**Otoczka Wypukła, algorytmy Jarvisa i Grahama**

**Ryszard Pręcikowski Wtorek 14:40B**

**12.11.2020**

Specyfikacja techniczna:

System operacyjny: Windows 10 64bit

Procesor: AMD Ryzen 3600 6-Core Processor

Środowisko: Python 3.8.3

Opis realizacji ćwiczenia:

Pierwszym krokiem w doświadczeniu było wygenerowanie czterech zestawów danych:

1. 100 losowych punktów o współrzędnych z przedziału [-100, 100],
2. 100 losowych punktów leżących na okręgu o środku (0,0) i promieniu R = 10,
3. 100 losowych punktów leżących na boku prostokąta o wierzchołkach (-10, 10), (-10,-10), (10,-10), (10,10),
4. zawierający 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.

Następnie, na podstawie funkcji generujących powyższe zbiory zostały utworzone cztery kolejne:

1. 500 punktów z przedziału [-10^5, 10^5],
2. 100 punktów, R = 1000, środek okręgu (10, 10),
3. 250 punktów, leżących na bokach prostokąta o wierzchołkach (-500, -500), (500, 500), (-500, 500), (500, -500),
4. Współrzędne kwadratu (-500, -500), (500, 500), (500, -500), (-500, 500) na bokach i przekątnych po 100 punktów.

Punkty zostały wygenerowane na płaszczyźnie, natomiast ich współrzędne to liczby zmiennoprzecinkowe.

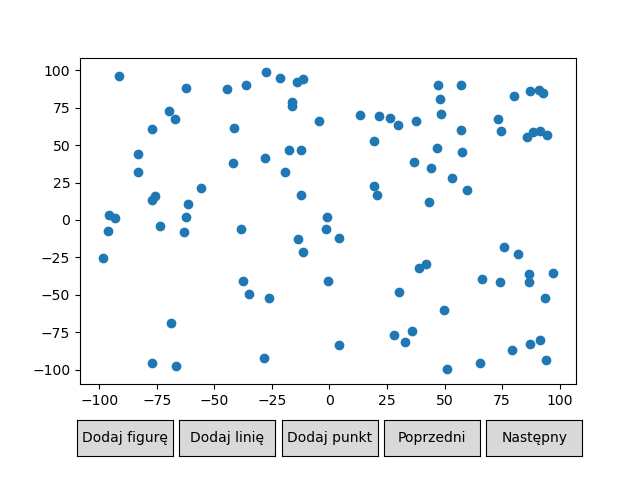
W kolejnym kroku dla każdego zestawu danych szukamy otoczki wypukłej za pomocą algorytmów Grahama oraz Jarvisa.

W obu algorytmach do wyznaczania wzajemnego położenia punktów używamy wyznacznika 3x3 w postaci:

W większości przypadków tolerancja dla zera wynosiła 10-10, ale w przypadku zmodyfikowanych zbiorów danych musiała zostać zwiększona w podpunktach a i c.

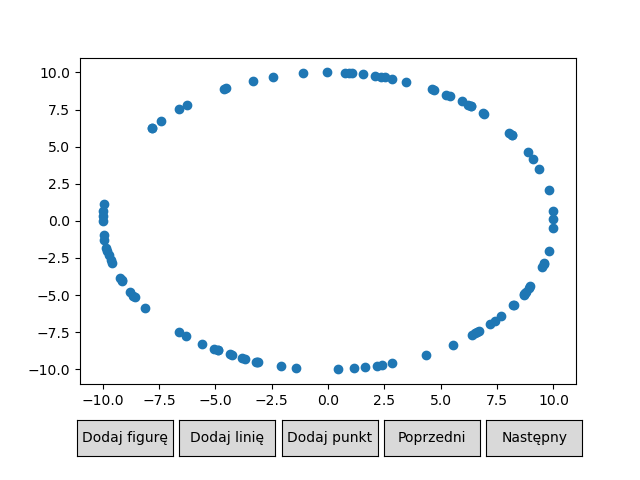
Wizualizacja zbiorów punktów:

