Conception & Pratique de l'Algorithmique

22/12/2023

Projet Individuel: Master 1

Rédaction d’un rapport de recherche sur l’algorithme Welzl résolvant le problème du cercle minimum.

Auteur : SAMOURA Amadou

ANALYSTE DEVELOPPEUR CHEZ TRANSACTIS & SOCIETE GENERALE

CFA INSTA
Institut National Supérieur des Technologies Avancées

TRANSACTIS

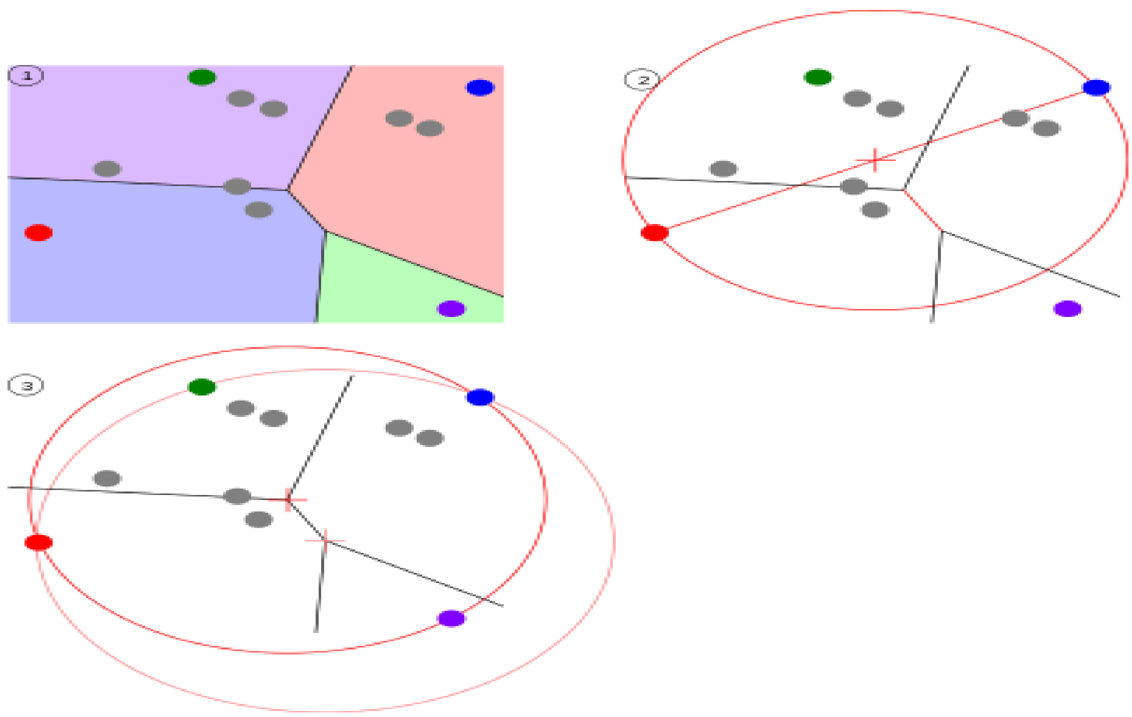
Sommaire :

1. Introduction

L'algorithme de Welzl, également connu sous le nom de "boule minimale englobante" (BME), est une technique algorithmique utilisée dans le domaine de la géométrie informatique. Développé par Emo Welzl, cet algorithme vise à trouver la plus petite sphère englobante qui contient un ensemble donné de points dans l'espace. Cette sphère est également appelée "boule minimale englobante" et est souvent utilisée dans des applications telles que la résolution de problèmes de géométrie ou la simplification de données pour des analyses ultérieures.

Historique de l'Algorithme

L'algorithme de Welzl a été introduit pour la première fois en 1991 par Emo Welzl dans un article intitulé "Smallest Enclosing Disks (balls and ellipsoids)". Depuis lors, il a suscité un grand intérêt dans la communauté informatique en raison de son efficacité et de son élégance conceptuelle. L'algorithme offre une solution efficace au problème de la boule minimale englobante en utilisant une approche probabiliste.

