

Optymalizacja wielokryterialna		2024/2025
Informatyka w Sterowaniu i Zarządzaniu	2 rok	Grupa 1
Temat: Ćwiczenie 2 - Optymalizacja wielokryterialna dyskretna		
Skład zespołu: Andrzej Janik Artur Mazurkiewicz		

1. Cel ćwiczenia

Celem ćwiczenia była implementacja i porównanie trzech algorytmów znajdowania punktów niezdominowanych w zbiorze wielowymiarowym. Realizowane algorytmy to:

1. **Algorytm bez filtracji punktów zdominowanych** – podstawowa metoda identyfikacji punktów niezdominowanych.
2. **Algorytm z filtracją punktów zdominowanych** – rozszerzenie podstawowej metody, eliminujące punkty zdominowane po każdej iteracji.
3. **Algorytm oparty o punkt idealny** – metoda wykorzystująca koncepcję punktu idealnego i sortowanie względem odległości od tego punktu.

2. Wstęp teoretyczny

Wielokryterialna optymalizacja dyskretna zajmuje się identyfikowaniem najlepszych rozwiązań w zbiorach punktów, które reprezentują możliwe opcje z uwzględnieniem wielu kryteriów. Punkty niezdominowane (znane również jako punkty Pareto-optymalne) to takie, które nie są w całości gorsze od innych w żadnym kryterium, a w co najmniej jednym są lepsze.

Algorytmy znajdowania punktów niezdominowanych różnią się podejściem do przetwarzania danych:

1. **Algorytm bez filtracji** porównuje każdy punkt ze wszystkimi innymi, prowadząc do złożoności obliczeniowej wprost proporcjonalnej do kwadratu liczby punktów.
2. **Algorytm z filtracją punktów zdominowanych** dodaje krok eliminacji punktów, które są zdominowane po wyznaczeniu każdego punktu niezdominowanego, co zmniejsza liczbę porównań w kolejnych iteracjach.
3. **Algorytm oparty o punkt idealny** definiuje punkt referencyjny (idealny), oblicza odległości od tego punktu dla wszystkich elementów, a następnie iteracyjnie wyznacza punkty niezdominowane, eliminując zdominowane elementy.

Zastosowanie tych metod ma istotne znaczenie w problemach decyzyjnych, takich jak optymalizacja projektów technicznych, alokacja zasobów, czy analiza danych wielokryterialnych. Porównanie tych algorytmów pozwala ocenić ich przydatność w różnych

warunkach oraz określić, w jakich przypadkach stosowanie zaawansowanych technik przynosi korzyści względem prostszych rozwiązań.

3. Aplikacja

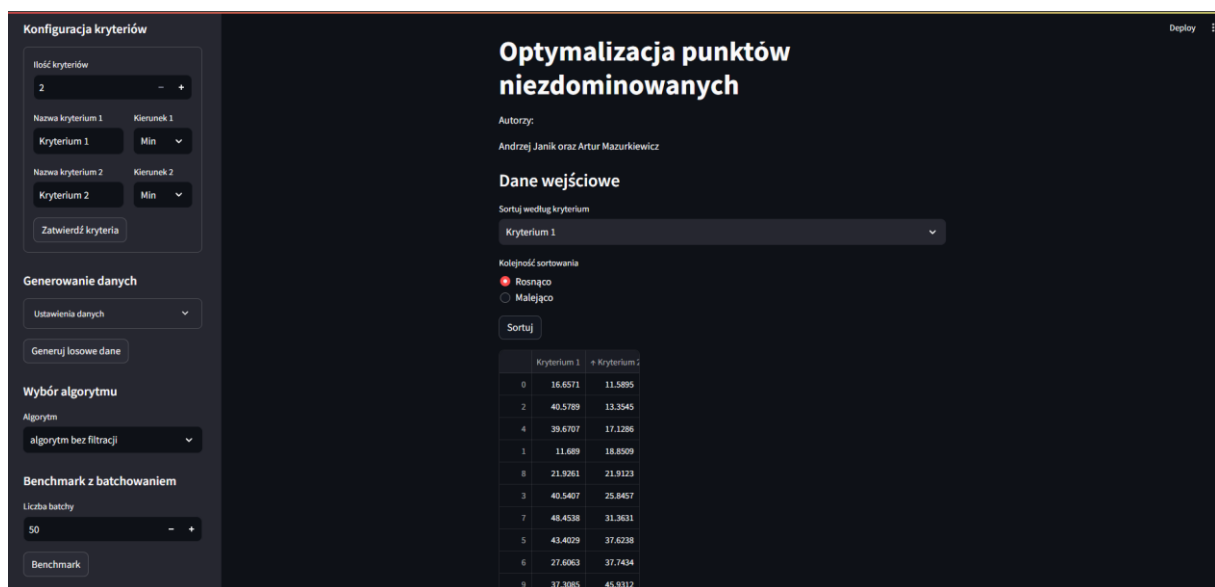
Aby uruchomić aplikację należy pobrać kod w postaci paczki .zip. Następnie należy zainstalować potrzebne biblioteki za pomocą:

```
pip install -r requirements.txt
```

