Décomposition de matroïdes orientés:

Rapport de stage n°1

MARIN Yann

21 février 2022

Ce premier document de stage sert à présenter le sujet et à rendre compte de mon avancement au bout d'un mois et demi.
Nous ferons donc une courte introduction du sujet après quoi nous présenterons les problèmes principaux puis mes différentes solutions. Enfin nous présenterons le travail qui reste à effectuer.
A noter qu'une majeure partie de ce document ne demande aucune notion sur les matroïdes orientés ni même un quelconque niveau en mathématiques, nous alternerons en effet entre une redaction générale et sa formalisation logique.
Par ailleurs, tout le code ainsi que des compte-rendus hébdomadaire et des feuilles de route de chaque algorithme sont disponnibles sur github : https ://github.com/YannMarin/stage-info mais ne sont aucunement nécessaire à la lecture de ce document.

Table des matières

0.1 Glossaire

— $E = \{1 :: n\}$ ensemble de n points.
— M un modèle labelisé par les points de E.
— $\mathcal{M}$ un ensemble de modèles.
— d la dimension d'un espace affine, par défaut d=4.
— B l'ensemble des $\binom{n}{d}$ d-uplets sur E.
— $\mathcal{B}$ l'ensemble des bases 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.
— $L = \{P \subseteq E | S_P \subseteq X\}$
— $\overline{X} = E - X$
— $\overline{L} = \{P \subseteq E | \exists x \in \overline{X} \text{ tq } x \in S_P\}$
— $max(L) = \{P \in L | \text{ si } P \subseteq P' \in L \text{ alors } P = P'\}$
— $X_P = X \cap S_P$
— SM = "Sous-modèle".
— SMC = "Sous-modèle constant".

1 Introduction

Etant donné E un certain ensemble de n de points caractéristiques d'un ensemble de *modèles* $\mathcal{M}$, nous cherchons à décomposer E en sous-parties appellées *sous-modèle* tel que chaque modèle M de $\mathcal{M}$ soit composé de ces sous-modèles et soit caractérisé par les intéractions entre les différentes parties.

Pour cela, nous allons regarder l'ensemble B de tous les d-uplets (en pratique avec d=4) de E. Ces d-uplets sont appellés des *bases* si les d points qui le compose sont affinnements indépendant (ie : ne sont pas tous dans le même hyperplan). Si c'est le cas, en dimension 3 ils forment alors un tétraèdre. Ces d-uplets sont munis d'un signe (-,0,+), 0 dans le cas où ils sont affinnement dépendants et + ou - selon l'orientation du tétraèdre dans le cas contraire.

Sans perte de généralité, on triera B et les d-uplets dans l'ordre lexicographique.

Mathématiquement, si l'on note $\mathcal{B}$ l'ensemble des bases sur E alors (E,$\mathcal{B}$) est un *matroïde orienté* et l'application $\mathcal{X}$ qui à tout d-uplet associe son signe est appellé *chirotope*. Pour le moment, nous n'aurons besoin que de cette notion de d-uplet signé.

Définition 1.1 (Base constante/d-uplet constant) *Une base $b \in \mathcal{B}$ est dite constante sur l'ensemble de modèles $\mathcal{M}$ si le signe de b est le même dans tous les modèles $M \in \mathcal{M}$.*
Même lorsque l'on parlera de d-uplet constant on ne parlera en fait que des bases dont le signe est constant.

Définition 1.2 (d-uplet non constant) *Soit $b \in B$ l'ensemble des d-uplets de E. b est dit non constant sur $\mathcal{M}$ si ce n'est pas une base ou si son signe n'est pas le même dans tous les modèles $M \in \mathcal{M}$*

Définition 1.3 (Sous-modèle constant) *Soit $P \subseteq E$ un sous-modèle. P est un sous-modèle constant ssi tout d-uplet $b \subseteq P$ est constant.*
On abrégera sous-modèle par SM et sous-modèle constant par SMC.

