Homework #5
CS675 Fall 2026

Due Wednesday Nov 18, by 5pm

General Instructions The following assignment is meant to be challenging. Feel free to discuss with fellow students, though please write up your solutions independently and acknowledge everyone you discussed the homework with on your writeup. I also expect that you will not attempt to consult outside sources, including generative AI models, for solutions to any of these homework problems — doing so would be considered cheating. Please see the course website for details on the academic integrity policy.

We request that you submit your homework as a pdf file, by email to the TA.

Finally, whenever a question asks you to "show" or "prove" a claim, please provide a formal mathematical proof.

Problem 1. (16 points)
In this problem, we will show how to implement Caratheodory's theorem efficiently, assuming only that the relevant extreme points form a solvable polytope. Formally, let $\mathcal{P} = \{x : Ax \preceq b\} \subseteq \mathbb{R}^n$ be a polytope given implicitly by a *separation oracle*: a procedure which takes as input $y \in \mathbb{R}^n$, and either certifies $y \in \mathcal{P}$ or else returns one of the constraints $Ax \preceq b$ which is violated at $x = y$. In the *contructive Caratheodory problem* for $\mathcal{P}$, we are given a point $y \in \mathbb{R}^n$, and we must output a representation of y as a convex combination of at most $n + 1$ vertices of $\mathcal{P}$, or else declare that $y \notin \mathcal{P}$. You may assume that $\mathcal{P}$ is full dimensional if this simplifies your proofs, though this is not necessary.

(a) [4 points]. Given $y \in \mathcal{P}$, show that there exist two points y_1 and y_2 on the boundary of $\mathcal{P}$, and $\alpha \in [0, 1]$, such that $y = \alpha y_1 + (1 - \alpha)y_2$. Show how to compute such a y_1, y_2, and α. The runtime of your algorithm should $\text{poly}(n, B, \langle y \rangle)$, plus $\text{poly}(n, B, \langle y \rangle)$ calls to the separation oracle, where B is an upperbound on the bit complexity of individual entries of A and b.

(b) [4 points]. Building on your solution to part (a), argue that you can also efficiently identify a constraint from the system $Ax \preceq b$ which is tight at y_1 (and similarly for y_2). The runtime of your algorithm should be as specified for part (a).
(**Hint:** When $y_1 \neq y_2$, you can compute a point y_1' just past y_1 on the line connecting y_2 to y_1 such that invoking the separation oracle for y_1' returns a constraint which is tight at y_1. When $y_1 = y_2$ a somewhat similar trick works. There are tedious numerical issues here which you are permitted to discuss informally.)

(c) [8 points]. Building on your solution to (a) and (b), describe a recursive algorithm which solves

the constructive Caratheodory problem. The runtime of your algorithm should be as specified for part (a).

Problem 2. (16 points)
In class, we stated a quite powerful solvability result for implicitly-described linear programs. Specifically, given access to a separation oracle for the set $P = \{x \in \mathbb{R}^n : Ax \preceq b\}$, and the guarantee that every number occurring in $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$ is a rational number with at most B bits in its binary representation, an optimal solution to the linear program $\max_{x \in P} c^\intercal x$ can be computed with $\text{poly}(n, B)$ overhead in runtime. Crucially, the runtime guarantee is independent of m, the number of constraints of the (hidden) linear program, which may be very large (say, exponential or doubly exponential in n). In this problem, you will prove that a solution to a (normalized) dual linear program can be obtained as well, despite the fact that the number of variables in the dual LP is very large.

(a) [4 points]. Solving the dual LP is hopeless if we can't express the solution succinctly. Given a primal LP of the form described above, write down the dual LP, and explain how an optimal dual solution may be represented in $\text{poly}(n, B)$ bits.

(b) [4 points]. Explain why, in general, given a linear program of the form $\max\{c^\intercal x : Ax \preceq b\}$ with the feasible set given by an arbitrary separation oracle, you could never hope to obtain the optimal dual solution, or any feasible dual solution for that matter.

