Midterm Exam, CS-450: Algorithms II, 2024-2025

Do not turn the page before the start of the exam. This document is double-sided and has 8 pages.

- You are only allowed to have an A4 page written on both sides.
- Communication, calculators, cell phones, computers, etc... are not allowed.
- The exam consists of two parts. The first part consists of multiple-choice questions (Problem 1), the second part consists of three open-ended questions (Problems 2, 3, 4).
- For the open-ended questions, your explanations should be clear enough and in sufficient detail that a fellow student can understand them. In particular, do not only give pseudocode without explanations. A good guideline is that a description of an algorithm should be such that a fellow student can easily implement the algorithm following the description.
- You are allowed to refer to material covered in the lectures including algorithms and theorems (without reproving them). You are however *not* allowed to simply refer to material covered in exercises.

Good luck!

Problem 1: Multiple Choice Questions (39 points)

For each question, select the correct alternative. Each question has **exactly one** correct answer. Wrong answers are **not penalized** with negative points.

1a. Matroids *(13 points).* Let $n \geq 2$, and consider the ground set $E = [n]$. Which of the following is **not** a matroid?

A. $M = (E, \mathcal{I})$ for $\mathcal{I} = \{X \subseteq E : |X| \leq \frac{n}{100}\}$

B. $M = (E, \mathcal{I})$ for $\mathcal{I} = \{\emptyset, \{1\}\}$

C. $M = (E, \mathcal{I})$ for $\mathcal{I} = \{\emptyset\} \cup \{X \subseteq E : 1 \in X\}$

D. $M = (E, \mathcal{I})$ for $\mathcal{I} = \{X \subseteq E : 1 \notin X\}$

Solution. Answer: C.

A. is an example of a k-uniform matroid

B. is a matroid (it is straightforward to check that the two axioms hold).

C. is **not** a matroid: the first axiom fails, because $\{1,2\} \in \mathcal{I}$, $\{2\} \subseteq \{1,2\}$, but $\{2\} \notin \mathcal{I}$.

D. is a matroid: If $Y \in \mathcal{I}$ and $X \subseteq Y$, then $1 \notin X$, so $X \in \mathcal{I}$, so the first axiom holds. To check the second axiom: Suppose $X, Y \in \mathcal{I}$ and $|X| < |Y|$. Pick any $i \in Y \setminus X$. Then $i \neq 1$ (since $i \in Y$), so $X \cup \{i\} \in \mathcal{I}$.

□

1b. Duality *(13 points).* What is the Dual of the following Linear Program:

$$\begin{array}{ll} \textbf{Minimize} & x_1 + 2x_2 \\ \textbf{Subject to} & 5x_1 + x_2 \geq 2 \\ & 2x_1 + x_2 \geq 1 \\ & \quad\quad 4x_2 \leq 10 \\ & \quad x_1, x_2 \geq 0 \end{array}$$

A.

$$\begin{array}{lr} \textbf{Maximize} & 2y_1 + y_2 - 10y_3 \\ \textbf{Subject to} & 5y_1 + 2y_2 \leq 1 \\ & y_1 + y_2 - 4y_3 \leq 2 \\ & y_1, y_2, y_3 \geq 0 \end{array}$$

B.

$$\begin{array}{lr} \textbf{Maximize} & 2y_1 + y_2 + 10y_3 \\ \textbf{Subject to} & 5y_1 + 2y_2 \leq 1 \\ & y_1 + y_2 + 4y_3 \leq 2 \\ & y_1, y_2, y_3 \geq 0 \end{array}$$

C.

$$\begin{aligned} \textbf{Maximize} \quad & 2y_1 + y_2 + 10y_3 \\ \textbf{Subject to} \quad & 5y_1 + 2y_2 \geq 2 \\ & y_1 + y_2 + 4y_3 \geq 1 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}$$

D. None of the above

Solution. The primal can be rewritten in the standard form

$$\begin{aligned} \textbf{Minimize} \quad & x_1 + 2x_2 \\ \textbf{Subject to} \quad & 5x_1 + x_2 \geq 2 \\ & 2x_1 + x_2 \geq 1 \\ & -4x_2 \geq -10 \\ & x_1, x_2 \geq 0 \end{aligned}$$

so the correct solution is A:

$$\begin{aligned} \textbf{Maximize} \quad & 2y_1 + y_2 - 10y_3 \\ \textbf{Subject to} \quad & 5y_1 + 2y_2 \leq 1 \\ & y_1 + y_2 - 4y_3 \leq 2 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}$$