Proposition 1.1 *Soit $P \subseteq E$. Alors $S_P \subseteq B$ l'ensemble des $\binom{|P|}{d}$ d-uplets dans P, détermine P.*

Remarque 1.1 *Soit $P \subseteq E$, P est constant si et seulement si tout d-uplet de S_P est constant. De plus, si l'on prend un certain ensemble S_P de $\binom{|P|}{d}$ d-uplets constants entièrement contenus dans un ensemble $P \subseteq E$ alors P est un sous-modèle constant.*

Proposition 1.2 (Hérédité) *Soit $P \subseteq E$ un sous-modèle constant et $P' \subseteq P$. Alors P est un sous-modèle constant. On dit que la propriété est héréditaire.*

Proposition 1.3 (Hérédité inverse) *Soit $P \subseteq E$ un sous-modèle non constant et P' tel que $P \subseteq P'$ alors P' est un sous-modèle non constant.*

Proposition 1.4 (Intersection) *Soit $P \subseteq E$ et $P' \subseteq E$ des sous-modèles constants. Alors $P \cap P'$ est un sous-modèle constant.*

Après ces définitions, notre objectif est de trouver une décomposition de E en sous-modèle constant significatifs. Pour cela la première étape est de trouver (et stocker) l'ensemble des sous-modèles constants maximaux pour l'inclusion.

1.1 Formalisation des questions du stage :

Soit $\mathcal{M}$ un ensemble de modèle sur $E = \{1 :: n\}$, et $X \subseteq \mathcal{B}$ un ensemble de bases constantes sur $\mathcal{M}$. Soit L l'ensemble des sous-modèles constants. On cherche :

1. $max(L) = \{P \in L |$ si $P \subseteq P' \in L$ alors $P = P'\}$, en d'autre termes les sous-modèles constants maximaux de E.
2. Etant donné max(L), trouver les recouvrements/partitions de E par le minimum de P de max(L). C'est à dire les recouvrements en sous-modèles maximaux avec le moins de partie possibles.
3. Etant donné une partition ou un recouvrement par sous-modèles fixes, trouver comment mesurer certaines caractéristiques entre les sous-parties.
4. Développer des liens concrets avec l'informatique graphique, la compression d'objet 3D et les moteurs d'applications intéractives.

Ces différentes questions mènnent elles mêmes à des sous-questions et à des pistes de recherches possibles. Cela sera évoqué dans les sections adéquates.

Ce stage est donc décomposable en trois parties :

1. Lister les SMC et les recouvrements (partie algorithmique).
2. Définitions de paramètres significatifs (partie mathématiques).
3. Optimisation et implémentation.

et ce rapport traite majoritairement de la première partie.

2 Un premier jeu de donnés expérimentales :

Dans ce stage, la plupart de nos expérimentations seront sur des donnés d'anthropologie. Ici, on regarde un ensemble de 290 modèles de crânes d'humains et de singes sur 16 points anatomiques significatifs (fournis par José Braga et Jacques Treil).

