Sorbonne Université
Faculté des Sciences et Ingénierie
Departement d'Informatique

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

Dans le cadre de l'UE d'Algorithmique Avancée

Auteur
Amine BENSLIMANE

Binh-Minh BUI-XUAN

Master 1 STL 2020/ 2021
10 janvier 2021

Table des matières

0.1 Introduction

Le problème du cercle minimum est un classique dans le jargon de l'algorithmique. Il a été soulevé par James Joseph Silvester en 1857. Depuis lors, plusieurs algorithmiciens et scientifiques au fil du temps ont contribué à la résolution de ce problème.

Nous trouvons dans la littérature plusieurs de ces algorithmes, nous citons notamment :

- L'algorithme naïf*
- L'algorithme de Chrystal [1]
- L'algorithme Ritter [2]
- L'algorithme de Welzel [3]

Dans ce travail, nous présentons un algorithme naïf et son implantation pour le calcul du cercle minimum. Ensuite nous ferons de même pour l'algorithme Wezl.

Ceci étant fait, on confrontera ces deux algorithmes à la même base de test choisie est celle de *VAROUMAS*.

On établira ensuite une étude comparative de l'efficacité en temps de notre algorithme naïf et de celui de Welzl, on illustrera l'ensemble de nos résultats par des diagrammes-bâtons.

On divisera notre travail en . chapitres comme suit :
- Le premier chapitre portera sur une étude théorique concise de l'état de l'art des divers algorithmes trouvés dans la littérature. - Le deuxième chapitre portera sur la définition de l'algorithme naïf ainsi que son implantation.
- Le troisième chapitre portera sur la définition de l'algorithme Welzl ainsi que son implantation puis le confronter avec l'algorithme naïf sur une même base de test en synthétisant l'ensemble des résultats que l'on juge pertinents obtenus. On finit ce chapitre avec quelques conclusions.

Nous avons choisi de rester sur le langage de programmation JAVA, puisque nos TMEs dans l'UE ont été faits sur celui ci. On profitera alors des connaissances acquises et de l'interface graphique "DiamRacer" utilisées dans cette UE.

* : Nous insinuons par "naïf" le fait que l'algorithme parcoure tout les points possibles sans aucun filtrage, c'est une expression très utilisée dans le jargon de l'informatique théorique.

Chapitre 1

État de l'art

1.1 Présentation du problème du cercle minimum

Nous définissons le problème du cercle minimum comme suit :

Definition 1.1.1 (Cercle minimum)**.** Le cercle minimum dans un plan est le cercle le plus petit contenant un ensemble de points d'un plan.

Voici une illustration du cercle minimum :

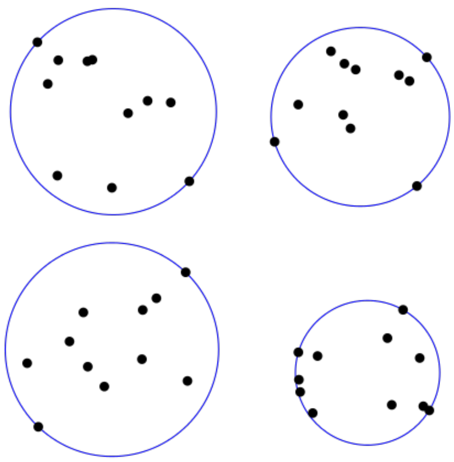

FIGURE 1.1 – Smallest circle probleme

Le premier a avoir soulevé ce problème est le mathématicien Anglais James Joseph Silvester en 1857 [4], dans un article publié dans le prestigieux journal Quarterly Journal of Pure and Applied Mathematics où il y dit :

"Il faut trouver le plus petit cercle qui contient un ensemble donné de points du plan".

Ce problème peut être mathématiquement formalisé en un problème de minimisation que l'on présentera dans la section ci-dessous.

1.2 Formalisation du problème du cercle minimum

Le problème du cercle minimum peut se ramener à un problème d'optimisation. En effet, il s'agit de minimiser le rayon r du cercle en considérant un ensemble fini de points a_i pour $i \in [1, .., n]$
On cherche donc à minimiser la quantité r en imposant que le cercle de centre x et de rayon r doit contenir tout les points, ce qui est formalise le problème (P) suivant :

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

Toutefois, nous nous penchons vers les approches algorithmiques de la résolution de ce problème.

