CR2 Décomposition de matroïdes orientés.

MARIN Yann

4 février 2022

Résumé

Dans cette deuxième semaine, j'ai commencé à faire une liste des différents types d'algorithmes pour résoudre le problème de recouvrement par sous-modèles fixes. Cette liste permettrait d'utiliser le bon algorithme selon ce que l'on sait sur l'ensemble de modèle que l'on souhaite traiter. Tous ces algorithmes ont fait ressortir un problème algorithmique commun que l'on expliquera dans la partie 2.

Table des matières

1 Liste d'algorithmes 1

1.1 Possibilité d'amélioration . . . 2

2 Problème algorithmique : combinaisons avec facteurs interdits. 3

2.1 Problème de génération de combinaisons uniques avec facteurs interdits . . . 3

3 Piste de chose à faire pour la semaine 3 : 4

Glossaire

- $E = \{1 :: n\}$ ensemble de n points.
- d la dimension d'un espace affine, par défaut d=4.
- B l'ensemble des $\binom{n}{d}$ d-uplets sur E.
- $X \subseteq B$ un ensemble de d-uplets.
- $P \subseteq E$ un ensemble de points.
- $S_P \subseteq B$ l'ensemble des $\binom{|P|}{d}$ d-uplets dans P.
- $L = \{P \subseteq E | S_P \subseteq X\}$
- $\overline{X} = E - X$
- $\overline{L} = \{P \subseteq E | \exists x \in \overline{X} \text{ tq } x \in S_P\}$
- $max(L) = \{P \in L | \text{ si } P \subseteq P' \in L \text{ alors } P = P'\}$

1 Liste d'algorithmes

Rappellons le problème donné :
Etant donné E et X, trouvé tous les P tels que $S_P \subseteq X$.
Il y à plusieurs façons de résoudre ce problème selon ce que l'on sait sur E,X,$\overline{X}$ et nos présomptions sur L et E-L. Dans tous les cas, nous sommes obligés de tester une propriété sur un très grand nombre de sous ensembles de E ou de X. Il nous faut donc trouver la manière la plus adapté.

tableau	Parcours de $\emptyset$ à L	Parcours de E à max(L)
Sur E	$\|E\| << \|X\|$, $\|L\| << \|E - L\|$, on utilise (P1)	Si $\|E\| << \|X\|$, $\|E - L\| << \|L\|$, on utilise (P1) ou (P2)
Sur X	$\|X\| << \|E\|$, $\|L\| << \|E - L\|$, on utilise (P1)	$\|X\| << \|E\|$, $\|E - L\| << \|L\|$, on utilise (P1)
Sur $\overline{X}$	$\|\overline{X}\| << \|E\|$, $\|L\| << \|E - L\|$	$\|\overline{X}\| << \|E\|$, $\|E - L\| << \|L\|$

TABLE 1 – Différents algorithmes et leur utilisation selon E,X,L, la liste détaillé est dans "Decomposition de matro ıdes orient es : Liste d'algorithme"

Dans tous les cas, nous avons deux ordres de parcourt :
-Soit on commence par les d-uplets de B pour arriver à L ("Parcours de $\emptyset$ à L").
-Soit on commence par E pour arriver à max(L). ("Parcours de E à max(L)").

Dans tous les cas, On doit parcourir un assez grand sous-ensemble de $2^{|E|}$, et de préférence de manière unique sans oublier de sous ensemble et sans en traité un deux fois. Un tel parcourt est d'ailleurs le problème de la section 2.

Dans chacun de ses sous-ensembles P, nous allons devoir vérifier l'une des deux propriétés suivantes :
-(P1) : Est-ce que $P \in L$?
Si oui, pour tout $P' \subseteq P$, alors $P' \in L$. (propriété héréditaire). Pour vérifier (P1) il suffit de tester $|S_P \cap X| = \binom{|P|}{d}$.
-(P2) : Est-ce que $P \in E - L$?
Si oui, pour tout $P \subseteq P'$, alors $P' \in E - L$.
Pour vérifier (P2) il suffit de regarder s'il existe $x \in \overline{X}$ tel que $x \in E - L$.