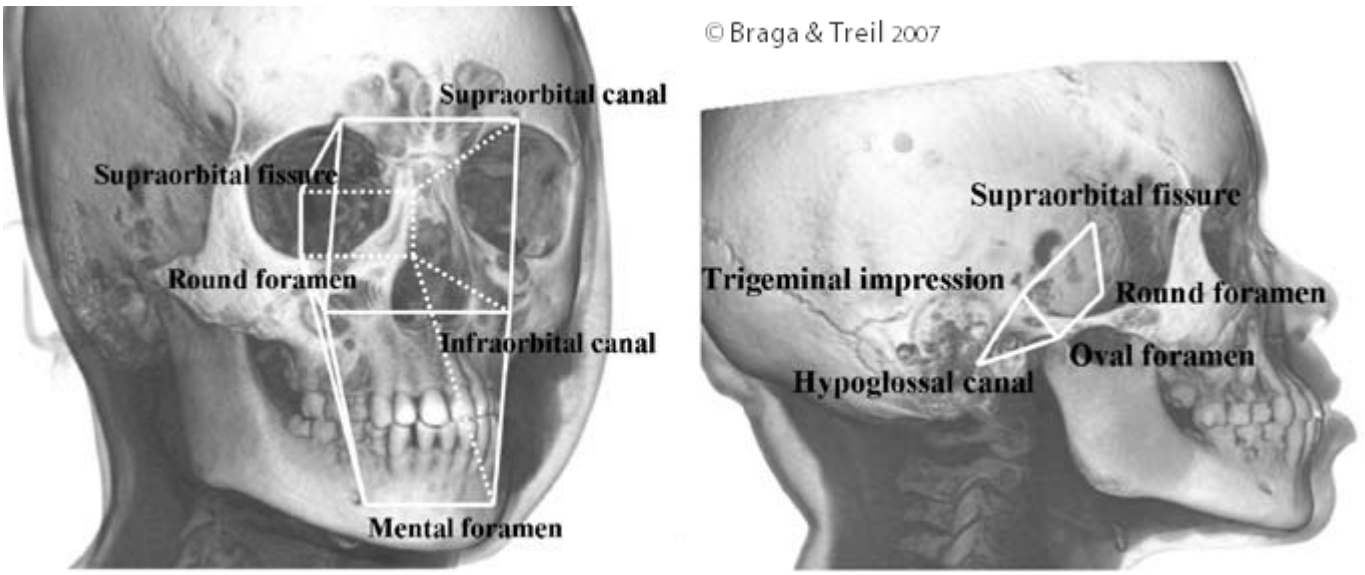


FIGURE 1 – 16 points anatomiques significatifs du crâne.

L'ensemble des points et des modèles sont modélisés et consultablent sur le logiciel OMSMO (https ://omsmo.lirmm.fr/). Le but de ce stage est d'y implémenter des programmes permettants d'analyser de tels ensemble de modèles, notamment en cherchant les recouvrements par sous-modèles constants maximum les plus significatifs.

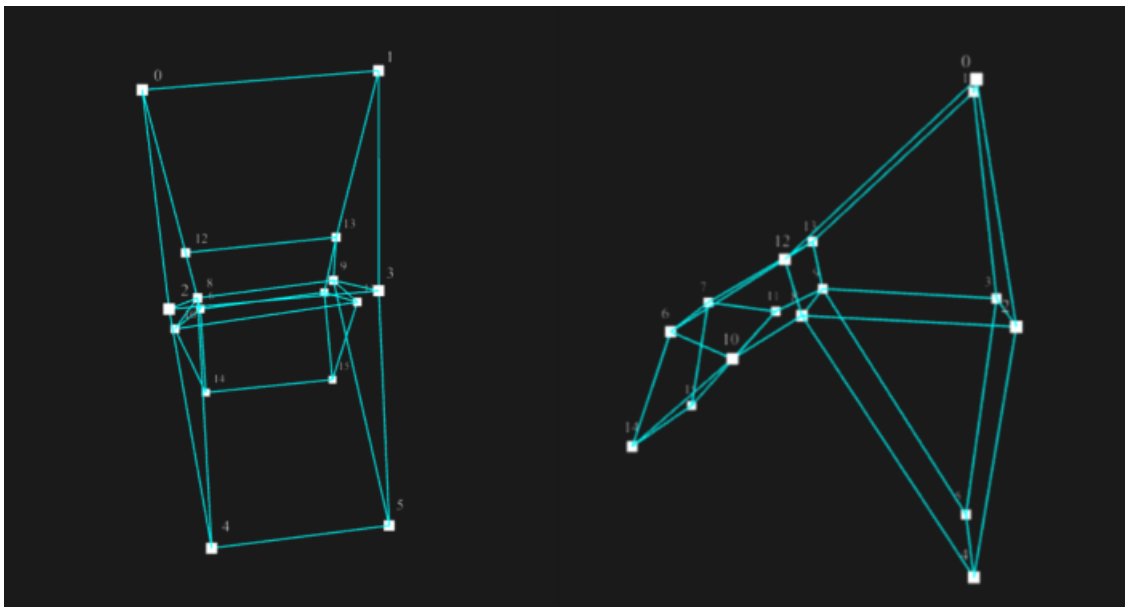


FIGURE 2 – Modélisation sur OMSMO.

3 Trouver les sous-modèles constants maximum

Problème 1 (Sous-modèles constants maximum) *Soit $\mathcal{M}$ un ensemble de modèle sur E=1 : :n, et $X \subseteq \mathcal{B}$. On cherche à trouver l'ensemble des sous-modèles constants, et à garder uniquement les sous-modèles constants maximum pour l'inclusion.*

Au total, nous avons 2^n sous-modèles sur E. Un nombre qui devient rapidement très grand lorsque n augmente. Trop même, surtout s'il faut tester si chacun de ces sous-modèles est constant ou non. Sur les 16 points on en a déjà 65536 et notre ambition est de passer à au moins 40 points tout en envisageant n=100. C'est pourtant une étape principale pour traiter les autres questions du stage. Il nous faut donc :

1. Minimiser le nombre de sous-modèle à tester.
2. Minimiser le nombre d'opération d'un test.
3. Minimiser le ratio espace/temps utilisé.