1.3 Cercle circonscrit d'un triangle

Pour traiter ce problème, il est important de définir au moins succintivement la notion de cercle circonscrit d'un triangle.

Definition 1.3.1 (Cercle circonscrit)**.** Le cercle circonscrit d'un triangle est l'unique cercle qui passe par ses trois points.

La figure 1.2 illustre le cercle circonscrit et sa construction euclidienne.

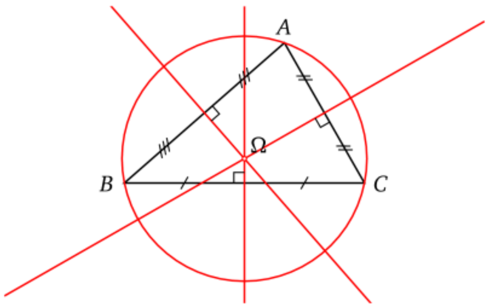

FIGURE 1.2 – Cercle circonscrit

1.4 Enveloppe convexe

Aussi, nous définissons l'enveloppe convexe d'un ensemble de point.

Definition 1.4.1 (Enveloppe convexe)**.** L'enveloppe convexe d'un ensemble E de point est le plus petit polygone convexe formé par les points de E.

La figure 1.3 ci-dessous illustre la notion d'enveloppe convexe. (dessinée en bleu)

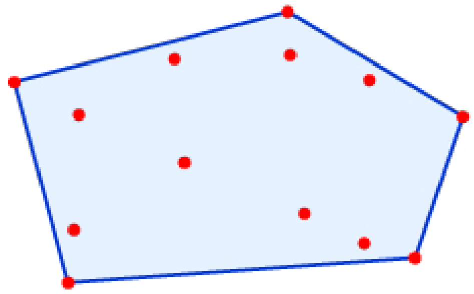

FIGURE 1.3 – Enveloppe convexe

1.5 Solutions algorithmiques connues de la littérature

1.5.1 Algorithme de Chrystal

Plusieurs algorithmiciens se sont penchés vers ce problème mais c'est au professeur M. Chrystal que revient le mérite d'avoir la première contribution avec l'algorithme qui porte son nom [1], il se base sur le fait de ne considérer que les points formant l'enveloppe convexe des points, voici un pseudo-code résumant son algorithme :

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

Require: Ensemble P de points
Ensure: Cercle minimum c contenant P

1. Calculer l'enveloppe convexe (H_i) tq $1 \leq i \leq m$;
2. Prendre arbitrairement un coté $[H_iH_j]$ de deux sommets adjacents $\in (H_i)$ et choisir un 3^e sommet H_k tel que l'angle opposé formé est minimal ;
3. Si l'angle formé est droit ou obtus alors renvoyer le cercle de diamètre $[H_iH_j]$;
4. Si les trois angles sont aïgues alors renvoyer cercle circonscrit du triangle ;
5. Si $H_iH_jH_k$ est obtus en H_i alors répéter 3 en considérant $[H_jH_k]$, si le nouvel angle est obtus en H_j alors répéter 3 et considérer $[H_kH_i]$

Le nombre d'opération dépend du premier côté choisi, toutefois on ne peut choisir le même coté deux fois ; par suite $\frac{1}{2}m(m-1)$ est une borne supérieure du nombre d'opérations $\implies$ complexité en $O(m^2)$.

1.5.2 Algorithme de Shamos et Hoey

En 1975, Michael Ian Shamos et Dan Hoey présentent un algorithme qui se base sur le paradigme "diviser pour régner" en utilisant le "diagramme de Voronoï" [5]
Nous présentons brièvement l'algorithme :

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

Require: Ensemble P de points
Ensure: Cercle minimum c contenant P

1. Déterminer l'enveloppe convexe et le diagramme de Voronoï des points les plus éloignés
2. Déterminer la paire la plus éloignée et essai du cercle dont elle est le diamètre
3. Déterminer les triplets les plus éloignés et essai des cercles circonscrits correspondants

L'algorithme de Shamos et Hoey s'éxecute en $O(nlog(n))$. [6]

Il y'a plusieurs autres algorithmes permettant le calcul du cercle minimum (ou même une approximation du cercle minimum), nous citons quelques uns avec leur complexités respectives :
- L'algorithme de Ritter en $O(n)$. [2]
- L'algorithme de Megiddo en $O(n)$. [7]
- L'algorithme de Welzl en $O(n)$. [3]

Nous concentrons notre travail à l'implémentation de l'algorithme de Welzl et à sa confrontation à une base de test.

