

Algorytmy Geometryczne

Sprawozdanie Projekt

Wyznaczanie minimalnego okręgu i
prostokąta zawierającego chmurę
punktów na płaszczyźnie

Aga Patro
czw_13.30_A

1. Specyfikacja sprzętu i narzędzia wykorzystane w realizacji

System: Debian Linux Parrot OS x64

Procesor: AMD Ryzen 5 4500U, 6 rdzeni, 6 wątków, 4.00GHz

Pamięć RAM: 16 GB

Środowisko: Jupyter Notebook

Język: Python 3

2. Temat ćwiczenia

Celem ćwiczenia było zapoznanie się z tematem wyznaczania minimalnego okręgu i prostokąta zawierającego chmurę punktów na płaszczyźnie.

3. Opis problemu

Zadawana jest chmura punktów na płaszczyźnie dwuwymiarowej.

Program ma wyznaczać:

- minimalny okrąg zawierający tę chmurę,
- prostokąt o minimalnym polu powierzchni zawierający tę chmurę.
- prostokąt o minimalnym obwodzie zawierający tę chmurę.

4. Sposób wykonania

By wykonać ćwiczenie podjęłam następujące kroki:

1. Zaimplementowałam funkcję generującą losowo zbiory punktów.
2. By wyznaczyć minimalny okrąg zaimplementowałam algorytm Welzl'a
3. By wyznaczyć minimalne prostokąty zaimplementowałam algorytm oparty o *rotating_calipers*.

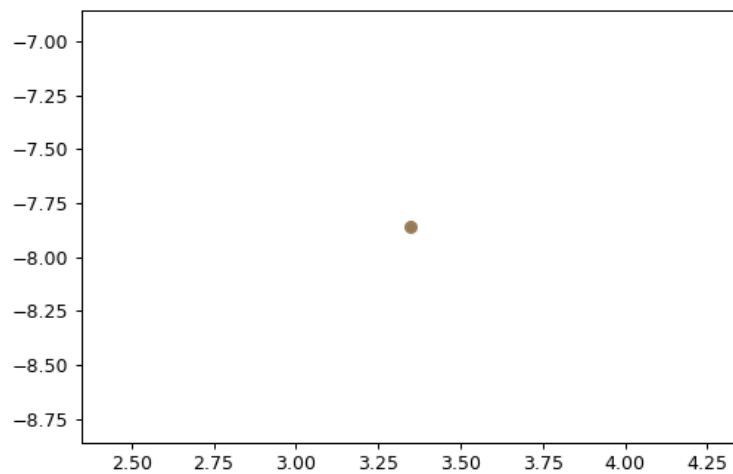
5. Realizacja ćwiczenia

5.1 Wyznaczanie minimalnego okręgu

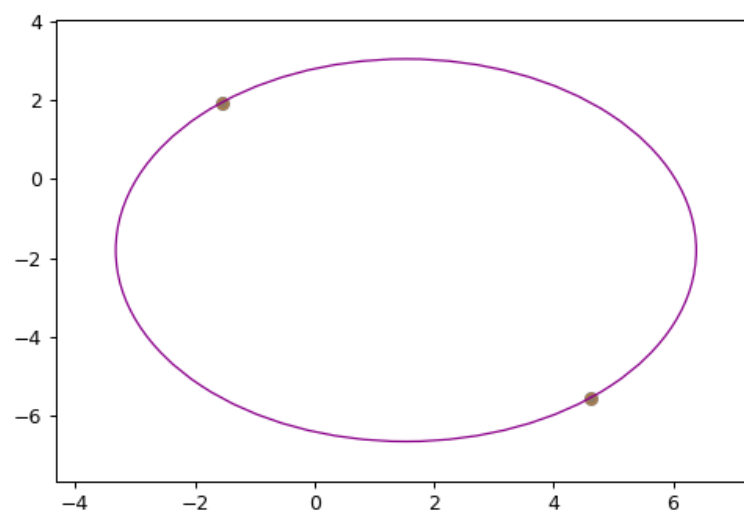
Muszę przyznać że nie udało mi się zrealizować prezentacji wizualizacji działającego programu. Jedyne co mogę pokazać to już znaleziony okrąg.

Algorytm powinien działać w czasie liniowym, aczkolwiek na niektórych $n > 10$, bardzo długo trwa mu wyznaczenie danego okręgu. Wydaje mi się że jest to spowodowane tym że algorytm szuka randomowo do skutku, aż znajdzie dany okrąg.

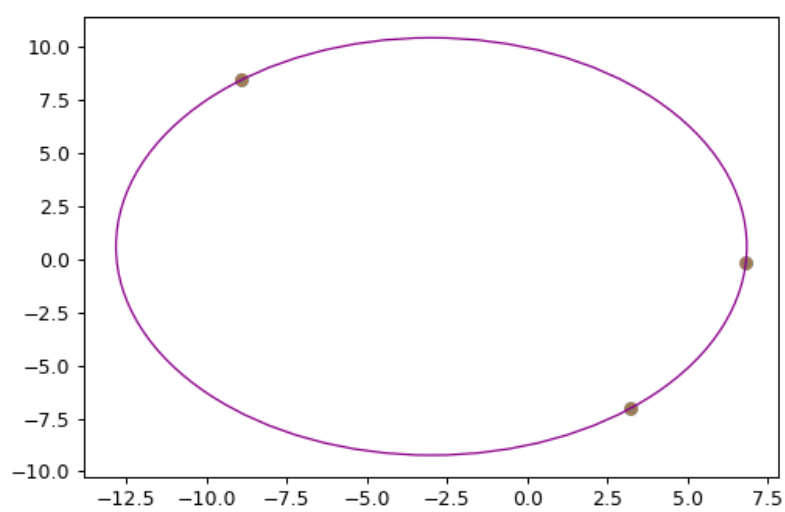
5.1.1 Wizualizacja testów dla wrażliwej ilości punktów



