Étude de laboratoire - ASD

Binôme A11
Simon Léo, Levy–Falk Hugo
Supélec, promo 2020

5 février 2018

Table des matières

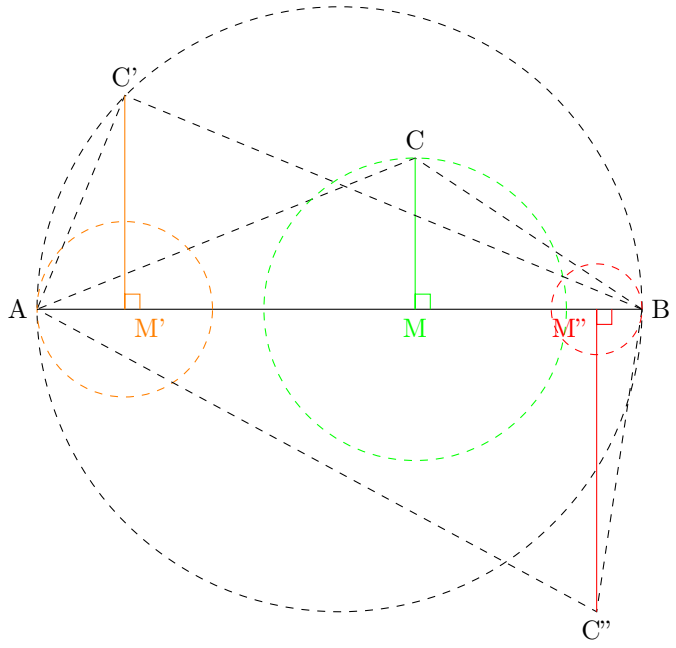

Figure 1 – Différentes positions possible de points par rapport à A et B

Première partie

Génération de carte routière réaliste

1 Condition pour un graphe de Gabriel

En notant $\mathcal{V} = \{P_i\}_{1 \leq i \leq n}$ un nuage de $n \in \mathbb{N}$ points dans le plan représentants des villes, on définit pour tout $A, B \in \mathcal{V}, d(A, B)$ la distance à vol d'oiseau entre les deux villes. On décide de placer une arête entre deux points A et B du plan si et seulement si,

$$\forall C \in \mathcal{V}, \forall M \in [A, B], d(M, V) \geq \min\{d(M, A), d(M, B)\} \tag{1}$$

Montrons que la condition 1 est équivalente à ce que pour toute paire de sommets (A, B) du nuage, $\{A, B\}$ forme une arête si et seulement si il n'existe pas de points $C \in \mathcal{V}$ dans le cercle de diamètre $[A, B]$. Un graphe vérifiant cette condition sera par la suite appelé *graphe de Gabriel*.

preuve : Soient $A, B \in \mathcal{V}$ et on appelle $\mathcal{C}$ le cercle de diamètre $[A, B]$. La figure 1 montre différentes positions possibles de points.

Supposons qu'il existe une arête reliant les deux points. Si il existe des points du nuage dans le disque ouvert délimité par $\mathcal{C}$, alors il existe un point M qui ne vérifie pas la condition 1, *absurde*.

Réciproquement, si tous les points de $\mathcal{V}$ sont à l'extérieur du cercle ouvert délimité par $\mathcal{C}$, (dans l'exemple C' et C''), alors pour tout $P \in \mathcal{V}$ $\{A, B\}$, le point $M \in [A, B]$ le plus proche de P vérifie la condition 1 (dans l'exemple, les points M' et M'').□

2 Mise en pratique : graphe de Gabriel et de voisinage relatif

2.1 Création de graphe de Gabriel et de voisinage relatif

En traçant les graphes de Gabriel (figure 2) et de voisinage relatif (figure 3), on observe que le second est inclus dans le premier.

2.2 Génération d'un réseau

En combinant la génération de graphe de Gabriel et de voisinage relatif, on peut générer un réseau routier, comme le montre la figure 4.