□

1c. Maximum Bipartite Matching *(13 points)*. Consider a graph $G = (V, E)$ that is bipartite, i.e. the vertices V are partitioned into the two disjoint sets A and B such that every edge is between a vertex in A and a vertex in B. Moreover, the graph G is d-regular, i.e. every vertex has degree d, with $d > 0$. What is the size of the minimum vertex cover in the graph?

A. $d \cdot |A|$

B. $\frac{1}{d} \cdot |B|$

C. $|A|$

D. $|A| + |B|$

Solution. First, for any subset $S \subseteq A$, the number of edges incident to S is $d \cdot |S|$. The neighbors of S in B, denoted $N(S)$, must also have $d \cdot |N(S)|$ edges. Since each edge incident to S is also incident to $N(S)$, we have $d \cdot |S| \leq d \cdot |N(S)|$, implying $|S| \leq |N(S)|$. By Hall's theorem, this guarantees a perfect matching of size $\min(|A|, |B|)$. Since the graph is regular, $d \cdot |A| = d \cdot |B|$, so $|A| = |B|$. Finally, by König's theorem, the size of the minimum vertex cover equals the size of the maximum matching, which is $|A|$. □

Problem 2: Knapsack polytope extreme points (20 points)

In the **Fractional Knapsack** problem, we are given n items, each having a value $v_1, \dots, v_n$ and a weight $w_1, \dots, w_n$, along with a maximum capacity C. Our goal is to pick fractions of these items, so as to maximize their total value, subject to the constraint that the selected fractions have a total weight of at most C. This problem is captured by the following linear program:

$$
\begin{aligned}
\text{Maximize } & \sum_{i=1}^{n} v_i x_i \\
\text{Subject to: } & \sum_{i=1}^{n} w_i x_i \leq C \\
& 0 \leq x_i \leq 1 \quad \forall i \in \{1, \dots, n\}
\end{aligned}
$$

where each variable x_i corresponds to the fraction of i-th item in our solution.

Your task is to prove that **all** extreme points of the feasible region, defined by the above constraints, have at least $n-1$ integral coordinates. In other words, you should prove that, for any extreme point solution x^*, we have $|\{i \mid 0 < x_i^* < 1\}| \leq 1$.

Solution. Let $\boldsymbol{x}$ be an extreme point of the above polytope and suppose that it has at most $n-2$ integral coordinates. Let i and j be two of its fractional coordinates, that is $0 < x_i, x_j < 1$.

Choose any ϵ such that $0 < \epsilon < \min(w_\ell(1 - x_\ell), w_\ell x_\ell : \forall \ell \in \{i, j\})$. Notice that, due to x_i and x_j being fractional, such an ϵ exists. Using our choice, we construct the following solutions to the linear program.

$$
\boldsymbol{y} = \begin{cases} x_\ell & \text{if } \ell \notin \{i, j\} \\ x_i + \frac{\epsilon}{w_i} & \text{if } \ell = i \\ x_j - \frac{\epsilon}{w_j} & \text{if } \ell = j \end{cases}
$$

$$
\boldsymbol{z} = \begin{cases} x_\ell & \text{if } \ell \notin \{i, j\} \\ x_i - \frac{\epsilon}{w_i} & \text{if } \ell = i \\ x_j + \frac{\epsilon}{w_j} & \text{if } \ell = j \end{cases}
$$

Observe that both solutions are feasible since all of their coordinates have values in $[0, 1]$ and they also satisfy the capacity constraint since

$$
\sum_{i=1}^{n} w_i y_i = \sum_{i=1}^{n} w_i z_i = \sum_{i=1}^{n} w_i x_i \leq C.
$$

We finish the proof by noticing that $\boldsymbol{x}$ can be written as

$$
\boldsymbol{x} = \frac{1}{2}\boldsymbol{y} + \frac{1}{2}\boldsymbol{z},
$$

which contradicts the fact that $\boldsymbol{x}$ is an extreme point of the polytope.

