

Décomposition de matroïdes orientés: Liste d'algorithme

MARIN Yann

26 janvier 2022

Résumé

Ce document est une liste d'algorithmes possible pour résoudre le problème suivant :

Etant donnée : $E=1 : n$ un ensemble de point, M un ensemble de modèle sur E , B l'ensemble des $\binom{n}{d}$ d-uplets et $X \subseteq B$.

Trouver : la liste L des $P \subseteq E$ tel que $B_P \subseteq X$ où $B_P \subseteq B$ est l'ensemble des d-uplets que l'on peut créer à partir de P . De tels P sont appelés des sous-modèles fixes.

Ce problème sera traité dans le cas général mais aussi dans le cadre qui nous intéresse, c'est à dire avec $d = 4$ et $n \simeq 100$.

1 Introduction

Glossaire :

- $E=1 : n$ un ensemble de points.
- B l'ensemble des $\binom{n}{d}$ d-uplets de E . On parlera parfois de "base", faisant référence au matroïde orienté affine sur E .
- $X \subseteq B$ un ensemble de d-uplet. On parlera parfois de "bases fixes", faisant référence aux bases de même signe.
- $P \subseteq E$ un ensemble de point. On parlera parfois de sous-modèles.
- $B_P \subseteq B$ l'ensemble des d-uplets de P .

2 Premiers algorithmes :

Dans ce qui va suivre nous présenterons différents algorithmes, pour la plupart très médiocres, mais qui permettent de cerner le problème.

2.1 Tester $\simeq 2^{|X|}$ sous-ensembles de X :

- Pour tout $X_i \subseteq X$, chercher $P_{X_i} \subseteq E$ tel que $X_i \subseteq B_{P_{X_i}}$.
- Tester si $|X_i| = |B_{P_{X_i}}|$.
 - Si oui, alors on ajoute P_{X_i} à la liste de sous modèles

Points positifs	Points négatifs
Efficace si $ X \ll n$.	Inefficace pour $ X \geq n$
Très simple à implémenter	$ X \ll n$ que si d est grand*.

TABLE 1 – *Pour couvrir E , il faut au moins $n-d+1$ d-uplets. Or dans le cas pratique on cherche un recouvrement en sous-modèles fixes, ce qui demande que X recouvre E . On aura donc presque toujours $|X| \gg n$

Cette algorithme demande de savoir parcourir les sous-ensembles de X pour les avoir une et une seule fois. On a donc deux choix :

1. On part de X et on enlève des éléments.
2. On part des singletons de X et on ajoute des éléments.

On utilise 1. si on suppose que L a beaucoup d'éléments et que les sous-modèles maximaux sont grand.

On utilise 2. dans le cas contraire.

Il faut encore trouver une manière de parcourir tous les X_i de façon à ce que l'on s'arrête lorsque l'on de changer X_i si $P_{X_i} \in L$.

2.2 Tester $\simeq 2^{|E|}$ sous-ensemble de E :

- Pour tout $P_i \subseteq E$, chercher $X_i \subseteq X$ tel que $X_i \subseteq B_{P_i}$ et X_i est maximal pour cette propriété.
($X_i = B_{P_i} \cap X$)
- Tester si $|X_i| = |B_{P_i}|$.
- Si oui, alors on ajoute P_{X_i} à la liste de sous modèles

Points positifs	Points négatifs
Efficace pour tout $ X $.	Inefficace pour $ E $ grand

Comme précédemment, pour faire cette algorithm il faut choisir un ordre de parcourt des P_i , et une manière de filtrer les sous-ensembles inutiles.

3 Le problème dual :

3.1 Enoncé du problème :

Etant donnée : $E=1$: n un ensemble de point, M un ensemble de modèle sur E , B l'ensemble des $\binom{n}{d}$ d-uplets et $X \subseteq B$, $\overline{X} = B - X$.

Trouver : La liste L des $P \subseteq E$ tel que $B_p \cap \overline{X} = \emptyset$ où $B_p \subseteq B$ est l'ensemble des d-uplets que l'on peut créer à partir de P . De tels P sont appelés des sous-modèles fixes.

3.2 Tester $\simeq 2^{|E|}$ sous-ensembles de E :

- Pour chaque $P_i \subseteq E$, chercher $\overline{X}_i \subseteq E$ tel que $\overline{X}_i \subseteq B_{P_{\overline{X}_i}}$.
- Si $|\overline{X}_i| = 0$ on ajoute P_i à L .

3.3 Tester $\simeq |\overline{X}|$ sous-ensembles sur $\simeq 2^{|E|}$ sous-ensembles de E :

- Pour tout $x \in \overline{X}$, éliminer tous les sous-ensembles $P \subseteq E$ tels que $x \in P$.
- L est la liste des P restants.

Un exemple de programme utilisant ce principe :

Algorithm 1 Algorithmme

Require: \overline{X}, E **Ensure:** L

```
 $L \leftarrow [E]$ 
for  $x \in \overline{X}$  do
  for  $P \in L$  do
    if  $x \in P$  then
      for  $p \in x$  do
        if  $P \cup p \notin L$  then
           $L \leftarrow L \cup (P \cup p)$ 
        end if
      end for
    end if
  end for
end if
end for
end for
```

Une autre version :

3.4 Faire $\simeq 2^{|E|-d}|\overline{X}||X|$ test :

Algorithm 2 Algorithmme

Require: \overline{X}, E, X **Ensure:** L

```
 $L \leftarrow [P_{X_i} \text{ for } X_i \in X]$ 
for  $p \in E$  do
  for  $P \in L$  do
    if  $P \cup p \notin L$  then
      if  $x \notin P \cup p$  for  $x \in \overline{X}$  then
         $L \leftarrow L \cup (P \cup p)$ 
      end if
    end if
  end for
end for
end for
```

4 Différentes méthodes importantes :

Pour utiliser au mieux les algorithmes précédents, il faut pallier à deux sous-problèmes importants :

- Trouver un parcourt des sous-ensemble de E pour tous les vérifier une et une seule fois.
- Trouver un un tel parcourt qui ne vérifie pas les sous-ensembles des sous-modèles maximaux.

4.1 Parcours de E :

On cherche une manière de visiter les sous-ensemble de E une et une seule fois, soit en partant de E,