Algorytmy Geometryczne
Labolatorium nr 2

Szymon Szarek

11 listopada 2023

1 Opis ćwiczenia

Celem ćwiczenia było wyznaczenie otoczki wypukłej zbioru punktów. Otoczka wypukła zbioru P to najmniejszy wielokąt wypukły taki, że każdy punkt ze zbioru P leży albo na brzegu wielokąta albo w jego wnętrzu. Do wyznaczenia otoczki wykorzystane były dwa algorytmy: Grahama i Jarvisa.

1.1 Algorytm Grahama

Kroki algorytmu prezentują się następująco:

1. niech p_0 będzie punktem w zbiorze Q z najmniejszą współrzędną y, oraz najmniejszą współrzędną x w przypadku, gdy wiele punktów ma tą samą współrzędną x
2. niech $\langle \mathtt{p_1, p_2, \ldots, p_m} \rangle$ będzie pozostałym zbiorem punktów w Q posortowanym zgodnie z przeciwnym ruchem wskazówek zegara wokół punktu p_0 (jeżeli więcej niż jeden punkt ma ten sam kąt to usuwamy wszystkie punkty z wyjątkiem tego najbardziej oddalonego od p_0)
3. stwórz pusty stos S
4. $\mathtt{PUSH(p_0, S)}$
5. $\mathtt{PUSH(p_1, S)}$
6. $\mathtt{PUSH(p_2, S)}$
7. **for** $i = 3$ **to** m
8. **while** kąt utworzony przez $\mathtt{NEXT-TO-TOP(S)}$, $\mathtt{TOP(S)}$ oraz p_i tworzy lewostronny skręt
9. $\quad\mathtt{POP(S)}$
10. $\mathtt{PUSH(p_i, S)}$
11. **return** S

Oznaczenia poszczególnych funkcji:

- $\mathtt{TOP(S)}$ zwraca punkt z góry stosu bez usuwania go ze stosu S
- $\mathtt{NEXT-TO-TOP(S)}$ zwraca punkt poniżej góry stosu bez usuwania go ze stosu
- $\mathtt{POP(S)}$ usuwa punkt z góry stosu
- $\mathtt{PUSH(p, S)}$ wstawia punkt p na stos S

Złożoność obliczeniowa algorytmu: $O(nlogn)$

1.2 Algorytm Jarvisa

1. Znajdź punkt i_0 z S o najmniejszej współrzędnej y; $i \leftarrow i_0$
2. **repeat**
3. **for** $j \neq i$ **do**
4. znajdź punkt, dla którego kąt liczony przeciwnie do wskazówek zegara w odniesieniu do ostatniej krawędzi otoczki jest najmniejszy
5. niech k będzie indeksem punktu z najmniejszym kątem. Zwróć (p_i, p_k) jako krawędź otoczki
6. $i \leftarrow k$
7. **until** $i = i_0$

Złożoność obliczeniowa algorytmu: $O(n^2)$
Gdy liczba wierzchołków jest ograniczona przez stałą k, złożonosc algorytmu wynosi $O(nk)$

2 Środowisko pracy

Ćwiczenie zostało napisane w języku Python za pomocą platformy Jupyter Notebook. Jako narzędzie graficzne do wizualizacji figur geometrycznych zostało użyte narzędzie przygotowane przez koło naukowe BIT, które wykorzystuje takie biblioteki jak np. *matplotlib* lub *numpy*.