3.1 Ordre de parcours des sous-modèles

Quel que soit l'algorithme que l'on envisage pour tester les sous-modèle, il nous faut une manière de tous les générer de façon unique. Et de préférence, si l'on trouve un sous-modèle constant, on ne doit pas tester les sous-modèles qu'il contient. De même si l'on trouve un sous-modèle non-constant, il est préférable de ne pas tester tous les sous-modèles qui le contiennent. Un parcourt en profondeur (c'est à dire taille par taille) semble le plus pertinent car cela nous permettrait de tester les sous-modèles de taille t à partir de ceux de taille t+1 (ou t-1). On a alors deux possibilité : soit de parcourir les sous ensembles dans l'ordre croissant de leur taille ou dans l'ordre décroissant.
Dans le premier cas, nous devons choisir pour chaque sous modèle quel point rajouter tandis que dans le second cas, quel point retirer.
Ces deux sens sont tout à fait analogue. En effet, générer les combinaisons $\binom{n}{t}$ revient à la même chose que de générer les combinaisons $\binom{n}{n-t}$.
Le problème revient alors à parcourir les $\binom{n}{t}$ parties de taille t dans l'ordre croissant, de façon à ce que si l'une de ces parties posséde la propriété à tester (constance ou non constance) alors on ne doit pas tester les sous-parties de taille t+1 qui la contiennent, plus formellement :

Problème 2 (génération de combinaison avec facteur interdit :) *Soit $\mathcal{A} = \{1 :: n\}$ un alphabet avec n symboles. On souhaite générer toutes les combinaisons possibles de manière unique en commençant par celles de taille t dans l'ordre croissant.*
A toute étape k, on doit pouvoir éliminer certaines combinaisons et dans ce cas, dans toute étape ¿k on ne doit pas trouver de combinaison qui contiennent ces combinaisons interdites.

Pour l'heure nous n'avons pas réussis à trouver de solution satisfaisante pour résoudre ce problème. Notre solution actuelle est de générer les combinaisons par ordre lexicographique de taille t+1 à partir des combinaisons non-interdites de taille t. Cette méthode permet pour chaque combinaison interdite d'éliminer les combinaisons qui sont inférieur dans l'ordre lexicographique.
Mais cette solution élimine très peu de combinaisons interdites par rapport à l'évolution du nombre de combinaisons lorsque n augmente.

3.2 Faire un algorithme selon nos connaissance sur les entrées :

Rappellons le problème donné :
Etant donné $E = \{1 :: n\}$ et X l'ensemble des d-uplets constants, trouver 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 $\overline{L}$. 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 adaptée.

tableau	Parcours de $\emptyset$ à L	Parcours de E à max(L)
Sur E	$\|E\| << \|X\|$, $\|L\| << \|\overline{L}\|$	Si $\|E\| << \|X\|$, $\|\overline{L}\| << \|L\|$
Sur X	$\|X\| << \|E\|$, $\|L\| << \|\overline{L}\|$	$\|X\| << \|E\|$, $\|\overline{L}\| << \|L\|$
Sur $\overline{X}$	$\|\overline{X}\| << \|E\|$, $\|L\| << \|\overline{L}\|$	$\|\overline{X}\| << \|E\|$, $\|\overline{L}\| << \|L\|$

TABLE 1 – Différents algorithmes selon E,X,L.

Dans tous les cas, nous avons deux ordres de parcourt :
-Soit on commence par les d-uplets constants 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)").
Une optimisation serait même de commencer par une taille t_init et de s'arrêter à une taille t_end soit dans l'ordre croissant soit dans l'ordre décroissant.

Dans la pratique, nous avons $|E| >> |X|$, $|X| \simeq |\overline{X}|$ et $|L| << |\overline{L}|$.

