Lab04-Matroid

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2021.

∗ If there is any problem, please contact TA Haolin Zhou.

∗ Name: WendiChen Student ID: 519021910071 Email: chenwendi-andy@sjtu.edu.cn

1. *Property of Matroid.*

   (a) Consider an arbitrary undirected graph $G = (V, E)$. Let us define $M_G = (S, C)$ where $S = E$ and $C = \{I \subseteq E \mid (V, E\backslash I) \text{ is connected}\}$. Prove that M_G is a **matroid**.

   Solution.
   Let us prove *Hereditary* first.
   If B satisfies that $B \in C$ and $(V, E\backslash B)$ is connected, for any $A \subset B$, we have $E\backslash B \subset E\backslash A$. Therefore, $(V, E\backslash A)$ is also connected and $A \in C$.
   Then we'll prove the *exchange property*.
   Consider $A, B \in C$ with $|A| < |B|$. Note that $(V, E\backslash B)$ is connected, then $|E\backslash B| \geq n-1$. Thus, we have $|E\backslash A| > |E\backslash B| \geq n-1$. That implies $(V, E\backslash A)$ has at least one circle. And there must exist one edge e on one circle of $(V, E\backslash A)$ which is not involved in $(V, E\backslash B)$(if not, all circles in $(V, E\backslash A)$ exist in $(V, E\backslash B)$, which means $|E\backslash A| \leq |E\backslash B|$ and contradicts). Thus, we have $e \in B\backslash A$ and $(V, E\backslash(A \cup \{e\}))$ is connected. Therefore, $E\backslash(A \cup \{e\}) \in C$.
   Thus, we successfully prove that M_G is a matroid.

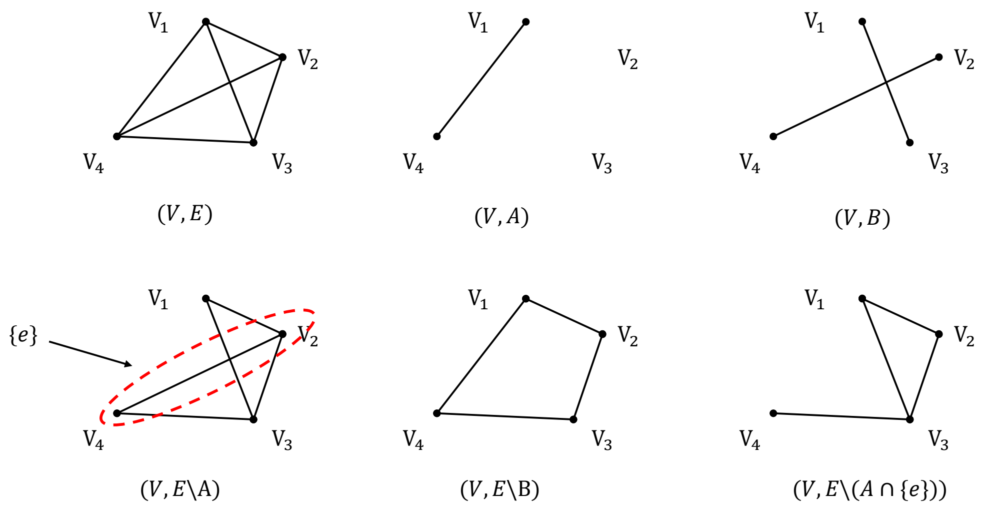

Figure 1: Proof for Cographic Matroid

□

   (b) Given a set A containing n real numbers, and you are allowed to choose k numbers from A. The bigger the sum of the chosen numbers is, the better. What is your algorithm to choose? Prove its correctness using **matroid**.

   Remark: Denote $\mathbf{C}$ be the collection of all subsets of A that contains no more than k elements. Try to prove $(A, \mathbf{C})$ is a matroid.

   Solution.

