Algorytmy Geometryczne

Ćwiczenie 2 - Otoczka wypukła

Jakub Własiewicz - Grupa 2 - Poniedziałek 13:00

2025-11-04

1. Dane techniczne

Program został uruchomiony na komputerze z następującymi specyfikacjami:

- **System Operacyjny** - Fedora Linux 43
- **Architektura Procesora** - x86_64
- **Procesor** - AMD Ryzen 7 7840HS
- **Język i wersja kompilatora** - Rust 1.93.0

Oraz użyte zostały następujące biblioteki:

- **config 0.15.18**
- **plotters 0.3.7**
- **rand 0.9.2**
- **serde 1.0.228**

Do obliczeń została użyta tolerancja dla zera $\varepsilon = 10^{-10}$, oraz liczby zmienno-przecinkowe o rozmiarze 64 bitów.

2. Opis ćwiczenia

Ćwiczenie polegało na wyznaczeniu otoczki wypukłej korzystając z algorytmu Grahama i algorytmu Jarvisa, wizualizacji wyników oraz porównania czasów wykonywania dla poniższych zbiorów:

- **Zbiór A** - 100 losowo wygenerowanych punktów o współrzędnych z przedziału $[-100, 100]$
- **Zbiór B** - 100 losowo wygenerowanych punktów leżących na okręgu o środku $(0, 0)$ i promieniu $R = 10$
- **Zbiór C** - 100 losowo wygenerowanych punktów leżących na bokach prostokąta o wierzchołkach $(-10, 10), (-10, -10), (10, -10), (10, 10)$
- **Zbiór D** - zawierający wierzchołki kwadratu $(0, 0), (10, 0), (10, 10), (0, 10)$ oraz po 25 punktów na dwóch bokach kwadratu leżących na osiach i po 20 punktów na przekątnych kwadratu.

3. Realizacja zadania

Oba algorytmy najpierw wyznaczały punkt p_0 najmniejszy względem współrzędnej y oraz najmniejszy względem x jeśli poprzednie współrzędne były równe.

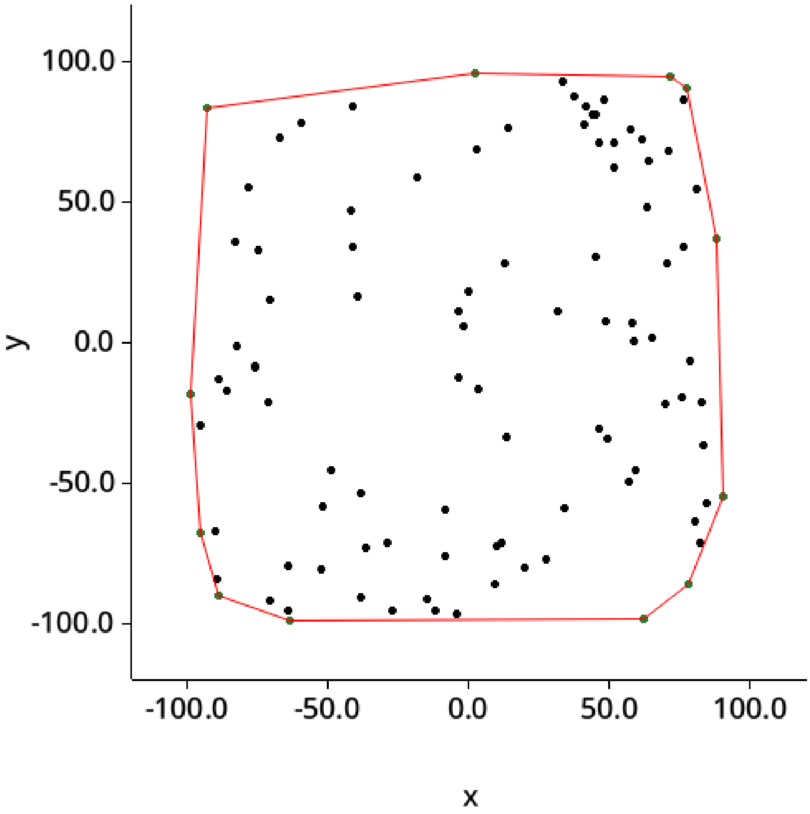
W algorytmie Grahama punkty zostały posortowane za pomocą funkcji bibliotecznej ze względu na kąt tworzący z poziomą osią oraz prostą od p_0 do rozpatrywanego punktu. Do tego celu wykorzystano wyznacznik macierzy 3x3. Z punktów współliniowych zostawiamy tylko ten z największą odległością od p_0.