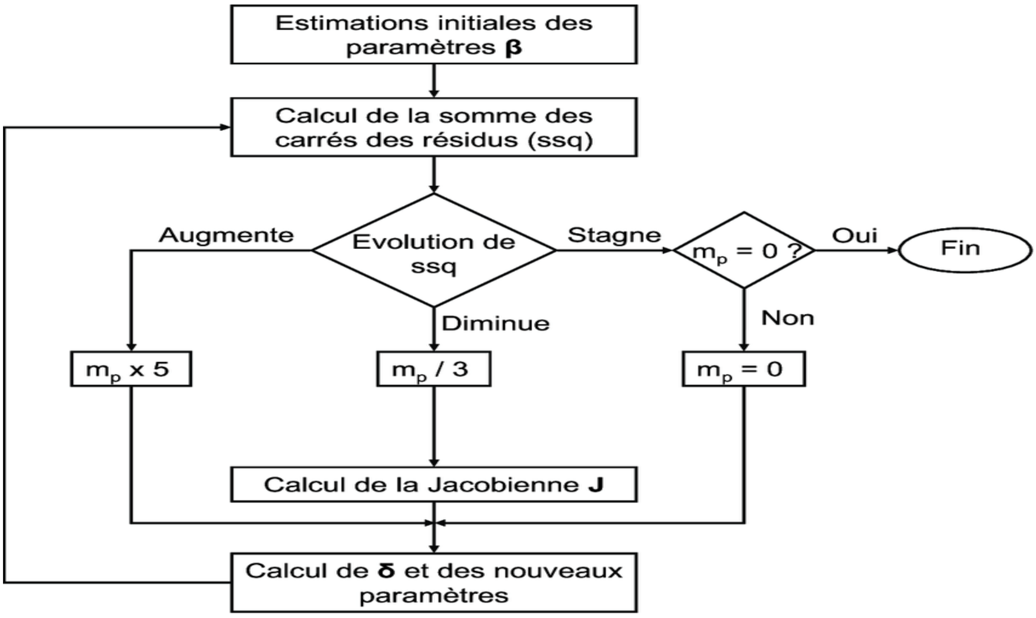


Fonctionnement de l'Algorithme :

L'idée principale derrière l'algorithme de Welzl est de construire récursivement la boule minimale englobante en utilisant une méthode d'échantillonnage aléatoire. Voici une brève description du processus :

- ✓ Si l'ensemble de points est vide, la boule minimale englobante est définie comme étant une sphère de rayon nul centrée à l'origine

- ✓ Si l'ensemble de points contient un seul point, la boule minimale englobante est une sphère centrée sur ce point avec un rayon nul.
- ✓
- ✓ Si l'ensemble de points contient deux points, la boule minimale englobante est une sphère dont le diamètre est le segment de droite entre ces deux points.
- ✓
- ✓ Pour des ensembles de points de taille supérieure à deux, l'algorithme utilise une approche probabiliste en sélectionnant un point au hasard et en construisant récursivement la boule minimale englobante pour l'ensemble de points restants. Ce processus est répété plusieurs fois pour améliorer la précision du résultat.



Avantages et Limitations

L'algorithme de Welzl présente plusieurs avantages, notamment sa simplicité conceptuelle, sa rapidité d'exécution dans la pratique, et sa garantie de trouver la solution optimale avec une probabilité élevée. Cependant, il est important de noter que son efficacité dépend de la nature des ensembles de points en entrée.

Une limitation majeure de cet algorithme réside dans sa dépendance à la nature aléatoire de l'échantillonnage, ce qui peut entraîner des performances variables dans certaines situations. De plus, bien que l'algorithme soit probabiliste, il a été démontré empiriquement qu'il offre des résultats fiables dans la plupart des cas.

Applications Pratiques

L'algorithme de Welzl trouve des applications pratiques dans divers domaines tels que la modélisation 3D, la reconnaissance d'objets, et la robotique. Sa capacité à calculer rapidement une enveloppe convexe autour d'un ensemble de points en fait un outil précieux pour résoudre des problèmes de géométrie complexe.

Conclusion

En résumé, l'algorithme de Welzl offre une approche novatrice et efficace pour résoudre le problème de la boule minimale englobante. Son utilisation dans des domaines variés témoigne de son importance pratique. Bien que certaines limitations subsistent, ces dernières sont généralement compensées par la rapidité d'exécution et la garantie de résultats probables. L'algorithme de Welzl reste donc un outil précieux dans la boîte à outils des informaticiens travaillant sur des problèmes géométriques complexes.

