Sorbonne Université

Rapport Final

18 Mars 2023

Rapport de recherche sur l'algorithme Welzl résolvant le problème du cercle minimum - Projet individuel de l'UE CPA

Auteurs :
Alina Novikova

Encadrants :
Binh-Minh BUI-XUAN

Table des matières

Introduction

Ce projet vise à analyser les performances de calcul de l'algorithme de Welzl et de l'algorithme naïf pour le problème du cercle minimum couvrant un ensemble de points dans un plan. Ce problème est d'une grande importance pratique dans de nombreuses applications telles que la géométrie computationnelle, la vision par ordinateur, la robotique et de nombreux autres domaines.

L'algorithme de Welzl est une méthode récursive qui résout ce problème en temps linéaire. Cette méthode a été largement étudiée dans la littérature et plusieurs analyses de sa complexité ont été réalisées.

Dans ce rapport, nous définirons un problème du cercle minimum impliquant un ensemble de points dans un plan, et fournirons un bref aperçu et une critique d'algorithmes bien connus dans la littérature tels que l'algorithme de Ritter [4], l'algorithme de Shamos et Hoey [3] et l'algorithme de Chrystal [6].

Nous présentons ensuite une confrontation d'algorithmes sur le jeu de test de Varoumas et analysons les performances de l'algorithme de Welzl en termes de temps de calcul pour différents ensembles de points dans le plan en comparant ses résultats avec ceux de l'algorithme du cercle couvrant minimum naïf. Ce travail se fera à l'aide du langage Java et du fichier fourni par l'encadrant pour effectuer les TME.

Nous explorons également les limites de performance de l'algorithme de Welzl dans des cas particuliers et discutons des améliorations possibles qui pourraient être apportées à cet algorithme.

1 Définition du problème et analyse théorique des algorithmes existants

1.1 Présentation du problème du cercle minimum

Le problème du cercle couvrant minimum, également connu sous le nom de problème de disque minimum, est un problème classique de géométrie algorithmique. Il est défini comme suit : *étant donné un ensemble de points sur le plan, trouver un cercle de rayon minimum couvrant tous les points.*

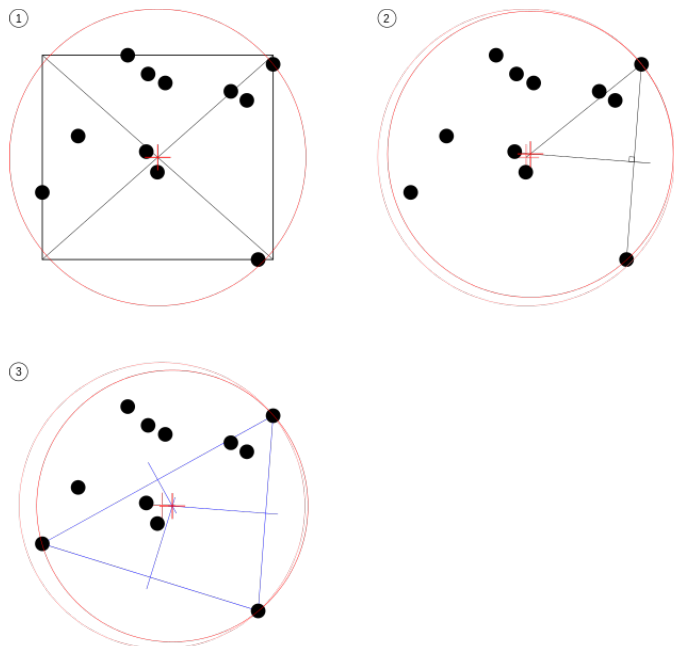

FIGURE 1 – Construction du cercle minimum [6]

Plus formellement, si P est un ensemble de n points dans le plan, le cercle minimum couvrant est défini comme le cercle de centre (cx, cy) et de rayon r, qui contient tous les points de P. Le but est de trouver les valeurs de cx, cy et r qui minimisent le rayon r du cercle.

