

Convergence de l'itération de Lloyd

Maxime Duval & Michaël Karpe

7 mai 2018

1 Vecteurs de relocalisation

1.1 Calcul

La visualisation des vecteurs de relocalisation s'obtient en traçant les segments entre les générateurs avant et après itération de Lloyd. Le résultat obtenu est présenté Figure 1, où l'on a tracé en rouge les générateurs des cellules à l'itération précédente.

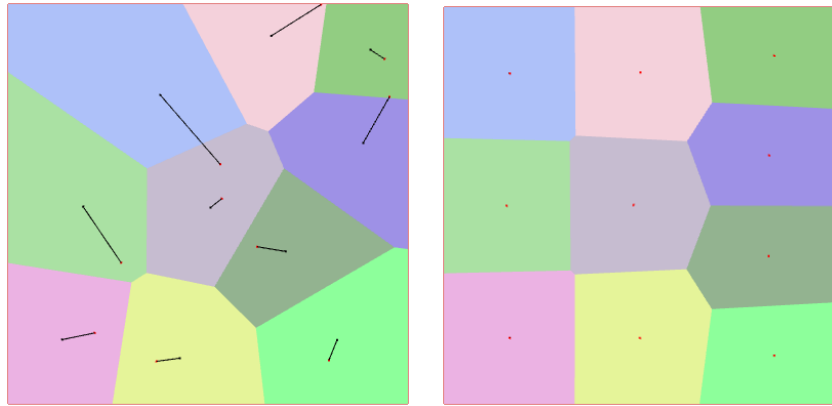


Figure 1: Visualisation des vecteurs de relocalisation (à gauche) et des cellules après convergence (à droite).

1.2 Convergence

Pour le tracé des courbes de convergence, nous avons sauvegardé les différentes valeurs dans des fichiers .txt et traité les données en langage Python, avec la bibliothèque matplotlib.pyplot. Les résultats obtenus sont présentés en Figure 2. On constate que cette évolution est globalement décroissante, avec une augmentation de l'énergie lorsque les cellules se "déloquent" après avoir convergé vers un minimum local d'énergie.

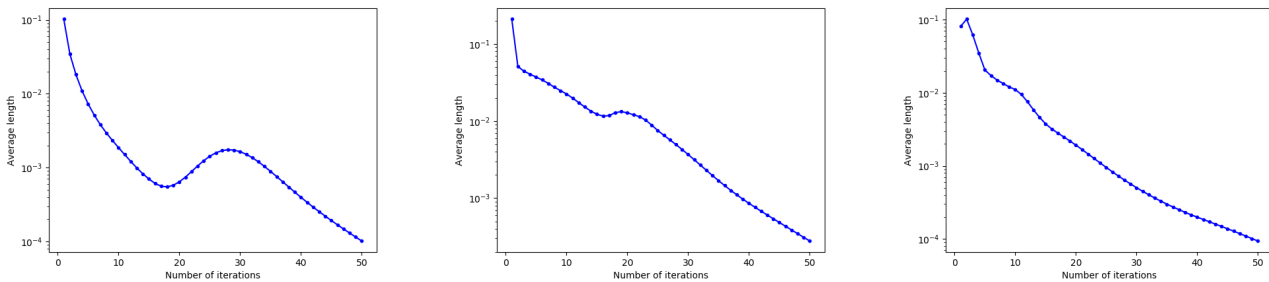


Figure 2: Evolution en échelle logarithmique de la norme moyenne des vecteurs de relocalisation en fonction du nombre d'itérations de Lloyd pour une initialisation uniforme (à gauche), dans un coin (au milieu) et sur une ligne (à droite).

En zoomant la première courbe de convergence entre 30 et 60 itérations et en la traçant en échelle réelle (et non pas en échelle logarithmique comme sur les différentes images), il apparaît clairement que la convergence de la longueur moyenne est quadratique (Figure 3).

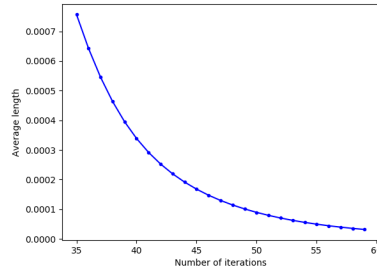


Figure 3: Visualisation de l'échantillonnage du domaine carré avec $N = 10000$ points.