Po udanej instalacji aplikację można uruchomić za pomocą wywołania następującej komendy w terminalu:

```
streamlit run app.py
```

Na Rys. 1. przedstawiono widok aplikacji po uruchomieniu.



Rys. 1. Widok aplikacji po uruchomieniu

Po lewej stronie GUI dostępny jest dostępny konfigurator, który umożliwia:

- Wybór liczby kryteriów oraz ich kierunek.
- Generowanie zestawu danych testowych zgodnie z następującymi rozkładami danych: Jednostajny, Gaussa, Ekspotencjalny, Poissona. Możliwe jest określenie liczby punktów.
- Wybór algorytmu wyznaczającego punkty niezdominowane.
- „Benchmark” czyli uruchomienie symulacji określoną ilość razy. Opcja ta umożliwia zapis wyników do pliku Excel, aby analiza wyników mogła być wygodniejsza.

W głównej części GUI znajdują się wygenerowane dane oraz przycisk do uruchomienia symulacji. Po wykonaniu symulacji prezentowane w tej części są wyniki optymalizacji – wyznaczone punkty niezdominowane, liczba porównań, czas obliczeń oraz wykres punktów w przestrzeni zależnej od liczby kryteriów.

4. Wyznaczanie punktów niezdominowanych

W tej części ćwiczenia postanowiono sprawdzić poprawność działania zaimplementowanych metod. Eksperymenty zostały przeprowadzone dla następujących ustawień konfiguratora:

Ilość kryteriów

2 - +

Nazwa kryterium 1 Kierunek 1

Kryterium 1 Min ▾

Nazwa kryterium 2 Kierunek 2

Kryterium 2 Min ▾

Zatwierdź kryteria

Rys. 2. Kryteria

Generowanie danych

Ustawienia danych ^

Ilość punktów

10 - +

Rozkład danych

Jednostajny ▾

Zakres wartości

10 50

0 100

Rys. 3. Generator danych testowych

	↑ Kryterium 1	Kryterium 2
5	11.3914	27.9208
0	12.0334	37.4386
2	16.555	22.6513
6	19.2995	46.3673
7	23.2945	35.2404
8	28.1424	32.7656
1	29.8158	36.6254
9	32.6969	20.4298
4	36.9576	12.4892
3	40.3487	10.1935

Rys. 4. Wygenerowane dane

Wyniki eksperymentu obliczeniowego dla powyższych ustawień:

- a) Algorytm bez filtracji

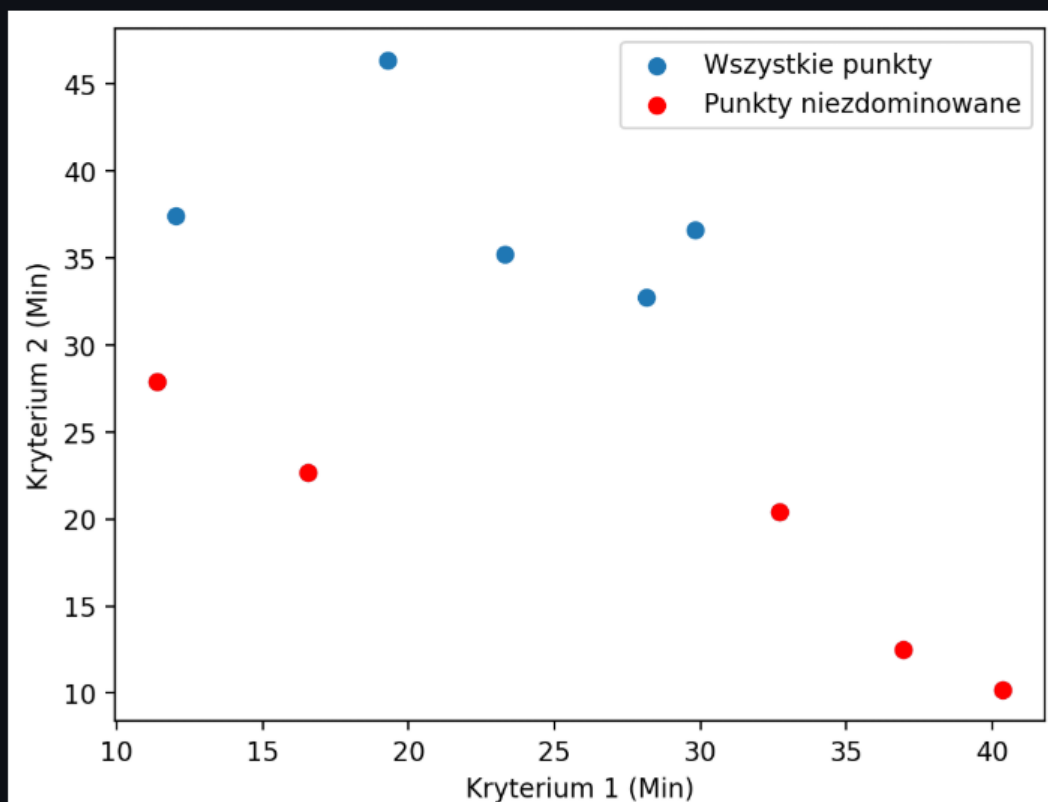
Punkty niezdominowane (Liczba: 5)

1: (16.555044570034397, 22.651347409796294)
2: (11.391415049594658, 27.920805418486687)
3: (32.69687275028819, 20.429755472862468)
4: (36.957634580040775, 12.489168338267994)
5: (40.34867676225315, 10.193459973016976)