1. Zbiór danych a (Rysunek 1)



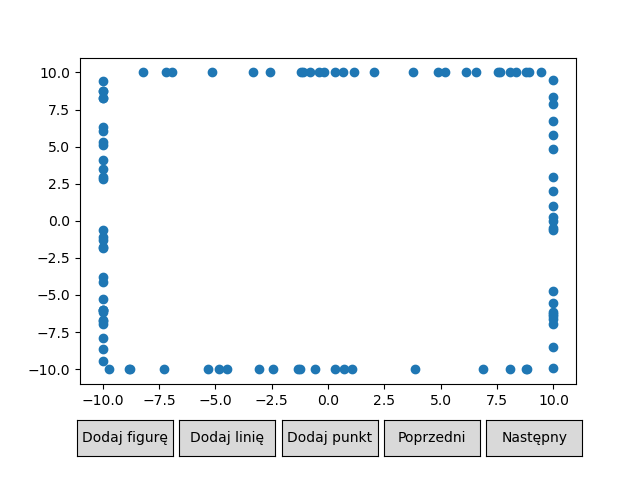
Rysunek 1 Zbiór danych a

1. Zbiór danych b (Rysunek 2)



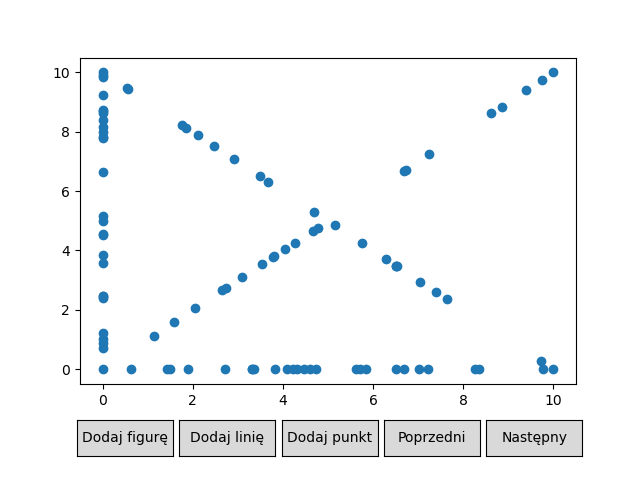
Rysunek 2 Zbiór danych b

1. Zbiór danych c (Rysunek 3)



Rysunek 3 Zbiór danych c

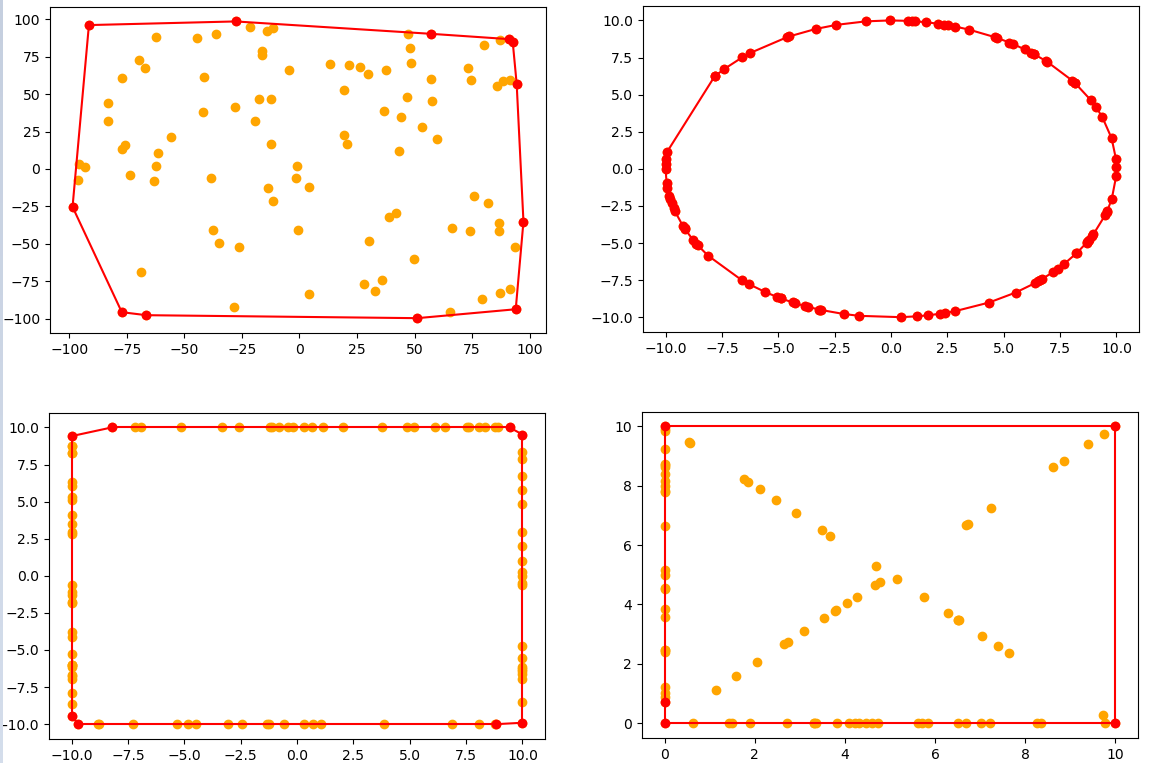
1. Zbiór danych d (Rysunek 4)