2 Distribution des énergies par cellule

2.1 Calcul

L'énergie globale d'une partition de domaine en N_c cellules \mathcal{C}_i est donnée par la formule suivante :

$$E_{dom} = \sum_{i=1}^{N_c} E_{cell} = \sum_{i=1}^{N_c} \int_{x \in \mathcal{C}_i} \|x - x_i\|^2 dx$$

Pour réaliser le calcul des intégrales, on procède par méthode de Monte-Carlo, en échantillonnant un nombre N de points x_j sur l'ensemble du domaine et en approximant l'intégrale par une somme discrète :

$$E_{cell} = \int_{x \in \mathcal{C}_i} \|x - x_i\|^2 dx = \sum_{x_j \in \mathcal{C}_i} \|x_j - x_i\|^2$$

Une représentation de cet échantillonnage est réalisée en Figure 4 pour $N = 10000$. Pour assez couvrir entièrement le domaine, nous avons choisi $N = 100000$.

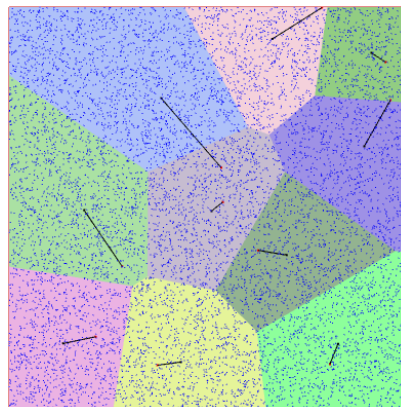


Figure 4: Visualisation des cellules après convergence dans un domaine en forme de sablier.

2.2 Convergence

Toujours dans le domaine carré avec une initialisation de 10 points, l'évolution de l'énergie des cellules est représentée Figure 5. Pour les 3 initialisations testées, on constate que les cellules convergent vers une énergie proche après une trentaine d'itérations. Cette convergence est rapide pour les initialisations uniforme et sur une ligne, avec une légère correction d'énergie à environ 23 itérations pour l'initialisation uniforme, en raison des cellules qui se bloquaient (ce que l'on retrouve avec la croissance de la norme moyenne des vecteurs de relocalisation autour du même nombre d'itérations).

Pour l'initialisation dans un coin, la convergence est plus lente car plus d'itérations sont nécessaires pour permettre aux 10 générateurs d'occuper l'espace entier du domaine. Cependant, on ne constate pas de saut majeur d'énergie qui serait dû à un blocage de cellules entre elles.

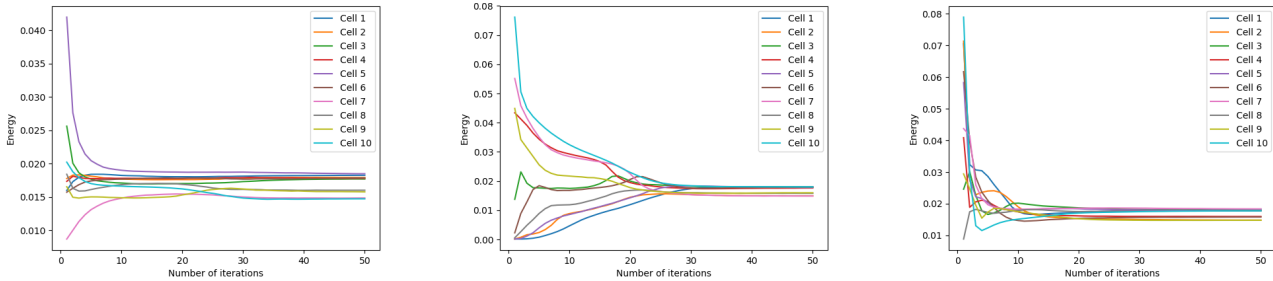


Figure 5: Evolution de l'énergie moyenne des cellules en fonction du nombre d'itérations de Lloyd pour une initialisation uniforme (à gauche), dans un coin (au milieu) et sur une ligne (à droite).

2.3 Influence du domaine

La forme du domaine influence beaucoup les résultats de convergence. Pour le domaine circulaire par exemple, qui est un domaine "régulier" dans le sens où les bords ne présentent pas de "coins" ou de rupture brutale de direction (les bords sont bien continus et dérivables), la convergence se fait sans problème au bout d'une dizaine d'itérations et n'est pas caractérisée par des sauts particuliers. Les résultats pour le domaine circulaire sont présentés Figure 6.

