Exercise Set II, Advanced Algorithms 2024

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 *(half a *)* Devise an algorithm for the following graph orientation problem:

Input: An undirected graph $G = (V, E)$ and capacities $k : V \to \mathbb{Z}$ for each vertex.

Output: If possible, an orientation of G such that each vertex $v \in V$ has in-degree at most $k(v)$.

An orientation of an undirected graph G replaces each undirected edge $\{u, v\}$ by either an arc (u, v) from u to v or by an (v, u) from v to u.

(Hint: reduce the problem to matroid intersection. You can also use bipartite matching...)

Solution: Consider the directed graph $D = (V, A)$ obtained from G by replacing every edge $\{u, v\} \in E$ by the two arcs (u, v) and (v, u). With the arc set A as ground set we define two partition matroids $\mathcal{M}_1$ and $\mathcal{M}_2$:

- To be independent in $\mathcal{M}_1$ one can take at most one of $\{(u, v), (v, u)\}$ for every $\{u, v\} \in E$, i.e.,

$$\mathcal{I}_1 = \{F \subseteq A : |F \cap \{(u, v), (v, u)\}| \leq 1 \text{ for all } \{u, v\} \in E\}\,.$$

  This matroid enforces the constraint that each edge should be oriented in one direction.

- To be independent in M_2, one can take at most $k(v)$ arcs among the set $\delta^-(v)$ of incoming arcs for every v:

$$\mathcal{I}_2 = \{F \subseteq A : |F \cap \delta^-(v)| \leq k(v) \text{ for all } v \in V\}\,.$$

  This matroid enforces the indegree restrictions of the orientation.

By the above definitions, there exists an orientation satisfying the required indegree restrictions if and only if there exists a common independent set to $\mathcal{M}_1$ and $\mathcal{M}_2$ of cardinality precisely $|E|$ (in which case we select either (u, v) or (v, u) but not both).

2 The first problem is difficult so you may want to skip that and solve (6b) assuming (6a) and then try (6a) if you have time.

2a *(*)* Consider a family $\mathcal{F}$ of subsets of the ground set E that satisfies: if $X, Y \in \mathcal{F}$ then either $X \cap Y = \emptyset$ (they are disjoint), $X \subseteq Y$ (X is a subset of Y), or $Y \subseteq X$ (Y is a subset of X). Show that for any positive integers $\{k_X\}_{X \in \mathcal{F}}$ (one for each set in $\mathcal{F}$) we have that $\mathcal{M} = (E, \mathcal{I})$ is a matroid, where

$$\mathcal{I} = \{S \subseteq E : |S \cap X| \leq k_X \text{ for every } X \in \mathcal{F}\}\,.$$

Such a matroid is called a laminar matroid.

Solution: (There are various proof of this fact. One is explained below.)

We show that $\mathcal{M}$ satisfies the two axioms I_1 and I_2 for matroids.

- Let $\emptyset \neq S \in \mathcal{I}$ and let $x \in S$. To show that $\mathcal{M}$ satisfies Axiom I_1, let $T \subsetneq S$. Since $S \in \mathcal{I}$, we have $|T \cap X| \leq |S \cap X| \leq k_X$ for each $X \in \mathcal{F}$. Hence, $T \in \mathcal{I}$.
- Let $A, B \in \mathcal{I}$ such that $|A| > |B|$. To show that $\mathcal{M}$ satisfies Axiom I_2, we show that there exist an element $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

  Suppose that $|B \cap X| < k_X$ (with strict inequality) for all $X \in \mathcal{F}$. In this case, we choose any $x \in A \setminus B$ and add it to B, and the resulting set $B \cup \{x\}$ is in $\mathcal{I}$.

  Now suppose that $|B \cap X| = k_X$ for some set $X \in \mathcal{F}$. Then, we cannot add any $x \in A \cap X$ to B as it would violate the constraint for X ($|(B \cup \{x\}) \cap X| > k_X$ for such an x). However notice that $|A \cap X|$ can also have at most k_X elements. Thus A has at least $|A| - k_X$ elements outside X and B has exactly $|B| - k_X$ elements outside X. Since $|A| > |B|$ we have $|A| - k_X > |B| - k_X$, and consequently, we have more elements in $A \setminus X$ than in $B \setminus X$. We generalize this idea formally below.