2. Enoncé du projet :

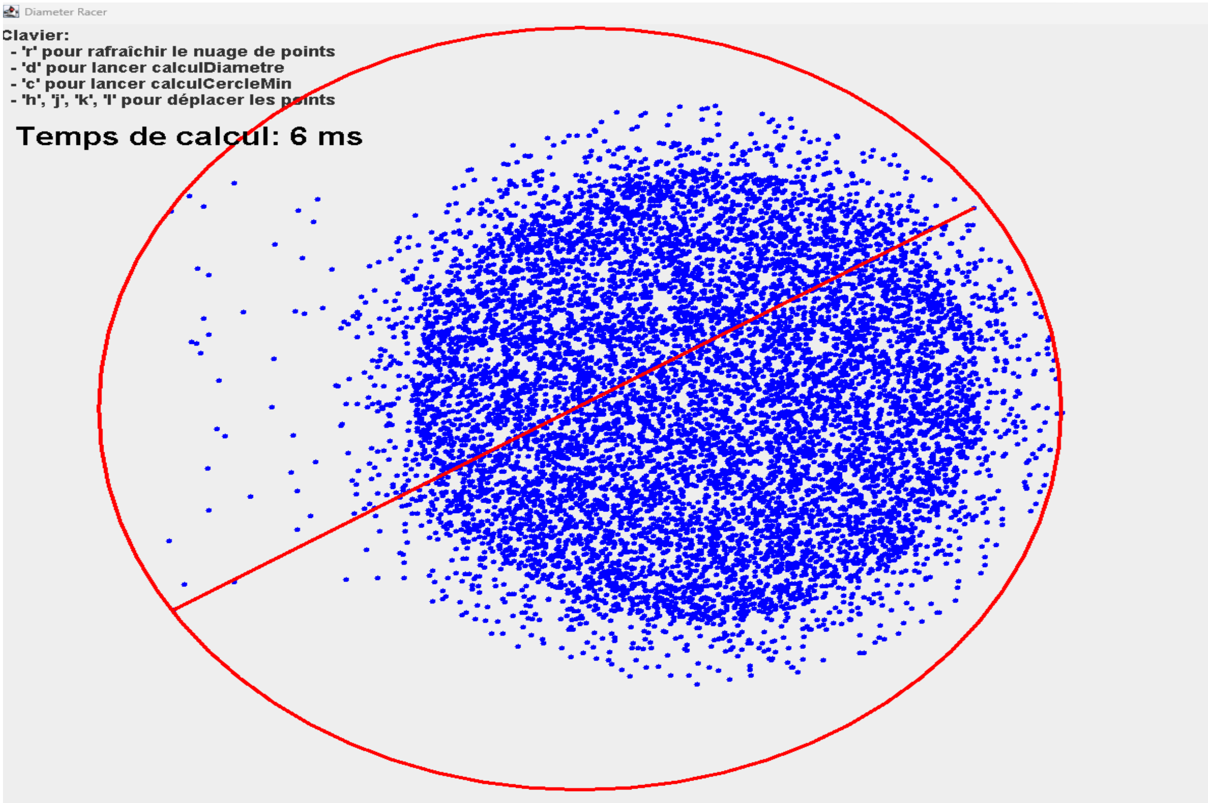
Il s’agit d’analyser la performance en temps de calcul de l’algorithme Welzl calculant un cercle couvrant un ensemble de points dans le plan.

a. Implanter un algorithme naıf calculant exactement le cercle minimum couvrant un ensemble donne de points dans le plan :

J’ai implémenté cet ensemble d'algorithmes géométriques en utilisant pour résoudre différents problèmes liés à des ensembles de points dans un plan.

1. `calculDiametre` :
   - Cette méthode prend une liste de points comme entrée.
   - Utilise l'algorithme de QuickHull (implémenté dans `tme1exercice8`) pour obtenir le polygone convexe enveloppant.
   - Puis, utilise l'algorithme de l'exercice 6 (`tme1exercice6`) pour calculer le diamètre du polygone convexe, c'est-à-dire la paire de points la plus éloignée.
2. `calculCercleMin` :
   - Prend une liste de points comme entrée.
   - Utilise également l'algorithme de QuickHull (`tme1exercice8`) pour obtenir le polygone convexe enveloppant.
   - Ensuite, applique l'algorithme de l'exercice 4 (`tme1exercice4`) pour calculer le cercle de rayon minimum couvrant le polygone convexe.
3. `tme1exercice4` :
   - Prend une liste de points comme entrée.
   - Implémente l'algorithme pour trouver le cercle circonscrit d'un ensemble de points.
   - L'algorithme commence par considérer les cercles ayant deux points comme diamètre, puis vérifie si tous les autres points sont à l'intérieur du cercle. Il retourne le cercle avec le rayon minimum qui couvre tous les points.
4. `tme1exercice5` :
   - Prend une liste de points comme entrée.
   - Implémente un algorithme pour calculer le cercle minimum en commençant par le plus petit cercle couvrant deux points, puis en ajoutant un point à la fois tout en maintenant la propriété du cercle couvrant minimum.
5. `tme1exercice6` :
   - Prend une liste de points comme entrée.
   - Implémente un algorithme pour trouver la paire de points la plus éloignée.
6. `tme1exercice7` :
   - Semble être un espace réservé (`return null`) et n'a pas d'implémentation dans le code que vous avez partagé.
7. `tme1exercice8` :
   - Prend une liste de points comme entrée.
   - Implémente l'algorithme QuickHull pour trouver le polygone convexe enveloppant.
   - Le polygone convexe résultant est construit récursivement en trouvant les points les plus éloignés d'une ligne donnée (segment de ligne entre deux points extrêmes) et en les ajoutant au polygone.