Le problème du cercle minimum couvrant a une longue histoire en mathématiques et en informatique. L'un des premiers intéressés par le problème du cercle couvrant en 1800 fut le mathématicien allemand Carl Friedrich Gauss, qui présenta le problème qui consiste à considérer un cercle tracé sur un quadrillage et à demander combien de nœuds du quadrillage sont dans le cercle.

Au fil des ans, de nombreux mathématiciens ont travaillé sur ce problème et ont proposé différentes méthodes pour le résoudre. En 1857, James Joseph Sylvester, un mathématicien britannique, a proposé un théorème important :

Theorem 1 *étant donné un ensemble fini de points dans le plan qui ne sont pas tous sur une seule ligne, il existe toujours un triplet de points qui ne sont pas alignés. Ce triplet peut être utilisé pour déterminer le cercle minimum couvrant de l'ensemble de points.*

Au début des années 1980, un algorithme efficace a été proposé par Emo Welzl, un mathématicien suisse, qui a introduit la notion de minidisque et a développé un algorithme qui utilise une approche récursive pour déterminer le cercle minimum couvrant d'un ensemble de points dans le plan.

1.2 Algorithmes connus de la littérature

Il existe plusieurs algorithmes décrits dans la littérature pour résoudre ce problème, chacun avec ses avantages et ses inconvénients en termes de complexité temporelle et spatiale. Nous allons nous concentrer sur les algorithmes suivants :
— L'algorithme Ritter
— L'algorithme de Shamos et Hoey
— L'algorithme de Chrystal

1.2.1 L'algorithme Ritter

L'algorithme de Ritter est une méthode classique pour trouver le cercle minimum couvrant d'un ensemble de points dans le plan. L'idée de base de l'algorithme est de calculer un cercle initial qui englobe tous les points, puis de réduire progressivement le rayon du cercle en cherchant les points les plus éloignés du cercle et en recalculant le cercle minimum couvrant.

Algorithm 1: Ritter - Calculer le cercle minimum d'un espace de points

```
Data: Initial sphere S, calculated with the pair of points with the maximum
      point-to-point separation as a diameter, point (x, y)
Result: Minimum covering circle S
for p=(x,y) in Points do
    dx ← x − S.x;
    dy ← y − S.y;
    old_to_p_sq ← dx ∗ dx + dy ∗ dy;
    rad_sq ← S.rad ∗ S.rad;
    if old_to_p_sq > rad_sq then
        /* Point is outside current sphere. update.                    */
        old_to_p ← old_to_p_sq;
        S.rad ← (S.rad + old_to_p)/2.0 ;
        rad_sq ← S.rad ∗ S.rad ;   /* update square of radius for next compare */
        old_to_new ← old_to_p − S.rad ;
        S.x ← (S.rad ∗ S.x + old_to_new ∗ x)/old_to_p ;
        S.y ← (S.rad ∗ S.y + old_to_new ∗ y)/old_to_p ;
    end
end
```

L'un des avantages de l'algorithme de Ritter est sa simplicité de mise en œuvre et sa facilité de compréhension.

Il s'exécute en temps $O(nd)$ sur des entrées constituées de n points dans un espace à d dimensions, ce qui le rend très efficace.

Cependant, l'algorithme de Ritter peut avoir des performances médiocres en termes de précision. Dans son rapport [4] Ritter écrit que la sphère calculée est environ 5% plus grande que la sphère idéale de rayon minimum.

Pour améliorer la précision de l'algorithme, nous pouvons utiliser l'optimisation locale : une fois que l'algorithme de Ritter a produit un résultat grossier, des techniques d'optimisation locale peuvent être appliquées pour améliorer le résultat. Cela implique d'ajuster le rayon et le centre du cercle pour minimiser la distance entre le cercle et les points.