Enfin, on doit choisir si on trouve nos sous-ensembles à partir de E (des points), de X (des d-uplets fixes) ou de $\overline{X}$.

Ces algorithmes n'utilisent aucune astuce, mais peut être qu'une recherche sur les d-uplets les plus utiles de X permettrait de les améliorer.

DIGRESSION :
Dans tous les cas, un problème majeur demeure : il nous faut parcourir soit L à partir des d-uplets de X avec la propriété (P2), soit E-L à partir de E avec la propriété (P1).
Dans le premier cas, on part d'ensemble à d sommets et on essaye de rajouter des sommets jusqu'à avoir max(L). En supposant que L et X couvrent E, on doit parcourir tous les sous-ensembles de E en éliminant à chaque fois ceux qui n'ont pas la propriété (P2) et leurs parents.
Cela revient à tester les $\binom{n}{k}$ pour 1¡=k¡=n en éliminant à chaque étape les ensembles ne satisfaisant pas (P2) et ceux qui les contiendrons dans la suite.

LOGIQUE :
En terme logique, on cherche à tester l'ensemble des $P \in L$ tels que si l'on test $P \notin L$ alors on ne test aucun des $P \subseteq P' \notin L$.

DIGRESSION :
Dans le second cas, on part des ensembles auxquels il manque un seul sommet, et on essaye d'en retirer jusqu'à avoir max(L). En supposant ue L et X couvrent E, on doit parcourir tous les sous-ensembles de E en éliminant à chaque fois ceux qui n'ont pas la propriété (P1) et leur enfants.
Cela revient à tester les $\binom{n}{n-k}$ pour 1¡=k¡=n en éliminant à chaque étape les ensembles ne satisfaisant par (P1) et ceux qui les contiendrons dans la suite.

LOGIQUE :
En terme logique, on cherche à tester l'ensemble des $P \notin L$ tels que si l'on test $P \in L$ alors on ne test aucun des $P' \subseteq P$.

En fait, comme $\binom{n}{k} = \binom{n}{n-k}$ c'est le même problème. Et le résoudre dans le cadre général est le sujet de la section 2.

1.1 Possibilité d'amélioration

Une possibilité d'amélioration serait d'aulieu de commencer de $E = \binom{n}{n}$ ou de $X \cup \overline{X} = \binom{n}{d}$ de commencer à $\binom{n}{k}$ pour un k arbitraire et d'augmenter (ou de diminuer) k.
Cela serait utile si on avait une taille minimum (ou maximum) de sous-modèles fixes. En effet, on pourrait tout de suite calculer les sous-ensembles de $L(k) = \{P \in L||P| = k\}$ en calculant les $\binom{n}{k}$ sous ensembles de E et en testant si chaque sous-ensemble est dans L(K) ou non. A partir de L(K), on peut

facilement avoir L(i) pour tout $i < k$. De même, à partir de $\overline{L}(K)$ on peut facilement avoir $\overline{L}(i)$ pour $k < i$.

