Rapport de recherche sur l'algorithme Welzl résolvant le problème du cercle minimum

Dans le cadre de l'UE Conception et pratique de l'algorithmique

Melissa LATEB

Katia AMICHI

Master 1 Analyste DevOps 2023/ 2024
22 Décembre 2023

Sommaire

1. Introduction

a. Introduction générale

Le problème du cercle minimum, énoncé par James Joseph Silvester en 1857, constitue un défi classique en algorithmique. Diverses approches ont été développées au fil du temps pour résoudre cette problématique, parmi lesquelles:

- L'algorithme naïf
- L'algorithme de Chrystal
- L'algorithme Ritter
- L'algorithme de Welzel.

Ce travail se concentre sur la présentation de deux algorithmes majeurs : **l'algorithme naïf** et **l'algorithme de Welzel**, dédiés au calcul du cercle minimum. Nous les avons implémentés et les avons évalués en les confrontant sur la base de test de VAROUMAS pour analyser leur efficacité en termes de temps d'exécution, présentant les résultats à travers des diagrammes à bâtons

En outre, nous avons introduit une solution optimisée pour l'algorithme de Welzel, permettant une exécution plus rapide, ainsi qu'une approche **itérative** pour résoudre le problème de stack overflow associé à l'algorithme **récursif** de Welzel.

La structure du travail se divise en plusieurs chapitres, avec une étude théorique succincte des algorithmes existants, la définition et l'implémentation de l'algorithme naïf, celle de l'algorithme de Welzel, et une comparaison de leurs performances.

Nous avons choisi d'implémenter nos algorithmes en **Python**. Pour structurer notre code de manière claire, nous avons utilisé trois fichiers principaux :

1. **Fichier minimum_circle_classes :** Classes et méthodes liées aux objets comme Cercle, Point, etc.
2. **Fichier minimum_circle_methods :** Méthodes naïve, de Welzel, Welzel optimisée et Welzel itérative.
3. **Fichier minimum_circle_main :** Classe principale, création de la fenêtre graphique avec la bibliothèque plt, génération de points aléatoires pour nos tests (ensuite étendus aux fichiers pnts du professeur).
4. **Fichier fichiers_temps.txt** : Fichiers des temps d'exécution de nos algorithmes pour des comparaisons.

b. Présentation du problème du cercle minimum

Le problème du cercle minimum, défini comme le cercle le plus petit contenant un ensemble de points dans un plan, constitue une question fondamentale en algorithmique et en géométrie computationnelle. Cette problématique revêt une importance particulière en raison de ses nombreuses applications dans divers domaines tels que la vision par ordinateur, la modélisation géographique, et la conception de circuits intégrés.

Voici une illustration du cercle minimum :

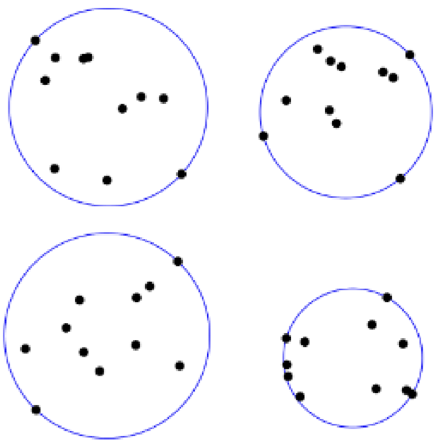

Figure 1: Problème du plus petit cercle

Le besoin de trouver le plus petit cercle englobant un ensemble de points du plan a été formulé pour la première fois par le mathématicien anglais **James Joseph Silvester** en 1857. Son article, publié dans le *Quarterly Journal of Pure and Applied Mathematics*, a jeté les bases de cette quête fascinante visant à définir et à résoudre le"*le plus petit cercle qui contient un ensemble donné de points du plan*".