W algorytmie Jarvisa szukamy punktów których kąt w odniesieniu do ostatniej krawędzi jest najmniejszy. Do tego również został wykorzystany wyznacznik 3x3. W przypadku współliniowości, rozpatrujemy tylko dalszy punkt.

Złożoność czasowa algorytmu Grahama to $O(n \log n)$, a Jarvisa $O(nk)$, gdzie n - liczba punktów w zbiorze, k - liczba punktów otoczki.

Na wszystkich poniższych rysunkach następujące kolory oznaczają:
- **Czarny** - punkty zbioru,
- **Zielony** - punkty zbioru zaliczone jako należące do otoczki
- **Czerwony** - boki otoczki,

3.1. Zbiór A



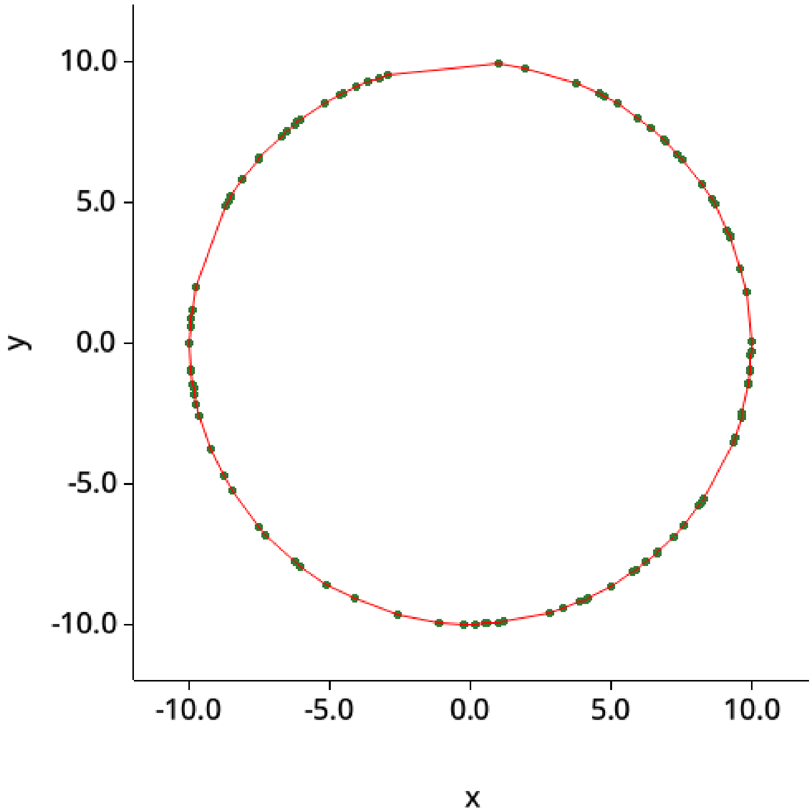
Rysunek 1: Otoczka wypukła dla zbioru A

Liczba punktów zbioru	Czas wykonania algorytmu Grahama [ms]	Czas wykonania algorytmu Jarvisa [ms]
10^2	0.004	0.002
10^3	0.047	0.032
10^4	0.717	0.552
10^5	10.065	5.287

Tabela 1: Czasy wykonania algorytmów dla różnych rozmiarów zbioru A

Algorytm Jarvisa działał szybciej dla wszystkich rozmiarów zbioru A.