Specyfikacja sprzętu:

- System Windows 11 x64
- Procesor AMD Ryzen PRO 5650U
- Pamięć RAM 16GB
- Python 3.9

3 Realizacja ćwiczenia

W ramach ćwiczenia na początku zostały wygenerowane 4 zbiory punktów typu *double*:

a 100 losowych punktów o współrzędnych z przedziału $[-100, 100]$,

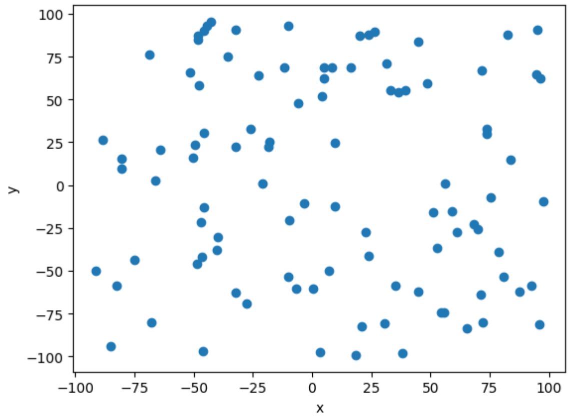
b 100 losowych punktów leżących na okręgu o środku (0,0) i o promieniu R=10,

c 100 losowych punktów leżących na bokach prostokąta o wierzchołkach (-10,10), (-10,-10), (10,-10), (10,10)

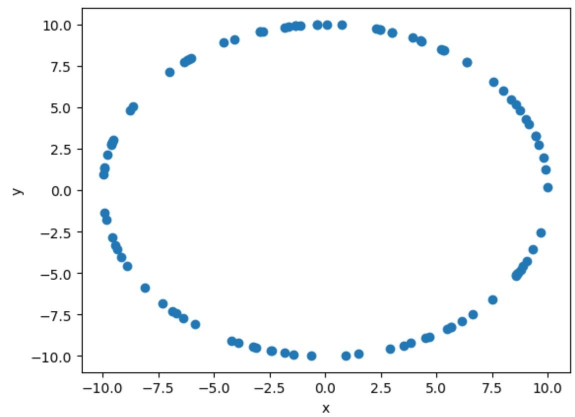
d wierzchołki kwadratu (0, 0), (10, 0), (10, 10), (0, 10) oraz punkty wygenerowane losowo w sposób następujący: po 25 punktów na dwóch bokach kwadratu leżących na osiach i po 20 punktów na przekątnych kwadratu.

Do wygenerowania zbiorów została użyta funkcja uniform z biblioteki random. Zbiór c został wygenerowany następująco: najpierw losowany był bok, na którym pojawi się punkt, a później losowany był punkt na prostej zawierającej ten bok ograniczonej przez wierzchołki prostokąta. Następnie zaimplementowano algorytmy Grahama i Jarvisa.