  Let $\mathcal{F}^\star = \{X \in \mathcal{F} : |B \cap X| = k_X\}$ be the collection of sets in $\mathcal{F}$ for which the the constraints are satisfied with equality. By our assumption above, $\mathcal{F}^\star$ is non-empty. Let Y_1 be the largest set in $\mathcal{F}^\star$. Let Y_2 be the next largest set in $\mathcal{F}^\star$ that is disjoint with Y_1. After Y_i is selected, let Y_{i+1} be the next largest set in $\mathcal{F}^\star$ that is disjoint from each Y_j for $j = 1, 2, \ldots, i$. Stop this procedure when no more such sets can be selected, and let Y_m be the last selected set. Any of the remaining sets in $\mathcal{F}^\star$ is completely contained inside one of the sets $Y_1, Y_2, \ldots Y_m$ (why?). Let $Y = \cup_{i=1}^m Y_i$. We show that the number of elements in $A \setminus Y$ is more than the number of elements in $B \setminus Y$. We have

  $$|A| = |(A \cap Y) \cup (A \setminus Y)| = \left|\left(\cup_{i\in[m]}(A \cap Y_i)\right) \cup (A \setminus Y)\right| = |A \setminus Y| + \sum_{i\in[m]} |A \cap Y_i|,$$

  and similarly, we have

  $$|B| = |B \setminus Y| + \sum_{i\in[m]} |B \cap Y_i|.$$

  But $|B \cap Y_i| = k_{Y_i}$ and $|A \cap Y_i| \leq k_{Y_i}$ for all $i = 1, \ldots, m$. Thus using $|A| > |B|$ we conclude that $|A \setminus Y| > |B \setminus Y|$. Choose an element $x \in (A \setminus Y) \setminus B$. By our selection of Y_i's, x is not in any of the sets in $\mathcal{F}^\star$, and therefore, adding it to B would not violate those constrains for sets in $\mathcal{F}^\star$. For all sets in $\mathcal{F} \setminus \mathcal{F}^\star$, the respective constraints have some slack and adding x to B would not violate those constraints either.

2b Argh! Buying the DVD rental shop was not such a great idea. After the explosion of more convenient streaming services, you are now forced to close your business venture. But what should you do with all your DVDs? To be exact, you have n DVDs and each one is placed in one of the following genres: action, comedy, drama, horror or adventure. As you are a very nice person, you decide to distribute these DVDs among your most loyal customers. You have m loyal customers and for each DVD i and customer j there is a positive weight $w(i, j)$ that models how interesting DVD i is for customer j. Your goal is to find an assignment of DVDs to loyal customers satisfying the following:

- Each DVD is assigned to at most one customer.

- Each customer receives at most 5 DVDs in total and no more than 2 DVDs of the same genre.
- The total weight (called the social welfare) of your assignment is maximized.

Show that the problem of distributing the DVDs as above can be formulated as that of finding a maximum weight independent set in the intersection of two matroids.

Solution: In this problem, we need to satisfy two conditions.

1. Each DVD is assigned at most one customer
2. Each customer receives at most 5 DVDs in total and no more than 2 DVDs of the same genre.

Let the customers be numbered $1, \ldots, m$ and DVDs be numbered $1, \ldots, n$. Let $E = \{(i, j) : i \in [m], j \in [n]\}$ be the set of possible edges, which will be the ground set for our matroids.

Let $S \subseteq E$ be any assignment of DVDs to customers that satisfies the two constraints.

Let $D_j = \{(i, j) : i \in [m]\}$ for all $j \in [n]$. In order to satisfy Condition 1, it is clear that, for all $j \in [n]$, $|S \cap D_j| \leq 1$. Hence, we define our first matroid as the following partition matroid.

$$\mathcal{I}_1 = \{S \subseteq E : |S \cap D_j| \leq 1 \text{ for all } j \in [n]\}.$$

For the second constraint, we use the result from **1a**. Let $G_1, \ldots G_5$ be a partition of $[n]$ corresponding to genres 'action', 'comedy', 'drama', 'horror' and 'adventure' respectively. For $i \in [n]$, let $C_i = \{(i, j) \in E : j \in [n]\}$ be the set of edges going from customer i to the set of DVDs. For $i \in [m], \ell \in [5]$, let $T_{i\ell} = \{(i, j) \in E : j \in G_\ell\}$ be the set of edges going from customer i to DVDs of genre G_ℓ. Note that C_i's are a partitioning of E and, for each i, $T_{i\ell}$'s are a partitioning of C_i.

If S satisfies Condition 2, it must be the case that, $|S \cap T_{i\ell}| \leq 2$ for all $\ell = 1, \ldots, 5$ and $|S \cap C_i| \leq 5$ for all $i \in [n]$.

Let $k_{T_{i\ell}} = 2$ for all $i \in [n], \ell \in [5]$ and let $k_{C_i} = 5$. Let $\mathcal{F} = \{T_{i\ell} : i \in [n], \ell \in [5]\} \cup \{C_i : i \in [m]\}$. Since C_i's are a partitioning of E and since for each i, $T_{i\ell}$'s are a partitioning of C_i, any $X, Y \in \mathcal{F}$ satisfies either $X \cap Y = \emptyset$ or $X \subseteq Y$ or $Y \subseteq X$. Thus, from **1a** the following is a matroid.

$$\mathcal{I}_2 = \{S \subseteq E : |S \cap X| \leq k_X \text{ for all } X \in \mathcal{F}\}.$$

From the above discussion it is clear that a solution S is feasible if and only if it is independent in both $\mathcal{I}_1$ and $\mathcal{I}_2$. Hence, the problem is equivalent to finding the *maximum weight independent set* in the intersection of the two matroids, $\mathcal{I}_1$ and $\mathcal{I}_2$.