3.2. Zbiór B



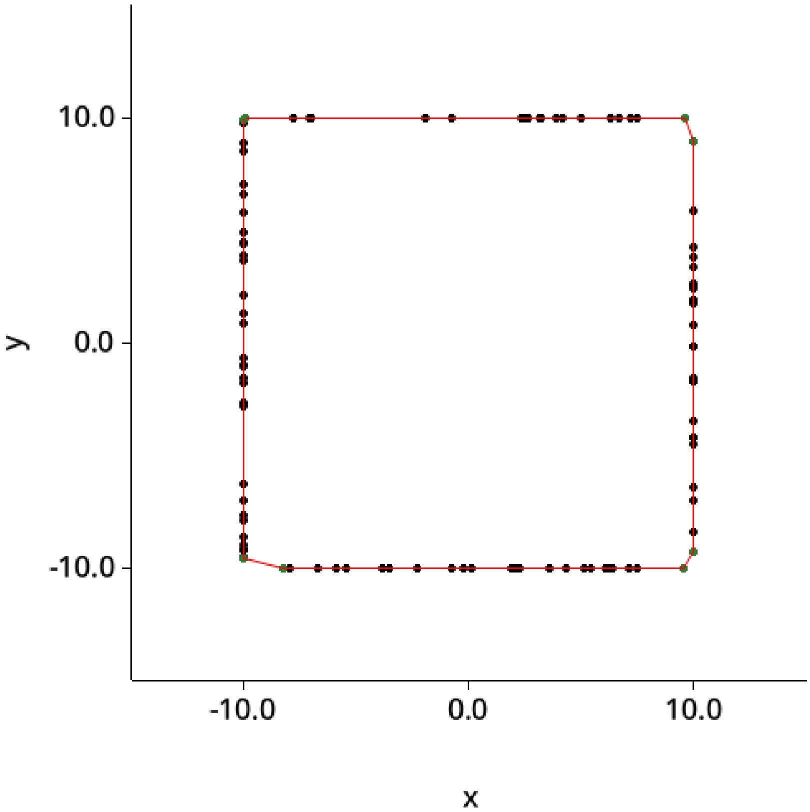
Rysunek 2: Otoczka wypukła dla zbioru B

Liczba punktów zbioru	**Czas wykonania algorytmu Grahama [ms]**	**Czas wykonania algorytmu Jarvisa [ms]**
10^2	0.004	0.017
10^3	0.042	1.750
10^4	0.662	163.230
$2 \cdot 10^4$	1.532	599.935

Tabela 2: Czasy wykonania algorytmów dla różnych rozmiarów zbioru B

Algorytm Jarvisa wyznaczył otoczkę wypukłą w znacznie dłuższym czasie w porównaniu do algorytmu Grahama. Wynika to z tego że wszystkie punkty zbioru należą do otoczki co powoduje kwadratową złożoność czasową.