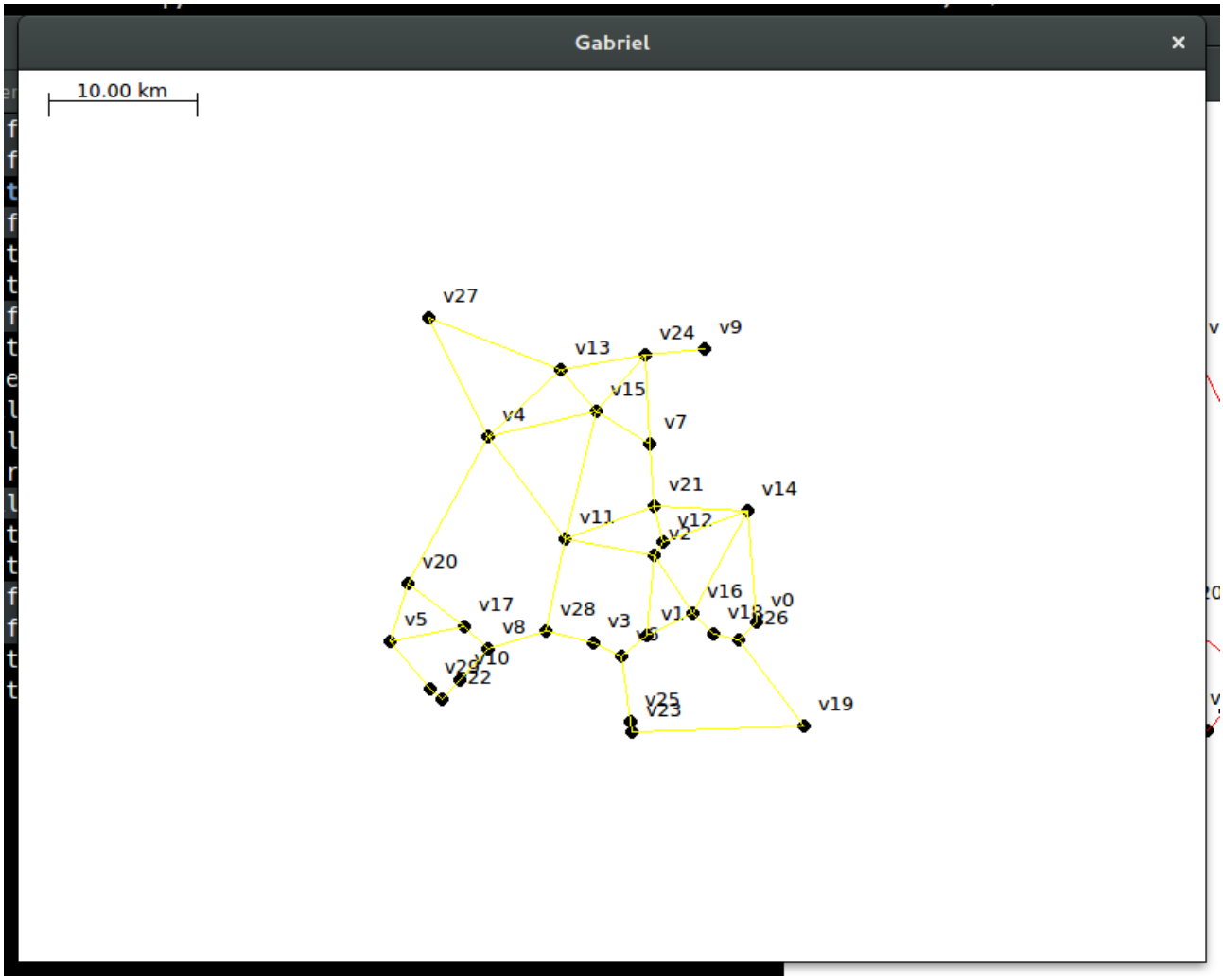

Figure 2 – Graphe de gabriel

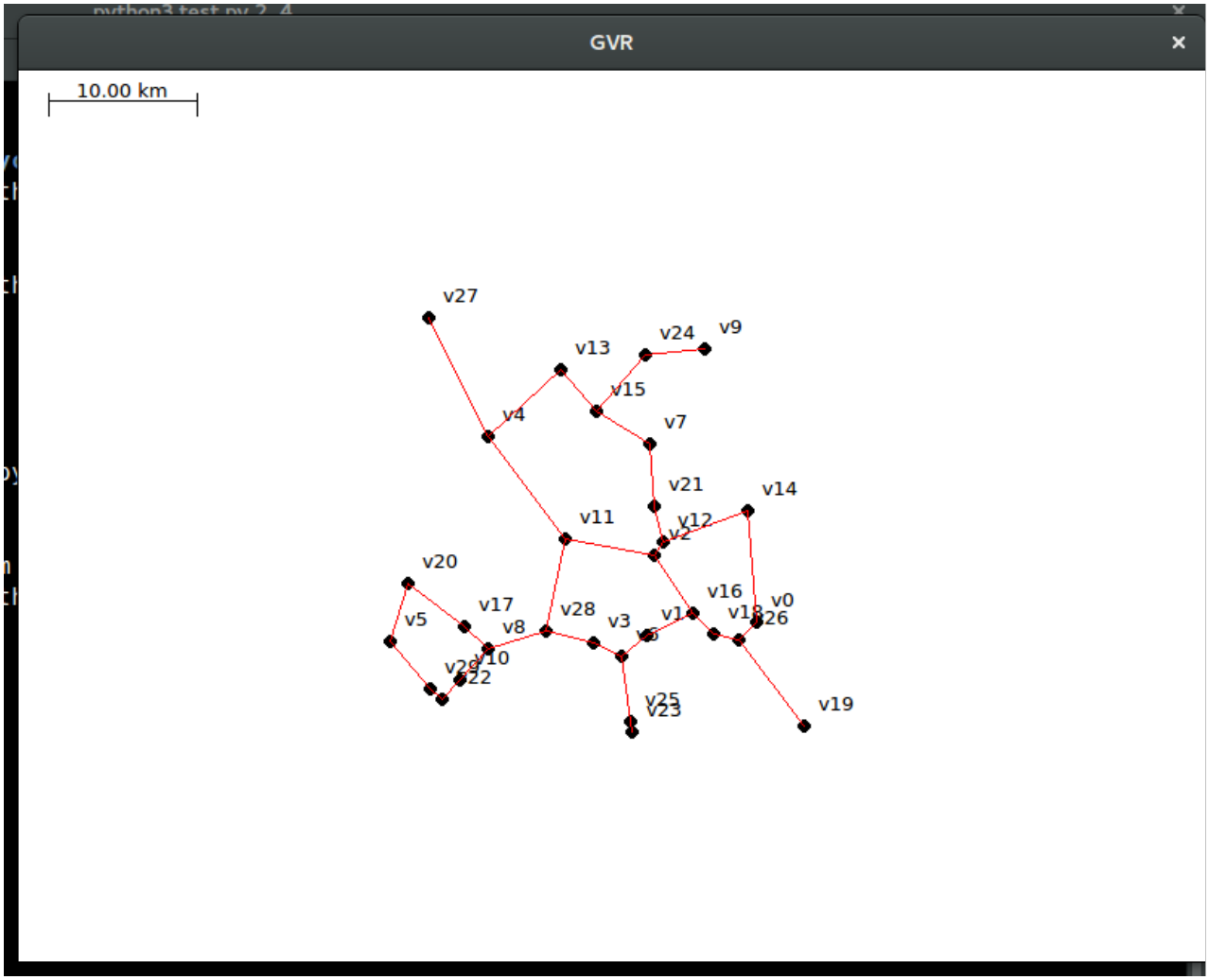

Figure 3 – Graphe de voisinage realtif

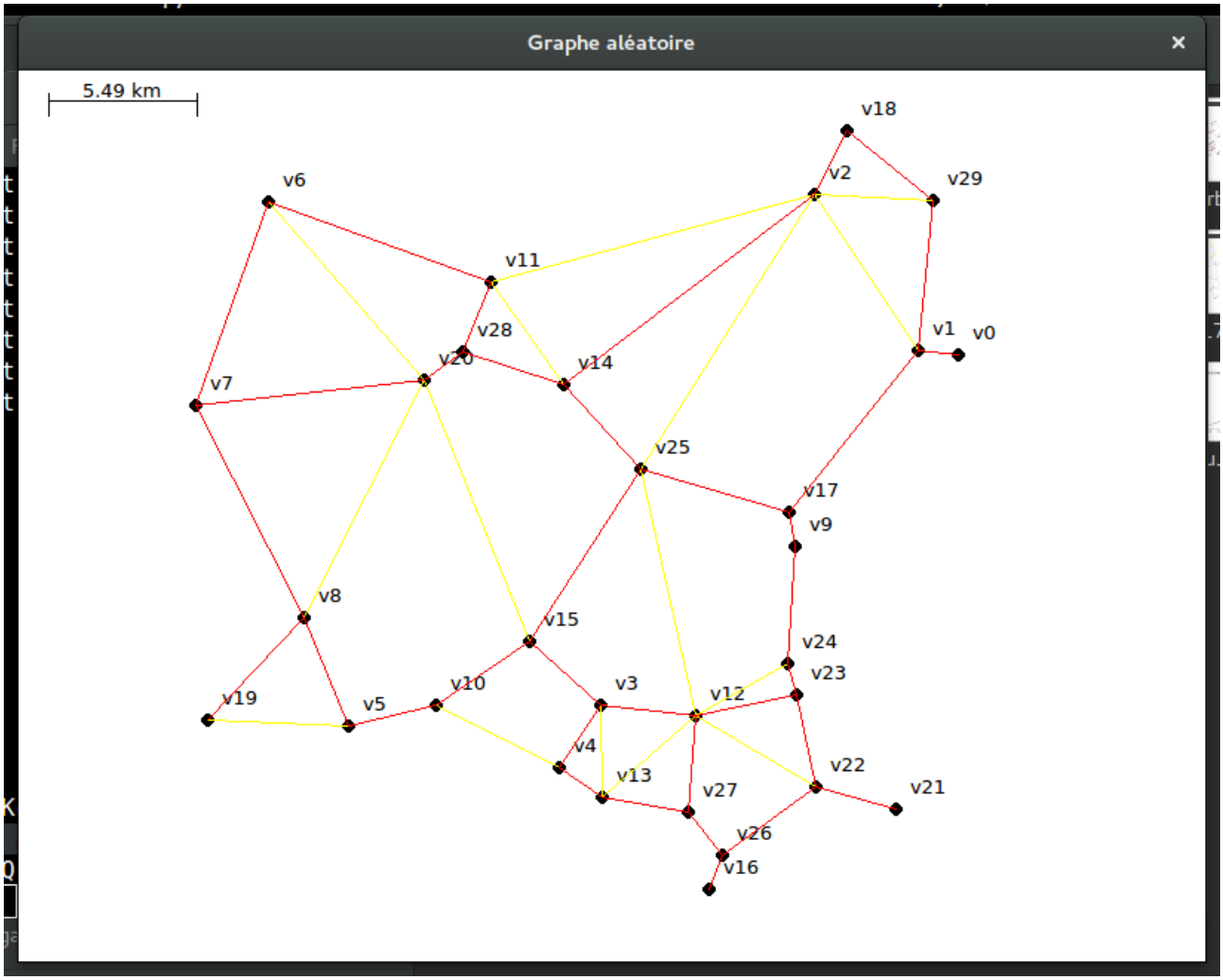

Figure 4 – Réseau généré

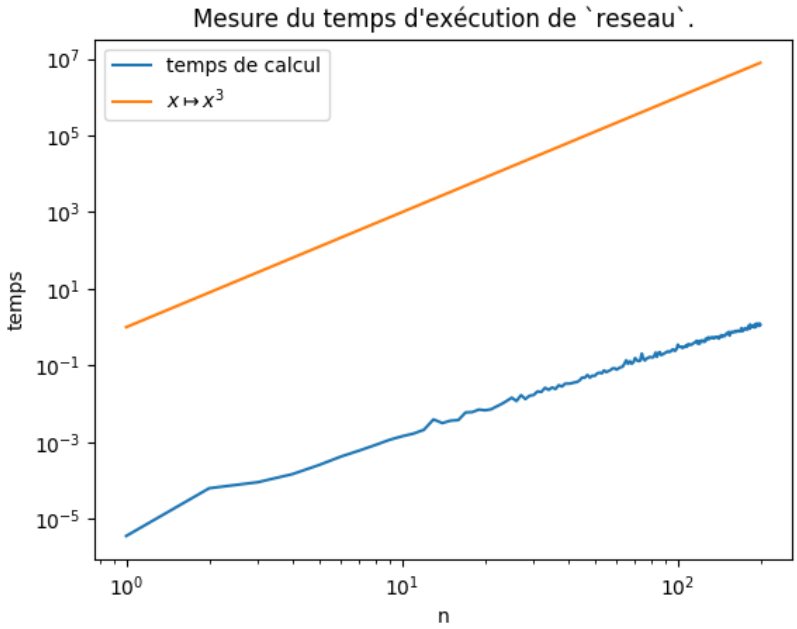

Figure 5 – Temps d'exécution du réseau naïf

2.3 Temps de génération d'un réseau

La figure 5 montre le temps nécessaire à la création d'un réseau. On remarque que la complexité est en $\Theta(n^3)$.

3 Triangulation de Delaunay

3.1 Pratique

3.2 Aspect théorique

3.2.1 Condition pour un graphe de Delaunay

Pour toute paire $\{A, B\}$ de points de $\mathcal{V}$, si on appelle $\mathcal{P}$ et $\mathcal{P}'$ les deux demis plans fermés de frontière (A, B) privés de A et B, alors $\{A, B\}$ est une arête de la triangulation de Delaunay sit et seulement si

$$\forall C \in \mathcal{P} \cap \mathcal{V}, \forall D \in \mathcal{P}' \cap \mathcal{V}, \widehat{ACB} + \widehat{ADB} \leq \pi$$

preuve : Supposons que $\{A, B\}$ soit une arête. Il existe donc un cercle $\mathcal{C}$ d'intérieur vide avec A et B sur le cercle. Cela correspond aux lignes $\mathcal{L}(\theta, A, B)$ et $\mathcal{L}(\pi - \theta, A, B)$. Quitte à renommer $\mathcal{P}$ et $\mathcal{P}'$, on peut supposer que le premier contour correspond à $\mathcal{P}$. On a donc :

$$\forall C \in \mathcal{P} \cap \mathcal{V}, \widehat{ACB} < \theta \tag{2}$$

$$\forall D \in \mathcal{P}' \cap \mathcal{V}, \widehat{ADB} < \theta - \pi \tag{3}$$

Ainsi :

$$\forall C \in \mathcal{P} \cap \mathcal{V}, \forall D \in \mathcal{P}' \cap \mathcal{V}, \widehat{ACB} + \widehat{ADB} \leq \pi$$

Supposons maintenant que :

$$\forall C \in \mathcal{P} \cap \mathcal{V}, \forall D \in \mathcal{P}' \cap \mathcal{V}, \widehat{ACB} + \widehat{ADB} \leq \pi$$

Posons :

$$m_p = \max_{C \in \mathcal{P} \cap \mathcal{V}} \widehat{ACB} \tag{4}$$

$$m_{p'} = \max_{D \in \mathcal{P}' \cap \mathcal{V}} \widehat{ADB} \tag{5}$$

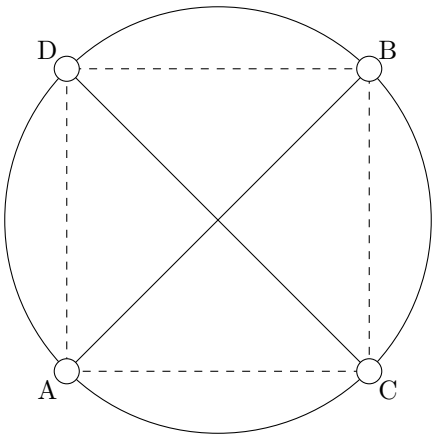

Figure 6 – Cas où le graphe de Delaunay n'est pas planaire.

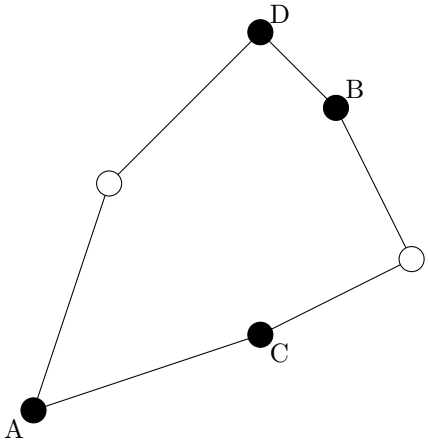

Figure 7 – Maille polygonale.

On a donc

$$m_p + m_{p'} \leq \pi$$