1.2.2 L'algorithme de Shamos et Hoey

Cet algorithme est basé sur une approche de type diviser-pour-régner et utilise le diagramme de Voronoï des points les plus éloignés. [3]

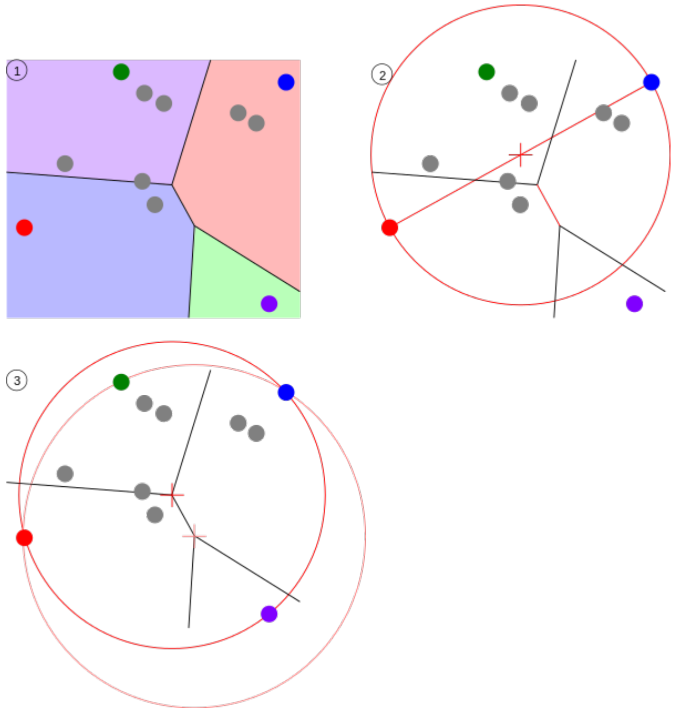

FIGURE 2 – Diagramme de Voronoï des points les plus éloignés [6]

Algorithm 2: Shamos et Hoey - Calculer le cercle minimum d'un espace de points

---

Data: Ensemble P de points
Result: Cercle minimum C contenant P
1. Détermination de l'enveloppe convexe (points de couleur) et du diagramme de Voronoï des points les plus éloignés ;
2. Détermination de la paire la plus éloignée et essai du cercle dont elle est le diamètre ;
3. Détermination des triplets les plus éloignés, et essai des cercles circonscrits correspondant ;

L'algorithme de Shamos et Hoey qui utilise le diagramme de Voronoï a une complexité temporelle de $O(n * log(n))$ sur des entrées constituées de n points.

Cependant, sa mise en œuvre peut être complexe, notamment la construction du diagramme de Voronoï.

Une amélioration possible de cet algorithme serait d'utiliser des structures de données plus efficaces pour trouver les points les plus éloignés, comme les arbres k-d.

1.2.3 L'algorithme de Chrystal

L'algorithme de Chrystal est une méthode pour calculer le cercle minimum d'un ensemble de points en utilisant l'enveloppe convexe de ces points. [6]

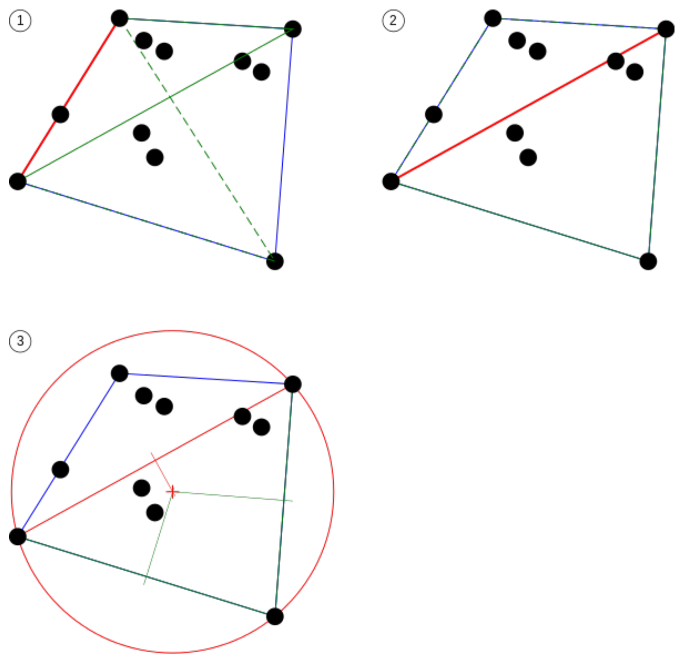

FIGURE 3 – L'enveloppe convexe [6]