Rysunek 4 Zbiór danych d

Analiza działania algorytmu Grahama

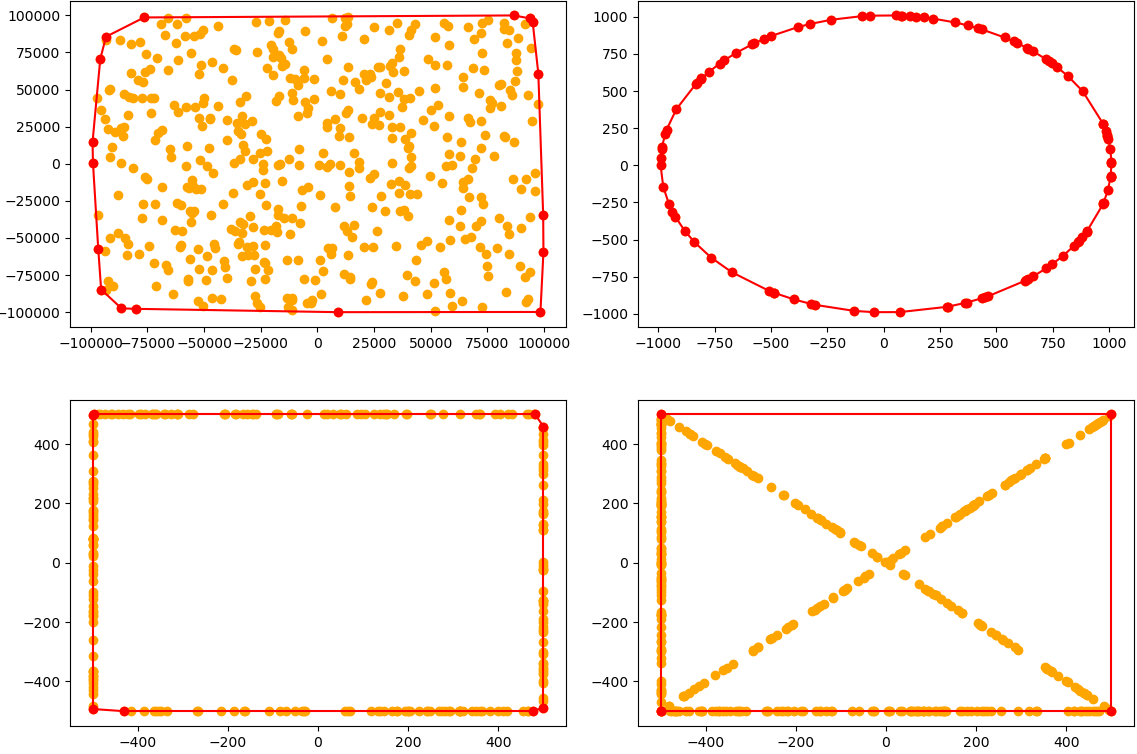
Dla podstawowych zestawów danych algorytm wygenerował otoczki poprawnie (Rysunek 5).



Rysunek 5 Otoczki wygenerowane przez algorytm Grahama dla podstawowych zbiorów danych

Jedynie dla zestawu D do otoczki został zakwalifikowany jeden punkt współliniowy, dla pozostałych uzyskaliśmy optymalne rozwiązania, bez punktów leżących na tej samej prostej.

Dla zmodyfikowanych zestawów danych algorytm również działa poprawnie (Rysunek 6).