Un critère supplémentaire important serait la taille maximal d'un sous-modèle maximum. Dans la théorie on a la borne suivante :

Proposition 3.1 *Pour $n > 5$ et $|X| = \frac{\binom{n}{4}}{2}$, $\frac{n}{2} + 1 > |P|$ pour tout $P \in L$.*

Démonstration :
$\binom{\frac{n}{2}+1}{4} > \frac{\binom{n}{4}}{2}$
$\Rightarrow (\frac{n}{2}+1)(\frac{n}{2})(\frac{n}{2}-2)(\frac{n}{2}-3)/4! > \frac{n(n-1)(n-2)(n-3)}{2*4!}$
$\Rightarrow (\frac{n+1}{2})(\frac{n}{2})(\frac{n-2}{2})(\frac{n-4}{2}) > \frac{n(n-1)(n-2)(n-3)}{2}$
$\Rightarrow (n+2)(n-4) > (n-1)(n-3) \Rightarrow n^2 - 2n - 8 > n^2 - 4n + 3 \Rightarrow n > 5.5$

Dans la pratique, sur l'exemple des 16 points les SMC maximaux sont de taille comprise entre 4 et 6. Sur d'autre exemple avec une trentaine de points, la taille maximal des SMC maximum est bien inférieur à n/2.

3.3 Deux algorithmes performants :

Nous avons développé deux algorithmes suffisament performant pour tester jusqu'à 30 points en moins de deux heures sur mon ordinateur personnel, en python et sans que le code ne soit optimisé. Ces deux algorithmes sont dans le programme du git et sont accessibles. Ils ont par ailleurs chacun une feuille de route consultable au même endroit.

3.4 Algorithme 1 : Décroissant sur E

Algorithm 1 TROUVER-SOUS-MODELES-FIXES

Require: X, E
Ensure: L
  FIFO=[(E,X)]
  while $|FIFO| > 0$ **do**
    for tout les sous ensembles P_i de FIFO[0][0] qui ne sont pas déjà dans FIFO **do**
      On calcul $X_i = X \cap S_{p_i}$
      if $|X_i| = \binom{|P_i|}{d}$ **then**
        $L = L \cup P_i$
      else
        $FIFO = FIFO \cup (P_i, X_i)$
      end if
    end for
    $FIFO = FIFO - FIFO[0]$
  end while

Où FIFO est une liste First-in First-Out.

Ce premier algorithme part de E pour trouver tous les sous-modèles constants maximaux. Il a des résultats correctes pour 16,20 et 23 points mais la mémoire utilisé explose pour 25 points.

3.5 Algorithme 2 : Croissant depuis X :

Algorithm 2 TROUVER-SOUS-MODELES-CONSTANTS

Require: X, E
Ensure: L
  FIFO=[(x,X-x) pour tout $x \in X$]
  while $|FIFO| > 0$ **do**
    P = FIFO[0][0]
    X = FIFO[0][1]
    p=P[-1] (le dernier élément de P).
    bool-est-max = true
    for i de p+1 à n **do**
      compteur = 0
      for $x \in X$ **do**
        if $x \in S_P$ **then**
          compteur +=1
          X=X-x
        end if
      end for
      if $\binom{|P|}{d} + compteur = \binom{|P|+1}{d}$ **then**
        bool-est-max = false
        $FIFO = FIFO \cup (P \cup p, X)$
      end if
    end for
    if bool-est-max = true **then**
      $L = L \cup P$
    end if
    FIFO = FIFO-FIFO[0]
  end while

Cet algorithme trouve tous les sous-modèles fixes à partir des d-uplets fixes. Il a des résultats correctes pour 16,20,23,25 points et prends quelques heures pour 30 points.

3.6 Comparaison des deux algorithmes :

En pratique, le deuxième algorithme est beaucoup plus rapide, mais cela vient du fait que sur nos exemples la taille maximale des SMC maximum est très faible. En théorie, le deuxième algorithme effectue des test beaucoup plus rapides mais il parcourt beaucoup trop de sous-modèle. Avec la connaissance de la taille maximale des SMC maximum le premier algorithme pourrait être nettement plus performant. En attendant, l'algorithme 2 est assez puissant pour envisager les 41 points.