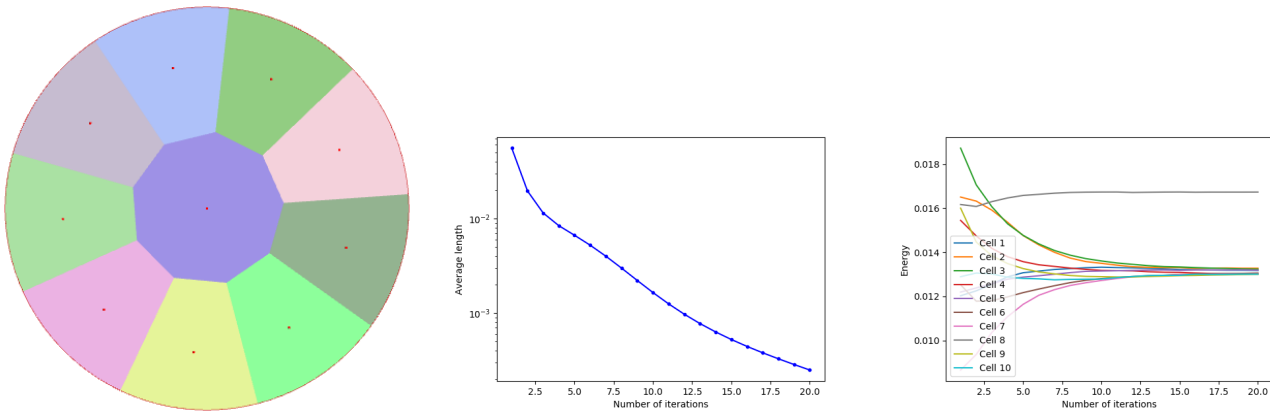


Figure 6: Visualisation des cellules après convergence dans un domaine circulaire (à gauche), évolution de la norme moyenne des vecteurs de relocalisation (au milieu) et de l'énergie moyenne des cellules (à droite) dans un domaine circulaire en fonction du nombre d'itérations de Lloyd pour une initialisation uniforme.

Pour les autres domaines fournis, on constate des comportements plus originaux, dus aux mêmes observations que formulées précédemment. Pour la forme en L, le coin intérieur du L va provoquer quelques augmentations de la

longueur moyenne, les cellules "franchissant" le coin se déplaçant rapidement d'un coup pour s'adapter à la forme, et la convergence est assez lente relativement aux formes carrée et circulaire.

Pour le domaine en forme de clé, la convergence est aussi plus lente en raison de la complexité du domaine, et les sauts de longueur moyenne sont plus importants en raison de cette complexité, car les cellules se déplacent beaucoup lorsqu'un "coin" est franchi et comblé par les cellules. L'ensemble des résultats est présenté dans les Figures 7 à 9.

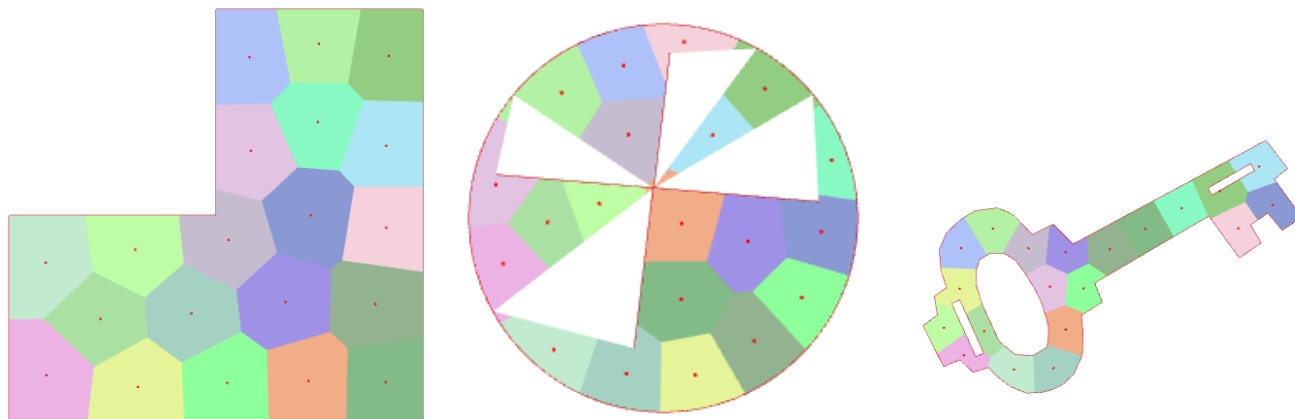


Figure 7: Visualisation des cellules après convergence dans différents domaines.