Algorithm 3: Chrystal - Calculer le cercle minimum d'un espace de points

---

Data: Ensemble P de points
Result: Cercle minimum C contenant P
1. Calculer l'enveloppe convexe H_i avec $1 \leq i \leq m$;
2. Prendre arbitrairement un coté $[H_j H_k]$ de deux sommets adjacents $\in H_i$ et choisir un 3^e sommet H_p tq l'angle opposé formé est minimal;
3. Si l'angle formé est droit ou obtus alors renvoyer le cercle de diamètre $[H_j H_k]$;
4. Si les trois angles sont aïgues alors renvoyer cercle circonscrit du triangle ;
5. Si $[H_j H_k H_p]$ est obtus en H_j alors répéter 3 en considérant $[H_k H_p]$, si le nouvel angle est obtus en H_k alors répéter 3 et considérer $[H_p H_j]$;

L'algorithme de Chrystal a une complexité en temps de $O(m^2)$ dans le pire des cas, où m est le nombre de points de l'enveloppe convexe.

Cet algorithme n'est pas optimal pour les ensembles de points qui contiennent des points très éloignés les uns des autres. Dans ces cas, le cercle minimum peut être beaucoup plus grand que la zone contenant tous les points, ce qui signifie que l'algorithme doit effectuer un grand nombre de calculs pour trouver le cercle minimum exact.

Une amélioration possible de cet algorithme serait de réduire la complexité en temps en utilisant une structure de données plus efficace pour stocker l'enveloppe convexe comme une liste doublement chaînée ou un arbre binaire de recherche. Cela réduirait la complexité en temps de la recherche des points dans l'enveloppe convexe de O(n) à O(log n).

1.2.4 Conclusion

On peut trouver plusieurs méthodes différentes dans la litterature pour calculer le cercle minimum d'un ensemble de points.

Nous avons constaté que l'algorithme de Ritter est le plus efficace en termes de temps d'exécution, suivi de près par l'algorithme de Shamos et Hoey. L'algorithme de Chrystal est le moins efficace des trois en termes de temps d'exécution, bien qu'il reste un outil utile pour résoudre des cas particuliers de cercle minimum.

Chaque méthode a ses avantages et ses inconvénients, en termes de complexité en temps, de précision et de performance pour différents types de données. Le choix de la méthode dépendra des besoins spécifiques de l'application et des caractéristiques de l'ensemble de points à traiter.

En plus des algorithmes déjà évoqués ci-dessus, nous présenterons également dans la prochaine partie du rapport une analyse de l'approche naïve et de l'algorithme de Welzl, qui est l'un des plus efficaces.

2 Algorithme Naïf

"Algorithme de contrôle" est une façon naïve de calculer le cercle minimum d'un ensemble de points. Cette méthode consiste à considérer toutes les combinaisons possibles de trois points parmi un ensemble de points donné et à calculer le cercle circonscrit de chaque combinaison. Ensuite, le cercle avec le plus petit rayon est considéré comme le cercle minimum.

Algorithm 4: Algorithme Naïf - Calculer le cercle minimum d'un espace de points

Data: Ensemble P de points
Result: Cercle minimum C contenant P
for *tout p dans Points* **do**
  for *tout q dans Points* **do**
    $C \leftarrow$ cercle de centre $\frac{p+q}{2}$ de diamètre $|pq|$;
    if *C couvre tous les points de Points* **then**
      retourner C ;
    end
  end
end
$resultat \leftarrow$ cercle de rayon infini;
for *tout p dans Points* **do**
  for *tout q dans Points* **do**
    for *tout r dans Points* **do**
      $C \leftarrow$ cercle circonscrit de p, q et r;
      if *C couvre tous les points de Points et C plus petit que resultat* **then**
        $resultat \leftarrow C$;
      end
    end
  end
end
retourner *resultat* ;

Bien que l'algorithme de contrôle soit facile à comprendre et à mettre en œuvre, il est très inefficace pour les grands ensembles de points. En effet, le nombre de combinaisons possibles de trois points augmente rapidement avec la taille de l'ensemble de points, rendant le temps d'exécution inacceptable pour des ensembles de points contenant un grand nombre de points.

