



# Projet Ingénierie Réseaux Clustering et routage dans un essaim de nano-satellites

Realisé par:

Mohamed EL YESSEFI  
Nabil AIT ALAYA  
Othman AIT OUAKRIM  
Yahya YOUNES

*2A-R*

2022-2023

# Contents

<b>1</b>	<b>Introduction et analyse du problème</b>	<b>3</b>
<b>2</b>	<b>State of the art</b>	<b>3</b>
<b>3</b>	<b>Caractérisation la dynamique de l'essaim</b>	<b>5</b>
3.1	Track d'un seul nano-sats et track des 100 nano-sats . . . . .	5
3.2	Identification des cliques . . . . .	5
3.3	Les composantes connexes . . . . .	8
3.4	Temps inter-contact . . . . .	10
<b>4</b>	<b>Proposition et mise en place des méthodes de clustering</b>	<b>10</b>
4.1	K-means . . . . .	11
4.2	Girvan-Newman . . . . .	12
4.3	DBSCAN . . . . .	14
<b>5</b>	<b>Evaluation des performances</b>	<b>15</b>
5.1	Comparaison d'énergie . . . . .	15
5.2	Comparaison de débit . . . . .	16
<b>6</b>	<b>Notre produit à élaborer</b>	<b>18</b>

# 1 Introduction et analyse du problème

La science des réseaux joue un rôle important dans le domaine des nanosatellites, qui offrent de nouvelles possibilités d'exploration spatiale, de communication et de collecte de données. Dans ce contexte, la gestion efficace des données des nanosatellites est un défi complexe. Ce projet se concentre sur le regroupement et le routage des nanosatellites pour une distribution optimale de la charge utile en tenant compte de la largeur de bande et de la consommation d'énergie.

Le défi n'est pas seulement de regrouper les satellites les plus proches géographiquement, mais aussi de partitionner l'essaim en groupes homogènes. Nous visons à développer une approche distribuée pour cette tâche, en considérant qu'il n'y a pas de coordinateur central. Les nanosatellites peuvent émettre à différentes distances en ajustant leur bande passante.

Les objectifs sont de caractériser la dynamique de l'essaim, de définir des objectifs de performance, de proposer des techniques de clustering et d'évaluer la performance de l'algorithme proposé. Pour atteindre les résultats souhaités, il est important d'analyser la trajectoire de mouvement de l'essaim de nanosatellites en utilisant des techniques de clustering et des méthodes distribuées appropriées.

## 2 State of the art

Les différentes méthodes de clustering dans le cadre des nano-satellites présentent des avantages et des inconvénients spécifiques :

### **K-means :**

*Fonctionnement* : L'algorithme K-means divise les données en K clusters en minimisant la somme des distances au carré entre chaque point de données et le centre de son cluster. L'algorithme alterne entre deux étapes : l'affectation des points aux clusters les plus proches et la mise à jour des centres des clusters en calculant la moyenne des points appartenant à chaque cluster.

Avantages :

- Facile à comprendre et à mettre en œuvre.
- Efficace pour les ensembles de données de grande taille.
- Convergence garantie.

Inconvénients :

- Sensible aux valeurs aberrantes.
- Nécessite de spécifier le nombre de clusters K au préalable.
- Peut converger vers des optima locaux.

## **Girvan-Newman :**

*Fonctionnement* : L'algorithme de Girvan-Newman est utilisé pour détecter les communautés dans les réseaux en utilisant des mesures de centralité et de coupure de graphes. Il s'appuie sur la suppression progressive des liens les plus "entre-essaims" (entre différentes communautés) pour identifier les sous-groupes cohérents.

Avantages :

- Identifie les structures communautaires dans les réseaux.
- Ne nécessite pas de spécifier le nombre de clusters à l'avance.
- Peut être utilisé pour des réseaux de grande taille.

Inconvénients :

- Coût computationnel élevé pour les grands réseaux.
- Peut générer des partitions non optimales dans certains cas.
- Sensible à la résolution des coupes de graphes.

## **DBSCAN :**

*Fonctionnement* : DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un algorithme de clustering qui se base sur la densité des points. Il identifie les zones denses de points dans l'espace des données en recherchant les régions où le nombre de points est supérieur à un seuil prédéfini. Les points qui se trouvent dans des zones moins denses sont considérés comme du bruit.

Avantages :

- Capable de trouver des clusters de formes arbitraires.
- Ne nécessite pas de spécifier le nombre de clusters à l'avance.
- Robuste aux valeurs aberrantes.

Inconvénients :

- Sensible au choix des paramètres (seuil de densité et rayon de voisinage).
- Peut avoir des difficultés avec des différences de densité significatives entre les clusters.
- Peut être moins efficace pour les ensembles de données de grande dimension.

### 3 Caractérisation la dynamique de l'essaim

#### 3.1 Track d'un seul nano-sats et track des 100 nano-sats

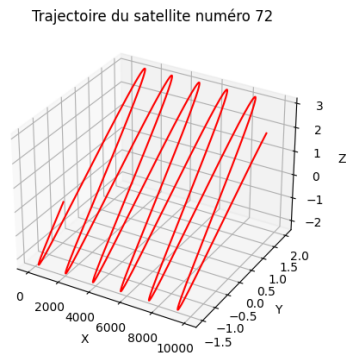


Figure 1: Trajectoire d'un seul satellite

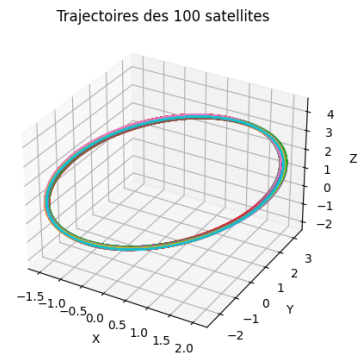


Figure 2: Trajectoires des 100 satellites

#### 3.2 Identification des cliques

Dans les essaims de nos nano-satellites, les cliques font référence à des regroupements spécifiques de nœuds(satellites) au sein de l'essaim. Une clique est un ensemble de nœuds satellites qui sont fortement connectés les uns aux autres. Cela signifie que chaque nœud de la clique est directement relié à tous les autres nœuds de cette même clique.