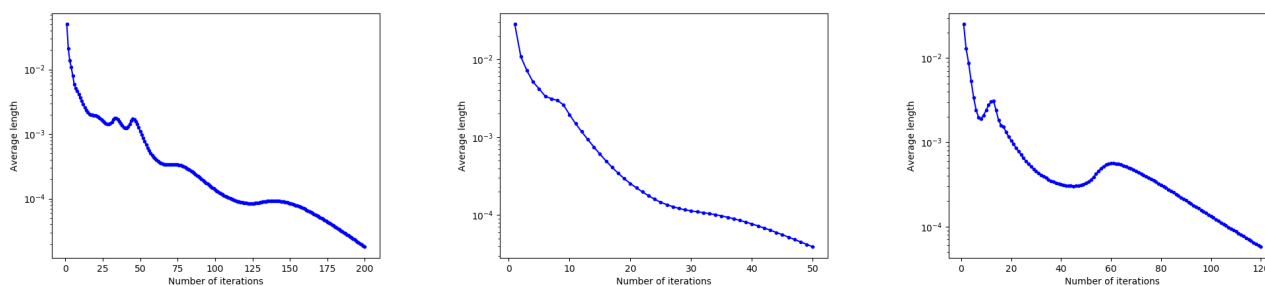


Figure 8: Evolution de la norme moyenne des vecteurs de relocalisation en fonction du nombre d'itérations de Lloyd pour un domaine en L (à gauche), un domaine comportant une croix (au milieu) et un domaine en forme de clé (à droite).

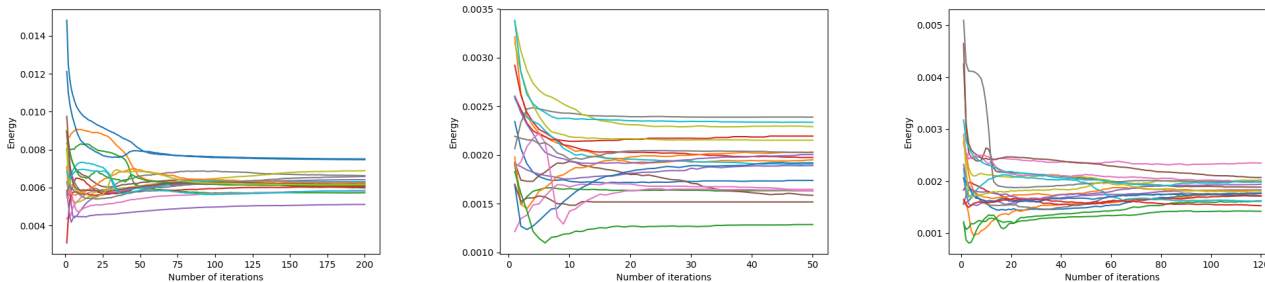


Figure 9: Evolution de l'énergie moyenne des cellules en fonction du nombre d'itérations de Lloyd pour un domaine en L (à gauche), un domaine comportant une croix (au milieu) et un domaine en forme de clé (à droite).

Pour les domaines en forme de sablier (codées en dur dans le code C++ en fonction d'un paramètre correspondant à la taille du goulet d'étranglement), cette difficulté pour l'itération de Lloyd à franchir des coins ou des passages étroits est nettement mise en évidence. On constate en effet que, de façon générale, plus le goulet d'étranglement est étroit, plus les cellules mettent du temps à franchir le goulet d'étranglement, et donc plus la remontée de longueur moyenne est importante lorsqu'il est franchi (Figure 11).

Initialisation L'observation sur l'ensemble des domaines nous amène à penser que pour des domaines assez simples et réguliers, une initialisation uniforme est assez efficace. Pour des formes de domaines plus complexes, il convient d'initialiser l'algorithme avec plus de générateurs aux bords irréguliers du domaine, ou aux endroits présentant des passages étroits.