D'après le théorème de l'ange inscrit, si on pose $\theta = \max\{m_p, m_{p'}\}$, les contours $\mathcal{L}(\theta, A, B) \cap \mathcal{P}$ et $\mathcal{L}(\pi - \theta, A, B) \cap \mathcal{P}'$ forment un cercle d'intérieur vide avec A et B sur le cercle. □

3.2.2 Le graphe de Delaunay est planaire

On remarque qu'il existe des configurations pour lesquelles cette assertion est fausse, par exemple la figure 6. On va donc supposer que les configurations défavorables (4 points sur un cercle) ont une probabilité nulle d'être rencontrées.

preuve Si l'on suppose qu'il existe deux arêtes $\{A, B\}$ et $\{C, D\}$ qui se croisent, on a alors $\widehat{ADB} + \widehat{ACB} < \pi$ et $\widehat{DBC} + \widehat{DAC} < \pi$. Or $ABCD$ est un quadrilatère, donc $\widehat{ADB} + \widehat{ACB} + \widehat{DBC} + \widehat{DAC} = 2\pi$, contradiction. □

3.2.3 Le graphe de Delaunay est une triangulation

Montrons que le graphe de Delaunay est une triangulation.

preuve Supposons qu'il existe une face polygonale à plus de 3 côtés dans le graphe de Delaunay. On choisit quatre points A, B, C, D de cette faces tels que $\{A, C\}$ et $\{B, D\}$ peuvent être des arêtes, mais pas $\{A, B\}$ et $\{C, D\}$. On a par exemple une maille semblable à la figure 7.