### Les cliques dans les essais du groupe

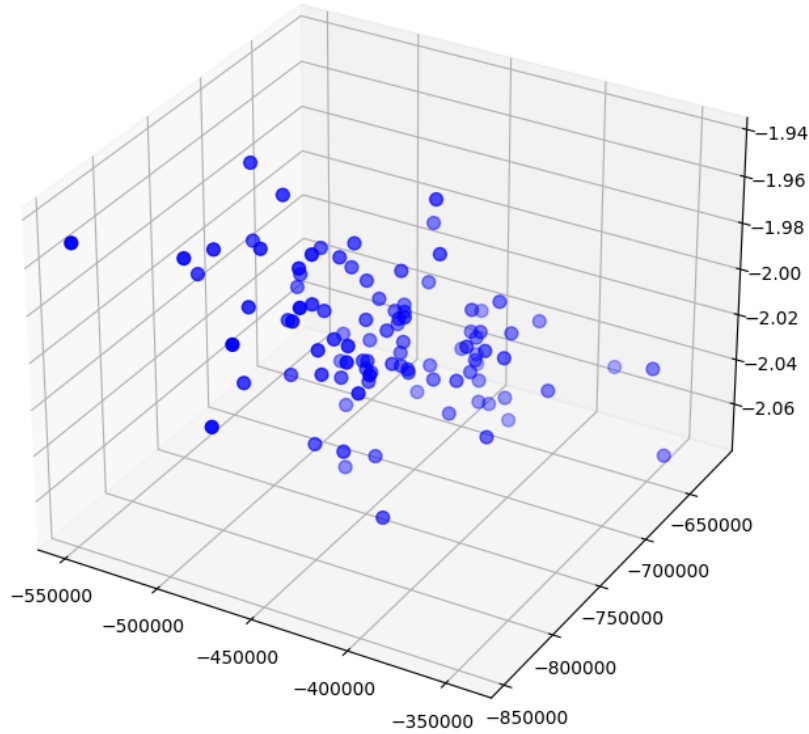


Figure 3: L'ensemble des cliques dans notre essaim de nano-satellites

Comme on travaille sur des données de 100 satellites on arrive pas à voir clairement les cliques dans l'essaim des nanosatellites on va plutôt afficher chaque clique séparément. Ci dessous 6 cliques prises au hasard de nos 104 cliques

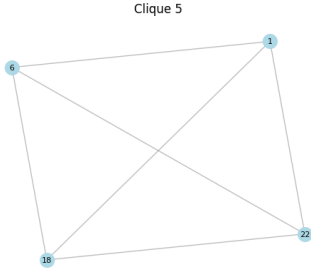


Figure 4: Clique numéro 5

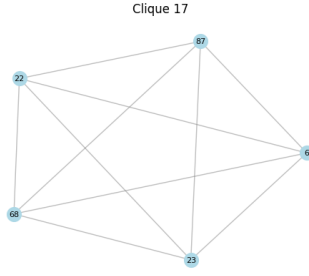


Figure 5: Clique numéro 17

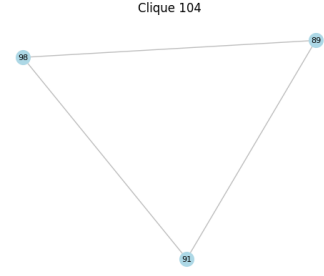


Figure 6: Clique numéro 104

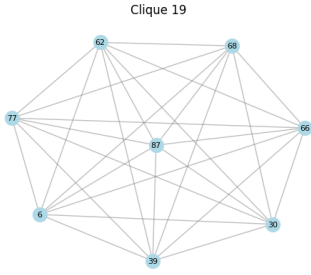


Figure 7: Clique numéro 19

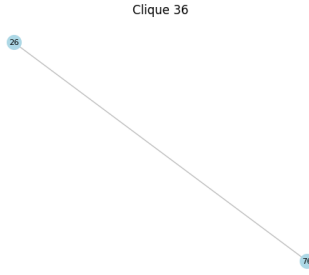


Figure 8: Clique numéro 36

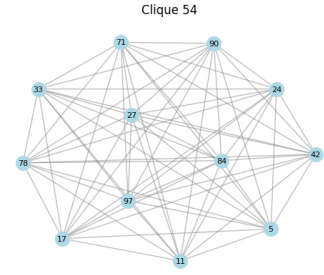


Figure 9: Clique numéro 54

L'utilisation des cliques dans le clustering des nano-satellites peut aider à économiser l'énergie et le débit de transmission. En effet, en identifiant les cliques, qui représentent des groupes de satellites fortement connectés, nous pouvons optimiser les communications et les échanges de données au sein de l'essaim. Par ailleurs cela permet de réduire le nombre de transmissions nécessaires, minimisant ainsi les dépenses énergétiques et évitant la congestion du réseau.

Cette approche permet d'optimiser l'utilisation des ressources de transmission, en particulier pour les satellites en périphérie et au cœur de l'essaim, tout en garantissant des communications efficaces et une utilisation plus efficiente de l'énergie.