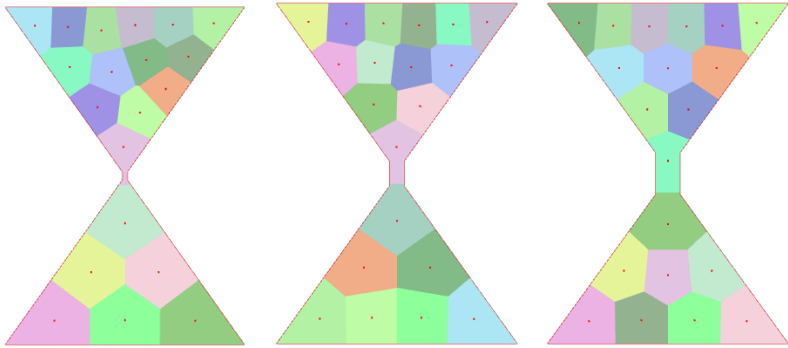


Figure 10: Visualisation des cellules après convergence dans un domaine en forme de sablier avec goulet d'étranglement de largeur croissante de gauche à droite.

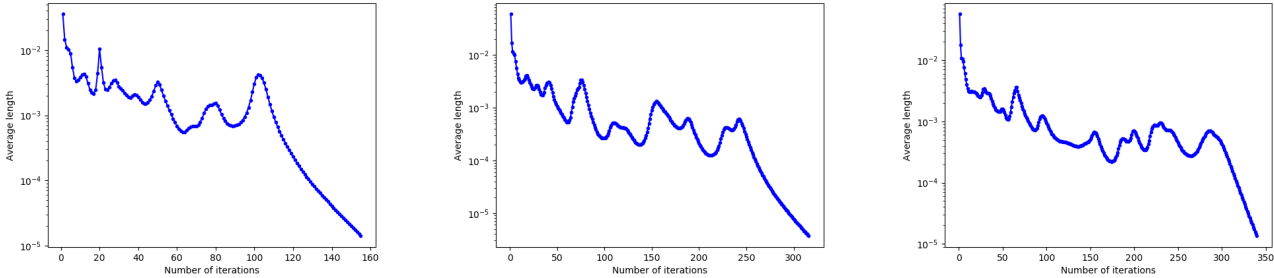


Figure 11: Evolution de la norme moyenne des vecteurs de relocalisation en fonction du nombre d'itérations de Lloyd pour un domaine en sablier avec goulet d'étranglement de largeur croissante de gauche à droite.

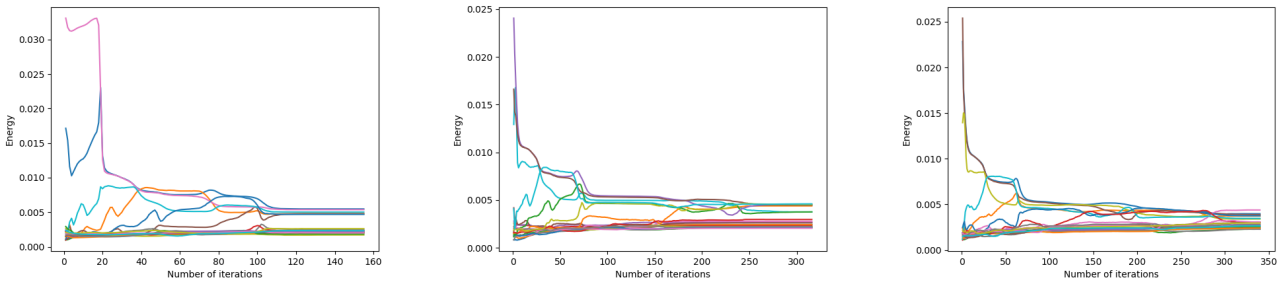


Figure 12: Evolution de l'énergie moyenne des cellules en fonction du nombre d'itérations de Lloyd pour un domaine en sablier avec goulet d'étranglement de largeur croissante de gauche à droite.

3 Dépassement des centres (overshooting) et inertie temporelle

3.1 Overshooting

L'overshooting consiste à déplacer les générateurs non pas aux centres de masse des cellules, mais un peu au-delà. Informatiquement, on considère le vecteur de déplacement v_N à l'itération N d'une itération de Lloyd classique, et au lieu de déplacer le générateur selon v_N , on le déplace selon $v_{N+1} = (1 + \alpha)v_N$, où $\alpha \in \mathbb{R}^+$.

