Lab04-Matroid

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

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

$*$ Name:Renyang Guan Student ID:519021911058 Email: guanrenyang@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. ***Hereditary property:***
   For $\forall A \subset B$, where $\forall B \in C$,

   $$B \backslash A \supset \emptyset \Rightarrow B \backslash A = \{e_1, e_2, \cdots, e_k\}(k \geq 1)$$

   Since $(V, E \backslash B)$ is connected, for $\forall D \subseteq B \backslash A$: $(V, (E \backslash B) \cup D)$ is connected.
   Considering $(V, (E \backslash B) \cup D) \subset (V, E \backslash A)$, $(V, E \backslash A)$ is connected.
   In other words, $A \in C$.
   The property of hereditary has been proved.

   ii. ***Exchange property:***

   Let us figure out a way to change graph G_B into $G_A = (V, E - A)$.
   For any edge e in E, it could be sorted into one of the categories below:
   Case 1: $e \notin A \cup B$
   *No-Changing.*The edge e will remain in G_A because $e \in E - a$.

   Case 2: $e \in A \cap B$
   *No-Changing.*The edge e is excluded from both G_A and G_B.

   Case 3: $e \in A \backslash B(e \in B \backslash A$ is discussed in this case in the form of $e')$
   *Change.*The edge e will be removed from G_B. Such case should be divide into sub-cases depending on whether e is a cut edge.
   Case 3-1: e is a cut edge of G_B.
   If e is removed from G_B, G_B will be disconnected. There must be one edge $e' \in B \backslash A$ added to the graph, such that e' connects the two connected components which e connected.
   Case 3-2: e is not a cut edge of G_B.
   e ought to be removed from G_B with no need to add a corresponding edge.

   Let us define such notations:
   E_1 denotes the set of e mentioned in *Case 3-1.*
   E_2 denotes the set of e' in corresponding with e.

   Since $|E_2| = |E_1| \leq |A| < |B|$ and $E_2 \subset B \backslash A$, for any edge $x \in B \backslash A - E_2$, x is not a cut edge, which means $E/(A \cup \{x\})$ is connected. In other words,

   $$A \cup \{x\} \in C$$

$\square$

(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. ***My algorithm to choose:***

Algorithm 1: My Algorithm

```
Input: A[1, ··· , n]
Output: A[1, ··· , k]
Sort all elements in A into ordering a_1 ≤ a_2 ≤ ··· ≤ x_n;
T ← ∅;
for i ← 1 to n do
    if |A ∪ {x_i}| ≤ k then
        A ← A ∪ {x_i};
```

Prove that (A, C) is a matroid.

i. ***Hereditary property:***
$\forall X \subset Y, \quad Y \in C$, since $|Y| \leq k$:
$$|X| \leq k$$
Because $Y \subset A$, such that
$$X \subset A$$
Combining the two formulas above,
$$X \in C$$
.
The property of hereditary has been proved.

ii. ***Exchange property:***
$\forall X, Y \in C$ and $|X| < |Y|$, such that
$$|X| < |Y| \leq k$$
Denote D as
$$D = Y \backslash X = \{d_1, d_2, \cdots, d_k\}(k \leq 1)$$
$\forall d \in D$:
$$|X \cup \{d\}| \leq k$$
Because $Y \subseteq A$ and $\{d\} \subseteq Y$:
$$X \cup \{d\} \subseteq A$$
Combining the formulas above:
$$X \cup \{d\} \in C$$
The property of exchange has been proved yet.

$\square$

$\square$

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. 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 final schedule is $< a_5, a_6, a_4, a_3, a_7, a_1, a_2 >$.
The optimal penalty is $\omega_1 + \omega_2 = 30$.

$\square$

   (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.
Lemma of Equivalence:
That the set A is independent is equivalent to that for $t = 0, 1, 2, \cdots, n, N_t(A) \leq t$, where $N_t(A)$ denotes the number of tasks in A whose deadline is t or earlier.

The algorithm to determine whether A is dependent in time $O(|A|)$

Algorithm 2: Algorithm of determining in time $O(|A|)$

Input: A set A of tasks
Output: *independent* or *unindependent*

1 $n \leftarrow |A|$;
2 $N[n+1] \leftarrow= \{0, 0, \cdots, 0\}$;
3 **for** $i \leftarrow 1$ **to** n **do**
4 $\quad d_i \leftarrow deadline\ of\ task\ A[i]$;
5 $\quad$ **if** $d_i \leq n$ **then**
6 $\quad\quad N[d_i] \leftarrow N[d_i] + 1$;
7 **for** $t \leftarrow 1$ **to** n **do**
8 $\quad N[t] \leftarrow N[t-1] + N[t]$;
9 $\quad$ **if** $N[t] > t$ **then**
10 $\quad\quad$ **return** *unindependent*
11 **return** *independent*

Description of the algorithm:
After analysing the property of $N_t(A)$ we could get such recursive formula easily:

$$N_t(A) = \begin{cases} 0 & t = 0 \\ N_{t-1}(A) + number\ of\ tasks\ whose\ dealine\ is\ t & t \geq 1 \end{cases}$$

The **for** loop starting in line 3, which loops for $|A|$ times, traverses the set A. It set the number of tasks whose deadline is t in $N[t]$.

The **for** loop starting in line 7, which loops for $|A|$ times, calculate $N_t(A)$ according to the recursive formula above and set $N_t(A)$ to $N[t]$. At the same time, it checks every $N_t(A)$ whether or not it is greater than t, where t is an integer ranges from 1 to n.

The total time complexity of the algorithm is $O(|A|)$.

□

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

Solution.