### 3.3 Les composantes connexes

Pour identifier les composantes connexes, on peut se baser sur plusieurs critères :

1. **Connexité** : Les nœuds d'une composante connexe sont tous connectés les uns aux autres, ce qui signifie qu'il existe un chemin entre chaque paire de nœuds de la composante.
2. **Distance** : Les nœuds d'une composante connexe sont généralement proches les uns des autres. Vous pouvez utiliser une mesure de distance, telle que la distance euclidienne ou la distance géodésique, pour évaluer la proximité entre les nœuds.
3. **Interactions fréquentes** : Les nœuds d'une composante connexe interagissent régulièrement les uns avec les autres. Vous pouvez prendre en compte la fréquence des interactions ou des échanges d'informations entre les nœuds pour déterminer les composantes connexes.
4. **Temps inter-contact** : Le temps inter-contact entre les nœuds peut être utilisé pour définir les composantes connexes. Si les nœuds ont des périodes d'interaction rapprochées dans le temps, cela peut indiquer qu'ils font partie d'une même composante connexe.
5. **Similarité** : Les nœuds d'une composante connexe peuvent présenter des similarités ou des caractéristiques communes. Par exemple, ils peuvent appartenir à la même classe ou partager des attributs similaires.

Dans notre implémentation du code Python, nous avons utilisé la distance entre les satellites comme critère pour déterminer les composantes connexes. Nous avons fixé un seuil de distance au-delà duquel deux satellites ne sont plus considérés comme connectés. Ainsi, si la distance entre deux satellites dépasse ce seuil, ils sont considérés comme appartenant à des composantes connexes distinctes.

Pour observer les changements des composantes connexes au fil du temps, nous avons effectué une itération sur les instants de temps. À chaque instant, nous avons recalculé les composantes connexes en fonction de la distance entre les satellites. Cela nous a permis d'observer comment les connexions entre les satellites évoluent au sein des essaims au fil du temps.

Ce choix de critère nous a permis d'analyser la dynamique des essaims de satellites en termes de composantes connexes, en identifiant les groupes de satellites qui restent connectés en fonction de leur proximité spatiale. Cela nous a également permis de quantifier les changements de connexions au fil du temps, fournissant des informations sur la dynamique et la cohésion de l'essaim.



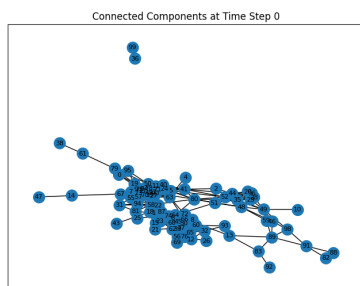


Figure 10: La composante connexe numéro 0

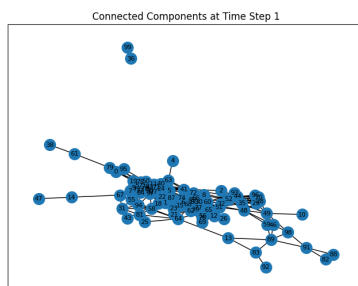


Figure 11: La composante connexe numéro 1

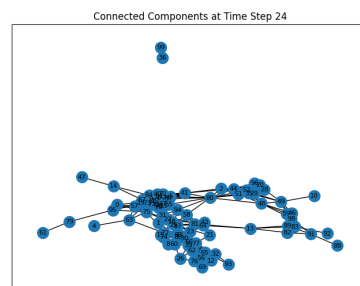


Figure 12: La composante connexe numéro 31

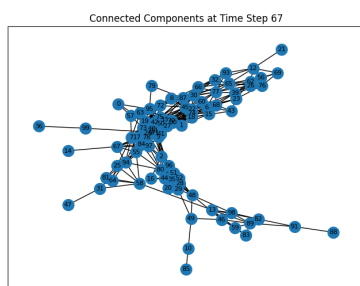


Figure 13: La composante connexe numéro 62

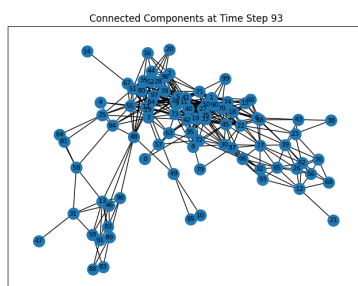


Figure 14: La composante connexe numéro 93

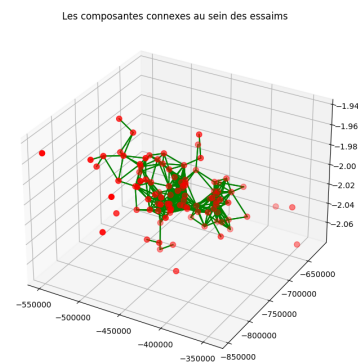


Figure 15: Les composantes connexes dans l'essaim

### 3.4 Temps inter-contact

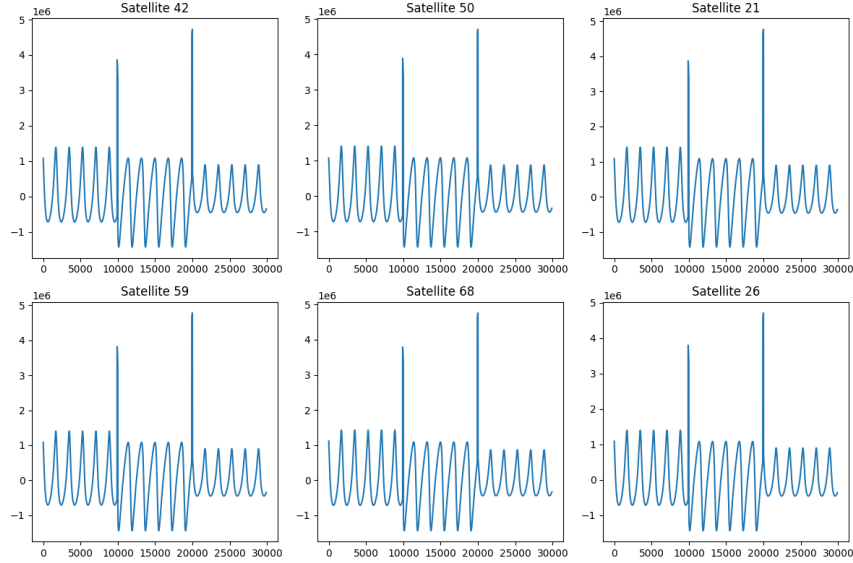


Figure 16: L'évolution du temps inter-contact entre les satellites au cours du temps