3 Spanning trees with colors. Consider the following problem where we are given an edge-colored graph and we wish to find a spanning tree that contains a specified number of edges of each color:

Input: A connected undirected graph $G = (V, E)$ where the edges E are partitioned into k color classes $E_1, E_2, \ldots, E_k$. In addition each color class i has a target number $t_i \in \mathbb{N}$.

Output: If possible, a spanning tree $T \subseteq E$ of the graph satisfying the color requirements:

$$|T \cap E_i| = t_i \qquad \text{for } i = 1, \ldots, k.$$

Otherwise, i.e., if no such spanning tree T exists, output that no solution exists.

Design a polynomial time algorithm for the above problem. You should analyze the correctness of your algorithm, i.e., why it finds a solution if possible. To do so, you are allowed to use algorithms and results seen in class without reexplaining them.

Solution: We solve this problem using matroid intersection. First observe that if the summation of the t_i for $1 \le i \le k$ is not equal to $n-1$ then there is no feasible solution since we know that the number of edge in any spanning tree is exactly $n-1$. Therefore, we assume $\sum_{1\le i\le k} t_i = n-1$. The ground set for both matroids that we use is the set of the edges E. First matroid that we use is the graphic matroid. The second matroid that we use is a partition matroid with following independent sets:

$$\mathcal{I} = \{F \subseteq E \mid |F \cap E_i| \le t_i, \quad \text{for } 1 \le i \le k\}$$

As shown in class the both above defined matroids are indeed matroid. Now assume that F is the maximum size independent set the intersection of these two matroids (we saw in the class how we can find F). If $|F| < n-1$ it is not possible to find a solution for our problem, since any solution to our problem corresponds to a solution in the intersection of these two matroids of size $n-1$. Moreover, if $|F| = n-1$, than F is a spanning tree and $|F \cap E_i| \le t_i$. Also, we know that $|F| = n-1$ and $\sum_{1\le i\le k} t_i = n-1$ and E_i's are disjoint. Therefore $|F \cap E_i| = t_i$, so we get the desired solution.

4 For a bipartite graph, devise an efficient algorithm for finding an augmenting path P (if one exists). What is the total running time of the AugmentingPathAlgorithm explained in the second lecture?

Solution: Let the bipartition of the bipartite graph $G = (V, E)$ be A and B, and let M be the current bipartite matching maintained by AugmentingPathAlgorithm. From G, obtain the directed graph where each edge in M is directed from the vertex in B to the vertex in A. All other edges $E \setminus M$ are directed from A to B. Now start a breadth-first-search from the unmatched vertices in A. Note that if an unmatched vertex in B is reached by the breadth-first-search, then we have found an augmenting path. The runtime analysis is

- $O(|E|)$ for the construction of the directed graph from G.
- $O(|V| + |E|)$ for the run of the breadth-first-search.

So we can find an augmenting path in time $O(|V|+|E|)$. Since we increase the size of the matching each time we find an augmenting path, the total running time is $O(k(|V|+|E|) = O(|V|(|V|+|E|)$ where k is the cardinality of a maximum matching.

5 You have just started your prestigious and important job as the Swiss Cheese Minister. As it turns out, different fondues and raclettes have different nutritional values and different prices:

Food	Fondue moitie moitie	Fondue a la tomate	Raclette	Requirement per week
Vitamin A [mg/kg]	35	0.5	0.5	0.5 mg
Vitamin B [mg/kg]	60	300	0.5	15 mg
Vitamin C [mg/kg]	30	20	70	4 mg
[price [CHF/kg]	50	75	60	—

Formulate the problem of finding the cheapest combination of the different fondues (moitie moitie & a la tomate) and Raclette so as to satisfy the weekly nutritional requirement as a linear program.

Solution: We have a variable x_1 for moitie moitie, a variable x_2 for a la tomate, and a variable x_3 for Raclette. The linear program becomes

$$\begin{aligned}
\text{Minimize} \quad & 50x_1 + 75x_2 + 60x_3 \\
\text{Subject to} \quad & 35x_1 + 0.5x_2 + 0.5x_3 \geq 0.5 \\
& 60x_1 + 300x_2 + 0.5x_3 \geq 15 \\
& 30x_1 + 20x_2 + 70x_3 \geq 4 \\
& x_1, x_2, x_3 \geq 0
\end{aligned}$$

6 Consider the following linear program for finding a maximum-weight matching:

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

(This is similar to the perfect matching problem seen in the lecture, except that we have inequality constraints instead of equality constraints.) Prove that, for bipartite graphs, any extreme point is integral.