On distingue trois classes de comportement de l'itération de Lloyd selon la valeur du paramètre d'overshooting (déterminées de façon approximative en testant numériquement, les valeurs limites de ces comportements changent notamment selon la forme du domaine et l'initialisation des générateurs) pour le domaine carré et pour une initialisation uniforme (Figures 13 à 15) :

- $\alpha \in [0, 1]$: la paramètre d'overshooting est assez petit : il permet d'accélérer la convergence de l'itération de Lloyd
- $\alpha \approx [1, 1 + \epsilon]$: le paramètre d'overshooting est un peu trop grand : l'itération aboutit à une situation où $(1 + \alpha)v_{N+1} = -(1 + \alpha)v_N$, ce qui entraîne l'alternance des générateurs entre deux configurations
- $\alpha \gg 1 + \epsilon$: le paramètre d'overshooting est beaucoup trop grand : l'itération de Lloyd diverge et a un comportement chaotique

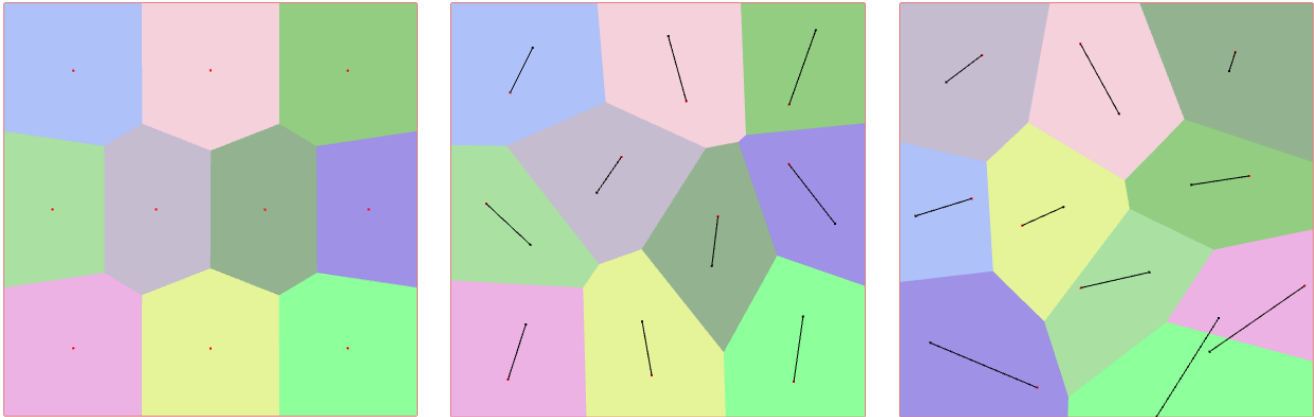


Figure 13: Visualisation des cellules après convergence avec un paramètre d'overshooting avec un paramètre d'overshooting croissant de gauche à droite ($\alpha \in \{0.5, 1.1, 1.5\}$).

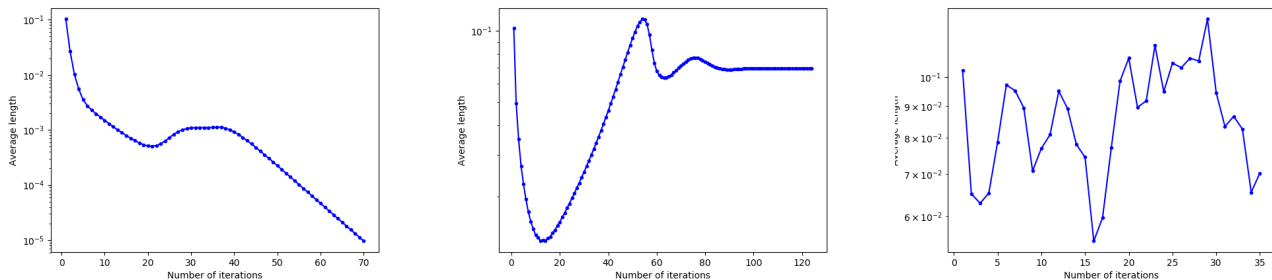


Figure 14: Evolution de la norme moyenne des vecteurs de relocalisation en fonction du nombre d'itérations de Lloyd pour un domaine carré avec un paramètre d'overshooting croissant de gauche à droite ($\alpha \in \{0.5, 1.1, 1.5\}$).