On a alors :

$$\widehat{ACB} + \widehat{ADB} > \pi \tag{6}$$

$$\widehat{CAD} + \widehat{CBD} > \pi \tag{7}$$

$$\tag{8}$$

Puisque $ACDB$ est un quadrilatère, la somme de ses angles devrait être de 2π, ce qui est absurde ici. □

3.2.4 Condition sur les faces

Montrons tout triplet $\{A, B, C\} \subseteq \mathcal{V}$ est une face de la triangulation si et seulement si le disque du cercle circonscrit au triangle ABC est vide de tout point de $\mathcal{V}$ autre que A, B, C.

preuve Supposons que le disque du cercle $\mathcal{C}$ circonscrit à ABC soit vide de tout point. On va montrer que l'arête A, B existe. On pose $\theta = \widehat{ACB}$. On a alors $\mathcal{C} = (\mathcal{L}(\theta, A, B) \cap \mathcal{P}) \cup (\mathcal{L}(\pi - \theta, A, B) \cap \mathcal{P}')$. Puisque le disque est vide,

$$\forall D \in \mathcal{P}' \cap \mathcal{V}, \widehat{ADB} < \pi - \theta$$

Donc, puisque C est le point de plus grand angle sur $\mathcal{P}$:

$$\forall E \in \mathcal{P} \cap \mathcal{V}, \forall D \in \mathcal{P}' \cap \mathcal{V}, \widehat{AEB} + \widehat{ADB} \leq \pi$$

Donc l'arête $\{A, B\}$ existe. On montre de la même façon que $\{A, C\}$ et $\{C, B\}$ existent.

Supposons que $\{A, B, C\}$ soit une face de la triangulation. En particulier, C est alors le point de plus grand angle $\widehat{ACB} = \theta$ sur $\mathcal{P}$. Puisque $\{A, B\}$ est une arête, d'après le lemme, tout point $D \in \mathcal{P}' \cap \mathcal{V}$ est d'angle $\widehat{ADB}$ plus petit que $\pi - \theta$. Donc un tel point D ne peut être dans le disque délimité par $(\mathcal{L}(\theta, A, B) \cap \mathcal{P}) \cup (\mathcal{L}(\pi - \theta, A, B) \cap \mathcal{P}')$. □

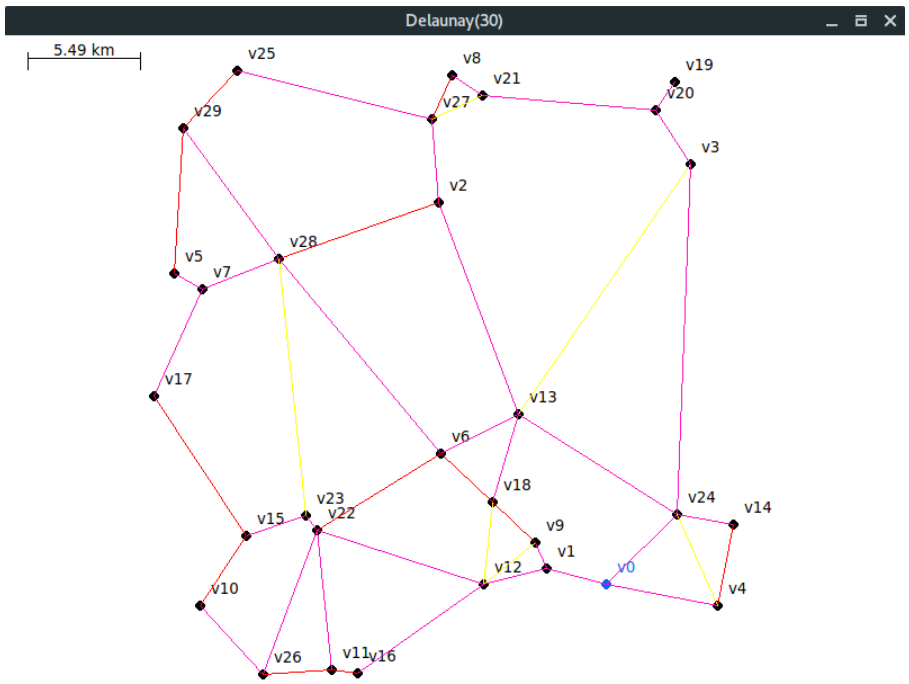

Figure 8 – Algorithme de Dijkstra avec des coûts constants

Deuxième partie

Algorithme de Dijkstra pour la recherche du plus court chemin

4 Algorithme de Dijkstra

4.1 Coûts constants

Après avoir implémenté l'algorithme de Dijkstra, on le teste avec des coûts constants (figure 8).