(a)

Definition 2 ((D, C))**.** *Let A denote any subset of S, in which any two triples $x, y \in A$, x and y are disjoint. C is the collection of all possible subset As.*

Proof of independent property of (D, C)**.**
$\forall A \subset B$, $\forall B \in C$, $\forall x, y \in A$

$$x, y \in B \Rightarrow x, y \text{ are disjoint} \Rightarrow A \in C$$

□

(b)

Algorithm 3: Greedy-Max For MAX-3DM

```
Input: A set of triples D
Output: A collection A of disjoint triples
Sort all elements in D into ordering c(x_1) ≥ c(x_2) ≥ ··· ≥ c(x_n);
A ← ∅;
n ← |D|;
for i ← 1 to n do
    if A ∪ x_i ∈ C then
        A ← A ∪ {x_i};
return A
```

(c) ***A Counter Example:***

$$X = \{1, 2\} \quad Y = \{3, 4\} \quad Z = \{5, 6\}$$

In such case:

$$D = \{(1,3,5), (1,3,6), (1,4,5), (1,4,6), (2,3,5), (2,3,6), (2,4,5), (2,4,6)\}$$

The function $C(*)$ is shown below:

$$\begin{aligned} (1,3,5) &\to 9 \quad (2,4,6) \to 1 \\ (2,4,5) &\to 8 \quad (1,3,6) \to 8 \\ (1,4,5) &\to 1 \quad (1,4,6) \to 1 \\ (2,3,5) &\to 1 \quad (2,3,6) \to 1 \end{aligned} \tag{1}$$

It is obvious that the optimal solution is $\mathcal{F} = \{(2,4,5),(1,3,6)\}$ where the maximum is 18 while the solution of Greedy-Max algorithm is $\mathcal{F} = \{(1,3,5),(2,4,6)\}$ where the sum is 10.

(d) According to the Theorem. 3, to prove the origin proposition we just need to prove the Lemma. 1 and Lemma. 2 below.

Lemma 1.
Let the set A satisfy:

$$A = \{(x,y,z)|x \in X, y \in Y, z \in Z, \text{ and } \forall (x_1,y_1,z_1),(x_2,y_2,z_2) \in A : x_1 \neq x_2\}$$

and $\mathcal{F}_x$ is the collection of all the subsets of A.
The system $(D, \mathcal{F}_x)$ is a matroid.

Lemma 2. *We define matroid $(D, \mathcal{F}_y)$ and $(D, \mathcal{F}_z)$ similar to that in Lemma. 1. $\mathcal{F}_x$, $\mathcal{F}_y$, and $\mathcal{F}_z$ satisfy:*

$$\mathcal{F}_x \cap \mathcal{F}_y \cap \mathcal{F}_z = \mathcal{F}$$

Proof of Lemma. 1.
Hereditary property:
$\forall B \in \mathcal{F}_x, \forall A \subset B$:

$$\begin{aligned} &\forall (x_1,y_1,z_1),(x_2,y_2,z_2) \in A \\ \Rightarrow\ &(x_1,y_1,z_1),(x_2,y_2,z_2) \in B \\ \Rightarrow\ &x_1 \neq x_2 \\ \Rightarrow\ &A \in \mathcal{F}_x \end{aligned}$$

Exchange property:
$\forall A, B \in \mathcal{F}_x$ and $|A| < |B|$, according to the definition of $\mathcal{F}_x$, we could easily get:

$$\forall (x_1,y_1,z_1),(x_2,y_2,z_2) \in A : x_1 \neq x_2$$

$$\forall (x_1,y_1,z_1),(x_2,y_2,z_2) \in B : x_1 \neq x_2$$

Let set $A_x = \{x|\ \forall (x,y,z) \in A\}$ and set $B_x = \{x|\ \forall (x,y,z) \in B\}$
Because $|A| < |B|$ and that the x value of every two different triples in A are different,which is similar to B, we have:

$$|A_x| < |B_x|$$

As a result of this,

$$\exists x \in B_x,\ \forall x' \in A_x :\ x \neq x'$$

(Otherwise B_x is a subset of A_x, which conflicts with $|A| < |B|$.)
and

$$\forall y \in Y,\ \forall z \in Z, (x,y,z) \notin A$$

According to the definition of B_x:

$$\exists (x,y,z) \in B \backslash A,\ \forall (x',y',z') \in A :\ x \neq x' \Rightarrow A \cup \{(x,y,z)\} \in \mathcal{F}_x$$

□

Proof of Lemma. 2.
Every element of $\mathcal{F}$ is in $\mathcal{F}_x$, $\mathcal{F}_y$, and $\mathcal{F}_z$

$\forall A \in \mathcal{F},\ \forall (x_1, y_1, z_1), (x_2, y_2, z_2) \in A$: $x_1 \neq x_2, y_1 \neq y_2, z_1 \neq z_2$

$$\begin{aligned} x_1 \neq x_2 &\Rightarrow A \in \mathcal{F}_x \\ y_1 \neq y_2 &\Rightarrow A \in \mathcal{F}_y \ \Rightarrow A \in \mathcal{F}_x \cap \mathcal{F}_y \cap \mathcal{F}_z \\ z_1 \neq z_2 &\Rightarrow A \in \mathcal{F}_z \end{aligned} \tag{2}$$

Any element not in $\mathcal{F}$ is not in $\mathcal{F}_x$(or $\mathcal{F}_y$ or $\mathcal{F}_z$):

$\forall A \notin \mathcal{F},\ \forall (x_1, y_1, z_1), (x_2, y_2, z_2) \in A :$

$$x_1 \neq x_2 \text{ or } y_1 \neq y_2 \text{ or } z_1 \neq z_2 \Rightarrow (A \notin \mathcal{F}_x) \vee (A \notin \mathcal{F}_y) \vee (A \notin \mathcal{F}_z)$$

□

□

Theorem 3. *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.