L'ensemble du code est orienté vers la résolution de problèmes géométriques classiques tels que le calcul du diamètre, du cercle circonscrit et du cercle couvrant minimum pour un ensemble de points dans le plan. L'utilisation de l'algorithme QuickHull est également présente pour trouver le polygone convexe enveloppant.



b. Implanter l’algorithme Welzl :

j’ai implementé cette partie de algorithmes pour calculer le diamètre d'un ensemble de points, ainsi que le cercle de rayon minimum englobant cet ensemble. Voici une explication détaillée des principales parties du code :

Partie Diamètre

1. `calculDiametre` **Function:**
    - Cette fonction prend une liste de points en entrée et retourne une ligne représentant le diamètre de l'ensemble, c'est-à-dire la paire de points ayant la distance maximale entre eux.
    - La fonction itère sur toutes les paires de points possibles, calculant la distance entre chaque paire et maintenant la paire avec la distance maximale.

Partie Cercle de Rayon Minimum

2. **`calculCercleMin` Function:**
   - Cette fonction prend une liste de points en entrée et retourne le cercle de rayon minimum englobant ces points.
   - Elle utilise l'algorithme de Welzl pour calculer ce cercle en deux variantes :
     - `calculCercleMin1` utilise également un algorithme naïf (`algoNaif`) basé sur toutes les combinaisons possibles de trois points pour déterminer le cercle minimal englobant.
     - `calculCercleMin2` utilise l'algorithme de Welzl lui-même (`calculateMinEnclosingCircle`).
3. **Algorithme Naïf (`algoNaif`):**
   - La fonction `algoNaif` effectue une recherche exhaustive en testant toutes les combinaisons de trois points pour déterminer le cercle minimal englobant.
   - Elle utilise des vérifications géométriques pour s'assurer que le cercle englobe effectivement tous les points.
4. **Algorithme de Welzl (`calculateMinEnclosingCircle`):**
   - Cet algorithme utilise l'algorithme de Welzl avec une variante (`welzlBMinidisk`) pour calculer le cercle de rayon minimum englobant un ensemble de points.
   - Il mélange aléatoirement les points, puis utilise `welzlBMinidisk` pour déterminer le cercle minimal englobant.
5. **Fonctions Utilitaires:**
   - Diverses fonctions utilitaires sont présentes, notamment celles pour calculer la distance entre deux points, vérifier si un point est à l'intérieur d'un cercle, calculer le centre et le rayon d'un cercle à partir de trois points, etc.

Partie Construction d'un Polygone Convexe

6. **`tme1exercice8` Function:**
   - Cette fonction utilise l'algorithme de QuickHull (`quickHull`) pour obtenir le polygone convexe englobant un ensemble de points.
   - Elle identifie les points extrêmes du polygone convexe à l'aide de la fonction `getExtremums`.

Partie Utilitaire pour les Points Extrêmes

7. **Fonctions pour les Points Extrêmes:**

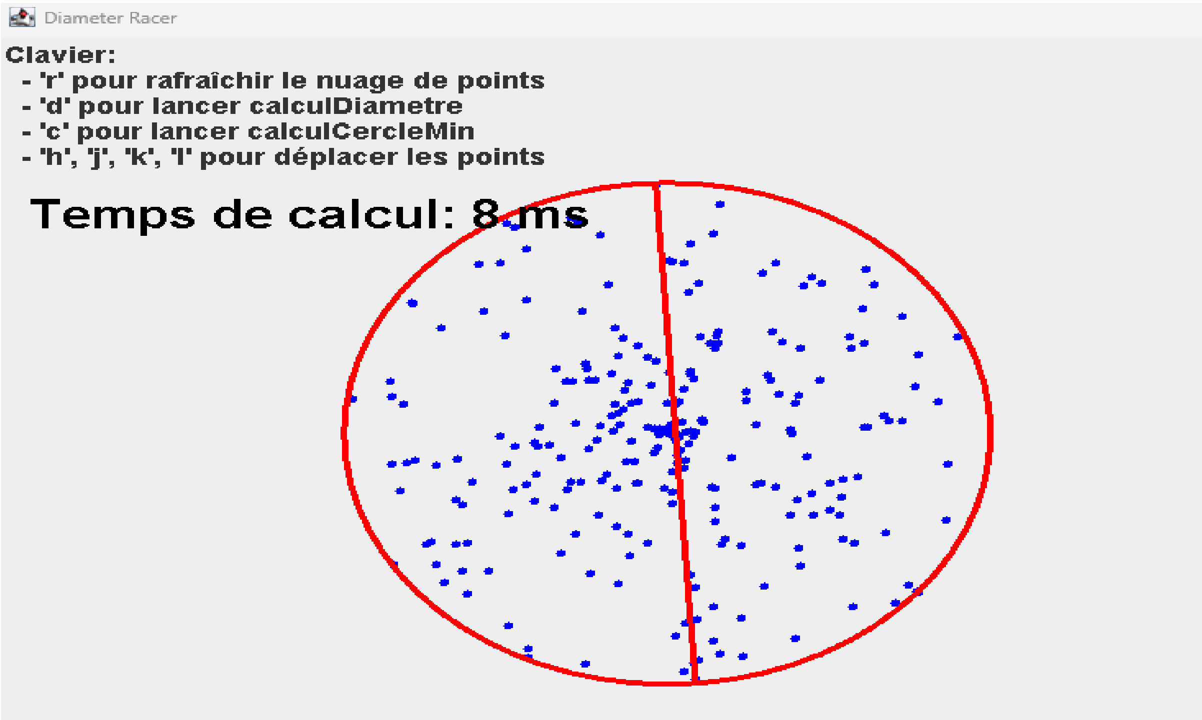
- `getExtremums`: Obtient les points les plus à gauche, à droite, en haut et en bas de l'ensemble de points.