En effet, pour calculer le cercle circonscrit, on doit tester pour tout triplet possible s'il couvre l'intégralité des points de P, et donc la complexité est de $O(n^3 * n) = O(n^4)$.

Cet algorithme est utile pour comprendre le concept de cercle minimum et pour des ensembles de points de petite taille, mais il n'est pas adapté pour des ensembles de points de grande taille et peut produire des résultats peu précis.

Ainsi, il est recommandé d'utiliser une méthode plus sophistiquée, telle que l'algorithme de Welzl, pour calculer le cercle minimum de manière efficace et précise.

3 Algorithme de Welzl

L'algorithme de Welzl [1] utilise une approche récursive pour calculer le cercle minimum en utilisant les points de l'ensemble. Pour faire cela, l'algorithme Welzl utilise une structure de données appelée "minidisk". Il s'agit d'un cercle qui couvre un sous-ensemble des points de l'ensemble original, et qui est construit de manière récursive à partir des points restants.

Il considère deux ensembles : P - Ensemble des points de départ ; R - Ensemble des points sur le cercle minimum. Au départ, P contient tout les points et R=0.

Algorithm 5: B_MINIDISK(P,R)

```
Data: P, R
Result: boundary_mindisk(P, R)
if P = ∅ OU | R |= 3 then
    D ← boundary_mindisk(P, R);
else
    choisir arbitrairement p ∈ P ;
    D ← B_MINIDISK(P - {p}, R);
    if D défini ET p ∉ D then
        D ← B_MINIDISK(P - {p}, R ∪ {p});
    end
end
return D ;
```

Ici, *boundary_mindisk*(P, R) est une fonction auxiliaire utilisée dans l'algorithme de Welzl pour calculer le cercle minimum d'un ensemble de points. Cette fonction calcule le cercle minimum d'un ensemble de points qui forment une frontière.

- Si la taille de "boundary" est égale à 0, cela signifie qu'il n'y a pas de points dans la frontière, et donc le cercle minimum est simplement un cercle de rayon 0.
- Si la taille de "boundary" est égale à 1, donc le cercle minimum est également un cercle de rayon 0 centré sur ce point unique.
- Si la taille de "boundary" est égale à 2, donc le cercle minimum est le cercle qui passe par ces deux points.
- Enfin, si la taille de "boundary" est égale à 3, donc le cercle minimum est le cercle circonscrit.

L'algorithme de Welzl est connu pour sa précision élevée et sa complexité en temps optimale de $O(n)$, où n est le nombre de points dans l'ensemble.

En termes de performances, l'algorithme de Welzl est très rapide et précis. En outre, il peut gérer des ensembles de points avec des configurations de points complexes, tels que des ensembles contenant des points de ligne ou des points de répétition. Il s'agit d'un algorithme bien équilibré qui offre des performances élevées pour une large gamme d'applications.

Nous passons maintenant à la section test.

4 Expérimentation

4.1 Tests

Nous avons décidé de rester dans le langage de programmation **Java**, car nos TME dans l'UE ont été réalisés dans ce langage. Ainsi, nous utiliserons les connaissances acquises et profitrons de l'interface graphique "Diameter Racer" utilisée dans ce cours.

Nous avons confronté l'algorithme naïf et l'algorithme de Welzl à la même base de test choisie est celle de **Varoumas Benchmark**, donc 1664 tests ont été effectués.

On précise que les tests ont été fait sur une machine dotée d'un processeur AMD Ryzen 5 3550H 2.10 GHz et 8GB de mémoire vive.

La figure 4 montre le diagramme de fréquence d'exécution basé sur les résultats de tous les 1664 tests pour les deux algorithmes. Ici vous pouvez voir que pour tous les tests, l'algorithme de Welzl montre un temps d'exécution extrêmement rapide, alors que l'algorithme naïf est en moyenne au niveau de 80ms.