4 Recouvrement/Partition par SMC :

Problème 3 (Recouvrement d'ensemble) *Soit $E = \{1 :: n\}$ et S un ensemble de parties de E. On cherche le nombre de parties minimum i de S tels que $S_1 \cup S_2 \cup ... \cup S_i = E$.*

Ce problème fait parties des 21 problèmes NP-complets de Karp. En réalité, nous ne nous intéressons pas réellement à la taille minimum des recouvrements mais à générer tous les

recouvrements par SMC maximum.

Problème 4 (Recouvrement par sous-modèles constants maximaux) *Soit $E = \{1 :: n\}$ $\mathcal{M}$ un ensemble de modèle et L l'ensemble des sous-modèles constants maximaux de E. On cherche l'ensemble des recouvrements de E par sous-modèles de L.*

Sur notre exemple de 16 points il n'y a pas de 2-recouvrements de E car la taille maximal des SMC maximum est 6. En revanche nous trouvons 40 3-recouvrements à partir des 768 SMC maximaux, dont 34 sont des 3-partitions.

Ce nombre, que nous trouvons trop important, nous à poussé à chercher des critères sur les SM plus significatifs.

4.1 Sous-modèles symétriques

Une première proposition est de regarder les sous-modèles symétrique car dans l'exemple des crânes les points d'intérets sont symétrique par paire par rapport à un axe médian du crâne.

Définition 4.1 (Sous-modèle symétrique intrinséque.) *Etant donné E=GCD une partition de E telle que G et D sont en bijection par une fonction s, alors un SM $P \subseteq E$ est dit symétrique intrinséque ssi pour tout $p \in P$ on a $s(p) \in P$. s(p) est appellé le point symétrique de p.*

Remarque 4.1 *Si les points de G et de D sont parfaitement symétrique, alors les d-uplets symétrique intrinséque sont forcément coplannaire. Dans ce cas les sous-modèles symétrique intrinséques ne sont donc jamais constant.*

Cette remarque nous a forcé à regarder d'autres définitions de symétrie :

Définition 4.2 (Paire de sous-modèles symétrique) *P et P' dans E sont dit symétriques ssi pour tout $p \in P$ on a $s(p) \in P'$ et pour tout $p \in P'$ on a $s(p) \in P$.On notera donc s(P) le symétrique de P.*

On peut distinguer trois types de paire de sous-modèles symétrique :

Remarque 4.2 *Un SM symétrique intrinséque P peut être vu comme une paire de sous-modèle symétrique (P,P).*

Définition 4.3 (Paire de sous-modèles symétrique séparés) *Une paire de SM symétrique (P,s(P)) est dites séparée si $P \subseteq GC$ et $s(P) \subseteq CD$ ou vice-versa.*

Définition 4.4 (Paire de sous-modèles symétrique croisés) *Une paire de SM symétrique (P,s(P)) est dite croisée si elle n'est pas séparée et que $P \neq s(P)$.*

Remarque 4.3 *La constance d'une paire de sous-modèles n'influe pas sur la symétrie d'une paire de sous-modèles. Ainsi en parlant de "Paire de sous-modèles constant symétrique" ou en parlant de "Paire de sous-modèles symétriques constant" voir même de "Paire symétrique de sous-modèles constant" on parlerait en fait de la même chose. Ce n'est pas le cas de la symétrie et de la maximalité d'une paire !*

Définition 4.5 (Paire symétrique de sous-modèles constants maximaux) *Une paire (P,P') de SMC maximaux est dites symétrique si P'= s(P)*
Dans notre exemple, 620 sous-modèles constants maximum sont dans une paire symétrique, donc on à 310 paires de SMC .
Ces 310 paires sont toutes des symétriques croisés.
Elles sont dans le fichier $16SMC\_Cr.txt$

Remarque 4.4 *On calcul les paires symétrique de sous-modèles constants maximaux à partir des sous-modèles constants maximaux.*

Définition 4.6 (Paire maximale de sous-modèles constants symétriques) *Une paire (P,s(P)) de SMC symétrique est dite maximum si (P,s(P)) est symétrique et que P et s(P) sont maximaux pour cette propriété.*
Dans notre cas on a 716 SMC qui sont dans des paires symétriques maximum, soit 358 paires. Toutes ces paires sont des symétriques croisées.
Ces 716 SMC sont dans le fichier $16PSMC.sym\_max.txt$