Partie Principale (`run` Function)

8. **`run` Function:**
   - La fonction principale exécute des tests en lisant des ensembles de points à partir de fichiers, calculant le temps d'exécution des algorithmes, et enregistrant les résultats dans un fichier CSV.
9. **Lecture de Points à partir de Fichiers:**
   - Les points sont lus à partir de fichiers externes (`test-i.points`), où `i` varie de 2 à `n`.

Partie Principale (`main` Function)

10. **`main` Function:**
    - La fonction `main` appelle la fonction `run` avec la valeur passée en argument (probablement la taille `n` des ensembles de points à tester).



c. Recuperer une base de test de taille consequence :

Partie Principale (`run` Function)

8. **`run` Function:**
   - La fonction principale exécute des tests en lisant des ensembles de points à partir de fichiers, calculant le temps d'exécution des algorithmes, et enregistrant les résultats dans un fichier CSV.

INPUT	COORDONNE	NAIF	WELZL
2	256	6	1
3	512	5	1
4	768	7	1
5	1024	4	0
6	1280	3	1
7	1536	4	1
8	1792	5	1
9	2048	4	2
10	2304	4	2
11	2560	4	2
12	2816	6	2
13	3072	4	2
14	3328	7	2
15	3584	4	2
16	3840	6	1
17	4096	5	1
18	4352	4	1
19	4608	5	1
20	4864	5	2

d. Confronter l'algorithme Welzl aux instances de la base de test :

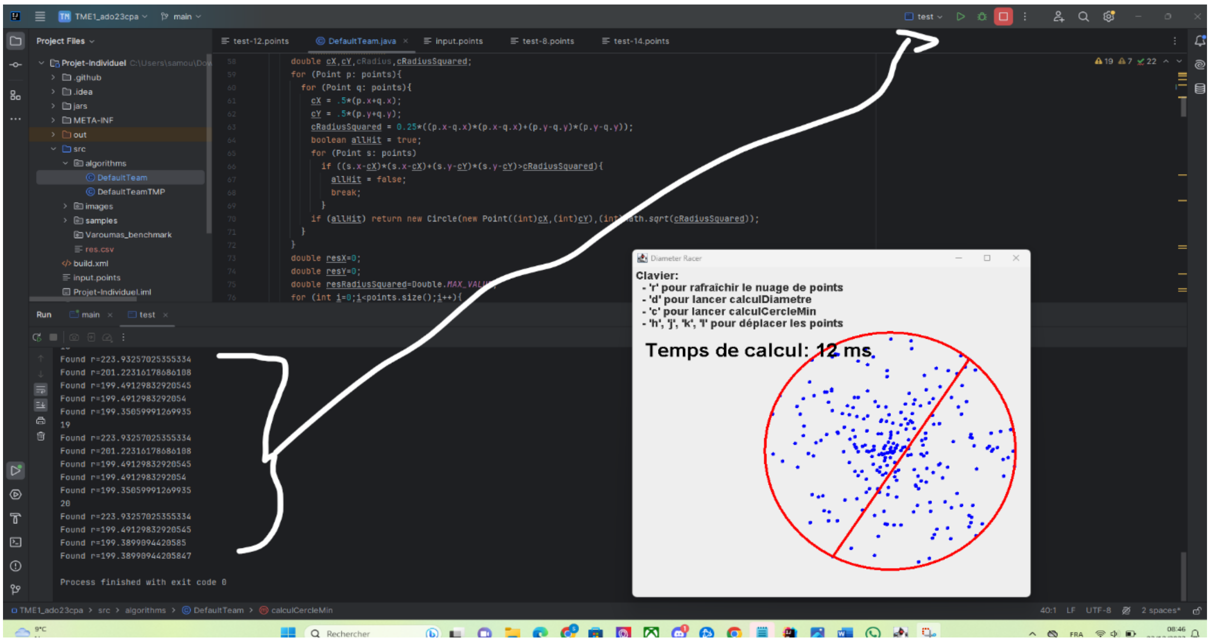
Partie Principale (`run` Function)