Liczba porównań: 60

Czas wykonania algorytmu: 0.0000 sekundy

Wizualizacja wyników



Rys. 5. Algorytm bez filtracji

b) Algorytm z filtracją punktów niezdominowanych

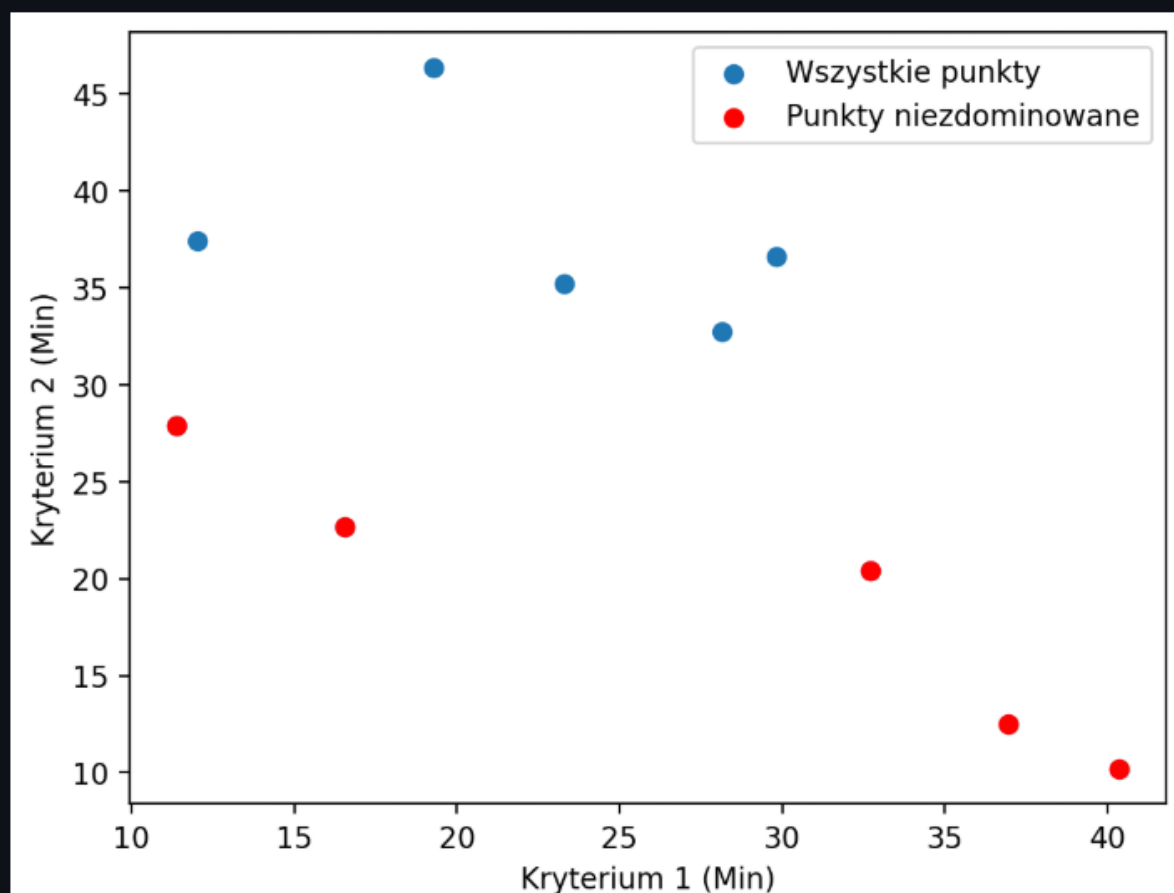
Punkty niezdominowane (Liczba: 5)

1: (11.391415049594658, 27.920805418486687)
2: (16.555044570034397, 22.651347409796294)
3: (40.34867676225315, 10.193459973016976)
4: (36.957634580040775, 12.489168338267994)
5: (32.69687275028819, 20.429755472862468)

Liczba porównań: 30

Czas wykonania algorytmu: 0.0000 sekundy

Wizualizacja wyników



Rys. 5. Algorytm z filtracją punktów niezdominowanych

c) Algorytm oparty o punkt idealny

Punkty niezdominowane (Liczba: 5)

1: (16.555044570034397, 22.651347409796294)

2: (11.391415049594658, 27.920805418486687)

3: (32.69687275028819, 20.429755472862468)