$\square$

Problem 3: Matching on general graphs (21 points)

In class you have seen the following linear program to solve maximum weight matching on bipartite graphs:

$$\begin{aligned} \text{Maximize} \quad & \sum_{e \in E} x_e w_e \\ \text{Subject to} \quad & \sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in V \\ & x_e \geq 0 \quad \forall e \in E \end{aligned} \tag{1}$$

Here, we use $\delta(v)$ to be the set of edges incident to vertex v, formally, $\delta(v) = \{e \in E : v \in e\}$.

In this problem you are supposed to prove that the **integrality gap** of this linear program on general graphs is 3/2. In the case of a maximization problem the **integrality gap** g is defined as

$$g = \max_{I \in \mathcal{I}} \frac{OPT_{LP}(I)}{OPT(I)},$$

where $\mathcal{I}$ is the set of all problem instances.

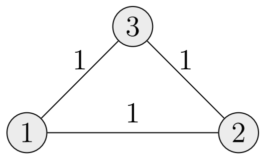
3a *(8 points)*. Show that the **integrality gap** g is at least 3/2.

3b *(13 points)*. Show that the **integrality gap** g is at most 3/2. You are allowed to use the following fact without proof. The extreme points of the following linear program are **integral**.

$$\begin{aligned} \text{Maximize} \quad & \sum_{e \in E} x_e w_e \\ \text{Subject to} \quad & \sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in V \\ & \sum_{\substack{e \in E, \\ e \subseteq S}} x_e \leq \frac{|S| - 1}{2} \quad \forall S \subseteq V, |S| \text{ is odd} \\ & x_e \geq 0 \quad \forall e \in E \end{aligned} \tag{2}$$

Hint: Let x be a solution to the linear program (1). Analyze a scaled version of x using (2).

Solution. **3a.** Consider the following example:



Here, the solution that sets $x_e = 1/2$ for every edge e has value 3/2, but any integral matching has only value 1.

3b. Consider the optimal solution x to the first linear program. The point $\frac{2}{3}x$ is feasible for the second linear program. Indeed,

$$\frac{2}{3} \cdot \sum_{\substack{e \in E, \\ e \subseteq S}} x_e \leq \frac{2}{3} \cdot \frac{1}{2} \cdot \sum_{v \in S} \sum_{e \in \delta(v)} x_e = \frac{|S|}{3} \leq \frac{|S|-1}{2}.$$

Here, we used the degree constraint from LP (1). Note that $\frac{2}{3}x$ achieves the weight

$$\frac{2}{3} \sum_{e \in E} x_e w_e.$$

Since the second linear program is integral, we know that there exists an integral matching with weight at least the weight of $\frac{2}{3}x$. In particular, for any instance I,

$$\frac{OPT_{LP}(I)}{OPT(I)} \leq \frac{\sum_{e \in E} x_e w_e}{\frac{2}{3} \sum_{e \in E} x_e w_e} = \frac{3}{2}.$$

$\square$

Problem 4: Pizzeria (20 points)

Your friend is building an online ordering platform for their pizzeria. Every evening, when the pizzeria opens at 18h00, the situation is the following.

- There are n pizzas requested.
- Each pizza $i \in [n]$ has a requested pick-up time t_i which is either 19h00, 20h00, 21h00.
- Each pizza $i \in [n]$ can give a profit p_i.
- There are 3 different kinds of pizza dough (regular, gluten free, whole grain). Each pizza $i \in [n]$ is requested to be made with one of these dough types, i.e. the set of pizzas $[n]$ can be partitioned into three sets $R, G, W \subseteq [n]$.

Design a polynomial time algorithm that:

- **takes as input** $n, (t_i)_{i\in[n]}, (p_i)_{i\in[n]}, R, G, W$ as described above, and two integers $d, h > 0$.
- **outputs** a subset $S \subseteq [n]$ of pizzas such that the profit $\sum_{i\in S} p_i$ is maximized, subject to the following constraints:
  (1) for each of the three types of dough R, G, W, there is only enough dough to make d pizzas from it, so S can contain at most d pizzas of each type;
  (2) it is possible to prepare all the pizzas in S so that
    - each pizza $i \in S$ is ready before its pick-up time t_i, and
    - at most h pizzas are prepared during each hour 18h00–18h59, 19h00–19h59, 20h00–20h59.