3.3. Zbiór C



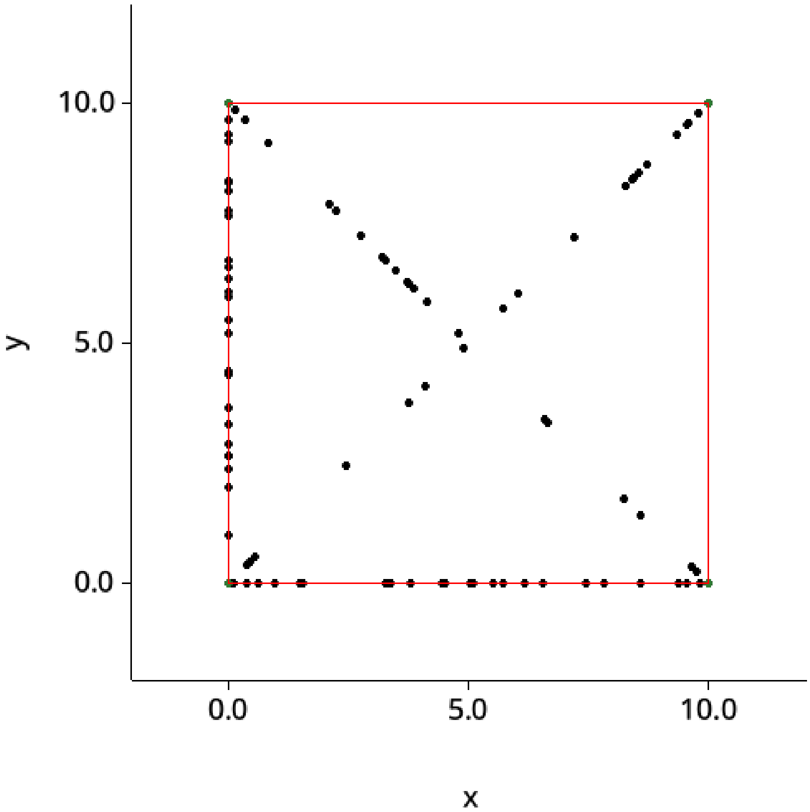
Rysunek 3: Otoczka wypukła dla zbioru C

Liczba punktów zbioru	Czas wykonania algorytmu Grahama [ms]	Czas wykonania algorytmu Jarvisa [ms]
10^2	0.003	0.002
10^3	0.049	0.021
10^4	0.756	0.405
10^5	12.579	4.503

Tabela 3: Czasy wykonania algorytmów dla różnych rozmiarów zbioru C

Tutaj algorytm Jarvisa wykonał się w krótszym czasie, wynika to z tego, że zbiór jest prostokątem bez gwarancji zawierania wierzchołków. W tym zbiorze kluczowym był wybór tolerancji dla zera, ponieważ dla niektórych wartości otoczka była wyznaczana błędnie.

3.4. Zbiór D



Rysunek 4: Otoczka wypukła dla zbioru D

Liczba punktów zbioru	**Czas wykonania algorytmu Grahama [ms]**	**Czas wykonania algorytmu Jarvisa [ms]**
94	0.004	0.001
904	0.065	0.010
9004	0.987	0.098
90004	13.517	1.002

Tabela 4: Czasy wykonania algorytmów dla różnych rozmiarów zbioru D

Każdy ze zbiorów zawiera 4 wierzchołki kwadratu zatem złożoność czasowa algorytmu Jarvisa staje się liniowa.

4. Wnioski

Oba algorytmy poprawnie wyznaczyły punkty otoczki danych zbiorów. Takie zbiory zostały zaproponowane z następujących powodów:

- **Zbiór A** - powszechny przypadek losowego zbioru punktów, pozwala na ogólne sprawdzenie poprawności,
- **Zbiór B** - bardzo specyficzny zbiór, najgorszy przypadek algorytmu Jarvisa,
- **Zbiór C** - duża liczba punktów współliniowych, sprawdza implementację tych przypadków,
- **Zbiór D** - podobnie jak w poprzednim z dodatkiem punktów na przekątnych, algorytm Grahama w tym przypadku sprawdza punkty przekątnej aż do wierzchołka

Algorytm Jarvisa pod względem czasu działania radził sobie znacznie lepiej niż algorytm Grahama, jedyny wyjątek stanowił zbiór B gdzie wszystkie punkty należały do otoczki wypukłej. Wynika to z tego, że każdy ze zbiorów A, C i D w miarę zwiększania liczebności staje się prostokątem. Algorytm Grahama ma lepszą złożoność czasową dla dużych zbiorów z dużą liczbą punktów otoczki, dodatkowym zbiorem mógłby być na przykład prostokąt z zygzakowatymi bokami.