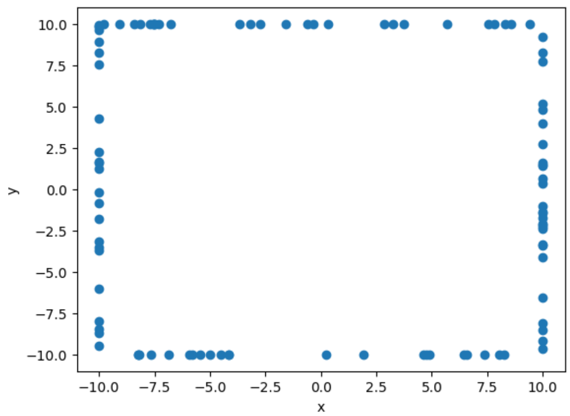
4 Wizualizacja zbiorów



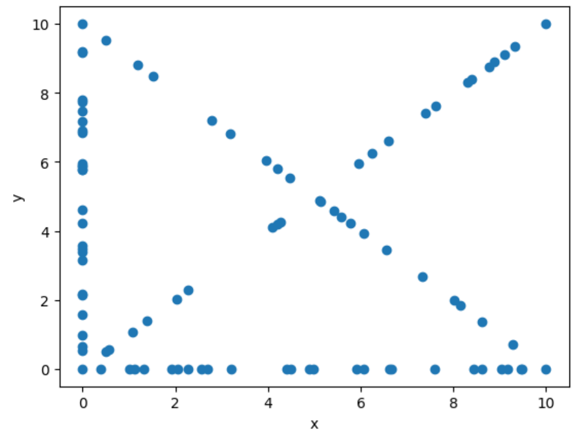
Wykres 1. Zbiór a



Wykres 2. Zbiór b



Wykres 3. Zbiór c



Wykres 4. Zbiór d

5 Wizualizacja działania algorytmów

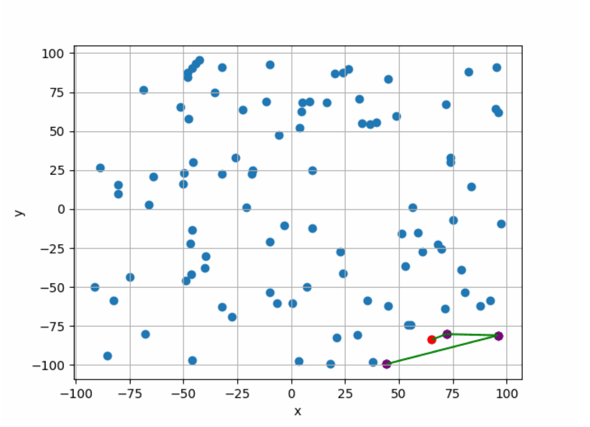
Otoczka została zaznaczona kolorem zielonym.
Legenda kolorów dla algorytmu Grahama:

- Fioletowy - punkt należący do otoczki,
- Czerwony - punkty aktualnie rozpatrywany,
- Czarny - punkt już rozpatrzony.

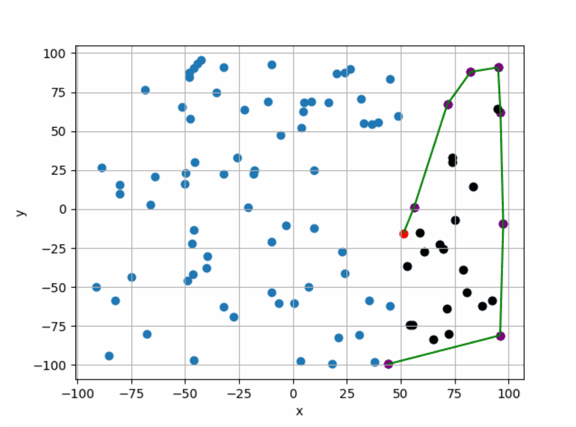
Legenda kolorów dla algorytmu Jarvisa:

- Czerwony - punkt należący do otoczki,
- Fioletowy - odcinek aktualnie rozpoznany jako kolejny do dołączenia do otoczki.

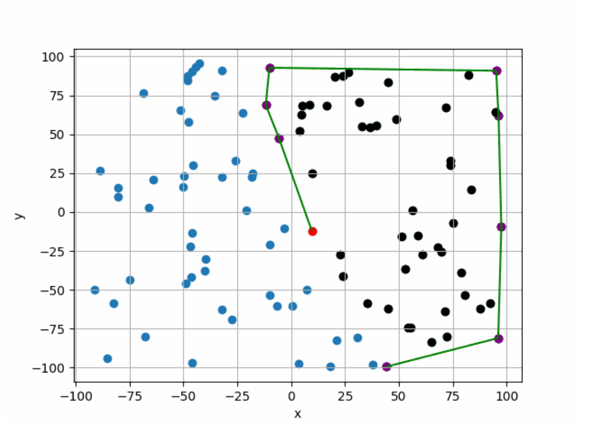
5.1 Algorytm Grahama dla zbioru a



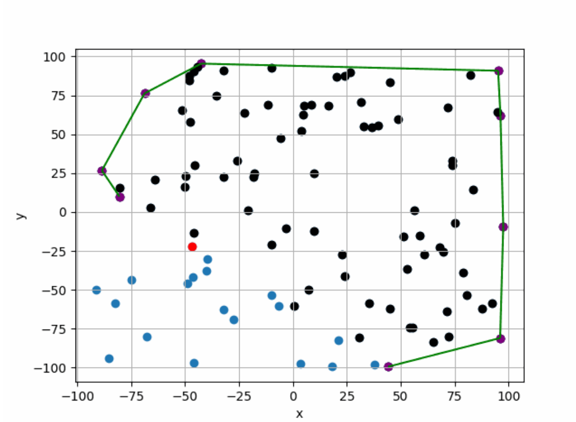
Wykres 5.