L'exploitation du temps inter-contact entre les satellites au sein d'un essaim de nano-satellites permet de planifier de manière efficace l'envoi de données. En analysant les variations du temps inter-contact, il est possible de déterminer les moments où la connectivité entre les satellites est optimale. En utilisant ces informations, des stratégies de routage peuvent être mises en place pour diriger les transmissions de données pendant les périodes de connectivité maximale.

De plus, en adaptant dynamiquement les ressources en fonction des variations du temps inter-contact, il est possible d'optimiser l'utilisation des capacités de transmission tout en minimisant la consommation d'énergie. Ainsi, l'exploitation du temps inter-contact permet d'améliorer significativement l'efficacité du réseau de nano-satellites en termes de débit de transmission et de gestion de l'énergie.

## 4 Proposition et mise en place des méthodes de clustering

Il existe trois catégories globales des méthodes de clustering, et on a élu une méthode état d'art parmi chaque catégorie : K-means (*Partitionnement*), Girvan-newmann (*Hierarchique*), et DBSCAN (*à Densité*) ; et on les a comparé suivant les critères du débit, et d'économie d'énergie.

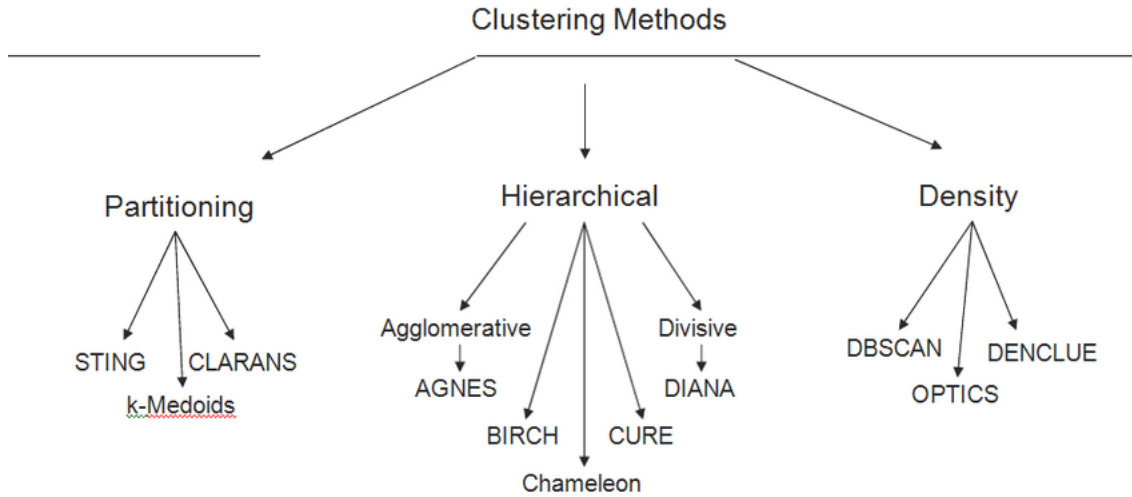


Figure 17: Les principaux catégories des méthodes de clustering

## 4.1 K-means

K-Means est un algorithme de clustering basé sur la partition qui vise à partitionner  $n$  nœuds d'un graphe en  $k$  clusters où chaque noeud appartient au cluster dont la moyenne est la plus proche.

L'algorithme commence par initialiser aléatoirement  $k$  centroïdes. Il effectue ensuite deux étapes de manière itérative : l'étape d'affectation, au cours de laquelle chaque point de données est affecté au centroïde le plus proche, et l'étape de mise à jour, au cours de laquelle de nouveaux centroïdes sont calculés sur la base de la moyenne de tous les points de la grappe.

Ce processus est répété jusqu'à ce qu'une condition d'arrêt soit remplie, par exemple lorsque les centroïdes ne changent pas ou peu entre les itérations, ou lorsqu'un nombre maximal d'itérations est atteint.

K-Means est un algorithme simple et rapide qui convient aux grands ensembles de données. Il fonctionne bien lorsque les grappes sont sphériques et de taille à peu près égale. Cependant, il peut donner de mauvais résultats lorsque les grappes ont des géométries compliquées ou des densités différentes.

Dans notre projet, il s'agissait de l'algorithme le plus simple à mettre en œuvre. Il est efficace en termes de coûts de calcul, ce qui rend K-means 'scalable'. En fait, les complexités théoriques de temps et d'espace de cet algorithme sont les suivantes :

Complexité temporelle :  $O(n \times K \times I \times d)$ , où  $n$  est le nombre de points de données,  $K$  est le nombre de clusters,  $I$  est le nombre d'itérations, et  $d$  est le nombre d'attributs (3 dans notre cas :  $x$ ,  $y$  et  $z$ ).