Remarque 4.5 *On calcul les paires maximales de sous-modèles constants symétriques à partir des sous-modèles constants maximaux qui ne contiennent que des d-uplets de X qui ont leur symétrique dans X. En pratique, on utilise l'un des deux algorithmes sur* $X_{symetrique}$ *l'ensemble des d-uplets qui ont leur symétrique dans X.*

Remarque 4.6 *Si (P,s(P)) est une paire symétrique de sous-modèles constants maximaux, alors (P,s(P)) est une paire maximum de sous-modèles constants symétrique. L'inverse n'est pas vrai.*

Remarque 4.7 (paire non symétrique de SM) *Si (P,s(P)) est une paire non symétriques de SM tel que P soit constant et maximum, Alors soit s(P) n'est pas constant soit s(P) n'est pas maximum. Dans ce cas, il existe une paire* $(P', s(P'))$ *tel que* $s(P) \subseteq s(P')$ *et* $s(P)'$ *est maximum où* $P \subseteq P'$ *et* P' *n'est pas constant.*

Définition 4.7 (paire maximum de SM non symétriques) *Une paire (P,s(P)) de SM est dites paire maximum de SM non symétriques ssi P ou s(P) est un SMC maximum et P ou s(P) n'est pas constant.*

4.2 Birecouvrement par paire maximale de SMC symétrique

Après avoir calculer les recouvrements par SMC maximum de E, on s'est intéressé aux recouvrements par paire maximale de SMC symétrique.

Remarque 4.8 *Deux paires distinctes de SMC symétrique peuvent recouvrir le même ensemble de points de E.*

Avec cette remarque, on a alors décidé de considérer les SM intrinséques composés par des paires maximales de SMC symétriques. Nous avons trouvé 106 tels SM, et 32 en prenant seulement ceux maximaux par l'inclusion. Avec ces 32 SM intrinséques on à trouver 77 birecouvrements de E, ce qui est toujours un nombre trop important. Cela nous a poussé à affiner encore plus nos critères de SM significatifs. (Le même raisonnement sur les paires maximales de SM symétrique a donner 122 SM intrinséques dont 33 sont maximaux et produisent 88 birecouvrements).

4.3 Sous-modèles intrinséques presque constants

Définition 4.8 *Un SM intrinséque est dit presque constant s'il a peu de d-uplet non constants en dehors des d-uplets symétrique intrinséques (les d-uplets qui ont deux sommets de G et leurs symétrique dans D).*

Ainsi nous pouvons atribuer à chaque SM intrinséque un score, et regarder en priorité les recouvrements qui maximise les scores des SM utilisés. Cette étape est encore en cours d'étude.

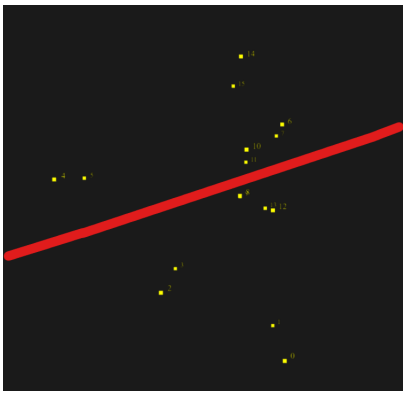
D'autre critères pourraient permetre de diminuer le nombre de recouvrements, notamment des critères mathématiques que l'on pourrait trouver en traitant la 3ème partie de ce stage.

4.4 Un premier birecouvrement intéressant ?

Une succession d'étapes a mené à un 2-recouvrement qui semble intéressant :

1. Calcul des SMC maximaux (768)
2. On ne regarde que les paires de SMC maximaux non symétriques (79)
3. On regarde les SM intrinséques correspondants (79), on ne garde que les **minimaux** (26).
4. On cherche les bipartitions de E avec ces 26 SM.