One can encode each of the constraints (1) and (2) via a matroid. You are allowed to use the following fact without proof. Remember that a family $\mathcal{F}$ of subsets of the ground set $[n]$ is called laminar if for all distinct $X, Y \in \mathcal{F}$, either $X \cap Y = \emptyset$, or $X \subseteq Y$, or $Y \subseteq X$. For a laminar family $\mathcal{F}$ and any set of positive integers $\{k_X\}_{X\in\mathcal{F}}$, one has that $\mathcal{M} = ([n], \mathcal{I})$ is a matroid, where

$$\mathcal{I} = \{T \subseteq [n] : |T \cap X| \le k_X \text{ for every } X \in \mathcal{F}\}.$$

Example. *Let us say we have $n = 30$ pizzas, and let $d = 7$ and $h = 3$. Suppose that 10 of the pizzas have $t_i =$19h00, 10 have $t_i =$20h00, and 10 have $t_i =$21h00.*

- *Let S_1 be a set consisting of 7 pizzas with $t_i =$21h00. Then S_1 **satisfies** constraint (2).*
- *Let S_2 be a set consisting of 1 pizza with $t_i =$19h00, 4 pizzas with $t_i =$20h00, and 2 pizzas with $t_i =$21h00. Then S_2 **satisfies** constraint (2).*
- *Let S_3 be a set consisting of 1 pizza with $t_i =$19h00 and 6 pizzas with $t_i =$20h00. Then S_3 **does not satisfy** constraint (2).*

Note that each of S_1, S_2, S_3 contains at most 7 pizzas. Hence, constraint (1) is satisfied by each of them, no matter what R, G, W are.

Solution. The problem can be solved by defining two matroids $\mathcal{M}_1 = ([n], \mathcal{I}_1)$ and $\mathcal{M}_2 = ([n], \mathcal{I}_2)$ and finding the maximum weight independent set in their intersection. The matroid $\mathcal{M}_1$ is meant to capture constraint (1) above, while $\mathcal{M}_2$ is meant to capture constraint (2). More precisely:

1. Let
$$\mathcal{I}_1 = \{S \subseteq [n] : |S \cap R| \leq d,\ |S \cap G| \leq d,\ |S \cap W| \leq d\}\,.$$
Since R, G, W partition $[n]$, we have that $([n], \mathcal{I}_1)$ is a partition matroid. By virtue of intersecting with this matroid, there will be enough dough to make each type of pizza in our output solution $S \subseteq [n]$.

2. Define
$$\begin{aligned}
T_1 &= \{i \in [n] : t_i = 19\text{h}00\}\,,\\
T_2 &= \{i \in [n] : t_i = 20\text{h}00\}\,,\\
T_3 &= \{i \in [n] : t_i = 21\text{h}00\}\,,\\
X_1 &= \{i \in [n] : i \in T_1\}\,,\\
X_2 &= \{i \in [n] : i \in T_1 \cup T_2\}\,,\\
X_3 &= \{i \in [n] : i \in T_1 \cup T_2 \cup T_3\}\,,
\end{aligned}$$
and let
$$\mathcal{I}_2 = \{S \subseteq [n] : \forall j \in \{1, 2, 3\},\ |S \cap X_j| \leq h \cdot j\}\,.$$
First observe that $\{X_1, X_2, X_3\}$ is a laminar family because $X_1 \subseteq X_2 \subseteq X_3$. Then $([n], \mathcal{I}_2)$ is a (laminar) matroid. By virtue of intersecting with this matroid, we are guaranteed that each pizza $i \in S$ can be prepared within its pick-up time t_i while only preparing h pizzas per hour. In particular, we will start preparing the pizzas with the following priority: first the pizzas in $S \cap T_1$, then the pizzas in $S \cap T_2$ and last the pizzas in $S \cap T_3$. We will always prepare exactly h pizzas each hour unless all the pizzas are already prepared. Assume for contradiction that there is a pizza $i \in S \cap T_j$ that does not meet its pick-up time. That means that there are more than $h \cdot j$ pizzas with higher or equal priority than i in S. This is a contradiction to the constraint that $|S \cap X_j| \leq h \cdot j$.

$\square$