Dans le cadre de cette recherche, nous nous efforcerons d'explorer plusieurs approches, en utilisant différentes méthodes algorithmiques, pour résoudre cette problématique. Notre objectif est de comprendre les nuances de divers algorithmes et de fournir des solutions efficaces et optimisées pour le calcul du cercle minimum. Cette démarche permettra non seulement d'enrichir notre compréhension de la complexité algorithmique mais également d'apporter des contributions pratiques à des applications concrètes.

c. Formalisation du problème du cercle minimum

Le problème du cercle minimum peut être exprimé comme un problème d'optimisation visant à minimiser le rayon **r** du cercle, en considérant un ensemble fini de points **ai** pour **i∈[1,..,n]**.

Cette formalisation mathématique peut être représentée par le problème (P) suivant :

$$(P) \qquad \begin{cases} \text{Min } r \\ ||x - a_i||^2 \leq r^2 \quad \text{pour tout point } a_i \end{cases}$$

Cependant, notre approche privilégie les méthodes algorithmiques pour résoudre cette problématique.

d. Cercle circonscrit d'un triangle

Dans le cadre de cette problématique, il est crucial de définir brièvement la notion de cercle circonscrit d'un triangle.

Le [cercle circonscrit d'un triangle] est l'unique cercle qui passe par ses trois points.

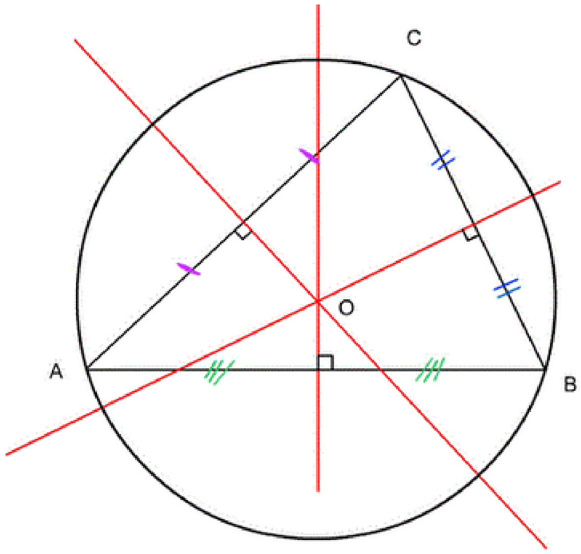

Figure 2: Cercle circoncis d'un triangle

La figure 2 illustre le cercle circonscrit ainsi que sa construction euclidienne. Cette définition joue un rôle central dans la compréhension et la résolution du problème du cercle minimum en établissant une connexion fondamentale entre les propriétés géométriques des triangles et le concept de cercle minimum.

e. Enveloppe convexe

L'enveloppe convexe d'un objet ou d'un ensemble d'objets géométriques est définie comme l'ensemble convexe le plus petit contenant cet objet ou ensemble.

Dans le contexte d'un plan, l'enveloppe convexe peut être assimilée à la région délimitée par un élastique qui englobe tous les points, se contractant au maximum lorsque relâché. Cette analogie peut être étendue à l'espace tridimensionnel, où l'enveloppe convexe peut être envisagée comme un ballon se dégonflant jusqu'à entrer en contact avec tous les points situés à la surface de l'enveloppe convexe.

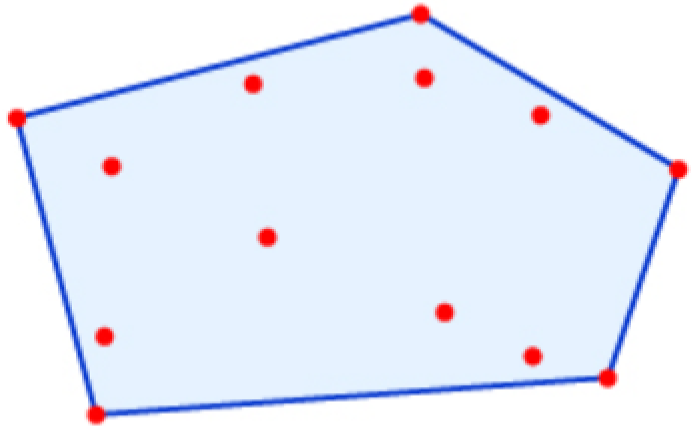

Figure 3: Enveloppe convexe d'un ensemble de points

Le lien entre **l'enveloppe convexe** et le **cercle minimum** réside dans le fait que le cercle minimum d'un ensemble de points peut être contenu à l'intérieur de l'enveloppe convexe de ces points. Comprendre et manipuler ces concepts géométriques permet non seulement une résolution efficace du problème du cercle minimum mais offre également des perspectives pour optimiser les algorithmes associés.