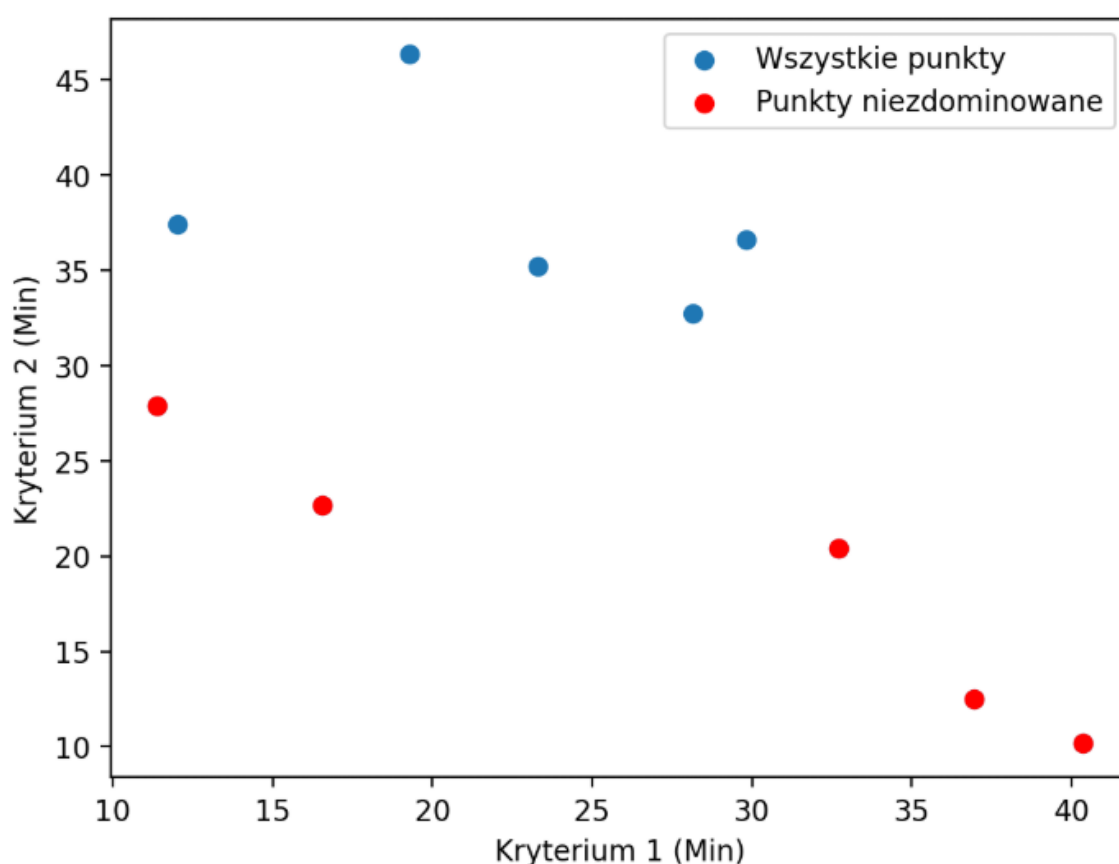
4: (36.957634580040775, 12.489168338267994)

5: (40.34867676225315, 10.193459973016976)

Liczba porównań: 60

Czas wykonania algorytmu: 0.0000 sekundy

Wizualizacja wyników



Rys. 6. Algorytm oparty o punkt idealny

Każdy algorytm zwrócił dokładnie te same punkty, czas obliczeń w każdym przypadku jest znikomy i nie różni się pomiędzy algorytmami. Jedyna zauważalna różnica to mniejsza liczba porównań (30) dla algorytmu z filtracją niż w pozostałych przypadkach (60 porównań).

5. Eksperymenty obliczeniowe

a) Przeprowadzono symulacje, dla $K1 \geq 50$ zbiorów 100 elementowych o liczbie kryteriów 2,3,4,5,10. Zastosowano dwa rozkłady: jednostajny na $[0, 2]$ oraz Gaussa(1,1).

1. Liczba kryteriów – 2, Rozkład jednostajny na [0, 2]:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	9	392.48	0.0001
1	algorytm z filtracją	6	261.8	0.0001
2	algorytm punkt idealny	2	644	0.0002

Rys. 7. Liczba kryteriów – 2, Rozkład jednostajny na [0, 2]:

2. Liczba kryteriów – 3, Rozkład jednostajny na [0, 2]:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	17	1,594.98	0.0003
1	algorytm z filtracją	10	863.52	0.0003
2	algorytm punkt idealny	12	1,604	0.0005

Rys. 8. Liczba kryteriów – 3, Rozkład jednostajny na [0, 2]:

3. Liczba kryteriów – 4, Rozkład jednostajny na [0, 2]:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	25	4,644.64	0.0008
1	algorytm z filtracją	25	3,051.76	0.0011
2	algorytm punkt idealny	23	3,054	0.0009

Rys. 9. Liczba kryteriów – 4, Rozkład jednostajny na [0, 2]:

4. Liczba kryteriów – 5, Rozkład jednostajny na $[0, 2]$:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	53	10,312.4	0.0016
1	algorytm z filtracją	26	6,325.7	0.0019
2	algorytm punkt idealny	47	4,826	0.0015

Rys. 10. Liczba kryteriów – 5, Rozkład jednostajny na $[0, 2]$:

5. Liczba kryteriów – 10, Rozkład jednostajny na $[0, 2]$:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	98	46,834.6	0.0053
1	algorytm z filtracją	92	44,830.2	0.0089
2	algorytm punkt idealny	95	10,274	0.0051