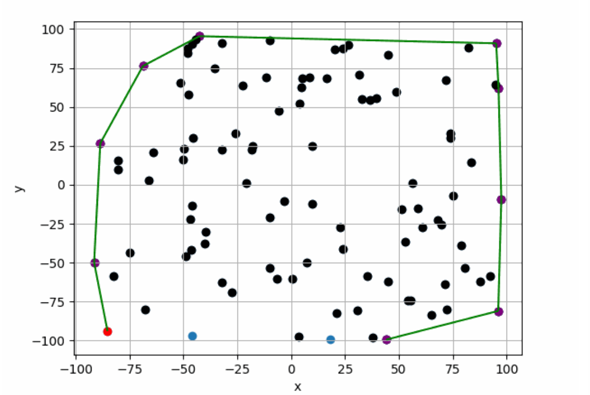
Wykres 6.



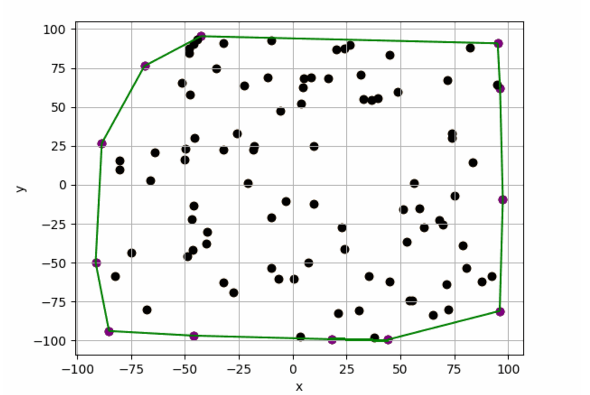
Wykres 7.



Wykres 8.

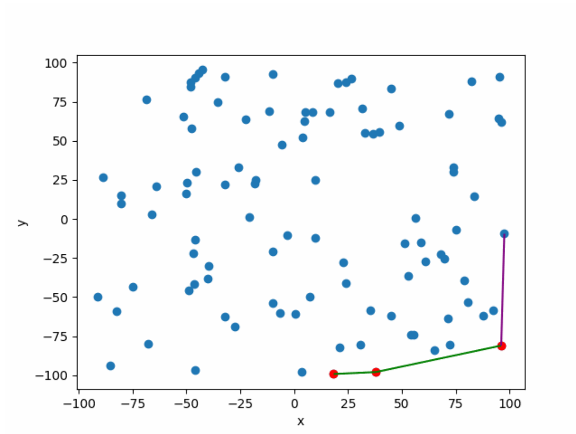


Wykres 9.

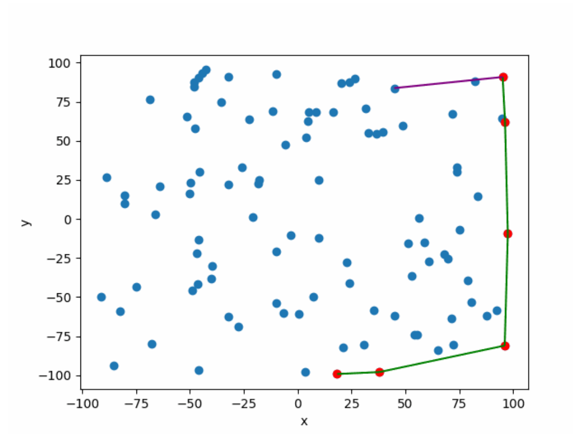


Wykres 10.