(c) [8 points]. In light of (b), we will instead solve the dual of a "normalized" primal LP. Assume that $b \succ 0$.[1] Now consider the normalized primal LP given by $\max\left\{c^\intercal x : A'x \preceq \vec{1}\right\}$ with $A'_{ij} = A_{ij}/b_i$, and note that it is equivalent to the original primal LP. Under the same assumptions described in the preamble, show how to obtain an optimal solution to the dual of the normalized primal with $\text{poly}(n, B)$ overhead in runtime.
(**Hint:** This should remind you of a problem on a previous homework. Use polars and your construction for Caratheodory's theorem from the previous problem.)

Problem 3. (10 points)
In class, we showed that a simple greedy algorithm maximizes a modular function over the independent sets of a matroid. Consider instead the problem of minimizing a modular function over the bases of a matroid. Show that a similar greedy algorithm succeeds for this problem.
Note: This is a generalization of the minimum spanning tree problem.
Hint: You can either mimic the proof presented in class for maximizing a modular function over the independent sets of a matroid, or for a shorter proof you can use a reduction to that problem!

[1] This is without loss of generality. You could use the ellipsoid method to compute a strictly feasible solution to the primal, then shift the origin so that it is strictly feasible, which would guarantee that $b \succ 0$. However, you do not have to prove any of this for this problem.

Problem 4. (9 points)
Let $\mathcal{X} = \{v_1, \ldots, v_m\}$ where $v_i \in \mathbb{R}^n$ for $i = 1, \ldots, m$. As explained in class, the set system over $\mathcal{X}$ whose feasible sets are the linearly independent subsets of $\mathcal{X}$ is a matroid. In this problem, we will explore whether other forms of independence also yield matroids.

(a) [3 points]. Suppose instead that we define $S \subseteq \mathcal{X}$ to be feasible if the vectors in S are independent with respect to convex combinations — i.e. no vector $v \in S$ can be written as a convex combination of the remaining vectors $S \setminus v$. Is the resulting set system a matroid? Prove or disprove.

(b) [3 points]. What if we define $S \subseteq \mathcal{X}$ to be feasible if the vectors in S are affinely independent — i.e. no vector $v \in S$ can be written as an affine combination of the remaining vectors $S \setminus v$. Is the resulting set system a matroid? Prove or disprove.

(c) [3 points]. Finally, what if we say $S \subseteq \mathcal{X}$ is feasible if the vectors in S are conically independent — i.e. no vector $v \in S$ can be written as a conic combination of the remaining vectors $S \setminus v$. Is the resulting set system a matroid? Prove or disprove.

Problem 5. (20 points)
In this problem, we will examine some miscellaneous properties of matroids. We use $\mathcal{M} = (\mathcal{X}, \mathcal{I})$ to denote an arbitrary matroid.

(a) [4 points]. Prove that the rank function $rank_{\mathcal{M}} : 2^{\mathcal{X}} \to \mathbb{N}$ of a matroid is monotone, normalized, and submodular.

(b) [4 points]. Prove the *strong exchange property* of matroids: If B and B' are bases of matroid $\mathcal{M}$, then for every $i \in B \setminus B'$ there is some $j \in B' \setminus B$ such that $(B \setminus \{i\}) \cup \{j\}$ is a basis of $\mathcal{M}$. Show that the exchange property is implied by the strong exchange property for every downwards-closed set system. (i.e. we could have equivalently defined matroids using the strong exchange property instead).

(c) [4 points]. Show that matroids are closed under *truncation*. Namely, if $\mathcal{M} = (\mathcal{X}, \mathcal{I})$ is a matroid, and k is a nonnegative integer, then the set system $\mathcal{M}_k = (\mathcal{X}, \mathcal{I}_k)$ with $\mathcal{I}_k = \{S \in \mathcal{I} : |S| \leq k\}$ is also a matroid.

