 Counterexamples

Problem 3. (10 points) The Greenfields Shire Council has decided to sell some of its unused land to raise revenue. The Greenfields surveyor has considered the council land $[0, C]^2$ (for some large C) and proposed n paddocks $\{p_1, p_2, \ldots, p_n\}$ that would be suitable to sell. Each paddock $p_i \subseteq [0, C]^2$ is a rectangular region that is not necessarily axis-aligned. The paddocks are not necessarily disjoint, and do not necessarily cover the entire council land. The surveyor has determined that paddock p_i could be sold for \$ v_i. It is not possible to sell two intersecting paddocks, as this would mean selling a piece of land twice. The council wants to maximize the profit they make from the sales so have come up with the following IP formulation of the problem:

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{i=1}^{n} v_i x_i & \\ \text{subject to} & x_i + x_j \leq 1 & \forall i, j \in [n] : p_i \cap p_j \neq \emptyset \\ & x_i \in \{0, 1\} & \forall i \in [n] \end{array}$$

a) Prove that this problem *is not* a matroid.
b) Prove that this formulation does *not* have an integral LP relaxation.
c) Write down an alternative IP formulation for this problem that has a *stronger* LP relaxation.
d) Prove that your proposed IP is valid and prove that its relaxation is indeed (strictly) stronger.

Solution 3.

a) Consider $n = 3$ paddocks p_1, p_2 and p_3 with $p_1 \cap p_2 \neq \emptyset$, $p_2 \cap p3 \neq \emptyset$ but $p_1 \cap p_3 = \emptyset$. Let $S = \{p_2\}$ and $T = \{p_1, p_3\}$. Then S and T are both feasible sets with $|S| < |T|$ but neither p_1 nor p_2 can be added to S without making it infeasible. See Figure 2.a.

b) Consider $n = 3$ paddocks such that each paddock intersects the other two. Let the value of each paddock be 1. The vector $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ is a feasible solution to the LP relaxation with value of 1.5. However it is clearly not possible to choose more than 1 paddock so the optimal integral solution has value 1. Since there is a fractional solution with value greater than any integral solution this LP relaxation cannot be integral. See Figure 2.b.

c) We will add an additional constraint requiring all 3-wise intersections to not add to more than 1.

$$
\begin{array}{lrl}
\text{maximize} & \displaystyle\sum_{i=1}^{n} v_i x_i & \\
\text{subject to} & x_i + x_j \leq 1 & \forall i, j \in [n] : p_i \cap p_j \neq \emptyset \\
& x_i + x_j + x_k \leq 1 & \forall i, j, k \in [n] : p_i \cap p_j \cap p_k \neq \emptyset \\
& x_i \in \{0, 1\} & \forall i \in [n]
\end{array}
$$

d) First we will show this formulation is valid, i.e. the set of integral solutions is the same. Each vector $x \in \{0,1\}^n$ is associated with a set of paddocks P_x where $p_i \in P_x \Leftrightarrow x_i = 1$. Every set of paddocks can be associated with an $x \in \{0,1\}^n$ in this way. The objective value of our IP, $v \cdot x$, is exactly the sum of the values of the paddocks in P_x, as intended. It remains to show the sets of feasible solutions are the same. If x is a feasible solution to the IP then we have at most one of x_i and x_j is 1 for all intersecting paddocks p_i and p_j. This ensures P_x does not contain any intersecting paddocks and so is feasible. If P_x is a feasible set of paddocks, and $p_i \cap p_j \neq \emptyset$ then at most one of p_i and p_j can be in P_x. This ensures only one of x_i and x_j can be 1, which ensures that both sets of constraints are satisfied.
Our LP relaxation has a superset of the constraints of the original LP relaxation, therefore any solution to our LP relaxation is also a solution to the original LP relaxation. Furthermore our counter-example given in part b) is feasible for the original LP relaxation but not for our LP relaxation. Hence our formulation has a strictly stronger LP relaxation.

Problem 4. (10 points) We study the problem of decomposing a given graph $G(V, E)$ into a small number of spanning trees. Let $\mathcal{T} = \{T \subseteq E : T \text{ is a spanning tree of } G\}$. We want to choose a small number of trees from $\mathcal{T}$ such that every edge is in exactly one tree we have chosen. This problem has a (very large) IP with the following LP relaxation.

$$\begin{aligned} \text{minimize} \quad & \sum_{T \in \mathcal{T}} x_T & \\ \text{subject to} \quad & \sum_{T \in \mathcal{T} : e \in T} x_T = 1 & \forall e \in E \\ & x_T \geq 0 & \forall T \in \mathcal{T} \end{aligned}$$

a) Suppose we are running simplex and need to choose an index to come into the basis. Show how to find an index T that minimizes reduced cost.

b) Write down the dual of this linear program.

c) Design a polynomial-time separation oracle for the dual program.

Solution 4.

a) The reduced cost of some variable indexed by $T \in \mathcal{T}$ is $c_T - c_B^T A_B^{-1} A_T$. Notice $c_T = 1$ and define $y^T = c_B^T A_B^T$. Our problem has $|E|$ constraints, so $|B| = |E|$ and y^T can be computed in polynomial time. The vector A_T has a coordinate for each $e \in E$ with $(A_T)_e = 1$ if $e \in T$ and $(A_T)_e = 0$ otherwise. Therefore $c_T - c_B^T A_B^{-1} A_T = 1 - y^T A_T = 1 - \sum_{e \in T} y_e$. Finding the T that minimizes reduced cost is therefore a matter of finding the maximum spanning tree with weights y_e.
Some canny students pointed out that every feasible solution of this LP has the same cost, so the reduced cost vector will always be non-negative. This question will be marked more generously in light of this potential confusion.

b)

$$
\begin{aligned}
\text{maximize} \quad & \sum_{e \in E} y_e \\
\text{subject to} \quad & \sum_{e \in T} y_e \leq 1 && \forall T \in \mathcal{T} \\
& y_e \ \text{ free} && \forall e \in E
\end{aligned}
$$

c) The separation oracle for this LP is the maximum spanning tree problem. Given a proposed solution y_e for $e \in E$ we compute the maximum spanning tree problem on the graph with edge weights y_e. If the returned tree T has weight greater than 1 we have a T such that $\sum_{e \in T} y_e > 1$ which gives us a violated constraint. If the maximum spanning tree has weight less than or equal to 1 then $\sum_{e \in T} y_e \leq 1$ for all $T \in \mathcal{T}$ so this y is feasible.

Advice on how to do the assignment

- Assignments should be typed and submitted as pdf (no pdf containing text as images, no handwriting).
- Start by typing your student ID at the top of the first page of your submission. Do not type your name.
- Submit only your answers to the questions. Do not copy the questions.
- Be careful with giving multiple or alternative answers. If you give multiple answers, then we will give you marks only for "your worst answer", as this indicates how well you understood the question.
- You can use the material presented in the lecture slides or lecture notes without proving it. You do not need to write more than necessary.
- When giving answers to questions, always prove/explain/motivate your answers.
- When giving an algorithm as an answer, the algorithm does not have to be given as (pseudo-)code.
- If you do give (pseudo-)code, then you still have to explain your code and your ideas in plain English.
- Unless otherwise stated, we always ask about worst-case analysis, worst case running times, etc.
- If you use further resources (books, scientific papers, the internet, ...) to formulate your answers, then add references to your sources and explain it in your own words. Only citing a source doesn't show your understanding and will thus get you very few (if any) marks. Copying from any source without reference is plagiarism.
- Finally, to make marking run more smoothly, please specify in with your Gradescope submission, which pages in your submission cover which problem.