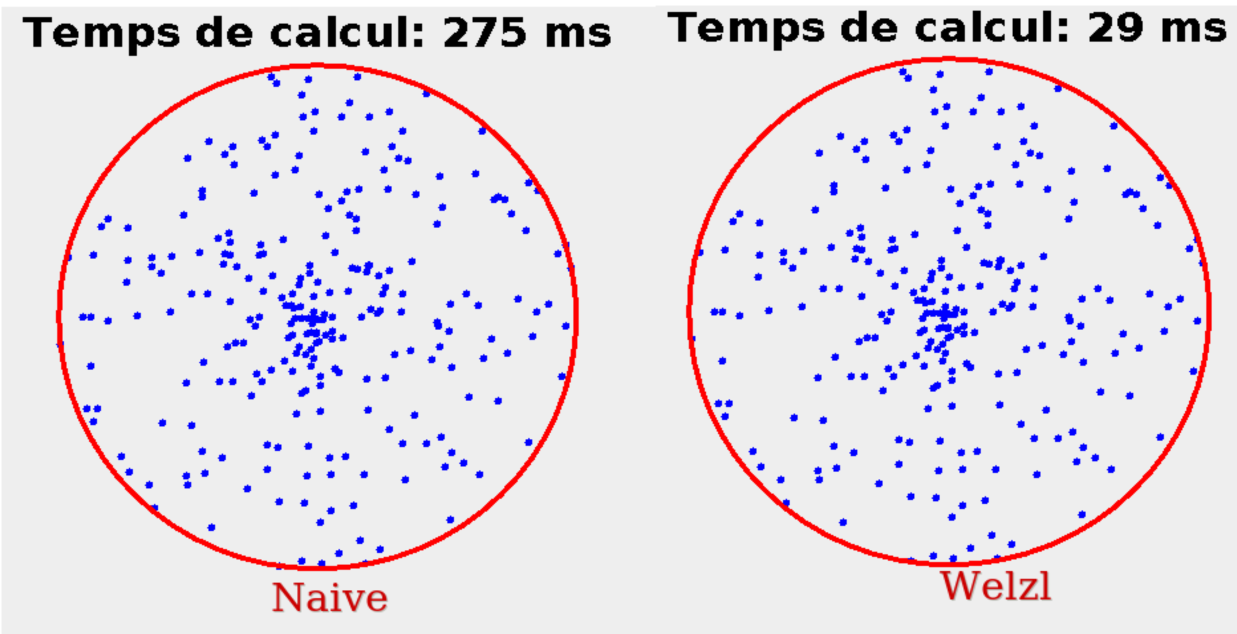
8. **`run` Function:**
   - La fonction principale exécute des tests en lisant des ensembles de points à partir de fichiers, calculant le temps d'exécution des algorithmes, et enregistrant les résultats dans un fichier CSV.
9. **Lecture de Points à partir de Fichiers:**