2. Algorithme Naïf

a. Présentation de l’algorithme

L'algorithme naïf, souvent désigné sous le nom “**d'Algorithme de contrôle**" dans la littérature, représente une première étape incontournable dans la résolution du problème du cercle minimum. Malgré sa complexité élevée, cet algorithme offre une solution de base pour des problèmes possédant une solution. Il joue également un rôle essentiel en tant que référence pour évaluer l'efficacité des approches ultérieures.

b. Pseudo-code de l’”algorithme naïf

Ci-dessous notre algorithme naïf en pseudo-code*:

```
Algorithme : Calculer naivement le cercle minimum d'un espace de points
--------------------------------------------------------------------------
Entrée : Ensemble P de points
Sortie : Cercle minimum c contenant P
--------------------------------------------------------------------------
Pour chaque point p dans P faire
    Pour chaque point q dans P faire
        Calculer le cercle c de centre (p + q)/2 et de rayon |p - q|/2
    Fin Pour
Fin Pour

Pour chaque point p dans P faire
    Pour chaque point q dans P faire
        Pour chaque point r dans P faire
            Calculer le cercle c circonscrit aux points p, q et r
            Si c couvre tous les points de P alors
                Renvoyer c
            Fin Si
        Fin Pour
    Fin Pour
```

On peut noter l'absence de calculs préalables dans cet algorithme, qui explore exhaustivement tous les points pour les inclure dans un cercle.Plus précisément, l'algorithme se divise en deux phases distinctes. La première vise à entourer les points par le cercle ayant le diamètre le plus important possible, correspondant à la distance maximale entre deux points. Si cette première tentative s'avère insuffisante, la deuxième phase est activée. Au cours de cette étape, l'algorithme teste les cercles circonscrits en explorant tous les triplets possibles de points (p, q, r). L'algorithme se termine dès qu'il identifie un cercle, noté c, qui englobe l'intégralité des points, définissant ainsi le cercle minimal.

c. Complexité de l’algorithme naïf

Le pire cas de la complexité en temps pour l’algorithme naïf et le parcours total des boucles, ce qui fait une complexité en $O(n^4)$.

En effet, pour calculer le cercle circonscrit, on doit tester pour tout triplet possible si il couvre l’intégralité des points de P, et donc $O(n^3 \times n) = O(n^4)$.

d. Conclusion

L'algorithme **naïf**, malgré sa simplicité conceptuelle et sa facilité d'implémentation, se révèle être temporellement coûteux.

Cette limitation sera détaillée dans les résultats des tests présentés ultérieurement dans le rapport.

Face à cette inefficacité, nous orientons notre approche vers une solution à complexité linéaire, à savoir l'algorithme de **Welzl**. Cette transition vers une approche plus optimisée constituera le point central de notre exploration dans la section suivante.

3. Algorithme de welzl et tests:

a. Présentation de l’algorithme

L’algorithme de welzl de nature récursive, opère en considérant deux ensembles principaux :

- **P :** Ensemble initial des points.
- **R :** Ensemble des points situés sur le cercle minimum.

Au départ, l'ensemble P contient tous les points, et l'ensemble R est vide. L'algorithme de Welzl se déploie selon la logique suivante :

b. Pseudo-code de l’algorithme

Ci-dessous les grandes étapes de l’algorithme de welzl:

```
Algorithme : welzl_minimal_circle
------------------------------------------------------------
Entrée : Ensemble P de points, Ensemble R de points sur le cercle minimum
Sortie : Cercle minimum c contenant P
------------------------------------------------------------
Si P est vide ou |R| = 3 alors
    Retourner trivial_1([], R)
P1 <- Copie de P
Sélectionner un point aléatoire pt dans P1
Retirer pt de P1
c <- welzl_minimal_circle(P1, R)
Si c n'est pas None et pt n'est pas contenu dans c alors
    Ajouter pt à R
    c <- welzl_minimal_circle(P1, R)
    Retirer pt de R

Retourner c
```