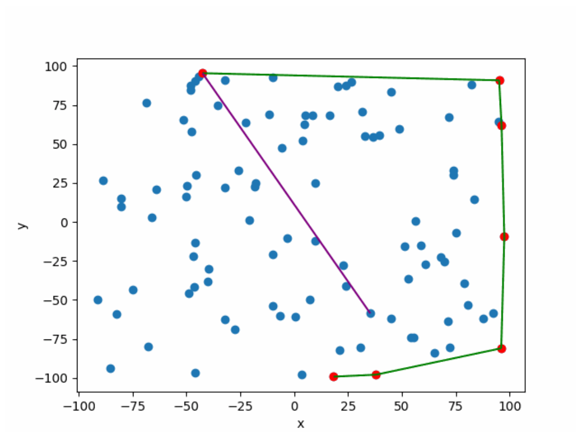
5.2 Algorytm Jarvisa dla zbioru a



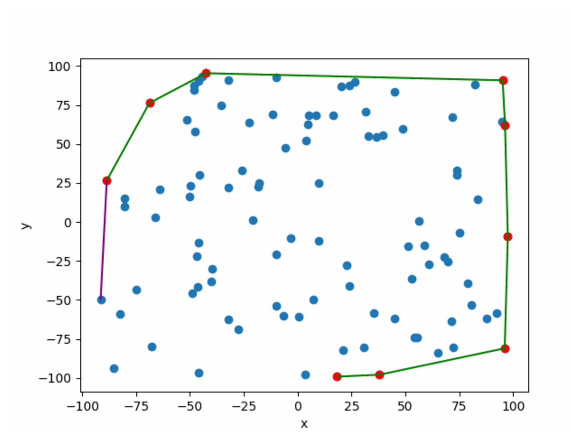
Wykres 11.



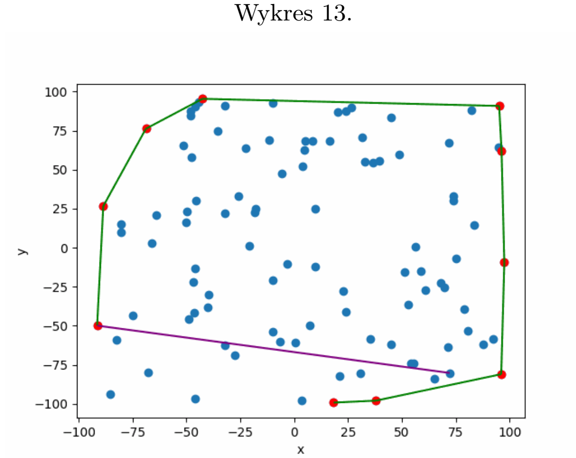
Wykres 12.



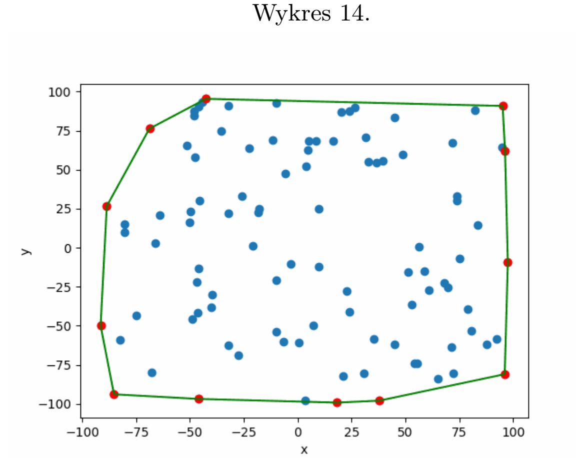
Wykres 13.



Wykres 14.



Wykres 15.