Complexité spatiale :  $O((n + K) \times d)$ ,

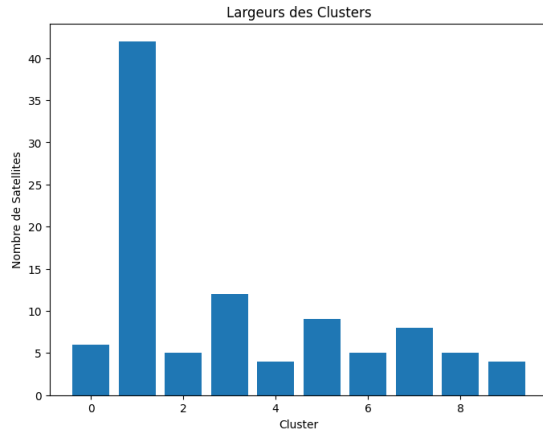


Figure 18: La largeur des clusters obtenus en utilisant l'algorithme de k-means

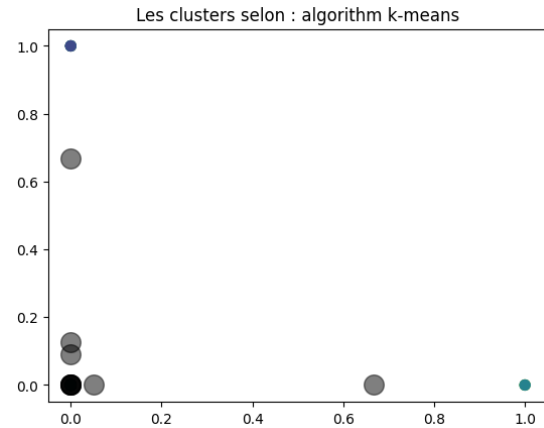


Figure 19: Les clusters obtenus en utilisant l'algorithme de k-means

Notons que K-means n'est qu'accessoirement utile dans notre cas puisque nous connaissons déjà le nombre de clusters  $K(\sqrt{N})$ , ce qui n'est pas toujours le cas. Cet algorithme est mal adapté à notre projet puisqu'il regroupe les nœuds en fonction de leurs distances les uns par rapport aux autres. Cependant, un autre algorithme utilisant des clusters plus hétérogènes permettrait d'améliorer les performances.

## 4.2 Girvan-Newman

Girvan-Newman est un algorithme de regroupement hiérarchique utilisé principalement pour détecter les communautés dans les réseaux. L'idée est de supprimer progressivement les arêtes du graphe original, en créant une séquence de graphes de plus en plus petits, chacun correspondant à une communauté dans le réseau.

L'algorithme identifie la "centralité d'intermédierité" de toutes les arêtes du réseau, c'est-à-dire la proportion des chemins les plus courts entre les paires de nœuds qui passent par cette arête. Les arêtes ayant une centralité d'intermédierité élevée sont susceptibles de relier différentes communautés. Ces arêtes sont ensuite supprimées, ce qui permet de diviser le graphe en communautés plus petites.

Ce processus est répété de manière itérative jusqu'à ce que toutes les arêtes soient supprimées, créant ainsi un dendrogramme de communautés.

En général, l'algorithme de Girvan-Newman est avantageux dans les cas où l'ensemble de données est mieux modélisé comme un réseau, ce qui est idéal pour notre projet puisque nous disposons d'un réseau de nano-satellites.

Nous devons noter que si nous avons pu nous en sortir en exécutant cet algorithme sur notre petit graphe de 100 nœuds, l'algorithme est en général coûteux en termes de calcul - le calcul de l'interdépendance des arêtes est coûteux, ce qui le rend moins approprié pour les très grands réseaux.

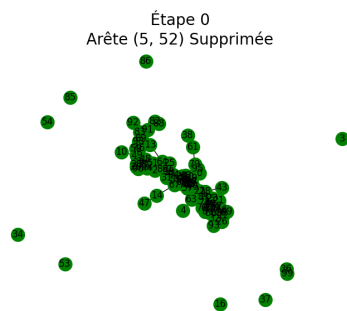


Figure 20: L'étape numéro 0 dans l'algorithme Girvan-Newman

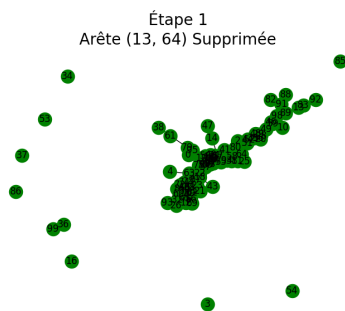


Figure 21: L'étape numéro 1 dans l'algorithme Girvan-Newman

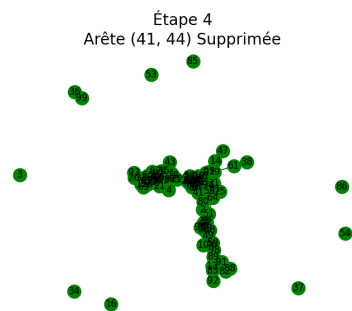


Figure 22: L'étape numéro 4 dans l'algorithme Girvan-Newman

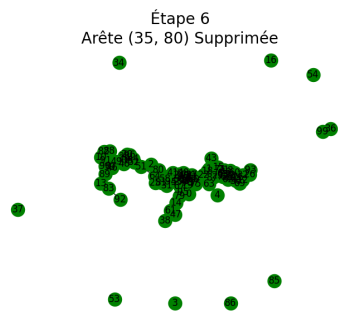


Figure 23: L'étape numéro 6 dans l'algorithme Girvan-Newman

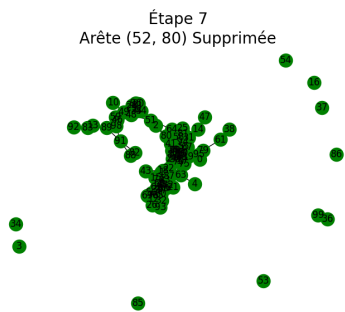


Figure 24: L'étape numéro 7 dans l'algorithme Girvan-Newman

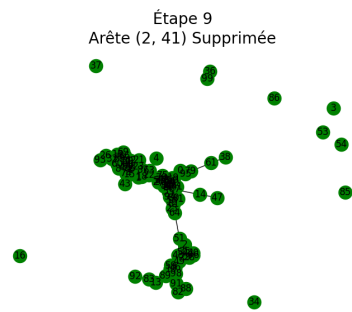


Figure 25: La dernière étape dans l'algorithme Girvan-Newman

En fait, avec une complexité temporelle de  $O(m^2n)$  dans le pire des cas, où  $m$  est le nombre d'arêtes et  $n$  le nombre de sommets, il s'agit de l'algorithme le plus coûteux en termes de calcul que nous ayons testé, mais on a constaté que c'est aussi celui qui a obtenu les meilleurs résultats en termes de performances.