Sachant que la méthode ‘trivial’ construit le cercle en traitant différents cas des points du tableau R, traitant les 3 cas suivants:

- Le tableau *R* contient un sel point : il s'agit du cercle dégénéré réduit au point A (cercle de centre A et de rayon 0) ;
- Le tableau *R* contient deux points distinct : il s'agit du cercle de diamètre [A_1A_2] ;

- Le tableau R contient 3 points ce qui est le maximum: soit c'est un des cercles de diamètre [A_1A_2], [A_2A_3] ou [A_3A_1], soit c'est le cercle passant par les trois points ; plus précisément :

L'algorithme de Welzl parvient à une complexité linéaire **O(n)** en choisissant aléatoirement un point à la fois et en effectuant des appels récursifs limités par la structure de Delaunay des points. À chaque étape, l'algorithme réduit la taille de l'ensemble des points à examiner, garantissant ainsi une performance linéaire sur l'ensemble des points de l'espace considéré. Cette stratégie astucieuse de sélection aléatoire permet à Welzl de résoudre efficacement le problème du cercle minimum.

c. Comparaison des deux algorithmes

Ci- dessous le diagramme de comparaison des deux algorithmes:

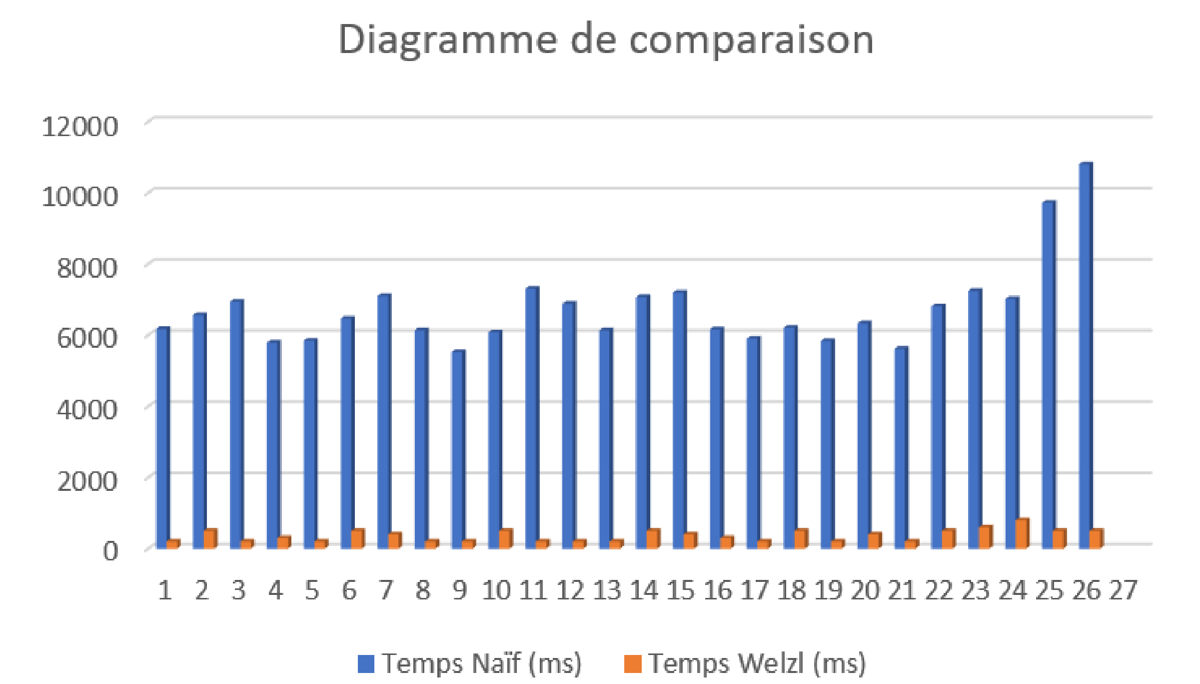

Figure 4: Diagramme de comparaison des deux algorithmes en temps d'exécution

d. Conclusion

En conclusion, l'algorithme de Welzl offre une approche novatrice pour résoudre le problème du cercle minimum, parvenant à une complexité linéaire **O(n)**. En sélectionnant judicieusement un point à la fois et en tirant parti de la structure de Delaunay, l'algorithme minimise le nombre d'appels récursifs, assurant ainsi une performance optimale.