4.2 Coûts variables

On implémente la variation du coût de la manière suivante : on stocke le mode de calcul des coûts dans la classe `Graphe` et les objets de type `Arete` vienne le lire pour donner leur coût avec la méthode décorée `cout`. Le résultat est montré par les figures 9 et 10.

4.3 Temps de calcul

On mesure le temps d'exécution de l'algorithme (figure 11).

5 Dijkstra avec tas

On implémente Dijkstra avec un tas et on compare les temps d'exécutions (figure 12). On remarque qu'ici le temps d'exécution de Dijkstra avec tas est supérieur. Cela peut être dû à l'implémentation. En effet Dijkstra sans tas utilise des fonctions `builtin` qui sont donc plus rapide. Étant donné que les temps d'exécutions, on peut penser que $m \in \Theta(n)$.

6 Algorithme A*

On implémente l'algorithme A* et on vérifie qu'il donne le même résultat que Dijkstra en visitant moins de sommets.

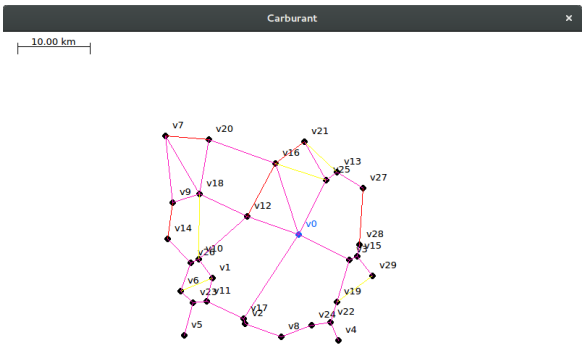

Figure 9 – Dijkstra avec le carburant comme coût

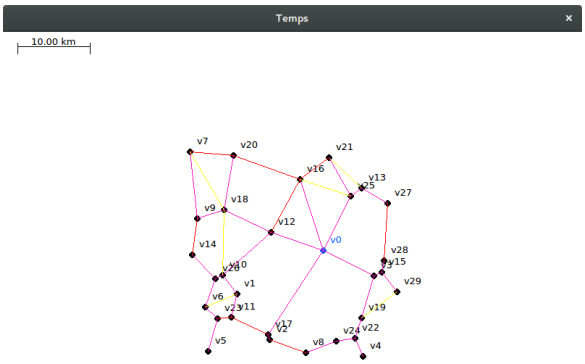

Figure 10 – Dijkstra avec le temps comme coût

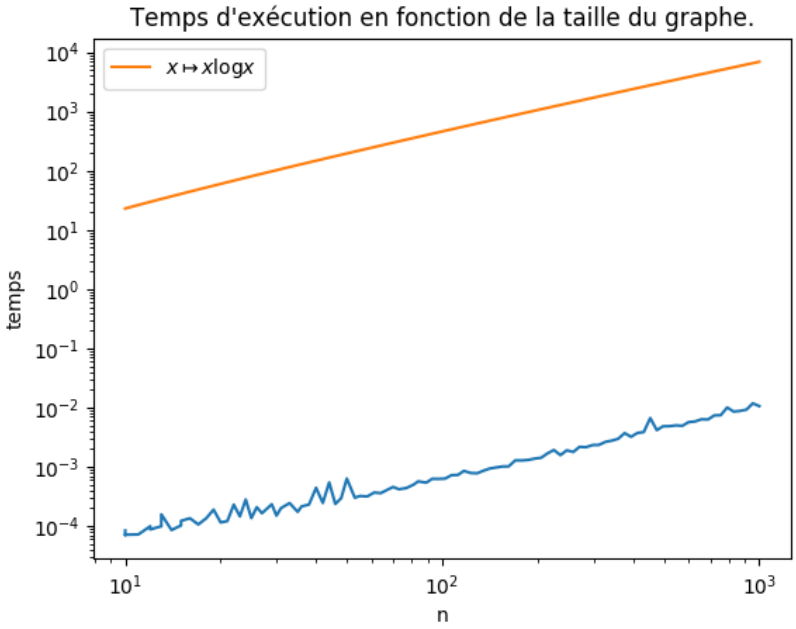

Figure 11 – Temps d'exécution de l'algorithme de Dijkstra

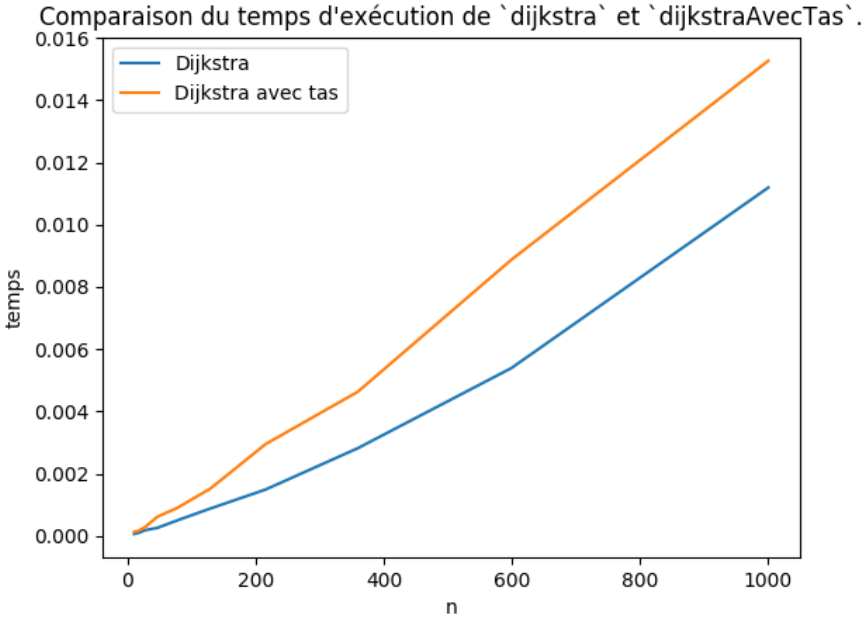

Figure 12 – Comparaison des temps d'exécution de Dijkstra et Dijkstra avec tas

On compare ensuite les temps d'exécutions des 3 algorithmes (figure 13). On remarque que A* est le plus efficace quand le nombre de points augmente.