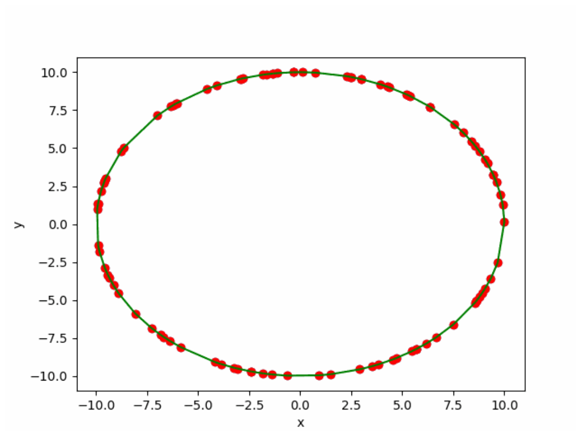
Wykres 16.

5.3 Otoczki dla zbiorów b, c, d

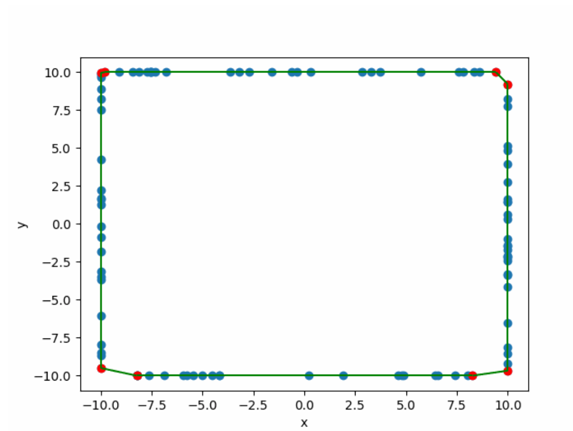
Oba algorytmy wyznaczyły taką samą otoczkę dla zbioru b. Jest to dość specyficzny zbiór, gdyż do otoczki należą wszystkie jego punkty. Algorytm Grahama pokazał tutaj znaczącą przewagę, ale więcej na ten temat w analizie czasu wykonywania algorytmów.

Zbiór c testował współliniowość punktów w otoczce. Oba algorytmy poradziły sobie z nim bardzo dobrze, jednak przewagę miał tutaj algorytm Jarvisa. Jest to spowodowane faktem, że liczba punktów w otoczce jest ograniczona. Teoretycznie jest możliwe, żeby w tym zbiorze znalazł się punkt idealnie na wierzchołku kwadratu, ale praktycznie jest to niemal niemożliwe.W większości przypadków otoczka zawierała będzie 8 punktów. Z perspektywy złożoności obliczeniowej nieważne czy znajdzie się wierzchołek czy nie, złożoność możemy wyznaczyć jako $O(8n)$.

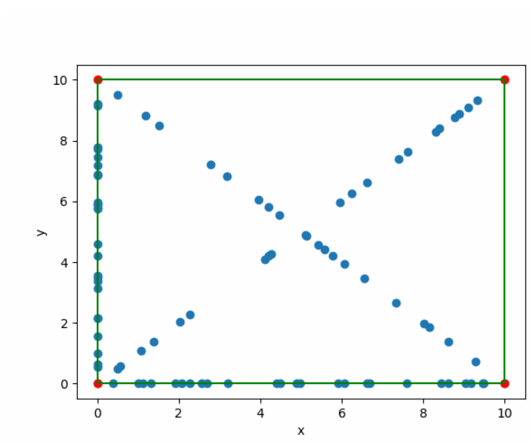
Zbiór d również testował współliniowość, jednak tym razem byliśmy pewni, że w zbiorze znajdą się wierzchołki prostokąta. Algorytm Grahama bardzo dobrze poradził sobie z wyeliminowaniem punktów współliniowych po wcześniejszym sortowaniu (wykres 20).