Algorithm 1: Greedy Algorithm to Choose k Numbers

Input: A set A containing n real numbers and an integer k

Output: A set of k numbers which has the biggest sum

Sort all n numbers in S into ordering $x_1 \geq x_2 \geq \cdots \geq x_n$;
$A \leftarrow \emptyset$;
for $i = 1$ *to* n **do**
  if $A \cup \{x_i\} \in C$ **then**
    $A \leftarrow A \cup \{x_i\}$;
return A;

As the remark described above, now we will prove that $(A, \mathbf{C})$ is a matroid.
Let us prove *Hereditary* first.
If B is a subset of B which contains no more than k elements, then for any $A \subset B$, $|A| < |B| \leq k$. Thus, $A \in \mathbf{C}$.
Then we will prove the exchange property.
Consider $A, B \in \mathbf{C}$ with $|A| < |B|$. There must exist one element e satisfying $e \in B \backslash A$. Then $|A \cup \{e\}| \leq |B| \leq k$. Thus $A \cup \{e\} \in \mathbf{C}$.
Therefore, we successfully prove that $(A, \mathbf{C})$ is a matroid. So, the Alg.1 is actually the *Greedy-MAX* algorithm, which performs an optimal solution for a matroid.

□

2. *Unit-time Task Scheduling Problem.* Consider the instance of the **Unit-time Task Scheduling Problem** given in class.

(a) Each penalty ω_i is replaced by $80 - \omega_i$. The modified instance is given in Tab. 1. Give the final schedule and the optimal penalty of the new instance using Greedy-MAX.

Table 1: Task

a_i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
ω_i	10	20	30	40	50	60	70

Solution.
Firstly, we sort the tasks non-increasingly and get the table below

Table 2: Sorted Task

a_i	7	6	5	4	3	2	1
d_i	6	4	1	3	4	2	4
ω_i	70	60	50	40	30	20	10

Then, we check whether $A \cup \{a_i\} \in \mathbf{C}$ for each a_i in order. At last, the *Greedy-MAX* selects $A = \{a_7, a_6, a_5, a_4, a_3\}$ and reject a_2, a_1. We sort A non-decreasingly by d_i and get the final schedule is $\langle a_5, a_4, a_3, a_6, a_7, a_1, a_2 \rangle$. The optimal penalty is $\omega_1 + \omega_2 = 30$. □

(b) Show how to determine in time $O(|A|)$ whether or not a given set A of tasks is independent. (**Hint**: You can use the lemma of equivalence given in class)

Solution.
According to the lemma of equivalence, we naturally introduce the the sum of prefixes to simply the process of judgement. We can simply calculate the number of each d_i and then use the sum of prefixes to determine whether A is independent. From Alg.2, we can find that the time complexity is obviously $O(k) = O(|A|)$.

Algorithm 2: Algorithm to Determine the Independence of Set A

Input: A set A of k tasks, where $A = \{x_1, \dots, x_k\}$
Output: A boolean that implies whether A is independent

$count[1, \dots, n] \leftarrow [0, \dots, 0]$;
$prefix \leftarrow 0$;
for $i = 1$ *to* k **do**
    $count[d_i] \leftarrow count[d_i] + 1$;
for $i = 1$ *to* k **do**
    $prefix \leftarrow prefix + count[i]$;
    if $prefix > i$ **then**
        return $False$;
return $True$;

□

3. *MAX-3DM.* Let X, Y, Z be three sets. We say two triples (x_1, y_1, z_1) and (x_2, y_2, z_2) in $X \times Y \times Z$ are *disjoint* if $x_1 \neq x_2$, $y_1 \neq y_2$, and $z_1 \neq z_2$. Consider the following problem:

Definition 1 (MAX-3DM)**.** *Given three disjoint sets X, Y, Z and a non-negative weight function $c(\cdot)$ on all triples in $X \times Y \times Z$,* ***Maximum 3-Dimensional Matching*** *(MAX-3DM) is to find a collection $\mathcal{F}$ of disjoint triples with maximum total weight.*

(a) Let $D = X \times Y \times Z$. Define independent sets for MAX-3DM.
(b) Write a greedy algorithm based on Greedy-MAX in the form of *pseudo code*.
(c) Give a counter-example to show that your Greedy-MAX algorithm in Q. 3b is not optimal.
(d) Show that: $\max_{F \subseteq D} \frac{v(F)}{u(F)} \leq 3$. (Hint: you may need Theorem 1 for this subquestion.)

Theorem 1. *Suppose an independent system $(E, \mathcal{I})$ is the intersection of k matroids $(E, \mathcal{I}_i)$, $1 \leq i \leq k$; that is, $\mathcal{I} = \bigcap_{i=1}^{k} \mathcal{I}_i$. Then $\max_{F \subseteq E} \frac{v(F)}{u(F)} \leq k$, where $v(F)$ is the maximum size of independent subset in F and $u(F)$ is the minimum size of maximal independent subset in F.*

Solution.