Solution: We prove that all the extreme points are integral by contradiction. To that end, assume that there exists an extreme point x^* that is not integral. Let $G = (V_1, V_2, E)$ be the given bipartite graph and let $E_f = \{e \in E \mid 0 < x_e^* < 1\}$. If E_f contains a cycle, then the proof follows in the same way as the proof in the lecture notes. Therefore, we assume that E_f does not contain any cycles. Consider any maximal path in E_f; let it have vertices $v_1, ..., v_k$ and edges $e_1, ..., e_{k-1}$. Choose any ϵ such that $0 < \epsilon < \min(x_{e_i}^*, 1 - x_{e_i}^* : i = 1, ..., k-1)$. Note that, since E_f only contains edges that are fractional, such an ϵ exists. Let y, z be the following two solutions to the linear program:

$$y = \begin{cases} x_e^* + \epsilon & \text{if } e \in \{e_1, e_3, e_5, e_7, ...\} \\ x_e^* - \epsilon & \text{if } e \in \{e_2, e_4, e_6, e_8, ...\} \\ x_e^* & \text{otherwise} \end{cases}$$

$$z = \begin{cases} x_e^* - \epsilon & \text{if } e \in \{e_1, e_3, e_5, e_7, ...\} \\ x_e^* + \epsilon & \text{if } e \in \{e_2, e_4, e_6, e_8, ...\} \\ x_e^* & \text{otherwise} \end{cases}$$

One can see that $x^* = \frac{y+z}{2}$.

We continue by showing that y is a feasible solution to the linear program. One can see that for any vertex $v \in G$ except v_1 and v_k we have $\sum_{e\in\delta(v)} y_e = \sum_{e\in\delta(v)} x^*_e$. So, we only need to show that the linear program constraint holds for v_1 and v_k. Let us first state two observations. First, by the definition of ϵ, we have that $0 \le x^*_{e_1} + \epsilon \le 1$, $0 \le x^*_{e_1} - \epsilon \le 1$, $0 \le x^*_{e_{k-1}} + \epsilon \le 1$, and $0 \le x^*_{e_{k-1}} - \epsilon \le 1$. Second, since the path is maximal and E_f does not contain any cycles, the degrees of v_1 and v_k in E_f are both one. Therefore $\sum_{e\in\delta(v_1)} y_e = y_{e_1}$ and $\sum_{e\in\delta(v_k)} y_e = y_{e_{k-1}}$. Putting together the previous two observations, we get that the linear program constraint also holds for v_1 and v_k, so y is a feasible solution.

We can similarly show that z is also a feasible solution. This shows that we can write x^* as a convex combination of y and z, which contradicts the fact that x^* is an extreme point.

7 *(half a *)* Use the integrality of the bipartite perfect matching polytope (as proved in class) to show the following classical result:

> The edge set of a k-regular bipartite graph $G = (A \cup B, E)$ can in polynomial time be partitioned into k disjoint perfect matchings.

A graph is k-regular if the degree of each vertex equals k. Two matchings are disjoint if they do not share any edges.

Solution: We show how to find such k disjoint perfect matchings in a k-regular bipartite graph in polynomial time.

Let $G_0 = (A \cup B, E)$ be a k-regular bipartite graph. Consider the LP for bipartite perfect matching on G_0. The LP is feasible because setting $x_e = 1/k$ for all $e \in E$ satisfies all the constraints (recall that each vertex of a k-regular graph is incident to exactly k edges). Now we find an extreme point solution to the LP in polynomial time, and due to the integrality of such solutions, we get a valid perfect matching M_1.

Notice that M_1, being a perfect matching, forms a 1-regular sub-graph of G. Therefore, if we remove the matching M_1 from the original graph G_0, we get a new $(k-1)$-regular graph $G_1 = (A \cup B, E \setminus M)$.

Now we repeat the process k times. Formally, at each iteration $i = 1, \ldots, k$, we start with $(k-i+1)$-regular graph G_{i-1}. By solving the bipartite perfect matching LP for G_{i-1} to get an extreme point solution, we obtain a perfect matching M_i. We remove M_i from G_{i-1} to obtain a $(k-i)$-regular graph G_i, which is a sub-graph of G_{i-1}.

Since we remove the already found perfect matchings at each iteration, the k-perfect matchings $M_1, \ldots, M_k$ are disjoint. Furthermore, since all graphs $G_1, \ldots, G_{k-1}$ are sub-graphs of the original graph G_0, the matchings $M_1, \ldots, M_k$ are all valid perfect matchings of G_0.