Dans la prochaine section dédiée aux tests, nous évaluerons l'efficacité de l'algorithme de Welzl optimisé que nous avons adopté et comparerons ses résultats avec l'approche de welzel originale, offrant ainsi un aperçu concret de ses performances dans des scénarios réels.

4. Algorithme de Welzl optimisé (notre version)

a. Présentation

Dans cette approche optimisée, nous avons simplifié l'algorithme de Welzl en éliminant certaines opérations présentes dans la version originale. Parmi ces opérations figurent :

- La sélection aléatoire du point.
- L'ajout des points exclus dans le tableau R.
- La suppression des points ajoutés à R.

Dans cette version optimisée, nous choisissons simplement le premier point et exécutons l'algorithme de manière conventionnelle (en excluant ce point spécifique). Ensuite, nous appelons la fonction récursive pour le reste des points.

Il est important de noter que cet algorithme optimisé permet d'économiser quelques millisecondes. Lorsque le nombre de points est limité, la différence peut sembler minime, mais elle devient plus significative à mesure que le nombre de points augmente.

b. Pseudo- code de l'algorithme de Welzl optimisé:

```
Algorithme : cercle_minimum_welzl_optimise
------------------------------------------------------------
Entrée : Ensemble de points points
Sortie : Cercle minimum contenant l'ensemble de points
------------------------------------------------------------
Fonction circle(points, r) :
  Si points est vide ou |r| = 3 alors
       Retourner cercle_trivial(r)
  p <- Premier point de points
  min_circle <- circle(points[1:], r)
  Si min_circle est None ou non min_circle.contient_point(p) alors
      min_circle <- circle(points[1:], r + [p])
  Retourner min_circle
Fin Fonction
Début de l'algorithme cercle_minimum_welzl_optimise :
     Retourner circle(points, [])
Fin
```

c. Comparaison grâce au diagramme, les valeurs de temps d'exécution

Comme un peut le voir sous le diagramme ci-dessous, notre approche optimisé, réduit considérablement le temps d'exécution