1.6 Conclusions

La littérature regorge de contributions résolvant le problème du cercle minimum. Bien sûr, nous n'avons pas encore présenté la solution algorithmique la plus simple pour résoudre ce problème, et qui est l'algorithme naïf, ceci se fera dans le chapitre suivant où nous détaillons nos travaux.
Enfin, nous présenterons l'algorithme Welzl et les résultats que nos obtenons lors de la phase des tests.

Chapitre 2

Algorithme Naïf

2.1 Présentation de l'algorithme

Aussi appelé dans la littérature "Algorithme de contrôle", l'algorithme *naïf* est incontournable. Il permet d'avoir une solution de base pour un problème qui admet une solution. Bien qu'il soit souvent trop coûteux en terme de complexité, c'est une instance simple à réaliser et qui constitue une base de comparaison pour quantifier l'efficacité des autres algorithmes traitant le même problème, c'est précisément la méthode de travail que nous choisissons.

2.2 Pseudo-code de l'algorithme naïf

Nous présentons ci-dessous notre algorithme naïf en pseudo-code (langage humain) :

Algorithm 3 Calculer naïvement le cercle minimum d'un espace de points

Require: Ensemble P de points
Ensure: Cercle minimum c contenant P
  for tout p dans P **do**
    for tout q dans P **do**
      $c \leftarrow$ cercle de centre $\frac{p+q}{2}$ de diamètre $|p \times q|$
    end for
  end for
  for tout p dans P **do**
    for tout q dans P **do**
      for tout r dans P **do**
        $c \leftarrow$ cercle circoncit de p, q et r

        if c couvre tout les points de P **then**
          renvoyer c
        end if
      end for
    end for
  end for

Vous remarquerez qu'aucun pré-calcule n'a été fait, l'algorithme parcours exactement tout les points pour tous les englober dans un cercle. Plus exactement, cet algorithme se compose de deux phases, la première vise à encercler les points par le cercle du plus grand diamètre possible (distance maximale entre deux points), si cela n'est pas suffisant on procède à la deuxième phase où l'on teste sur les cercles circonscrit en parcourant tout les triplets (p, q, r) possibles, on s'arrête dès que l'on trouve un cercle c qui englobe tout les points, c'est le cercle minimale.

Notre algorithme s'appuie sur deux lemmes principaux qui garantissent *(par abus de langage)* "la minimalité" du cercle retourné.
Voici les deux lemmes pour les deux phases utilisées dans les deux phases respectives :

Lemma 2.2.1. Si un cercle de diamètre égale à la distance de deux points couvre tout les autres points, alors ce cercle est minimal.

Lemma 2.2.2. Il existe un unique cercle passant par trois points non-colinéaires. *(en 2d)*

2.3 Compléxité 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 compléxité 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 $O(n^3 \times n) = O(n^4)$.

2.4 Conclusions

L'algorithme naïf représente une solution facilement compréhensible et simple à réaliser mais très coûteuse en temps, nous avons essayé d'exécuter une instance de 10 000 points sans pouvoir avoir un résultat (même après 30 minutes avec un I3 7th et 8GB de Ram). Nous présentons les résultats des tests plus tard dans le rapport. Passons maintenant à une solution à compléxité linéaire, qui est l'algorithme de Welzl.