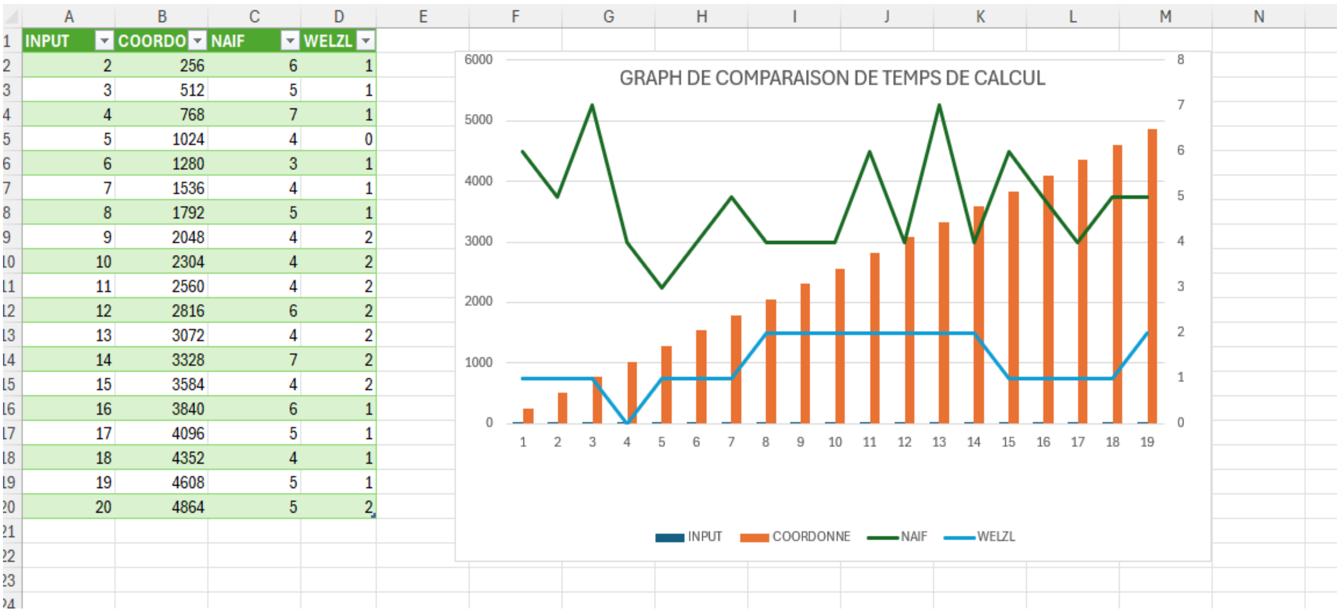
- Les points sont lus à partir de fichiers externes (`test-i.points`), où `i` varie de 2 à `n`.

Partie Principale (`main` Function)

10. **`main` Function:**
    - La fonction `main` appelle la fonction `run` avec la valeur passée en argument (probablement la taille `n` des ensembles de points à tester).
    - J'ai configuré une instance d'environnement permettant d'effectué et afficher le resultat des deux methode.
11. Voir image