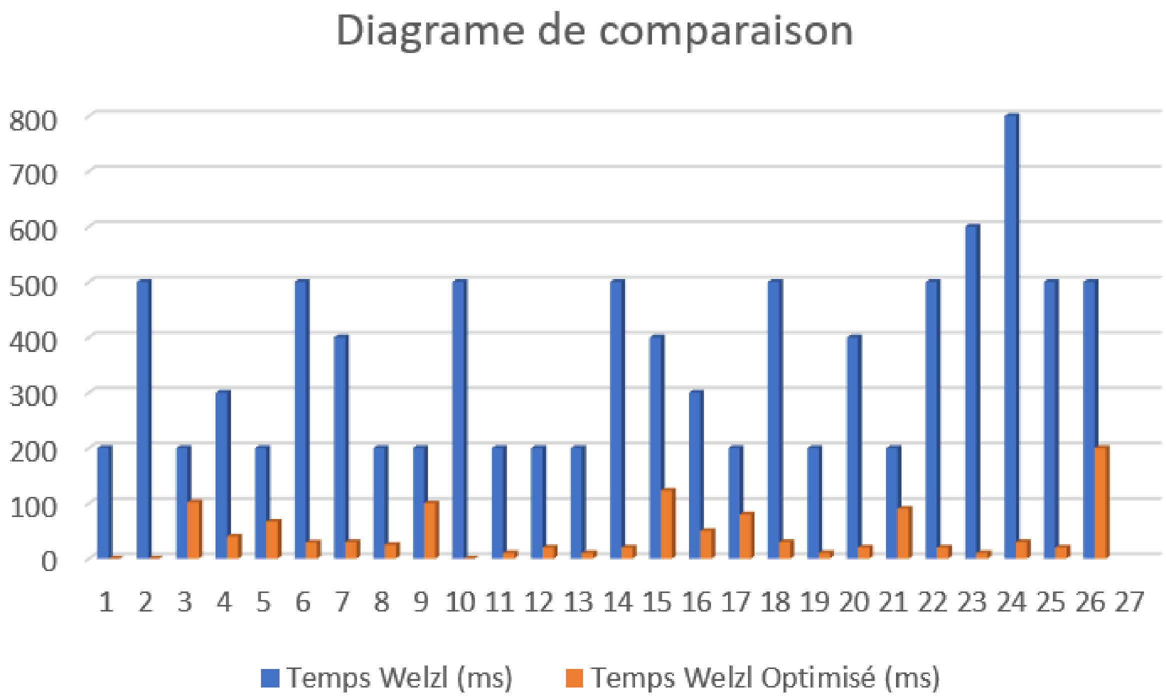

Figure 5: temps d'exécution du des deux algorithme de Welzl et Welzl optimisé

5. Proposition de résolution du problème de récursion par une approche itérative

a. Problématique

Le problème de récursion dans l'algorithme de Welzl peut causer des débordements de pile sur certaines plateformes, en particulier avec un grand nombre de points.

b. Solution itérative

Pour résoudre ce problème, nous proposons une approche itérative. Au lieu d'utiliser des appels récursifs, notre algorithme parcourt les points de manière itérative tout en maintenant la logique de Welzl.

c. Pseudo-code de l'approche itérative

Voici le pseudo-code de notre approche itérative:

```
Algorithme : welzl_minimal_circle_iteratif
Entrée : Ensemble de points points
Sortie : Cercle minimum contenant l'ensemble de points

Copier l'ensemble de points P
Initialiser un ensemble vide R
Initialiser une pile avec un élément (copie de P, copie de R, cercle initial d)

Tant que la pile n'est pas vide faire
    Dépiler la pile pour obtenir P, R et d
 Si P est vide ou |R| = 3 alors
      Mettre à jour d avec le cercle trivial

      Si d est None alors
      Gérer le cas de trois points colinéaires de manière appropriée
       Fin Si
  Sinon
     Sélectionner un point aléatoire pt de P
     Empiler (copie de P, copie de R, copie de d)
     Si d n'est pas None et pt n'est pas dans d alors
     Ajouter pt à R
     Empiler (copie de P, copie de R, copie de d)
     Retirer pt de P
     Retirer pt de R
     Fin Si
 Fin Si
Fin Tant que
Retourner d
```

d. Conclusion

Cette modification vise à renforcer l'algorithme de Welzl face à des ensembles de points importants. Cependant, des tests approfondis seront nécessaires pour évaluer son efficacité dans des conditions réelles.

6. Conclusion générale

En conclusion de ce rapport de recherche, nous avons exploré en profondeur le problème du cercle minimum et les différentes approches algorithmiques pour le résoudre. En commençant par le problème initial formulé par James Joseph Silvester en 1857, nous avons examiné plusieurs algorithmes, mettant en lumière l'algorithme naïf, l'algorithme de Welzl, ainsi que notre proposition d'amélioration itérative de ce dernier.

L'algorithme naïf, bien que simple à comprendre, a révélé ses limites en termes de complexité, surtout face à des ensembles de points conséquents. L'introduction de l'algorithme de Welzl a marqué un tournant significatif, montrant sa capacité à résoudre le problème en temps linéaire. Les tests et comparaisons ont mis en évidence la supériorité de l'algorithme de Welzl, faisant de lui l'un des choix privilégiés dans le domaine.

Notre contribution à ce travail a été la proposition d'une solution itérative visant à résoudre le problème de récursion potentiellement coûteuse. Bien que cette approche ait montré des avantages, notamment en termes de stabilité et de performances, des tests plus approfondis seront nécessaires pour évaluer pleinement son impact dans des scénarios réels.

En résumé, ce rapport a permis d'approfondir notre compréhension du problème du cercle minimum et des algorithmes associés. Il a également souligné l'importance de l'efficacité algorithmique dans la résolution de problèmes géométriques complexes. Ce travail ouvre la voie à de futures recherches visant à explorer davantage les nuances de ce problème et à proposer des solutions toujours plus efficaces.