(a) Donate $\mathbf{C}$ be the collection of all subsets of D which satisfies the following property. For every $A \in \mathbf{C}$, if $a = (x_1, y_1, z_1) \in A$ and $b = (x_2, y_2, z_2) \in A$, a and b are disjoint. Now let us prove that $(D, \mathbf{C})$ is an independent system.
Assume $B \in \mathbf{C}$, then for any $A \subset B$, if $a \in A \subset B$ and $b \in A \subset B$, a and b are disjoint. Thus, we have $A \in \mathbf{C}$, which implies $(D, \mathbf{C})$ is an independent system.

(b)

Algorithm 3: Greedy Algorithm to Choose Disjoint Triples

Input: A set D of triples, where $D = \{a_1, a_2, \ldots, a_n\}$
Output: A set F of disjoint triples with maximum total weight

Sort all n triples in D into ordering $c(a_1) \geq c(a_2) \geq \cdots \geq c(a_n)$;
$F \leftarrow \emptyset$;
for $i = 1$ *to* n **do**
  if $F \cup \{a_i\} \in \mathbf{C}$ **then**
    $F \leftarrow F \cup \{a_i\}$;
return F;

(c) Set $X = \{1, 2\}, Y = \{3, 4\}, Z = \{5, 6\}$ and $c(a = (x, y, z)) = y + z + \frac{x}{yz}$. If we use *Greedy-MAX* algorithm, we would choose $(2, 4, 6)$ and $(1, 3, 5)$, which generate an overall weight of $18 + \frac{1}{12} + \frac{1}{15} = 18.15$. However, if we choose $(2, 4, 5)$ and $(1, 3, 6)$, which is also feasible, we get an overall weight of $18 + \frac{1}{10} + \frac{1}{18} = 18.156 > 18.15$. Thus, the *Greedy-MAX* algorithm is not optimal.

(d) According to Theorem 1, what we need to prove is that in the independent system $(D, \mathbf{C})$, $\mathbf{C}$ is the intersection of 3 matroids.

Define $\mathbf{C_x}$ as the collection of all subsets of D which satisfies the *x-independent* property. That means for every $A \in \mathbf{C_x}$, if $a = (x_1, y_1, z_1) \in A$ and $b = (x_2, y_2, z_2) \in A$, we have $x_1 \neq x_2$. Then let us prove that $(D, \mathbf{C_x})$ is a matroid.
We prove *Hereditary* first.
If $B \in \mathbf{C_x}$, then for any $A \subset B$, if $a = (x_1, y_1, z_1) \in A \subset B$ and $b = (x_2, y_2, z_2) \in A \subset B$, it obviously follows $x_1 \neq x_2$. Thus, $A \in \mathbf{C_x}$.
Then let us prove the *exchange property*.
Consider $A, B \in \mathbf{C_x}$ with $|A| < |B|$. By the definition of $\mathbf{C_x}$, there are $|A|$ different values of x_i in A and $|B|$ different values of x_i in B. Thus, there exit a triple $e \in B$ whose x-dimension value is different from any triple in A, and obviously $e \in B \backslash A$. Therefore, $A \cup \{e\} \in \mathbf{C_x}$.

In the same way, we can define $\mathbf{C_y}$ and $\mathbf{C_z}$ and they are both matroid. Next, we will prove $\mathbf{C} = \bigcap_{i \in \{x,y,z\}} \mathbf{C_i}$.

In fact, we have

$$\begin{aligned}
\mathbf{C} &= \{A \mid A \subset D \text{ and } A \text{ is } x\text{-independent}, y\text{-independent and } z\text{-independent}\} \\
&= \{A \mid A \subset D \text{ and } A \text{ is } x\text{-independent}\} \cap \{A \mid A \subset D \text{ and } A \text{ is } y\text{-independent}\} \\
&\qquad \cap \{A \mid A \subset D \text{ and } A \text{ is } z\text{-independent}\} \\
&= \bigcap_{i \in \{x,y,z\}} \mathbf{C_i}
\end{aligned} \tag{1}$$

According to Theorem 1, we have $\max\limits_{F \subseteq D} \frac{v(F)}{u(F)} \leq 3$.

□

Remark: You need to include your .pdf and .tex files in your uploaded .rar or .zip file.