### 4.3 DBSCAN

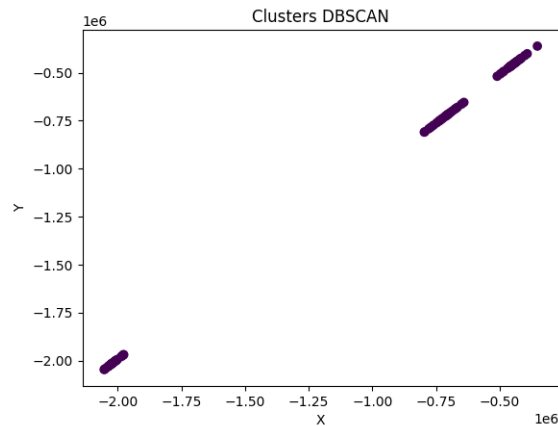


Figure 26: les clusters obtenus en utilisant l'algorithme DBSCAN

DBSCAN est un algorithme de regroupement basé sur la densité qui sépare les régions à forte densité des régions à faible densité. Contrairement à K-Means, DBSCAN ne demande pas à l'utilisateur de spécifier le nombre de grappes.

L'algorithme fonctionne en définissant un rayon spécifié "eps" et un nombre minimum de points requis pour former une région dense "minPts". L'algorithme commence par un point de départ arbitraire et s'il y a des "minPts" à moins de "eps", un nouveau groupe est créé. L'algorithme ajoute ensuite à la grappe tous les points situés à une distance "eps". Si un point situé à l'intérieur de la distance "eps" a également au moins des "minPts" à l'intérieur de la distance "eps", la grappe s'agrandit pour inclure ce point également. Le processus se poursuit jusqu'à ce que plus aucun point ne puisse être ajouté à la grappe, puis un nouveau point non visité est sélectionné et traité, ce qui permet de découvrir d'autres grappes ou bruits.

En général, DBSCAN est idéal pour les tâches où l'on ne connaît pas le nombre de grappes, ce qui est exagéré pour nos besoins spécifiques. Mais il est utile de savoir qu'il peut trouver des grappes de forme arbitraire et qu'il est moins affecté par le problème de l'initialisation.

Il convient de noter que les paramètres "eps" et "minPts" doivent être ajustés - un mauvais choix peut donner de mauvais résultats de regroupement. De plus, cet algorithme a des difficultés à trouver des grappes de densités variables.

Cet algorithme est efficace, à la fois en termes d'espace et de complexité temporelle : dans le pire des cas, la complexité temporelle est de :  $O(n^2)$  et la complexité spatiale est de :  $O(n)$ .

## 5 Evaluation des performances

Dans cette partie on va évaluer la consommation d'énergie et de débit en fonction de l'algorithme de clustering qu'on a utilisé. Pour cela, on prend une référence d'énergie et de débit qui est égale à zéro et de plus la distance augmente entre les satellites, le plus ils auront besoin d'énergie pour envoyer les données. On trace à côté de l'énergie et du débit l'évolution du coefficient de clustering CC pour voir comment les clusters évoluent au cours des étapes de l'algorithme

### 5.1 Comparaison d'énergie

**Pour l'algorithme de Girvain Newman :** On remarque que l'énergie consommée par les satellites est inversement proportionnelle à l'évolution des clusters au sein de l'essaim, ce qui se peut s'exploiter pour l'envoi des données entre les clusters. Autrement dit, au lieu d'envoyer les données par tous les satellites à tous les autres satellites, on peut désigner un satellite de périphérie qui s'occupe de l'envoi dans un cluster, car l'algorithme de Girvain Newman nous permet de voir les liens le plus chargé et les enlever donc former des communautés similaires et moins distantes entre elles puisque les critères qu'on a pris ici est la distance