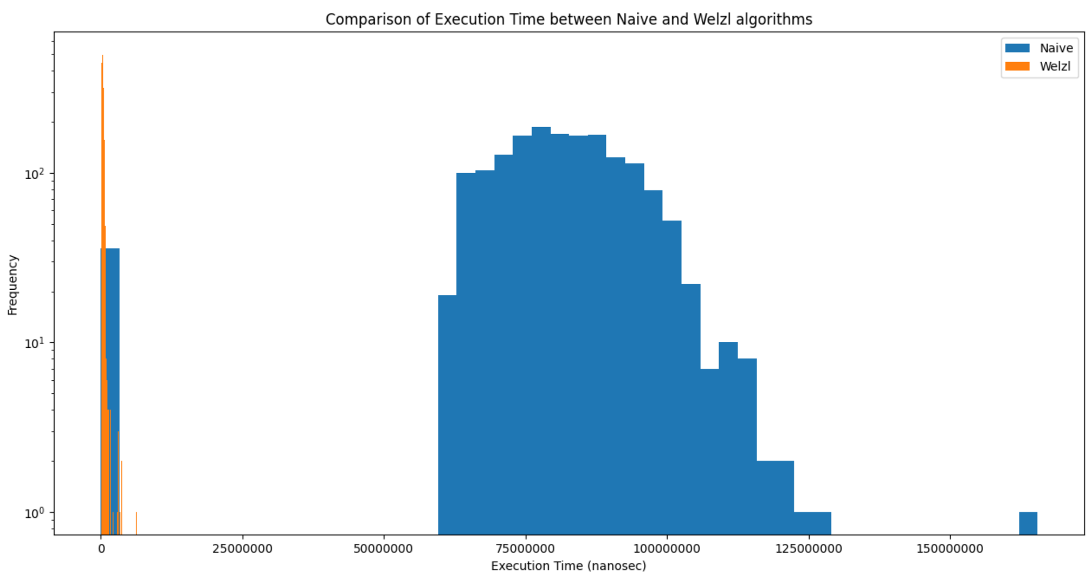

FIGURE 4 – Fréquance de temps d'exécution de l'algorithme naïf et de l'algorithme de Welzl

La figure 5 montre un graphique du temps d'exécution moyen basé sur les résultats des 1664 tests pour les deux algorithmes.

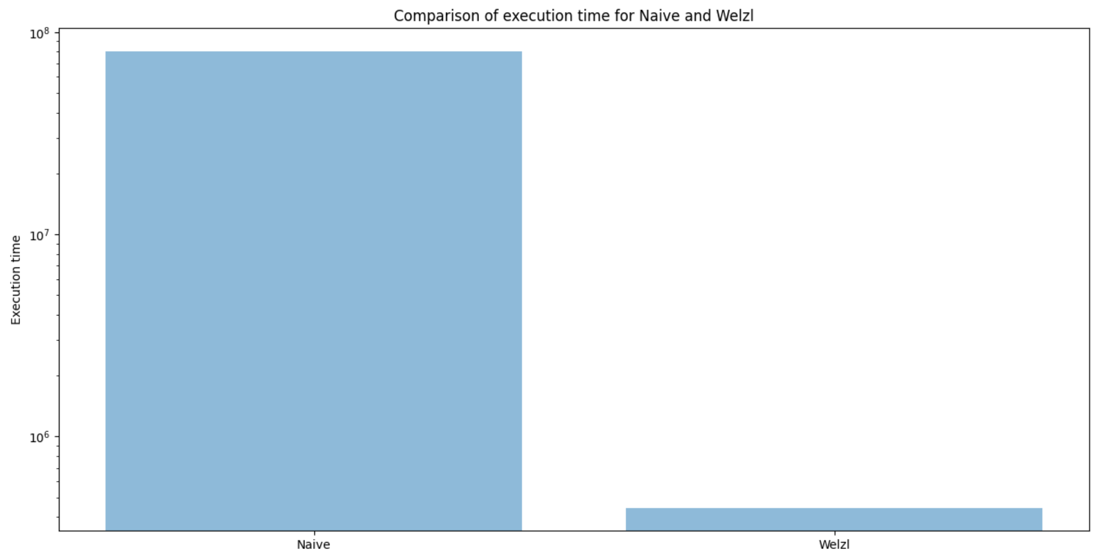

FIGURE 5 – Temps d'exécution moyen

La figure 6 montre un test pour lequel nous avons combiné plusieurs fichiers de test pour obtenir un nombre de points différent, soit 256, 512, ..., 2560. Ici vous pouvez voir que pour l'algorithme naïf, le temps d'exécution augmente avec l'augmentation du nombre de points, alors que pour l'algorithme de Welzl cet effet est presque négligeable.