On trouve un unique 2-recouvrement de GC qui est d'ailleurs une 2-partition. Cette bipartition est (0,2,8,12)(4,6,10,14). En considérant les symétriques de ces parties, on trouve une bipartition de E : $P_1 = (0, 1, 2, 3, 8, 9, 12, 13)$ et $P_2 = (4, 5, 6, 7, 10, 11, 14, 15)$.
Dans chacune de ces parties, on cherche alors une paire maximale de SMC non symétrique



entiérement contenu dedans. Pour P_1 on trouve (0,3,7,9,12,13), son symétrique est constant (1,2,8,12,13) mais pas maximum, en effet par exemple (1,2,8,11,12,13) est un SMC maximum, son symétrique est (0,3,9,10,12,13) qui n'est pas constant, la seule paire maximale de SMC non symétrique est donc **((1,2,8,11,12,13),(0,3,9,10,12,13))**. Mais (1,2,8,11,12,13) n'est pas entièrement contenu dans P_1. Dans P_2 on trouve la base (5,7,11,15) dont le symétrique (4,6,10,14) n'est pas constant c'est donc une paire maximale de SMC non symétrique : **((5,7,11,15),(4,6,10,14))**.
A première vu cette bipartition parait intéressante puisqu'elle est unique et qu'elle sépare correctement les 16 points. Mais il est difficile de savoir si ces parties ont réellement une propriété ou si c'est simplement un hasard de trouver cette bipartition.

5 Conclusion

Cette première partie du stage à permit de développer la partie algorithmique nécessaire pour pouvoir tester les différents recouvrement sur 16 points. Néanmoins une optimisation du code et son implémentation sur le logiciel OMSMO sont nécessaires avant d'essayer d'autre jeux de données. Ce travail préliminaire permet de passer à l'étape suivante, plus mathématiques, qui est de mesurer l'intéractions entre deux sous-modèles constants. Une première piste est de commencer par des sous-modèles symétriques en 2 dimensions. Les résultats d'un telle partie pourrait permettre ensuite de comparer l'approche des modèles 3D par décomposition de matroïdes orientés avec les approches plus habituels. Dans le même genre d'idée, on pourrait comparer les test d'intéractions simples (intersections par exemple) sur des modèles simples comme des cubes ou des boites AABB et comparer les résultats avec des méthodes de moteur d'applications intéractives. Des test en 2 dimensions pourraient aussi mener à des pistes en analyse d'image.
Dans l'imédiat, les pistes les plus envisageables dans l'ordre de leur apparition dans ce compte rendu sont (avec un score de priorité allant de 1 à 5 et un score de difficulté de **A** = Très simple à **E**=Peut être impossible) :

- Chercher une solution au problème 2.**(1/5),E** Cela améliorait nettement les résultats de l'algorithme mais nous pensons qu'en l'état actuel une optimisation du code suffit pour traiter nos jeux de données.
- Chercher une borne pour la taille maximale des SMC maximum. **(3/5),D**

- Tester d'autre parcourt des sous-modèles (en augmentant la taille de deux par deux par exemple, et en cherchant rétroactivement si nous n'avons pas rater un SMC maximum). **(2/5),B**
- Tester l'algorithme 2 sur 41 points (deuxième jeu de données non décris ici). **(4/5),A**
- Développer le critère de "presque constance" de SM intrinséques. **(5/5),B**
- Chercher en dimension 2 et 3 des critères pour mesurer l'interaction entre deux SMC maximaux. **(5/5),C**

6 Partie bibliographique :

- Ce sujet de stage fait suite à un TER qui abordait principalement la question de lister l'ensemble des SMC :
  - K. Planolles. *Outils préliminaires pour la décomposition de matroïdes orientés.* Rapport TER M1 Université de Montpellier. 23 mai 2021.
- Une grande partie de mes connaissances sur le sujet des matroïdes orientés vient du cours de master de J. Ramirez Alfonsin et du document :
  -E. Gioan, J. Ramirez Alfonsin. *Eléments de théorie des matroïdes et matroïdes orientés.* Chapitre 2 de Informatique Mathématique - Une photographie en 2013 (Philippe Langlois, ed.), Presses Universitaires de Perpignan, pages 47-95, 2013.
- Une notion qui peut être intéressante mais n'est pas mentionné ici à été développer dans mon stage de master mathématiques informatique avec J. Ramirez Alfonsin (voir CR1 pour plus de détails) :
  -Y. Marin. *Sur un matroïde de pavage.*Rapport de stage M2 Université de Montpellier.
- Dans *The art of computer programming*, deux chapitres sont dédiés à la génération de combinaisons et à la génération de partitions d'ensembles :
  -Knuth, D. E. (1997). *The art of computer programming (3rd ed.).* Addison Wesley.