Lab04-Matroid

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

* Name: Haoyi You   Student ID: 519030910193   Email: yuri-you@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.

   i. Heredity
   Assume $I_1 \subseteq I_2 \subseteq E, I_2 \in C$. That means $(V, E\backslash I_2)$ is connected. We have $E\backslash I_2 \subseteq E\backslash I_1$, so we know that $(V, E\backslash I_1)$ is connected.

   ii. Exchange property
   Assume $|V| = n, |E| = m, I_1 \subseteq E, I_2 \subseteq E, |I_1| > |I_2|$. If $\forall x \in I_1, \{x\} \cup I_2$ is not connected, then $\forall x \in I_1 - I_2$, x is connected to v_x that $d(v_x) = 1\ in\ E - I_2$. Since $E - I_2$ is connected, the x is only connected to one v_x that $d(v_x) = 1\ in\ E - I_2$. We divide the set of V into two parts. V_1 includes the v that satisfies $d(v) = 1$,others are in V_2.
   $\forall x \in I_1 - I_2$ two vertices of x are v_1, v_2. From the proof above, we assume $v_1 \in V_1, v_2 \in V_2$. Since $E - I_1$ is connected and (v_1, v_2) is not in $E - I_1$, there exists another vertex v_3 in V_2 that satisfies (v_3, v_1) is in $E - I_1$. So there at least $|V_1|$ edges that are in $E - I_1$ but not in $E - I_2$,these edges are all belongs to $I_2 - I_1$.
   As a result, $|I_1| - |I_2| = |I_1 - I_2| - |I_2 - I_1| \le |V_1| - |V_1| = 0$, which is against with the assumption $|I_1| > |I_2|$.

   □

   (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. Let $\mathbf{C}$ be the collection of all subsets of A that contains no more than k elements. Here we prove $(A, \mathbf{C})$ is a matroid.

   i. Heredity

$$\begin{gathered} \forall E \in \mathbf{C} \Rightarrow |E| \le k \\ \forall E_1 \subseteq E, |E_1| \le |E| \Rightarrow E \in \mathbf{C} \end{gathered} \tag{1}$$

   ii. Exchange property
   If $E_1, E_2 \in \mathbf{C}$, assume $|E_1| = a > b = |E_2|$. $E_1 - E_2$ is not an empty set. So there exists an element $x \in E_1 - E_2$. $|E_1 \cup \{x\}| = b + 1 \le a \le k$, so $E_1 \cup \{x\} \in \mathbf{C}$

   So the algorithm can refer to Algorithm **??**:

Algorithm 1: Choose k Numbers with Biggest Sum

Input: Array data$[n]$, integer k
Output: Array answer$[k]$

```
quicksort(data)
Initialize Answer[k]
for i = 1 to k do
    Answer[i] ← data[i]
return Answer;
```

$\square$

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. **??**. 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 solution is in Tab.**??**

   Table 2: Answer

t	1	2	3	4	5	6	7
$task$	5	4	6	3	7	2	1

   Only task 1 and 2 are delayed,which means the optimal penalty is 30. $\square$

   (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. Assume there are n elements in A, so we only need to ensure the complexity of algorithm is $O(n)$.
   According to the equivalent theorem,we only need to check $\forall t, 1 \leq t \leq n$,whether $N_t(A) \leq t$ is correct or not. Here is the Algorithm **??**

   Algorithm 2: Determine whether A is independent

   Input: deadline array$\{d_1, d_2 ... d_n\}$
   Output: bool flag

   ```
   end_number[n] ← [0, 0....0]
   for i = 1 to n do
       end_number[d_i] ← end_number[d_i] + 1
   work_number ← 0
   for i = 1 to n do
       work_number ← work_number + end_number[i]
       if work_number > t then
           return False
   return True
   ```

   There are only 2 loops with n times,so the complexity is $O(n)$. $\square$

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. **??** is not optimal.

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

Solution.

(a) i. Definition:
Assume $\mathbf{C} \subseteq D$. $\mathbf{C}$ is an independent set if $\forall a, b \in \mathbf{C}$, a, b are disjoint.

ii. Proof(Hereditary):
If $\mathbf{C}$ is independent, $\forall \mathbf{C}' \subseteq \mathbf{C}, \forall a, b \in \mathbf{C}'$,$a, b$ is also $\in \mathbf{C}$,so a, b are disjoint.

Here we can get an independent system $(E, \mathcal{C})$.

(b)

Algorithm 3: Greedy Algorithm

Input: A set $D = \{a_1, a_2, \ldots, a_n\}$, a value function $c()$
Output: A set *Answer* of with maximum total weight

```
quicksort(D, c())
Answer ← ∅;
for i = 1 to n do
    if Answer ∪ {a_i} is independent then
        Answer.push(a_i)
return Answer;
```

(c) Let $\mathbf{D} = \{1, 2\} \times \{1, 2\} \times \{1, 2\}$, $c(x, y, z) = xyz + |x - y| + |y - z| + |z - x|$.
If we use greedy algorithm, we will first choose $(2, 2, 2)$ with $c(2, 2, 2) = 8$,and then choose $(1, 1, 1)$.The sum is $8 + 1 = 9$.
However,if we choose $(2, 2, 1)$ and $(1, 1, 2)$. The sum is $c(2, 2, 1) + c(1, 1, 2) = 6 + 4 = 10$.

(d) i. First we define a independent system that $(E, \mathcal{I}_1)$ that $\forall F \in \mathcal{I}_1 \iff F \subseteq E$ && $\forall a, b \in F, a = (x_1, y_1, z_1), b = (x_2, y_2, z_2), x_1 \neq x_2$.According to this we can define another two independent systems $(E, \mathcal{I}_2)$ with $y_1 \neq y_2$ and $(E, \mathcal{I}_3)$ with $z_1 \neq z_2$ in the same way.

ii. Here we will prove all of this independent system are matroids. Obviously, We only need to prove $(E, \mathcal{I}_1)$ is a matroid.

iii. Hereditary: in the same way as the $(E, \mathcal{C})$.

iv. Exchange property:
If there are two sets X, Y with $|X| = x > y = |Y|$, both of them are independent set.Since $x > y$ and $\forall a(x_a, y_a, z_a), b(x_b, y_b, z_b) \in X$,$x_a \neq y_a$,there are x categories of different $x_i (i \in X)$ in X. But there are less than x element in Y, so there exists an element $t \in X, \forall s \in Y, x_t \neq x_s$,which indicates $Y \cup \{t\}$ is also independent.

v. Refer to **??**, $\mathcal{C} = \mathcal{I}_1 \cap \mathcal{I}_2 \cap \mathcal{I}_3$. So $\max\limits_{F \subseteq D} \frac{v(F)}{u(F)} \leq 3$.

□

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.