Chapitre 3

Algorithme de Welzl et tests

3.1 Présentation de l'algorithme de Welzl

Emo Welzl, professeur et chercheur en Informatique publie en 1991 son algorithme qui prend son nom [3] pour résoudre le problème du cercle minimum dans un espace de points donné dénombrable et fini à coût linéaire.
L'algorithme de Welzl est récursif, il considère deux ensembles :
- P : Ensemble des points de départ
- R : Ensemble des points sur le cercle minimum

Il convient qu'au départ, P contient tout les points et R=0. Voici l'algorithme :

Algorithm 4 MINIDESK(P,R)

Require: Ensemble P de points de départ, Ensemble R de points sur le cercle minimum
Ensure: Cercle minimum c contenant P noté MINIDESK(P,R)
  if P = ∅ OU |R|=3 **then**
    D ← trivial(∅,R)
  else
    choisir arbitrairement p ∈ P
    $D \leftarrow$ MINIDESK$(P - \{p\}, R)$
    if D *défini ET* $p \notin D$ **then**
    $D \leftarrow$ MINIDESK$(P - \{p\}, R \cup \{P\})$

  end if
  return D

Par abus de langage, on écrit $trivial(\emptyset, R)$ pour dire que dans ce cas, le résultat est immédiat : c-à-d soit il n'y a aucun point et donc aucun calcul à faire, soit il existe déjà trois points délimitant le cercle et donc il suffit de calculer le cercle circonscrit présenté dans la section 1.3.
Les cas triviaux écartés, on choisit par hasard un point p in P, on le retire de P et on stock le résultat de MINIDESK$(P - \{p\}, R)$ dans une variable D, si D n'est pas définie, il suffit

de rajouter le point dans R (le point est exactement sur le cercle minimal). Lorsqu'on aura $|R| = 3$, on atteindra un cas trivial.

Welzl prouve que son algorithme est en $O(n)$. [3]
Nous passons maintenant à la section test.

3.2 Test de l'algorithme naïf vs algorithme de Welzl

Nous présentons dans cette section les résultats obtenus par nos algorithmes. La base de test étant Varoumas Benchmark.
La figure ci-dessous synthétise les résultats que nous avons obtenus pour l'algorithme naïf

Notez que les abscisses sont 1, 2, 3..., 9. Cela désigne combien d'instance de 256 points ont été traitées. C'est à dire 256, 512, ..., 2048 points.