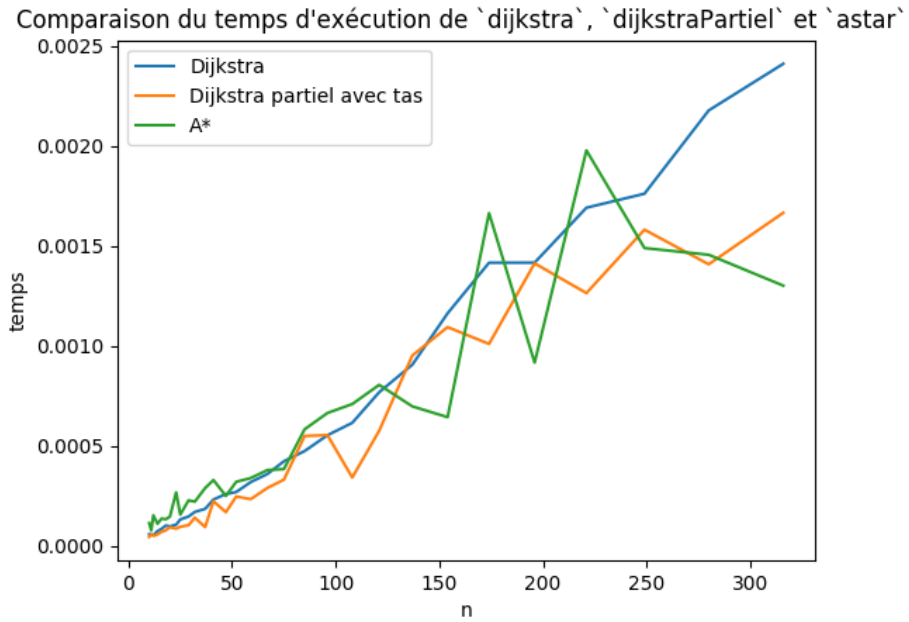

Figure 13 – Comparaison des trois algorithmes (temps moyens sur 10 exécutions)

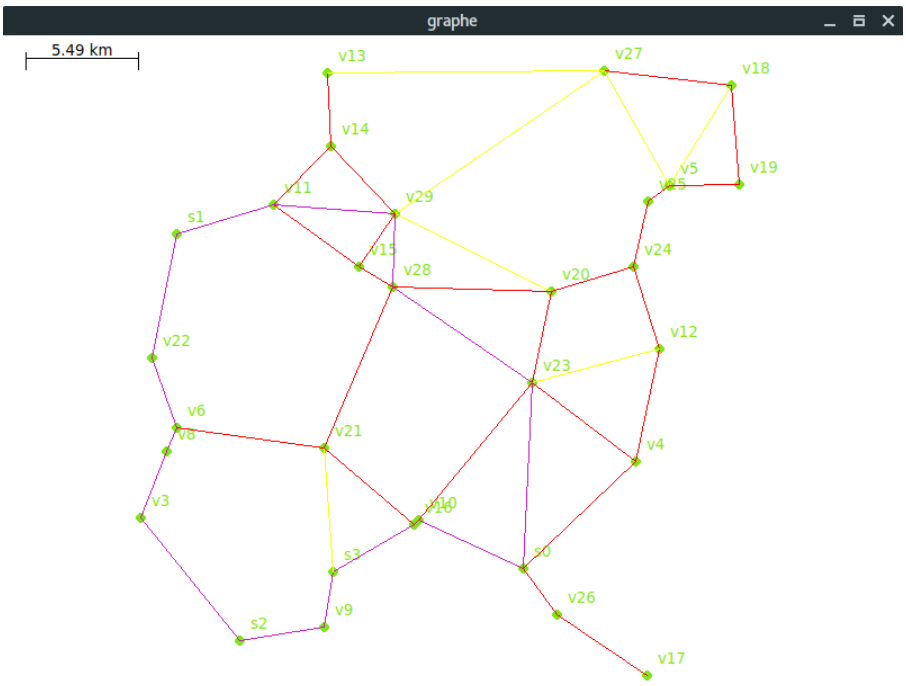

Figure 14 – Tournée optimale obtenue par backtracking

Troisième partie
Problème du voyageur de commerce

7 Résolution par Backtracking

7.1 Résolution

On résout le problème du voyageur de commerce par backtracking. Un état correspond à un parcours partiel de la tournée (représenté par une liste d'indice des sommets de la tournée). L'ensemble des extensions possibles d'un état correspond à l'ajout d'un sommet non encore visité. On élague si le cout minimum est inférieur au cout actuel auquel on ajoute le cout de l'ajout d'un sommet.

On obtient le résultat de la figure 14.

7.2 Temps d'exécution

En mesurant le temps d'exécution de l'algorithme, on note son caractère exponentiel (figure 15), ce qui le rend inutilisable en pratique.