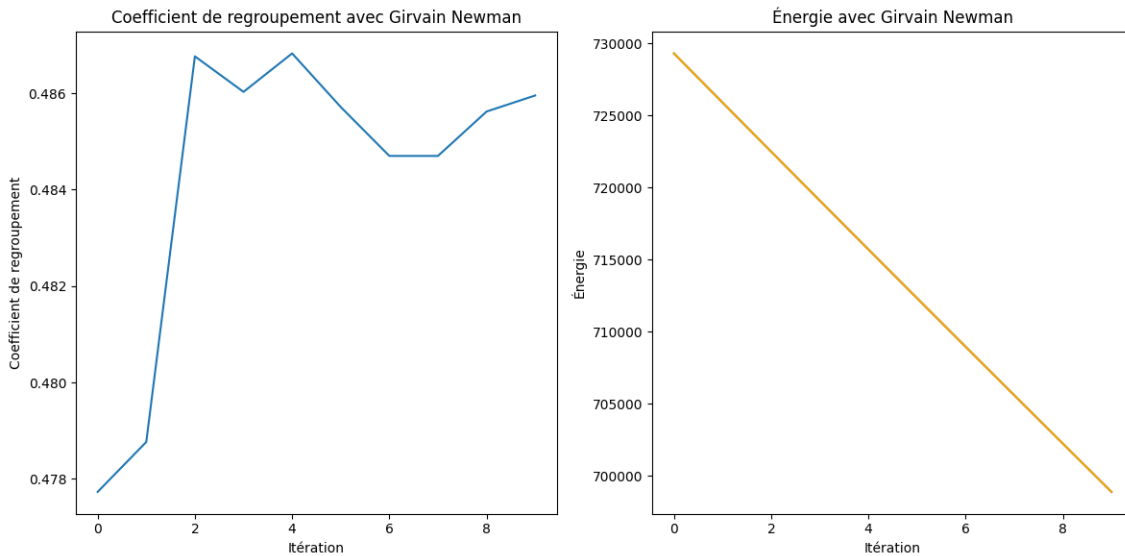


Figure 27: Évolution de CC et de l'énergie en appliquant l'algorithme de clustering

**Pour l'algorithme de K-means :** On constate que la consommation d'énergie des satellites reste constante tout au long de l'évolution des clusters au sein de l'essaim. Cette caractéristique peut être justifiée par le fait que les données spatiales qu'on a ne sont pas exploitées de manière à bien afficher les similarités entre les satellites, par exemple au lieu de prendre comme métrique de clustering la distance on pourrait prendre la similarité : c.-à-d. qu'un type de satellites qui se situent dans une partie  $p$  fréquentent tous au moyen un nombre  $n$  des autres satellites et donc prendre cette similarité comme critère de clustering pourrait améliorer le clustering par l'algorithme k-means

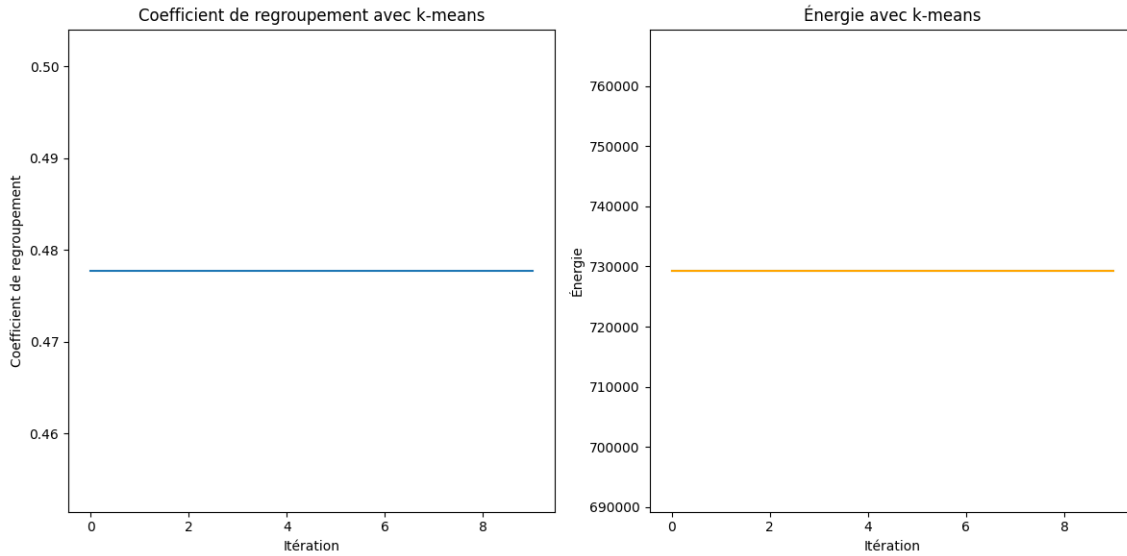


Figure 28: Évolution de CC et de l'énergie en appliquant l'algorithme de clustering

## 5.2 Comparaison de débit

**Pour l'algorithme de Girvain Newman :** L'algorithme de Girvain Newman présente une opportunité intéressante pour améliorer le débit de transmission entre les clusters de nano-satellites au sein de l'essaim. En exploitant les caractéristiques évolutives des clusters, il est possible de maximiser le débit en adoptant une approche sélective dans l'envoi des données. Plutôt que de transmettre les données entre tous les satellites de manière indiscriminée, une stratégie consiste à désigner un satellite périphérique spécifique au sein de chaque cluster pour assurer la transmission. L'algorithme de Girvain Newman facilite l'identification des liens les plus sollicités, permettant ainsi de former des communautés plus cohérentes et proches les unes des autres. Cette approche ciblée favorise une utilisation plus efficace des ressources et conduit à une augmentation significative du débit de transmission entre les clusters de nano-satellites.