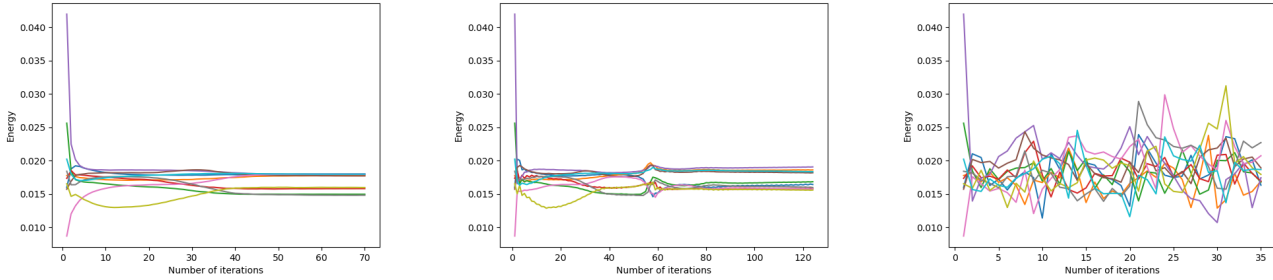


Figure 15: Evolution de l'énergie moyenne des cellules en fonction du nombre d'itérations de Lloyd pour un domaine carré avec un paramètre d'overshooting croissant de gauche à droite ($\alpha \in \{0.5, 1.1, 1.5\}$).

Ainsi, si le paramètre est trop important, l'itération de Lloyd risque de ne pas converger. Pour prévenir ce problème, on peut imposer un paramètre α assez petit (au moins strictement inférieur à 1, inférieur à 0.5 si l'on veut être sûr de converger y compris dans des cas particuliers).

Un autre risque est que si le paramètre α est trop grand, l'itération de Lloyd fasse sortir les générateurs du domaine. Pour pallier ce problème informatiquement, on pourrait utiliser la fonction `inside` de `cdt.h`, pour appliquer l'overshooting à la relocalisation d'un générateur seulement si le nouveau générateur est bien à l'intérieur (`inside`) du domaine.

3.2 Inertie temporelle

L'inertie temporelle consiste à garder en mémoire les centres de masse des itérations précédentes, et à déplacer les générateurs avec une combinaison linéaire des n vecteurs déplacements précédents. Informatiquement, on déplace un générateur à l'itération N selon le vecteur $v_{N+1} = \sum_{i=N-n}^N w_i v_i$, où les w_i sont des poids données par l'utilisateur aux vecteurs de relocalisation des itérations précédentes.

On distingue trois classes de comportement de l'itération de Lloyd selon la valeur des différents coefficients (classes déterminées de façon approximative en testant numériquement, les valeurs limites de ces comportements changent notamment selon la forme du domaine et l'initialisation des générateurs) pour le domaine carré et pour une initialisation uniforme, avec un nombre $n = 5$ de vecteurs déplacements pris en compte (Figures 16 à 18):

- $\sum_{i=N-n}^N w_i \leq 1$: les précédents vecteurs de relocalisation sont relativement peu pris en compte : la convergence est accélérée car les générateurs se déplacent plus vite, puisque les premières itérations concernent des déplacements plus importants
- $\sum_{i=N-n}^N w_i \approx 1 + \epsilon$: les précédents vecteurs de relocalisation sont un peu trop pris en compte, et les générateurs se déplacent un peu trop : leurs positions alternent entre 2 configurations (comportement oscillant)
- $\sum_{i=N-n}^N w_i \gg 1 + \epsilon$: les premiers vecteurs de relocalisation sont beaucoup trop pris en compte, et l'addition des vecteurs avec de trop grands coefficients produit des vecteurs de plus en plus grands : l'itération de Lloyd diverge et les générateurs sortent du domaine