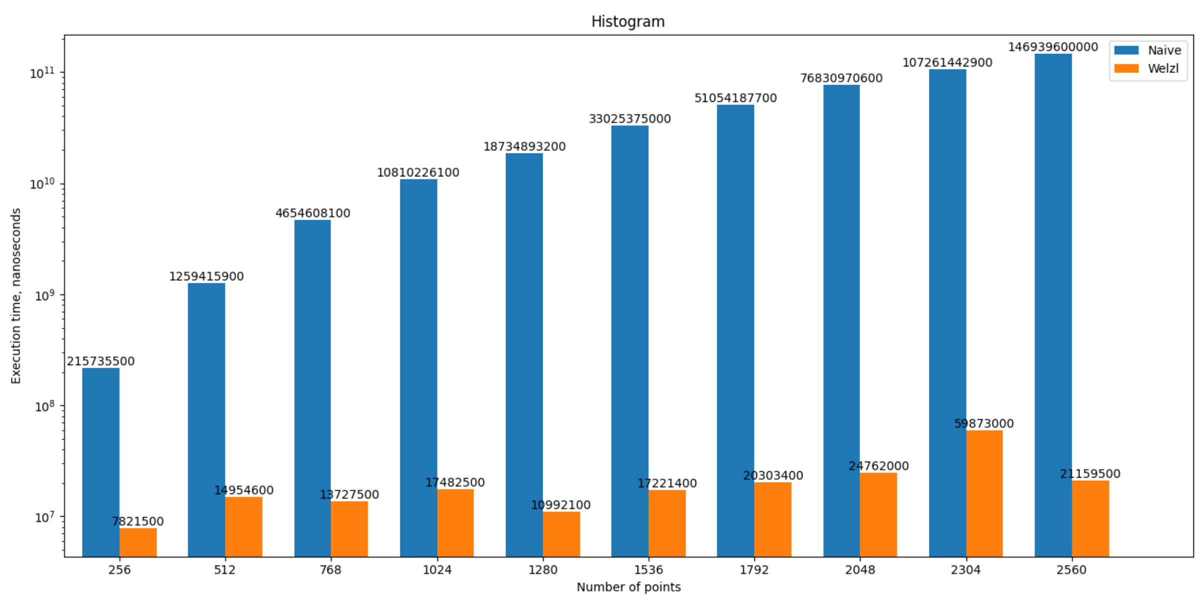

FIGURE 6 – Construction du cercle minimum

La figure 7 montre la valeur moyenne (en bleu) du rayon calculée pour tous les tests pour l'algorithme naïf et pour l'algorithme de Welzl, ainsi que la valeur de l'écart type (barre noire). On peut voir que l'écart type pour l'approche naïve est sensiblement plus grand, ceci s'explique par le fait que l'algorithme de contrôle n'est pas très précis et peut produire des cercles minimums qui ne sont pas optimaux. En effet, considérer uniquement les combinaisons de trois points peut ne pas prendre en compte des configurations de points qui nécessitent plus de trois points pour définir un cercle minimum précis.