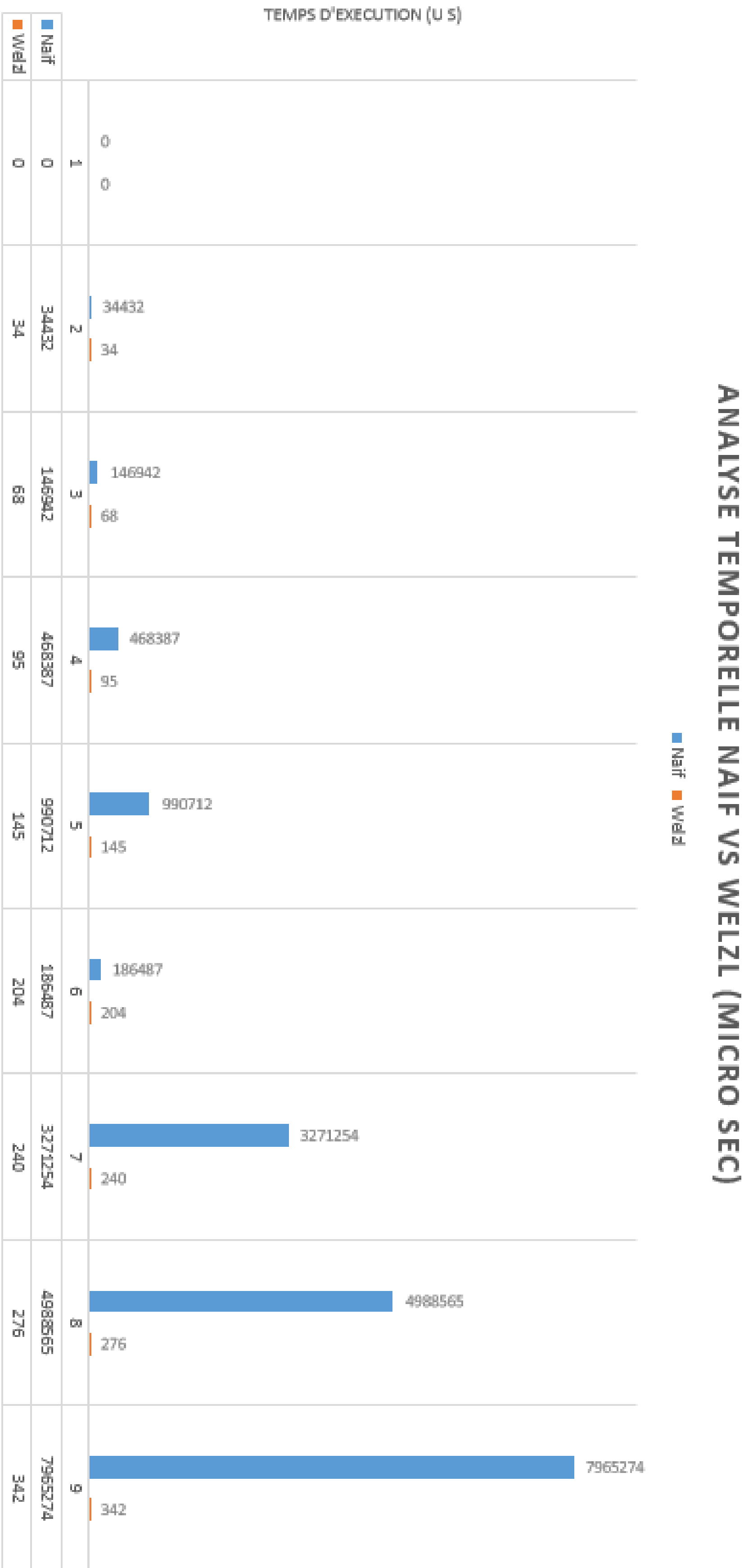

FIGURE 3.1 – Naif vs Welzl

3.3 Conclusions

La correction de l'algorithme Welzl par rapport à l'algorithme naïf est flagrante. C'est l'un des algorithmes les plus efficaces de la littérature pour ce problème.

On précise que les tests ont été fait sur une machine dotée d'un processeur I3 7th Gen avec 8GB de mémoire vive, bien qu'elle soit puissante, on n'est pas parvenue à executer l'algorithme naïf sur des instances de points assez conséquente (10 000 par exemple).

L'algorithme de Welzl quant à lui présente des résultats très efficaces et est l'un des algorithmes les plus utilisés dans l'industrie.
Ce travail a permis de montrer la puissance d'une solution algorithmique et la différence que l'on peut avoir entre une solution réalisable (Algorithme naïf) et une solution efficace (Algorithme Welzl).

Avec ce travail, nous avons enrichis nos connaissances dans le jargon de la géométrie algorithmique en générale et au problème du cercle minimum en particulier.

Bibliographie

[1] CHRYSTAL. « On the problem to construct the minimum circle enclosing n given points in a plane ». In : *Proceedings of the Edinburgh Mathematical Society* 3 (1884), p. 30-33. DOI : 10.1017/S0013091500037238 (pages 1, 4).

[2] Jack RITTER. « An Efficient Bounding Sphere ». In : (déc. 1990). DOI : 10.1016/B978-0-08-050753-8.50063-2 (pages 1, 5).

[3] Emo WELZL. « Smallest enclosing disks (balls and ellipsoids) ». In : *New Results and New Trends in Computer Science*. Sous la dir. d'Hermann MAURER. Berlin, Heidelberg : Springer Berlin Heidelberg, 1991, p. 359-370. ISBN : 978-3-540-46457-0 (pages 1, 5, 8, 9).

[4] J. J. SYLVESTER. « A question in the geometry of situation ». In : *Quarterly Journal of Pure and Applied Mathematics* (1857), p. 1-57 (page 2).

[5] D. Hoey. M. SHAMOS. « Closest-Point Problems ». In : *16th Annual Symposium on Foundations of Computer Science* (1975), p. 151-162 (page 4).

[6] L. GUIBAS, D. KNUTH et M. SHARIR. « Randomized incremental construction of Delaunay and Voronoi diagrams ». In : *Algorithmica* 7 (2005), p. 381-413 (page 5).

[7] N. MEGIDDO. « Linear-Time Algorithms for linear programming in R^3 and related problems ». In : *Siam J. Comput.* 12.4 (1983), p. 759-775 (page 5).