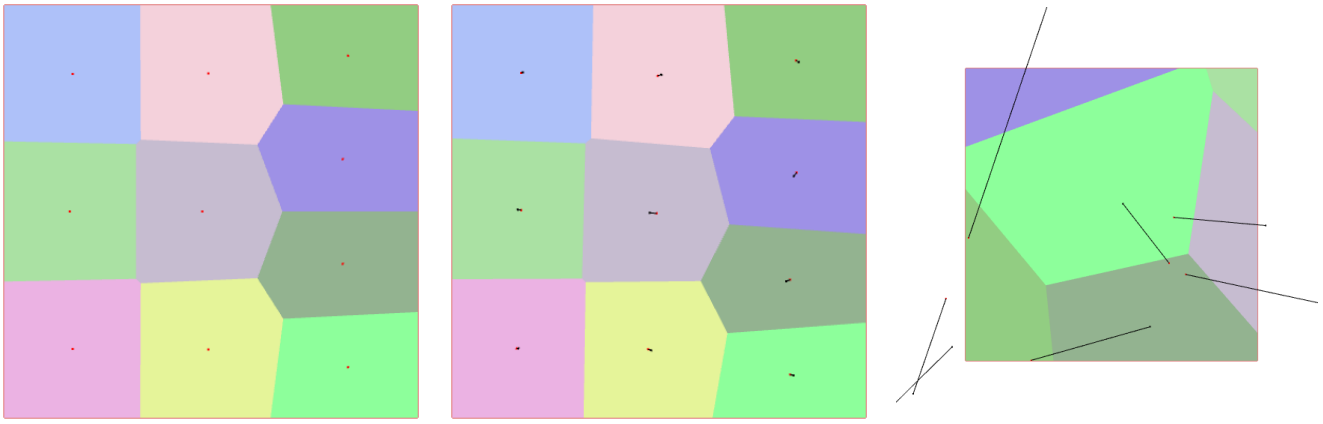


Figure 16: Visualisation des cellules après convergence avec inertie temporelle des 5 derniers déplacements, avec des poids $w_i \in \{0.15, 0.30, 0.50\}$.

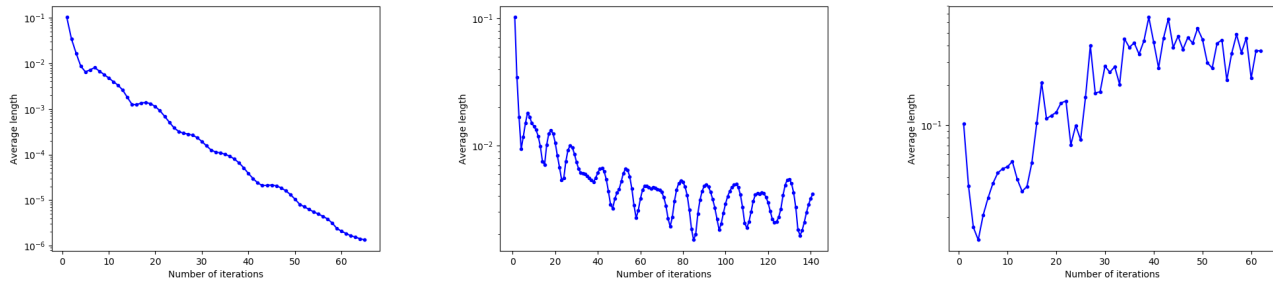


Figure 17: Evolution de la norme moyenne des vecteurs de relocalisation en fonction du nombre d'itérations de Lloyd pour un domaine carré avec inertie temporelle des 5 derniers déplacements, avec des poids $w_i \in \{0.15, 0.30, 0.50\}$.

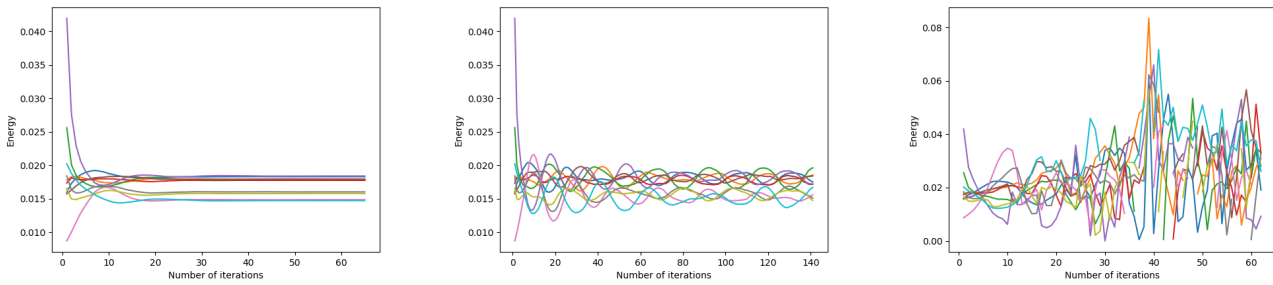


Figure 18: Evolution de l'énergie moyenne des cellules en fonction du nombre d'itérations de Lloyd pour un domaine carré avec inertie temporelle des 5 derniers déplacements, avec des poids $w_i \in \{0.15, 0.30, 0.50\}$.

La convergence est plus rapide lorsque les coefficients ne sont pas pris trop grands. En effet, la prise de comptes des "grands" vecteurs de déplacements précédents permet aux générateurs de se déplacer plus vite.

Cependant, de façon analogue au paramètre d'overshooting trop grand, des coefficients trop grands provoquent une divergence de l'algorithme (et même avec des coefficients petits, on constate une convergence avec de légères oscillations sur l'image de gauche en Figure 17). Pour pallier cela, on impose des coefficients petits, ou on vérifie que les générateurs restent à l'intérieur du domaine avec inside.