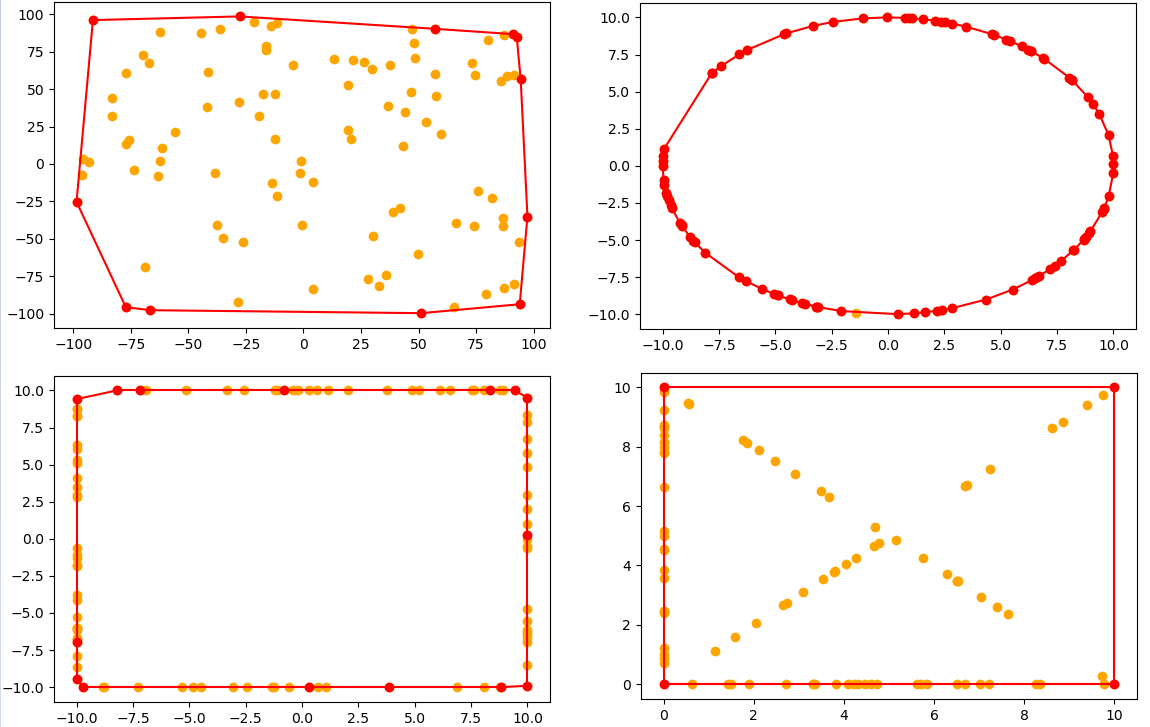
Rysunek 6 Otoczki wygenerowane przez algorytm Grahama dla zmodyfikowanych zbiorów danych

W tym przypadku dla każdego zestawu danych otrzymaliśmy otoczki bez punktów współliniowych.

Analiza działania algorytmu Jarvisa.

Dla podstawowych zbiorów danych algorytm generował poprawnie otoczki (Rysunek 7). Jedynie dla zbioru C przydzielane są punkty współliniowe, ale jest ich jedynie siedem. Jest to spowodowane najprawdopodobniej tolerancją zera.

W zestawie B jeden punkt nie jest zaliczany do otoczki.