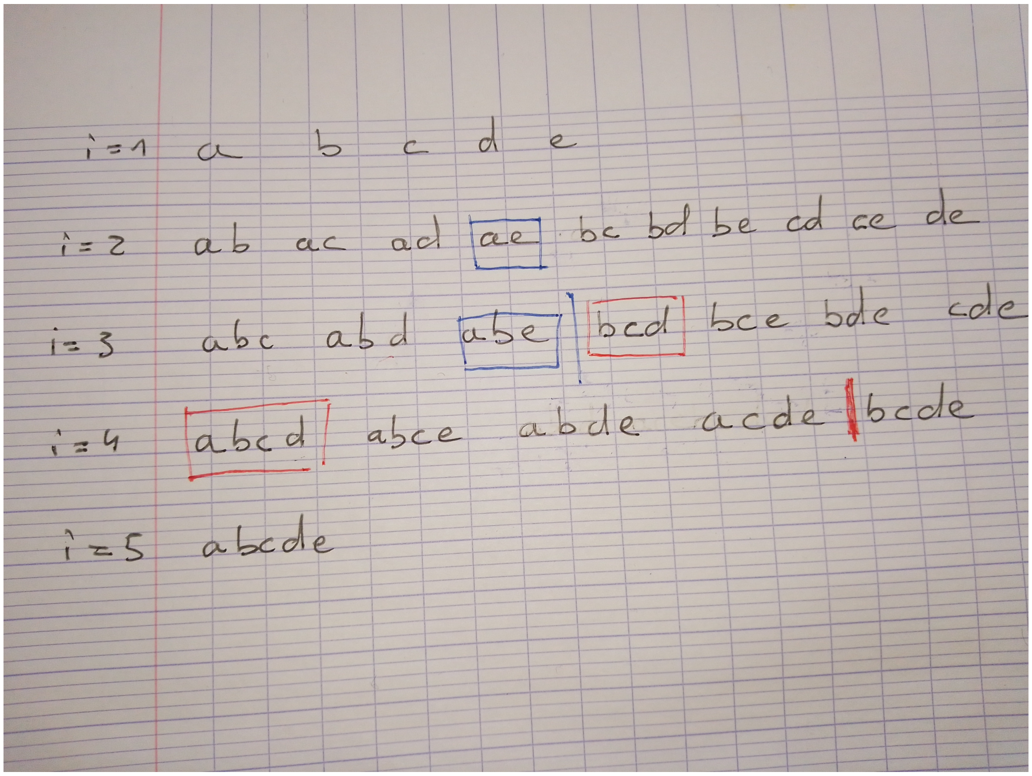
2 Problème algorithmique : combinaisons avec facteurs interdits.

2.1 Problème de génération de combinaisons uniques avec facteurs interdits

Soit $A = \{a, b, c...\}$ un ensemble ordoonée $a < b < c...$, on cherche à générer les $2^{|A|}$ sous-ensembles en commençant par les ensembles de taille 1, puis 2 etc... à chaque étape, on vérifier sur ces ensembles une propriétés (P) qui est héréditaire. Et si $m \in A^k$ satisfait la propriété (P), alors on ne doit pas retrouver $m' \in A^i$ tel que $m \subseteq m'$.

Le problème de générer de façons unique tous les sous-ensemble de A est facile, la difficulté est de supprimer les facteurs interdits.

En effet, on peut facilement générer tous les mots de taille k à partir des mots de taille k-1 rangés lexicographiquement. Mais interdire un sous-modèle est bien plus compliqué.



Par exemple ici, nous avons générer les sous-ensembles de A=a,b,c,d,e dans l'ordre lexicographique. On peut remarquer qu'ils sont alors rangés par préfixes. Lorsque l'on retire un motif (ici par exemple ae) on peut voir qu'il n'apparait plus dans les mots qui sont formés à partir des préfixes suivants (à droite de la ligne bleu). Par contre il apparait dans ceux générer par les préfixes précédents (à gauche de la ligne bleu). Une solution serait de faire de ce motif le premier préfixe de l'alphabet, mais cela ne marche pas pour plusieurs motifs à interdire.

A défaut d'une solution, des optimisations informatiques sont possibles, on pourrait par exemple ranger les facteurs interdits dans une liste et vérifier pour chaque sous ensemble s'il n'est pas contenu dans un élément de la liste.

3 Piste de chose à faire pour la semaine 3 :

En plus de faire tout ce que je n'ai pas fais dans ce qui était dans les pistes de chose à faire de la semaine 2 :

— Continuer de lister les différents algorithmes et les implémenter.
— Continuer de réfléchir au problème de motifs interdits.
— développer l'idée de la partie 1.1.
— Continuer de lire le livre de Knuth sur la génération de combinaisons.
— Réfléchir, étant donné deux sous modèles fixes et un d-uplet non fixe qui possède 3 points dans l'un et 1 points dans l'autre les différentes implications du signe. Et dans un cadre plus général, les liens entre d-uplets non fixe et rapport entre les sous-modèles fixes.