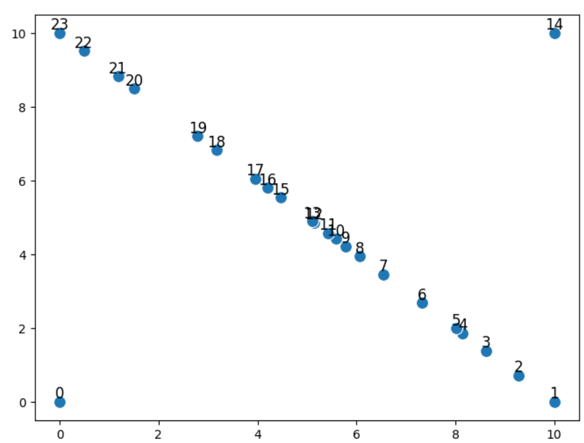
Wykres 17. zbiór b



Wykres 18. zbiór c



Wykres 19. zbiór d



Wykres 20. Zbiór d po usunięciu punktów współliniowych w algorytmie Grahama wraz z indeksem każdego punktu w tablicy.

6 Różnice w czasach wykonywania

W celach przetestowania algorytmów pod względem czasu wykonywania stworzyłem kolejno zmodyfikowane zbiory wyjściowe a, b, c, d. Algorytm Jarvisa poradził sobie lepiej w czterech na trzy przypadki. Może być do dość zadziwiające, ale po wstępnej analizie zbiorów wszystki staje się jasne. W zbiorze a różnice pomiędzy tymi dwoma algorytmami były najmniejsze ze wszystkich. Nie jesteśmy w stanie ustalić liczby otoczki dla tego zbioru, ale z bardzo dużym prawdopodobieństwem możemy stwierdzić, że jest ona dość mała w porównaniu to liczby punktów w całym zbiorze. Dla przykładu w oryginalnym zbiorze a nałożonym odgórnie w ćwiczeniu liczba punktów wynosiła 100, a w otoczce znalazło się 12 punktów. Nie są to oczywiście własności proporcjonalne - dla zbioru a o liczbie punktów 80000 liczba punktów w otoczce wyniosła zaledwie 27. Z racji tego, że ilość punktów w otoczce nie zmienia się tak bardzo (okolice 30), algorytm Jarvisa budował coraz to większą przewagę nad algorytmem Grahama.
W pochodnych zbioru b widać mocną przewagę algorytmu Grahama. Jest to również spowodowane ilością punktów w otoczce. Na okręgu wszystkie punkty należą do otoczki, dlatego też algorytm Jarvisa możemy oszacować na dokładnie $\Theta(n^2)$.
W pochodnych zbioru c oraz d również mogliśmy ograniczyć liczbę punktów w otoczce. W zbiorach d są to 4 punkty, ponieważ wiemy, że wierzchołki prostokąta należa do tego zbioru, a w zbiorach c, tak jak wcześniej wspominałem, jest to 8 punktów. Z tego względu algorytm Jarvisa również miał przewagę czasową w tych zbiorach.
Zbiory te nie zostały wybrane przypadkowo. Dzięki nim możemy zauważyć, że algorytm z gorszą pesymistyczną złożonością obliczeniową może w praktyce okazać się dużo szybszy od innych poprzez proste założenia lub wnioski zastosowane na danych wejściowych.

Zbiór a			
Przedział	Liczba punktów	Graham [s]	Jarvis [s]
[-300,300]	1000	0,011	0,005
	2000	0,024	0,007
	4000	0,068	0,026
	10000	0,130	0,053
	20000	0,278	0,142
	40000	0,619	0,273
	80000	1,300	0,494

Tabela 1

Zbiór b			
Środek i promień okręgu	Liczba punktów	Graham [s]	Jarvis [s]
O(0,0), R=30	1000	0,011	0,264
	2000	0,024	1,177
	4000	0,052	3,765
	10000	0,132	26,254
	20000	0,295	101,123
	40000	0,675	—
	80000	2,444	—

