Lab04-Matroid

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

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

$*$ Name:Beichen Yu   Student ID:519030910245   Email: polarisybc@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**.

   Proof. (Hereditary) If removing I does not disconnect the graph G, then removing any subset of I will not disconnect G either. So it holds hereditary.

   (Exchange property) For convenience, let $n = |V|$. Let $A \in C$, which means $(V, E\backslash A)$ is a connected graph. So $|E\backslash A| \geqslant n - 1$. Consider a set B that $|B| < |A|$, then we have $|E\backslash B| \geqslant |E\backslash A|$. So $|E\backslash B| \geqslant n$, which means there is at least a circle in $(V, E\backslash B)$.

   Because $|E\backslash B| \geqslant |E\backslash A|$ and both of them is connected, and they have the same number of vertices, so there are more circles in $(V, E\backslash B)$ than in $(V, E\backslash A)$. Then we can find a edge e in a circle in $(V, E\backslash B)$ that is not used in $(V, E\backslash A)$.

   That means $e \in A$ while $e \notin B$, which is $e \in A\backslash B$. As e is in a circle, we can remove it from $(V, E\backslash B)$ without disconnecting the graph. So $B \cup e \in C$. That means exchange property is held.

   As it holds both hereditary and exchange property, it is indeed a 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**.

   Solution. Obviously we should sort the n real numbers in descending order at first, and then choose the first k numbers. The sum of the chosen numbers is the biggest.

   Denote $\mathbf{C}$ be the collection of all subsets of A that contains no more than k elements. Now we will try to prove that $(A, \mathbf{C})$ is a matroid.
   (Hereditary) If $I \in \mathbf{C}$, then I contains no more then k elements. Obviously any subset of I contains no more then k elements, so $(A, \mathbf{C})$ is hereditary.
   (Exchange property) Denote $A \in \mathbf{C}, B \in \mathbf{C}$ and $|A| < |B|$. We can find a element x that $x \in A$ and $x \notin B$ because $|A| < |B|$. According to the definition we have $|A| < |B| < k$, so $|A| < k - 1$. So $|A \cup x| < k, A \cup x \in \mathbf{C}$. That means $(A, \mathbf{C})$ is exchange property.
   As $(A, \mathbf{C})$ holds both hereditary and exchange property, it is indeed a matroid.

   Next we prove that our solution is just equal to Greedy-MAX algorithm.
   Find the smallest element s in the A. And next we denote the weighted function $c : A \to \mathbb{R}^+$. For each element $x \in A$, we denote $c(x) = x - s + 1$. Then the weight function extend to subset of A by summation:

$$c(I) = \sum_{x \in I} c(x)$$

Then we can using the Greedy-MAX algorithm. First sort all elements in descending order of $c(x)$, and add elements to the solution set S constantly until the solution set $S \notin \mathbf{C}$, which means there is already k elements in S. Because the descending order of $c(x)$ is the same as the descending order of x, so our solution is the same as the Greedy-MAX algorithm.

□

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. The Greedy-MAX selects a_1, a_2, a_3, a_4, then rejects a_5, a_6, and finally accepts a_7. So the final schedule is $\langle a_2, a_4, a_1, a_3, a_7, a_5, a_6 \rangle$. The final optimal penalty is $80 - \omega_5 + 80 - \omega_6 = 80 - 50 + 80 - 60 = 50$. □

   (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 given in class, the set A is independent means for $t = 0, 1, 2, \cdots, n, N_t(A) \leqslant t$. We can assume that there are n elements in set A.
   First of all, we should use an array $a[t]$ to record the number of tasks whose deadline is t. This step takes $O(n)$ to finish.
   Then we should calculate the partial sum of $a[t]$ and record it in $b[t]$. The value recorded in $b[t]$ is in fact the value of $N_t(A)$. It takes another $O(n)$.
   Finally we need to check for $t = 0, 1, 2, \cdots, n$, whether $b[t] \leqslant t$, and it takes $O(n)$ as well.
   So the time complexity is equal to $O(n + n + n) = O(n) = O(|A|)$. □

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: $\max\limits_{F \subseteq D} \frac{v(F)}{u(F)} \leq 3$. (Hint: you may need Theorem 1 for this subquestion.)

   Solution. (a) A set $I \subset D$ is an independent set if and only if for any two triples in I are disjoint.

(b)

Algorithm 1: Greedy-MAX

```
Sort all the triples in $D$ by weight function $c(\cdot)$ so that
  $c(d_1) \geqslant c(d_2) \geqslant ... \geqslant c(d_m)$;
$S \leftarrow \varnothing$;
for $i = 1$ to $m$ do
    if for any triple $d$ in $S$, $d$ and $d_i$ are disjoint then
        $S \leftarrow S \cup \{d_i\}$
return $S$;
```

(c) We set $A = \{1, 2\}, B = \{3, 4\}, C = \{5, 6\}$, and $c(2, 3, 5) = 4, c(1, 3, 5) = 3, c(2, 4, 6) = 2$, and for any other triple $d \in D$, $c(d) = 0$. If using the Greedy-MAX algorithm, we will choose $c(2, 3, 5) = 4$ and $c(1, 4, 6) = 0$ with total weight is equal to 4. However if we choose $c(1, 3, 5) = 3, c(2, 4, 6) = 2$, and the total weight is 5.

(d) For $i \in \{1, 2, 3\}$, we denote that a set F_i is an i-th independent set if and only if for any two triples in F_i, the the numbers in the i-th dimension are not equal. We denote $\mathcal{I}_i$ as a collect of all i-th independent set. Then we try to prove that $M_i = (D, \mathcal{I}_i)$ is a matroid.

(Hereditary) A set $A \in \mathcal{I}_i$ means $\forall d$ in A, the numbers in the i-th dimension are not equal. So for any subset of A, the numbers in the i-th dimension are not equal as well. So M_i satisfies hereditary.

(Exchange property) Suppose $A, B \in \mathcal{I}_i$ and $|A| < |B|$. We can make sure that there must be a triple d in B that is different with any triple in A in the i-th dimension, otherwise we will find $|A| \leqslant |B|$. Then we have $A \cap d \in \mathcal{I}_i$ because in the i-th dimension d is different with any triple in A. Then M_i satisfies hereditary.

So $M_i = (D, \mathcal{I}_i)$ is indeed a matroid.

If we the independent set we defined in (a) I, then we get an independent system $(D, \mathcal{I})$. Obviously $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3$. Now according to **Theorem 1**, we have $\max\limits_{F \subseteq D} \frac{v(F)}{u(F)} \leq 3$.

□

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 $\max\limits_{F \subseteq E} \frac{v(F)}{u(F)} \leq k$, 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.*

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