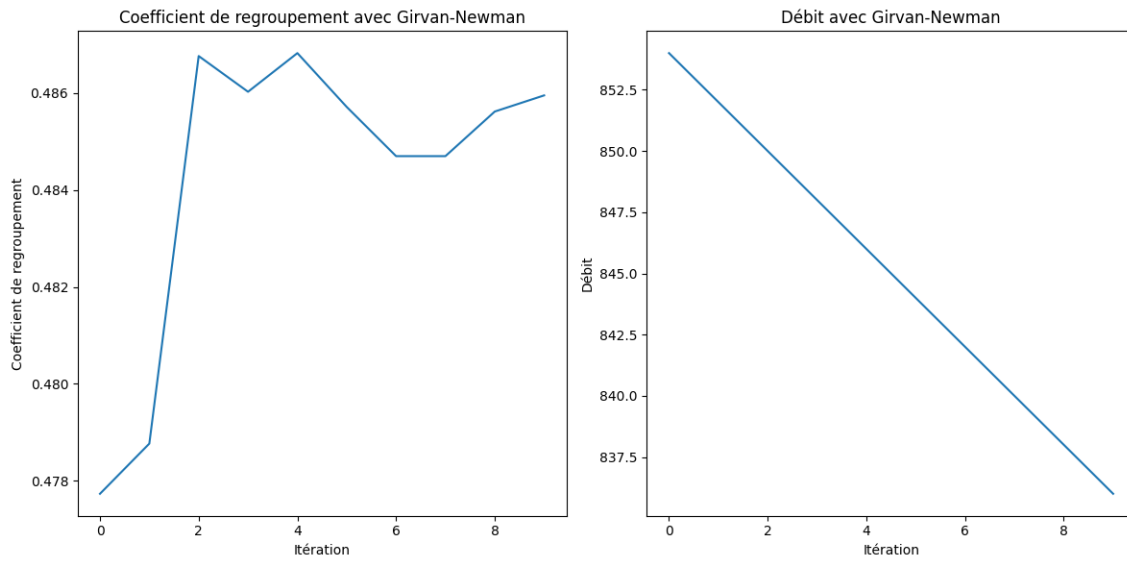


Figure 29: Évolution de CC et du débit en appliquant l'algorithme de clustering

**Pour l'algorithme de K-means :** On remarque que le débit des satellites reste constant lors de l'évolution des clusters au sein de l'essaim. Cette caractéristique met en évidence que les satellites maintiennent une capacité de transmission de données constante, indépendamment des regroupements au sein du système

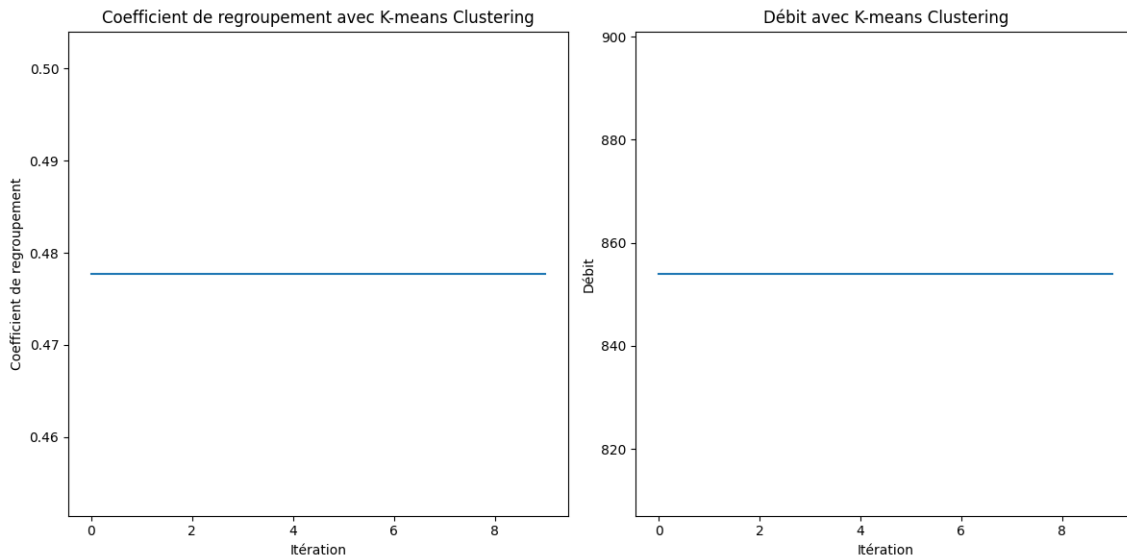


Figure 30: Évolution de CC et du débit en appliquant l’algorithme de clustering

## 6 Notre produit à élaborer

Notre produit pour ce projet est une page web sur laquelle vous pouvez entrer vos données, telles que Traces.csv, et définir certaines contraintes, notamment la portée maximale d’un satellite et la nécessité que tous les autres satellites reçoivent l’information. Notre équipe a travaillé ensemble pour concevoir cette page web et l’améliorer continuellement afin d’optimiser le processus de regroupement des satellites.

La page web offre aux utilisateurs le choix entre trois algorithmes de regroupement : Girvain-Newman, k-means et DBSCAN. En analysant les données fournies, la page web compare l’impact sur le débit pour chaque algorithme de regroupement et identifie celui qui offre les meilleures performances en termes de débit.

Notre objectif collectif était d’améliorer l’efficacité et l’optimisation des regroupements de satellites en utilisant les informations fournies. Nous avons travaillé en étroite collaboration pour développer une interface conviviale et intuitive, permettant aux utilisateurs de prendre des décisions éclairées quant au choix de l’algorithme de regroupement le plus adapté à leurs besoins.

Ce produit représente le fruit de notre collaboration et de notre engagement à fournir des solutions pratiques et efficaces dans le domaine des nano-satellites. En utilisant notre page web, vous pourrez optimiser vos ressources et améliorer les performances de votre système de nano-satellites en choisissant l’algorithme de regroupement le plus approprié pour vos données.

The image shows a web application interface for satellite selection. The background is a dark space scene with a blue horizon line, a satellite, and asteroids. A semi-transparent white form is centered on the page. The form has the following sections:

- Title:** "Find what fits your Sats the best"
- Section 1:** "La portée maximale de mes satellites:" followed by a text input field containing "50" and two green buttons labeled "-" and "+".
- Section 2:** "Choisissez un algorithme:" followed by a dropdown menu showing "Girvain-Newman".
- Section 3:** "Importer les données:" followed by a button labeled "Choisir un fichier" and the text "Aucun fichier choisi".
- Submit:** A green button labeled "Submit".

In the bottom left corner, the text "ENSEEIH" is visible, followed by "Parcours R" and "2022 - 2023" on separate lines.

Figure 31: La page web qu'on a créé pour utiliser nos algorithmes et trouver le meilleur compromis entre l'énergie et le débit