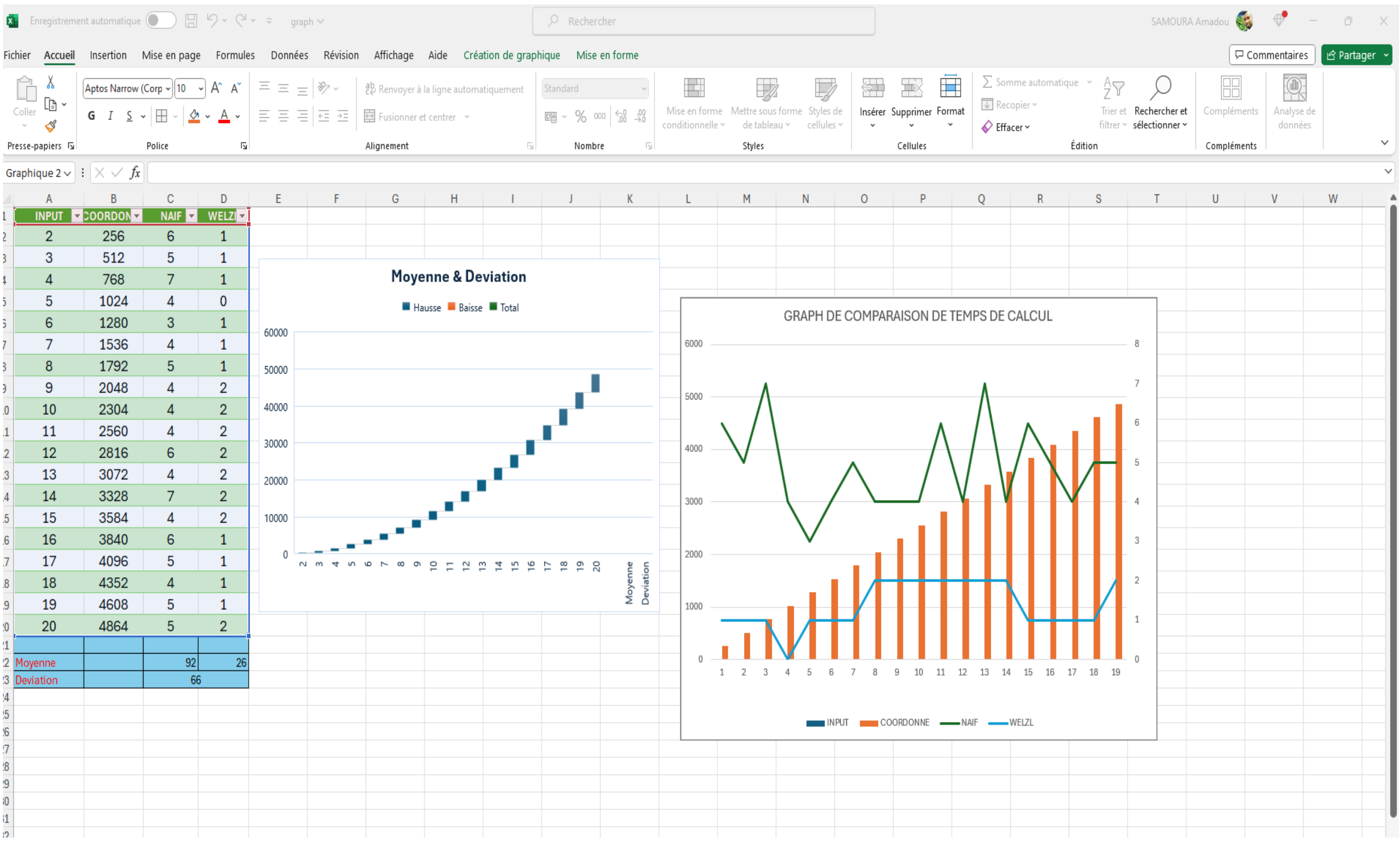




INPUT	COORDONNE	NAIF	WELZL
2	256	6	1
3	512	5	1
4	768	7	1
5	1024	4	0
6	1280	3	1
7	1536	4	1
8	1792	5	1
9	2048	4	2
10	2304	4	2
11	2560	4	2
12	2816	6	2
13	3072	4	2
14	3328	7	2
15	3584	4	2
16	3840	6	1
17	4096	5	1
18	4352	4	1
19	4608	5	1
20	4864	5	2

e. Etablir au moins un diagramme-batons (moyenne + deviation standard) :

Pour établir un diagramme représentant la moyenne et la déviation entre l'implémentation de l'algorithme Welzl et l'implémentation de l'algorithme naïf, vous devrez collecter des données expérimentales pour différentes tailles d'entrée (nombre de points) et mesurer les temps d'exécution correspondants.



f. Question bonus : Quelle est votre impression par rapport aux notions similaires en 3D :

Les notions similaires en 3D pour la recherche du cercle ou de la sphère minimale englobante impliquent généralement des concepts tels que la sphère englobante minimale (Minimum Bounding Sphere ou MBS en anglais). En 3D, l'idée est de trouver la sphère de rayon minimum qui englobe un ensemble de points dans l'espace tridimensionnel. Les algorithmes pour résoudre ce problème peuvent différer de ceux en 2D, mais certains principes de base persistent.

Sphère Englobante Minimale :

En 3D, la sphère englobante minimale est définie par son centre (coordonnées tridimensionnelles) et son rayon.
L'objectif est de trouver la sphère de rayon minimum qui englobe tous les points de l'ensemble donné.

Algorithme de Welzl en 3D :

L'algorithme de Welzl peut être étendu à la 3D pour résoudre le problème de la sphère englobante minimale en utilisant une approche similaire à celle en 2D.
Il implique la randomization et la récursion pour exclure des points et trouver la sphère minimale.

Approches spécifiques à la 3D :

En 3D, les calculs géométriques deviennent plus complexes, mais certaines méthodes, comme l'algorithme de Ritter, ont été proposées pour résoudre efficacement ce problème.
L'algorithme de Ritter combine des étapes d'heuristique avec des calculs géométriques pour converger rapidement vers une solution.

Complexité accrue :

Les calculs en 3D sont généralement plus intensifs que ceux en 2D en raison de la nature tridimensionnelle de l'espace.

Applications :

Les sphères englobantes minimales en 3D ont des applications dans la modélisation géométrique, la vision par ordinateur, la robotique et d'autres domaines où la compréhension de la structure spatiale des données est cruciale.
Globalement, les concepts en 3D sont une extension naturelle de ceux en 2D, mais la complexité des calculs géométriques et la nécessité de traiter avec des structures spatiales plus complexes rendent ces problèmes plus exigeants. Les algorithmes et les approches dépendront souvent du contexte d'application spécifique et des exigences de performance.

Dépôt GitHub :

Voir le lien ici puis les détails de démarrage du projet sont dans le fichier ReadMe.

https://github.com/samoura-amadou/CDA-PROJET-INDIVIDUEL.git

En résumé, le projet met en œuvre des algorithmes (naïf et welzl) pour calculer le diamètre d'un ensemble de points et le cercle de rayon minimum englobant cet ensemble. Il effectue des tests sur ces algorithmes avec des ensembles de points de tailles différentes et enregistre les résultats dans un fichier CSV.