(d) [4 points]. Recall that we call a set $C \subseteq \mathcal{X}$ a *circuit* of the matroid $\mathcal{M}$ if it is a minimal dependent set in $\mathcal{M}$. Note that a circuits of a graphical matroid correspond to cycles in the graph. Prove the *circuit property* of matroids: if $\mathcal{M} = (\mathcal{X}, \mathcal{I})$ is a matroid with distinct weights $w \in \mathbb{R}^{\mathcal{X}}$ assigned to its ground set, and C is a circuit of $\mathcal{M}$, then the element $i \in C$ of minimum weight is not in any maximum-weight basis of $\mathcal{M}$.

(e) [4 points]. We call a set $C \subseteq \mathcal{X}$ a *cut* of the matroid $\mathcal{M}$ if every basis of $\mathcal{M}$ intersects C, and no proper subset of C has this property. Note that a cut of a graphical matroid correspond to our notion of a cut in a graph — namely, a family of edges who's removal disconnects some connected component of the graph. Prove the *cut property* of matroids: if $\mathcal{M} = (\mathcal{X}, \mathcal{I})$ is a matroid with

distinct weights $w \in \mathbb{R}^{\mathcal{X}}$ assigned to its ground set, and C is a cut of $\mathcal{M}$, then the element $i \in C$ of maximum weight is in every maximum-weight basis of $\mathcal{M}$.

Problem 6. (15 points)
Given two matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ with a common ground set $\mathcal{X}$, and a weight vector $w \in \mathbb{R}^{\mathcal{X}}$, the *maximum-weight common independent set* of $\mathcal{M}_1$ and $\mathcal{M}_2$ is the set $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $\sum_{i \in S} w(i)$. We discussed in class how the maximum-weight common independent set can be computed in time poly($|\mathcal{X}|$), given only independence-oracle access to the matroids $\mathcal{M}_1$ and $\mathcal{M}_2$. Recall that this followed from the integrality of the polytope $\mathcal{P}(\mathcal{M}_1) \cap \mathcal{P}(\mathcal{M}_2)$, where $\mathcal{P}(\mathcal{M}_1)$ and $\mathcal{P}(\mathcal{M}_2)$ are the two matroid polytopes.

We will now consider a related problem, the *minimum-weight common-basis* problem, show how to solve it efficiently, and show that it can be viewed as a generalization of the shortest path problem in directed graphs. Given two matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$, let $\mathcal{B}_1 \subseteq 2^{\mathcal{X}}$ denote the family of all bases (maximum-cardinality independent sets) of $\mathcal{M}_1$, and similarly $\mathcal{B}_2$ for $\mathcal{M}_2$. Given a weight vector $w \in \mathbb{R}^{\mathcal{X}}$, the minimum-weight common basis of $\mathcal{M}_1$ and $\mathcal{M}_2$, if it exists, is the set $B \in \mathcal{B}_1 \cap \mathcal{B}_2$ minimizing $\sum_{i \in B} w(i)$. Note that for a common basis to exist, both matroids must have the same rank (though this is not sufficient).

(a) [8 points]. Show that the minimum-weight common-basis problem can be solved in polynomial time in $|\mathcal{X}|$, given only independence-oracle access to the two matroids $\mathcal{M}_1$ and $\mathcal{M}_2$. Your algorithm should either output a minimum-weight common-basis of $\mathcal{M}_1$ and $\mathcal{M}_2$ if one exists, or else assert that there is no common basis.
(**Hint**: You can start by showing that that the convex hull of common bases is a face of the matroid intersection polytope)

(b) [7 points]. Show that the shortest path problem in weighted directed graphs can be reduced, in linear time, to the minimum-weight common-basis problem when the graph has no negative cycles.
(**Hint**: Add self-loops to all nodes other than the source and destination, and define two partition matroids such that their common bases are precisely a path combined with some of the self-loops.)