5.1.1.1 otrzymany okrąg z $n = 1$

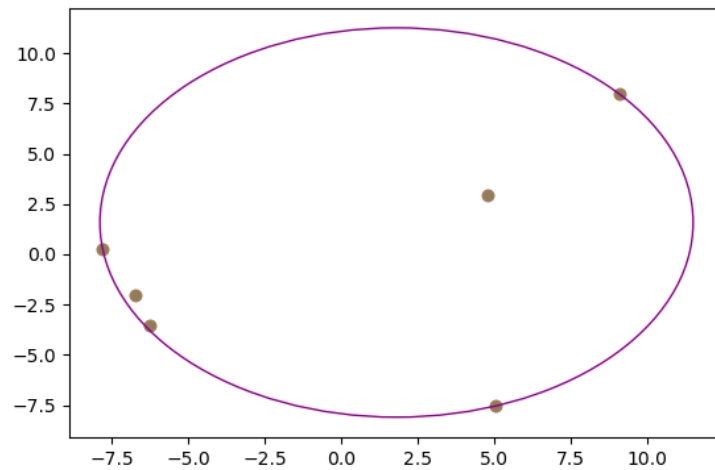


5.1.1.2 otrzymany okrąg z $n = 2$

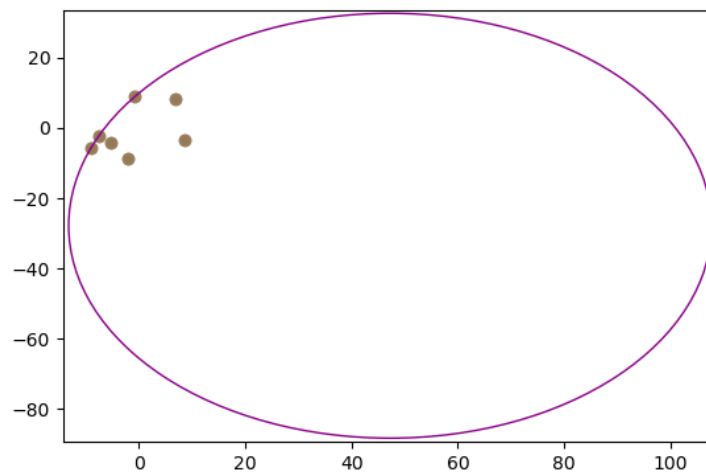


5.1.1.3 otrzymany okrąg z $n = 3$

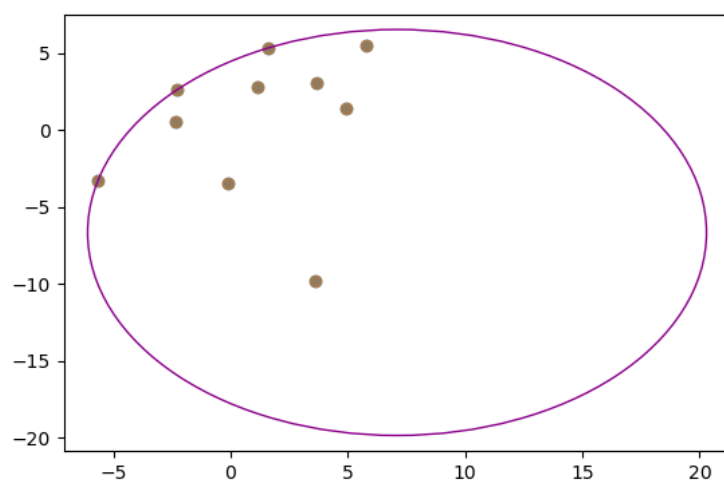
5.1.2 Wizualizacja testów dla większej ilości punktów



5.1.2.1 otrzymany okrąg z $n = 6$



5.1.2.2 otrzymany okrąg z $n = 7$



5.1.2.3 otrzymany okrąg z $n = 10$

5.1.2. Uzyskane wyniki

Dla wszystkich przetestowanych (niestety małych) danych program spełnia swoje zadanie, to jest wyznacza najmniejszy okrąg zawierający w sobie dany zbiór punktów. Algorytm jest poprawny.

5.2 Wyznaczanie minimalnych prostokątów

Tak jak w przykładzie z okręgiem, nie udało mi się zrealizować wizualizacji algorytmu. Ponadto funkcja do rysowania otrzymanego prostokąta rysuje zły prostokąt tj. zwracane współrzędne nie pasują do prostokąta narysowanego przez funkcję rysującą, i nie potrafię powiedzieć dlaczego. Dlatego nie mogłam przeprowadzić testów wizualnych by sprawdzić poprawność algorytmu. Jedyne co, to mogę wierzyć mojemu kodowi że zwraca dobry prostokąt.

Natomiast z moich obserwacji wynika, że algorytm wyznaczający najmniejsze prostokąty działa szybciej niż algorytm Welzl'a, oraz nie jest tak wrażliwy na dużą ilość danych.