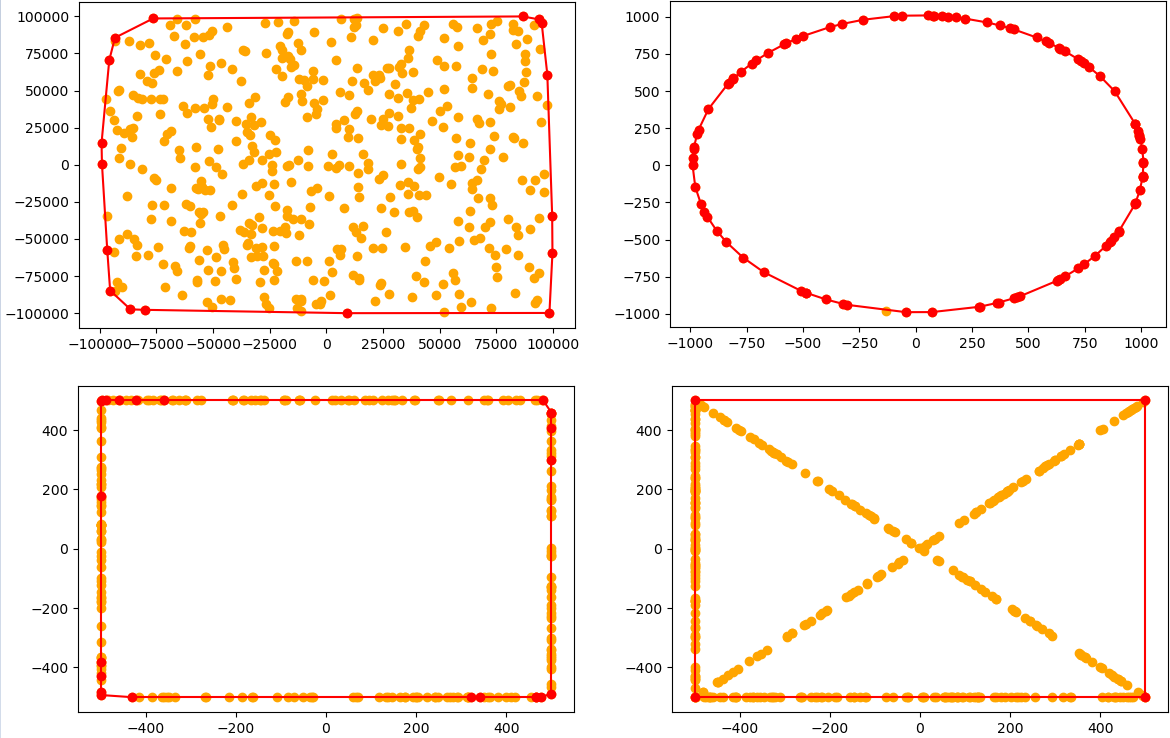


Rysunek 7 Otoczki wygenerowane przez algorytm Jarvisa dla podstawowych zbiorów danych

Dla zmodyfikowanych zestawów danych algorytm również działa poprawnie (Rysunek 8).

Zachodzi sytuacja analogiczna jak dla zestawu podstawowego.

Dla zbioru C znajdowane jest tym razem 15 punktów współliniowych, a w zbiorze B jeden punkt nie jest zaliczany do otoczki.



Rysunek 8 Otoczki wygenerowane przez algorytm Jarvisa dla zmodyfikowanych zbiorów danych

Porównanie jakości generowanych otoczek.

W Tabeli 1 przedstawiona została ilość punktów zawierających się w otoczkach generowanych przez algorytmy.

|  |  |  |  |
| --- | --- | --- | --- |
| zbiór | liczność zbioru | Graham | Jarvis |
| a | 100 | 12 | 12 |
| am | 500 | 17 | 17 |
| A | 500000 | 33 | 33 |
|  |  |  |  |
| b | 100 | 100 | 99 |
| bm | 100 | 100 | 99 |
| B | 5000 | 5000 | 5000 |
|  |  |  |  |
| c | 100 | 8 | 15 |
| cm | 250 | 8 | 23 |
| C | 200000 | 8 | 46 |
|  |  |  |  |
| d | 90 | 5 | 4 |
| dm | 400 | 5 | 4 |
| D | 400000 | 5 | 4 |

Tabela 1 Ilość punktów zakwalifikowana do otoczki, a - zbiór a, am- zmodyfikowany zbiór a, A - dodatkowy, duży zbiór

Dla losowo rozłożonych danych oba algorytmy generują otoczki tak samo liczne.

W przypadku rozłożenia punktów na kwadracie algorytm Grahama generuje otoczki bez punktów współliniowych, natomiast te z algorytmu Jarvisa zawierają od kilku do kilkunastu takich punktów.

W zbiorze D algorytm Jarvisa znajduje poprawnie cztery narożniki, algorytm Grahama przypisuje do otoczki dodatkowe, pojedyncze punkty.

Porównanie prędkości Algorytmów.

W tabeli 2 został przedstawiony czas działania algorytmów dla wszystkich zestawów danych.

|  |  |  |
| --- | --- | --- |
| zbiór | Graham | Jarvis |
| a | 0,003 | 0,001 |
| am | 0,017 | 0,008 |
| A | 9,55 | 13,04 |
|  |  |  |
| b | 0,004 | 0,009 |
| bm | 0,004 | 0,129 |
| B | 0,06 | 10,34 |
|  |  |  |
| c | 0,003 | 0,002 |
| cm | 0,006 | 0,007 |
| C | 4,25 | 9,03 |
|  |  |  |
| d | 0,120 | 0,000 |
| dm | 0,007 | 0,002 |
| D | 14,15 | 2,02 |

Tabela 2 Czas działania algorytmów dla różnych zestawów danych, w sekundach

Im mniej punktów zawiera się w otoczce tym przewaga algorytmu Grahama maleje. Jest to zgodne ze złożonością obliczeniową tych algorytmów. Algorytm Grahama ma złożoność nlogn, natomiast Jarvisa pesymistyczną n^2, jeśli jednak liczba punktów otoczki „k” jest znana, to złożoność wynosi k\*n. Największą przewagę algorytmu Jarvisa widzimy, gdy do otoczki należą jedynie 4 punkty, natomiast algorytm Grahama ma największą przewagę w przypadku, gdy do otoczki należą wszystkie punkty.