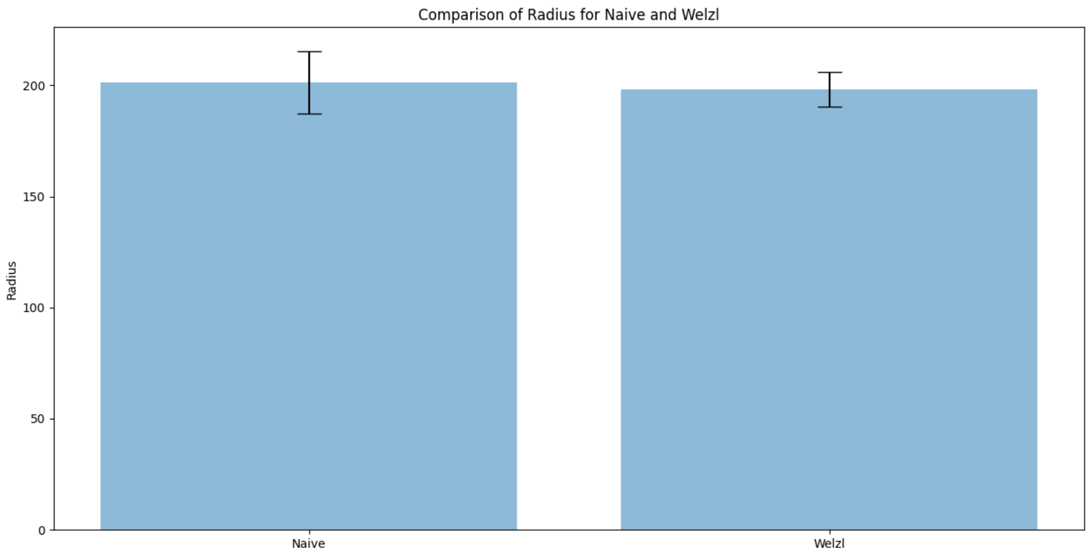

Figure 7 – Construction du cercle minimum

4.2 Conclusion

En conclusion, nous avons confronté l'algorithme de Welzl pour trouver un cercle minimum avec l'algorithme naïf de contrôle. Nous avons constaté que l'algorithme de Welzl est beaucoup plus efficace que l'algorithme naïf en termes de temps d'exécution. En effet, l'algorithme de Welzl a une complexité temporelle de $O(n)$, tandis que l'algorithme naïf a une complexité temporelle de $O(n^4)$. Nous avons également noté que l'algorithme de Welzl est plus robuste en présence de données bruyantes et de cas extrêmes.

En termes de perspectives, le problème de cercle minimum reste un sujet de recherche actif dans le domaine de la géométrie algorithmique. De nombreux travaux récents ont cherché à améliorer l'efficacité et la précision des algorithmes existants, ainsi qu'à explorer de nouvelles approches pour résoudre ce problème. Par exemple, certains chercheurs ont proposé des approches basées sur l'apprentissage automatique pour trouver des cercles minimums dans des ensembles de données complexes. [2] [5] [7]

Enfin, nous pouvons conclure que l'algorithme de Welzl est un outil puissant pour résoudre le problème de cercle minimum. Son efficacité et sa robustesse en font un choix naturel pour les applications pratiques. Cependant, il est important de continuer à explorer de nouvelles approches et techniques pour améliorer la résolution de ce problème et répondre aux défis posés par des cas complexes et bruyants.

Références

[1] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). https://www.stsci.edu/~RAB/Backup%20Oct%2022%202011/f_3_CalculationForWFIRSTML/Bob1.pdf, 1991.

[2] Ercan, M. F., Qiankun, A. L., Sakai, S. S., and Miyazaki, T. Circle detection in images : A deep learning approach. In *Global Oceans 2020 : Singapore – U.S. Gulf Coast* (2020), pp. 1–5.

[3] Michael Ian Shamos, Dan Hoey. Closest-point problems. http://cs470.cs.ua.edu/spring2017/Shamos%20Hoey%201975%20Closest%20Point.pdf, 1975.

[4] Ritter, J. An efficient bounding sphere. https://www.researchgate.net/publication/242453691_An_Efficient_Bounding_Sphere, 12 1990.

[5] Song, H., Saitoh, T., and Taniguchi, T. A machine learning approach to minimum enclosing ball problems. *Journal of Intelligent Manufacturing 32*, 1 (2021), 121–129.

[6] Wikipedia. Problème du cercle minimum. https://fr.wikipedia.org/wiki/Probl%C3%A8me_du_cercle_minimum, 2023.

[7] Zhang, X., Zhao, Y., and Lu, W. A deep learning approach to the minimum enclosing ball problem. *IEEE Transactions on Neural Networks and Learning Systems 30*, 1 (2019), 191–202.