8 Résolution approchée

On résout le problème de manière approchée en construisant un graphe de coût. Ce graphe est complet. On utilise alors l'algorithme de Prim pour trouver un arbre couvrant minimal et ainsi en déduire un ordre de parcours des points de la tournée. Les figures 16 et 17 montrent que les algorithmes peuvent donner des solutions différentes.

La figure 18 donne le temps d'exécution de la résolution approchée. On remarque que cette solution semble beaucoup plus rapide que la solution naïve.

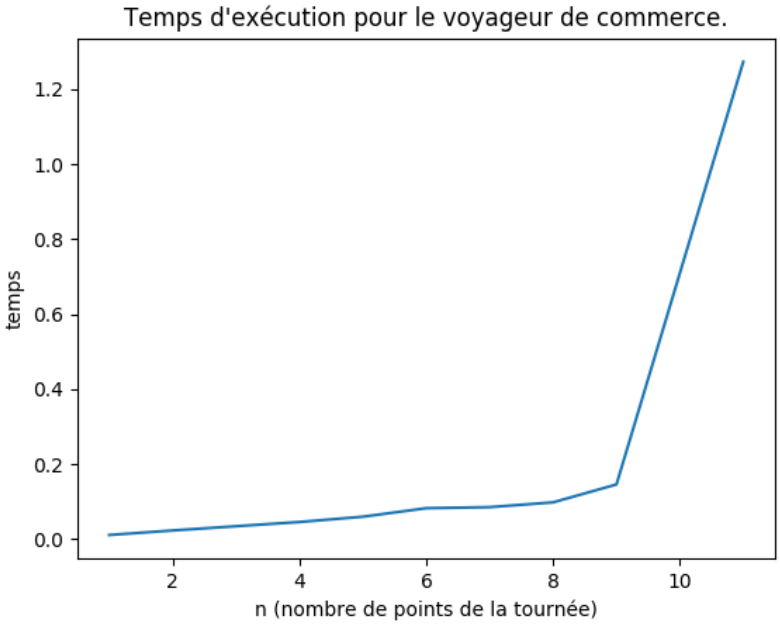

Figure 15 – Temps d'exécution de l'algorithme par backtracking

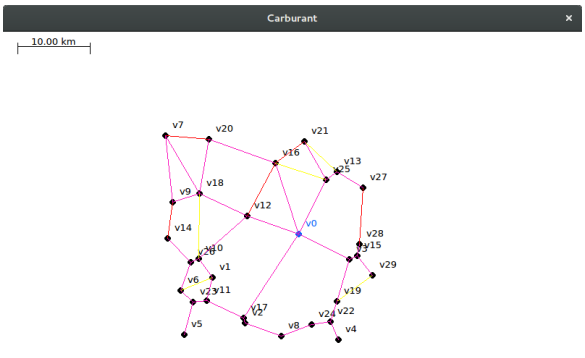

Figure 16 – Voyageur de commerce approché

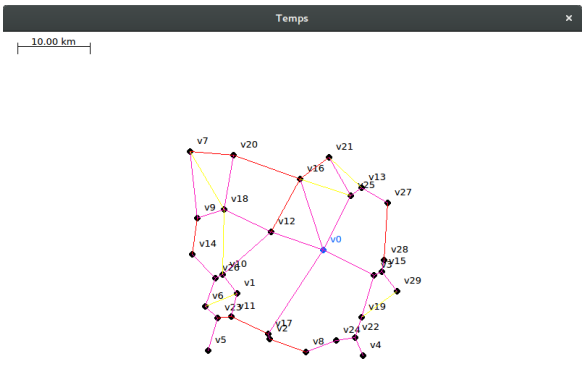

Figure 17 – Voyageur de commerce naïf

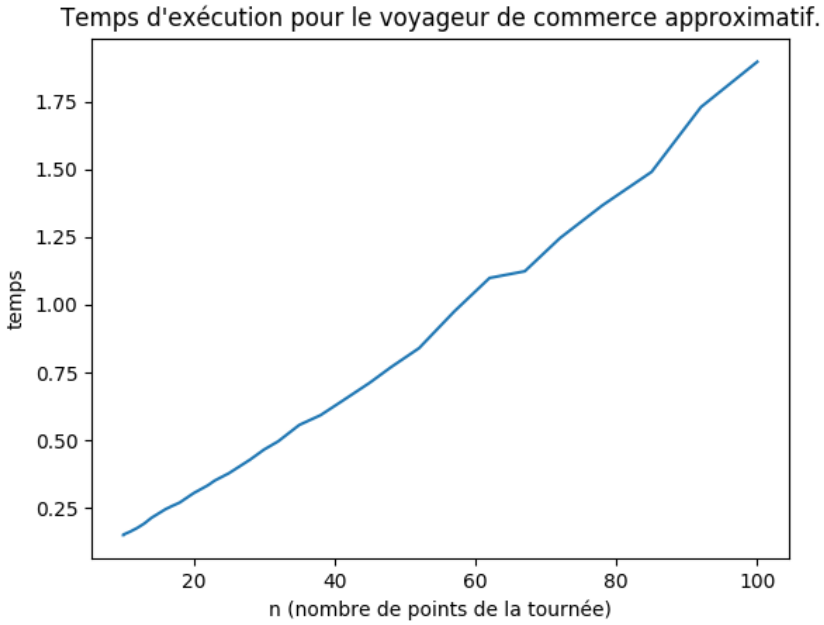

Figure 18 – Temps d'exécution de la résolution approchée

Table des figures