Łukasz Stępień gr. IV

Algorytmy geometryczne

dokumentacja projektu:

Wyznaczanie minimalnego okręgu i prostokąta zawierającego chmurę punktów na płaszczyźnie

1. Dane techniczne:
    - Język implementacji: Python
    - Środowisko programistyczne: Jupyter Notebook
    - System operacyjny: Microsoft Windows 10 Pro x64
    - Procesor: Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz, 2904 MHz

2. Wymagania systemowe:

    - System *Windows 10* lub nowszy
    - Zainstalowane środowisko *Jupyter Notebook*
    - Zainstalowany interpreter języka *Python*
    - Zainstalowane biblioteki języka *Python*: *Numpy*, *Matplotlib*

3. Działanie programu:

    Pierwszy moduł to konfiguracja narzędzia graficznego, następnie implementacja algorytmu Grahama wyznaczającego otoczkę wypukłą danego zbioru punktów. Potem zamieszczone zostały funkcję odpowiadające za wyznaczanie minimalnego okręgu i prostokąta zawierającego chmurę punktów na płaszczyźnie wraz z wizualizacją. Następnie przedstawione są testowane zbiory punktów oraz wizualizacja na nich przebiegu algorytmów.

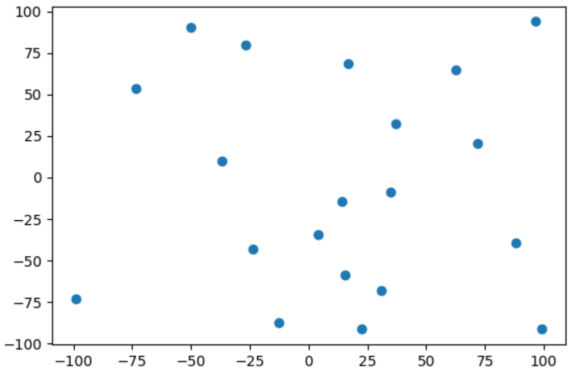
4. Sposób uruchamiania:

    Plik z programem należy uruchomić w środowisku Jupyter Notebook, następnie przyciskiem *run* uruchomić wszystkie komórki po kolei. Wizualizację oraz wyniki będą się pojawiać pod odpowiednią podpisanymi komórkami

5. Opis problemu:
    Wyznaczanie minimalnego okręgu i prostokąta zawierającego chmurę punktów na płaszczyźnie.

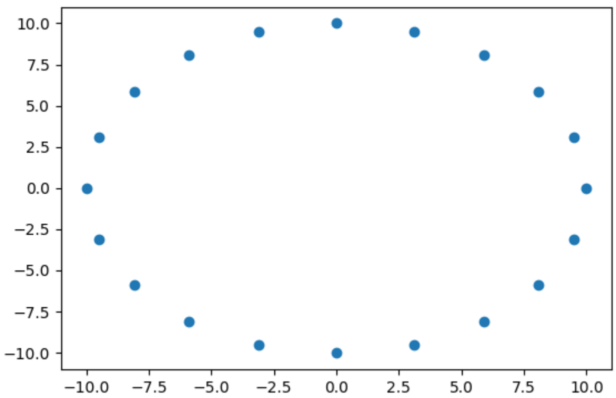
6. Zestawy danych testowanych:

- 20 losowo wygenerowanych punktów o współrzędnych z przedziału [-100, 100]:



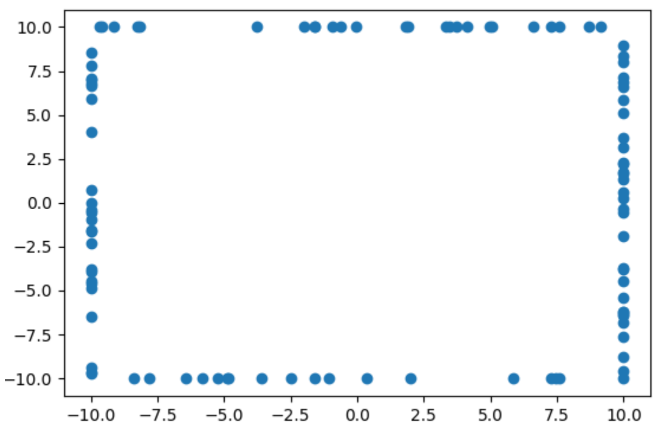
Wykres 1 Zestaw danych A

- 20 losowo wygenerowanych punktów leżących na okręgu o środku (0,0) i promieniu R=10:



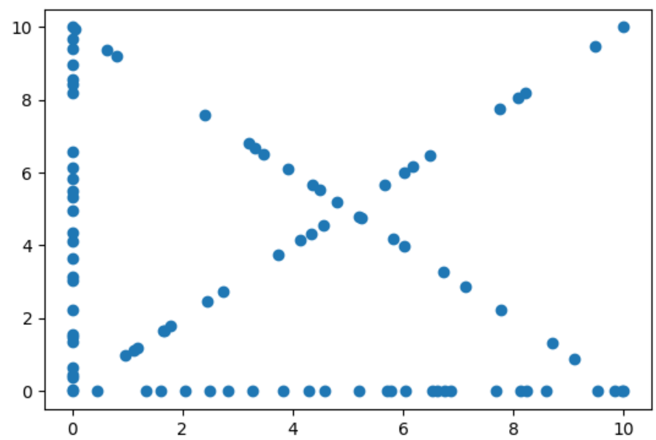
Wykres 2 Zestaw danych B

- 100 losowo wygenerowanych punktów leżących na bokach prostokąta o wierzchołkach (-10, 10), (-10,-10), (10,-10), (10,10):



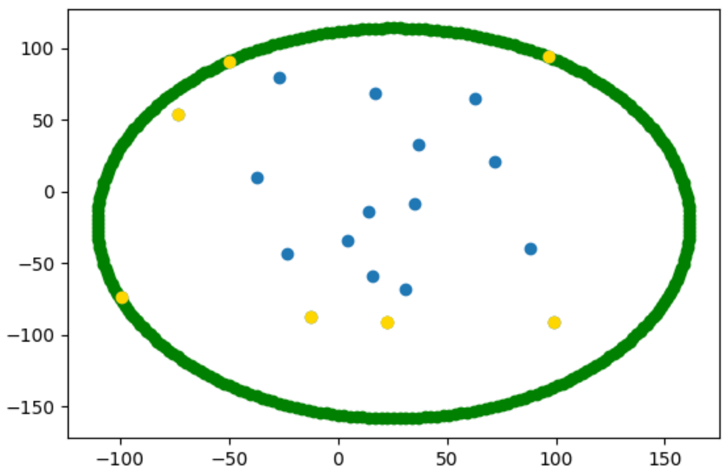
Wykres 3 Zestaw danych C

- wierzchołki kwadratu (0, 0), (10, 0), (10, 10), (0, 10) oraz punkty wygenerowane losowo w sposób następujący: po 25 punktów na dwóch bokach kwadratu leżących na osiach i po 20 punktów na przekątnych kwadratu.

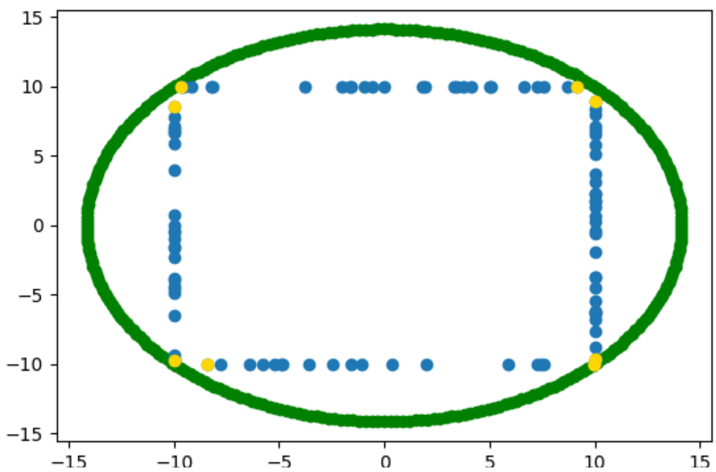


Wykres 4 Zestaw danych D

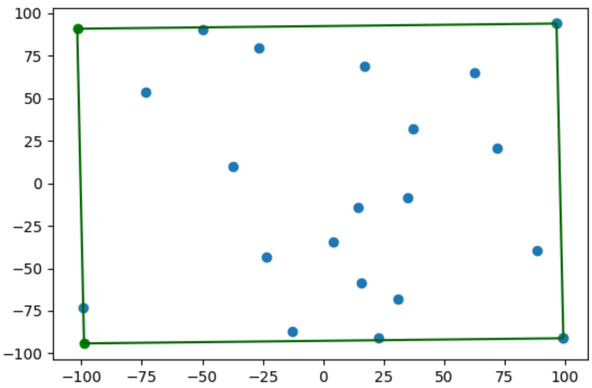
7. Wizualizacja wyników algorytmów dla kilku zestawów danych:



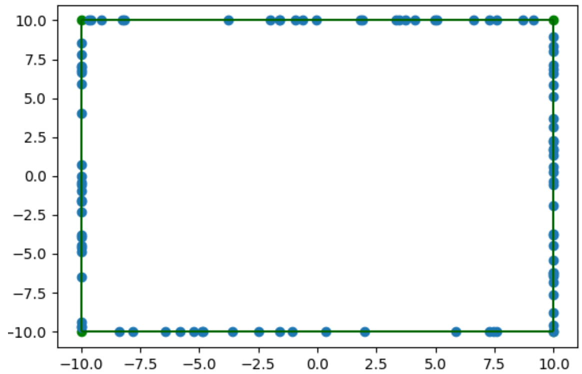
Wykres 5 Zestaw A - minimalny okrąg



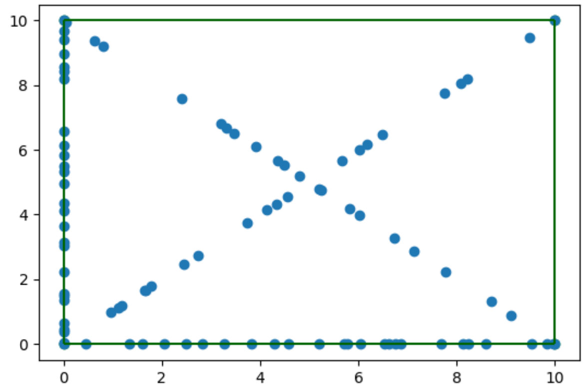
Wykres 6 Zestaw C - minimalny okrąg



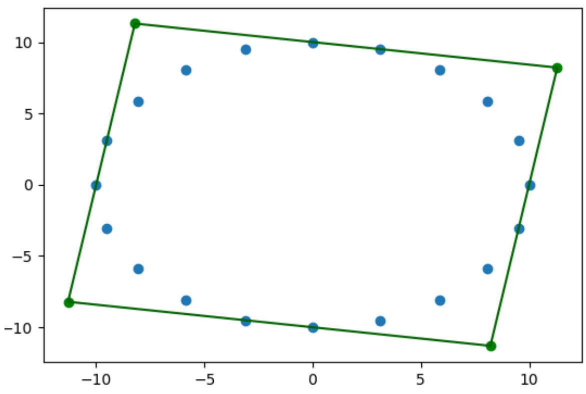
Wykres 7 Zestaw A - minimalny prostokąt (pole)



Wykres 8 Zestaw D - minimalny prostokąt (pole)



Wykres 9 Zestaw C - minimalny prostokąt (obwód)



Wykres 7 Zestaw B - minimalny prostokąt (obwód)

8. Opis przebiegu algorytmów.

Każdy algorytm rozpoczyna się od wyznaczenia otoczki wypukłej zbioru punktów na płaszczyźnie za pomocą algorytmu Grahama, którego złożoność obliczeniowa wynosi O(nlogn).

Algorytm wyznaczający minimalny okrąg zawierający chmurę punktów na płaszczyźnie to zoptymalizowany algorytm *brute-force.* Na początku przeszukujemy każdą parę punktów w otoczce, tworzymy okrąg przez nie opisany i sprawdzamy, czy zawiera wszystkie punkty z zadanego zbioru punktów. Jeżeli tak, to porównujemy jego promień z poprzednim najmniejszym okręgiem (na początku to okrąg o promieniu nieskończonym) i odpowiednio aktualizujemy wynik. Następnie przeszukujemy każdą trójkę punktów i powtarzamy te same kroki co w przypadku dwóch punktów. Złożoność algorytmu to $O(h^4+n\log n)$, gdzie n to liczebność wyjściowego zbioru punktów, a h to liczebność punktów w otoczce wypukłej. Jeżeli ilość punktów w otoczce jest mała, to $h^4<n\log n$, z czego wynika, że złożoność algorytmu dla większości zbiorów to O(nlogn).

Algorytm wyznaczający minimalny prostokąt pod względem pola lub obwodu zawierający chmurę punktów na płaszczyźnie korzysta z twierdzenia, że musi on zawierać przynajmniej jeden bok otoczki wypukłej. Jego działanie polega na iteracji po wszystkich bokach otoczki, wyznaczeniu w każdej najbardziej odległych punktów względem obróconego układu współrzędnych oraz wyznaczenie dzięki temu prostokąta i poprzez porównywanie odpowiednio pól lub obwodów wyznaczenie najmniejszego.
Złożoność obliczeniowa to O(h+nlogn), gdzie h<=n, czyli O(nlogn).

9. Podsumowanie:

Program poprawnie wyznacza minimalny okrąg oraz prostokąt zawierający chmurę punktów na płaszczyźnie. Zamieszczone wizualizację pozwalają łatwo śledzić przebieg każdego algorytmu. Program jest łatwy do użytkowania oraz nie wymaga profesjonalnych zdolności programistycznych.

10. Dane bibliograficzne:

- https://gist.github.com/kchr/77a0ee945e581df7ed25
- https://github.com/cansik/LongLiveTheSquare
- https://www.geeksforgeeks.org/minimum-enclosing-circle/?ref=lbp
- https://en.wikipedia.org/wiki/Minimum_bounding_box
- https://pl.wikipedia.org/wiki/Algorytm_Grahama