Tabela 2

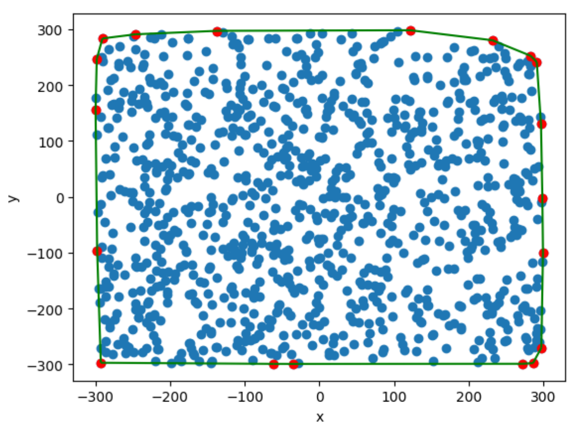
Zbiór c			
Wierzchołki prostokąta	Liczba punktów	Graham [s]	Jarvis [s]
(0,0),(50,0),(50,30),(0,30)	1000	0,00168	0,00034
	2000	0,00065	0,00016
	4000	0,00102	0,00016
	10000	0,00056	0,00017
	20000	0,00055	0,00020
	40000	0,00056	0,00016
	80000	0,00056	0,00017

Tabela 3

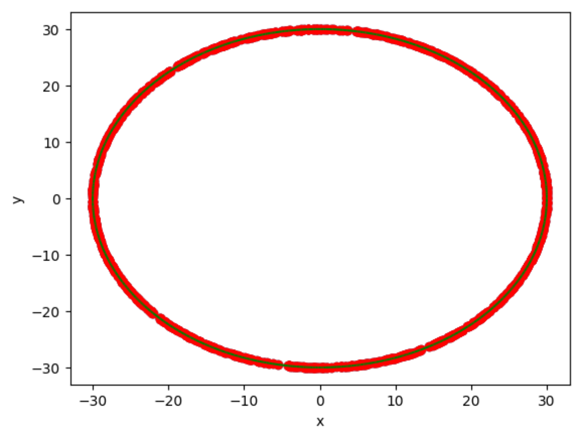
Zbiór d			
Wierzchołki prostokąta	Liczba punktów	Graham [s]	Jarvis [s]
(0,0),(50,0),(50,30),(0,30)	1000	0,01864	0,00005
	2000	0,04653	0,00005
	4000	0,09478	0,00009
	10000	0,24756	0,00021
	20000	0,60162	0,00041
	40000	1,49040	0,00078
	80000	4,61499	0,00162

Tabela 4

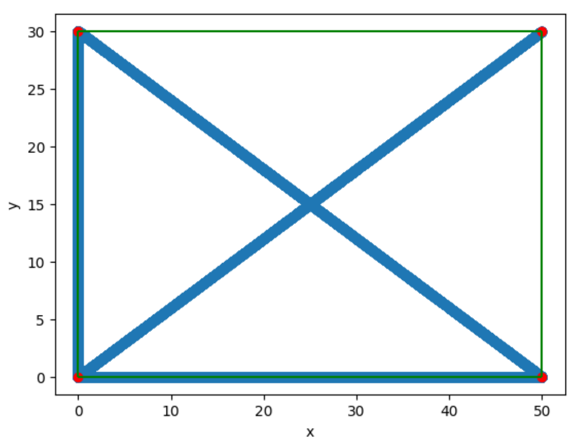
6.1 Wizualizacja otoczek dla zbiorów testowych



Wykres 21. Otoczka dla zbioru a o mocy 1000



Wykres 22. Otoczka dla zbioru b o mocy 1000



Wykres 23. Otoczka dla zbioru c o mocy 1000

7 Wnioski

Po analize czasów wykonywania dla zbiorów testowych trudno jest określić jednoznacznie lepszy algorytm. Jednym z powodów dlaczego algorytm Grahama tak źle poradził sobie w testach może być implementacja. Po wieloktornych próbach optymalizacji tego algorytmu nie udało się poprawić wyników w zbiorze testowym a, w którym w teorii Graham powinien poradzić sobie lepiej niż Jarvis. Wybór algorytmu jest więc trudny, chyba że znalibyśmy ograniczenia górne dla liczby punktów w otoczce. Jeżeli jednak nie mamy żadnych informacji na temat zbioru wejściowego rozsądnym będzie wybrać algorytm Grahama ze względu na lepszą złożoność pesymistyczną.