Rys. 11. Liczba kryteriów – 10, Rozkład jednostajny na $[0, 2]$:

6. Liczba kryteriów – 2, Rozkład Gaussa(1,1):

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	1	226.32	0.0001
1	algorytm z filtracją	1	198	0
2	algorytm punkt idealny	1	218	0.0002

Rys. 12. Liczba kryteriów – 2, Rozkład Gaussa(1,1):

7. Liczba kryteriów – 3, Rozkład Gaussa(1,1):

Wyniki benchmarku

	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	4	696.24	0.0002
1	algorytm z filtracją	12	446.76	0.0002
2	algorytm punkt idealny	1	662	0.0003

Rys. 13. Liczba kryteriów – 3, Rozkład Gaussa(1,1):

8. Liczba kryteriów – 4, Rozkład Gaussa(1,1):

Wyniki benchmarku

	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	33	2,869.76	0.0006
1	algorytm z filtracją	20	1,382.4	0.0005
2	algorytm punkt idealny	22	1,734	0.0007

Rys. 14. Liczba kryteriów – 4, Rozkład Gaussa(1,1):

9. Liczba kryteriów – 5, Rozkład Gaussa(1,1):

Wyniki benchmarku

	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	27	7,240.7	0.0013
1	algorytm z filtracją	29	4,138.7	0.0014
2	algorytm punkt idealny	39	3,494	0.0013

Rys. 15. Liczba kryteriów – 5, Rozkład Gaussa(1,1):

10. Liczba kryteriów – 10, Rozkład Gaussa(1,1):

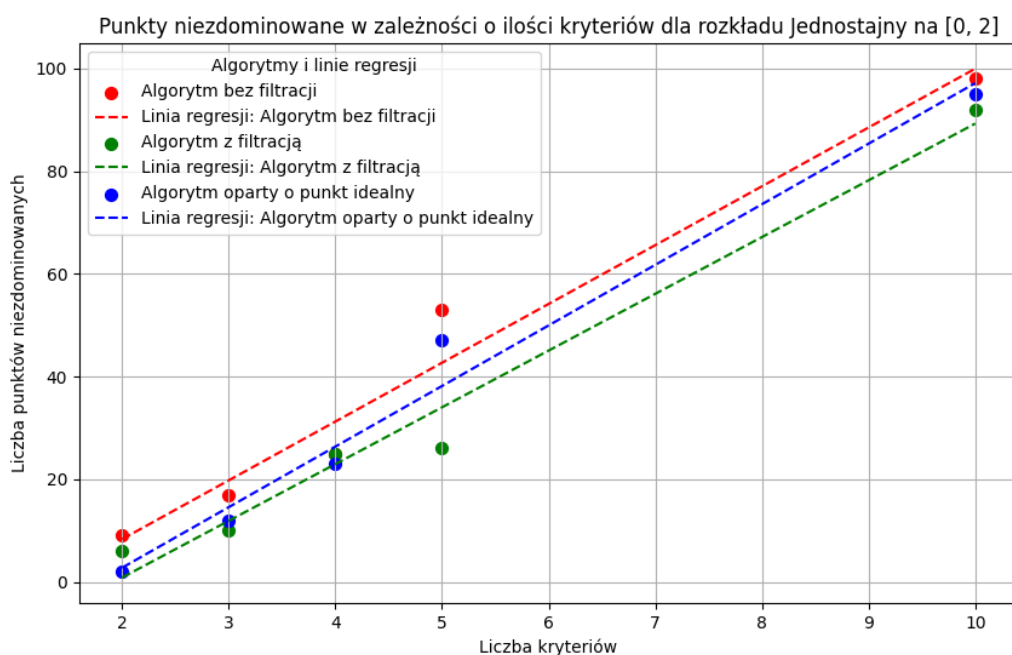
Wyniki benchmarku

	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	98	45,361.2	0.0058
1	algorytm z filtracją	93	41,411.2	0.0085
2	algorytm punkt idealny	81	9,930	0.0046

Rys. 16. Liczba kryteriów – 10, Rozkład Gaussa(1,1):

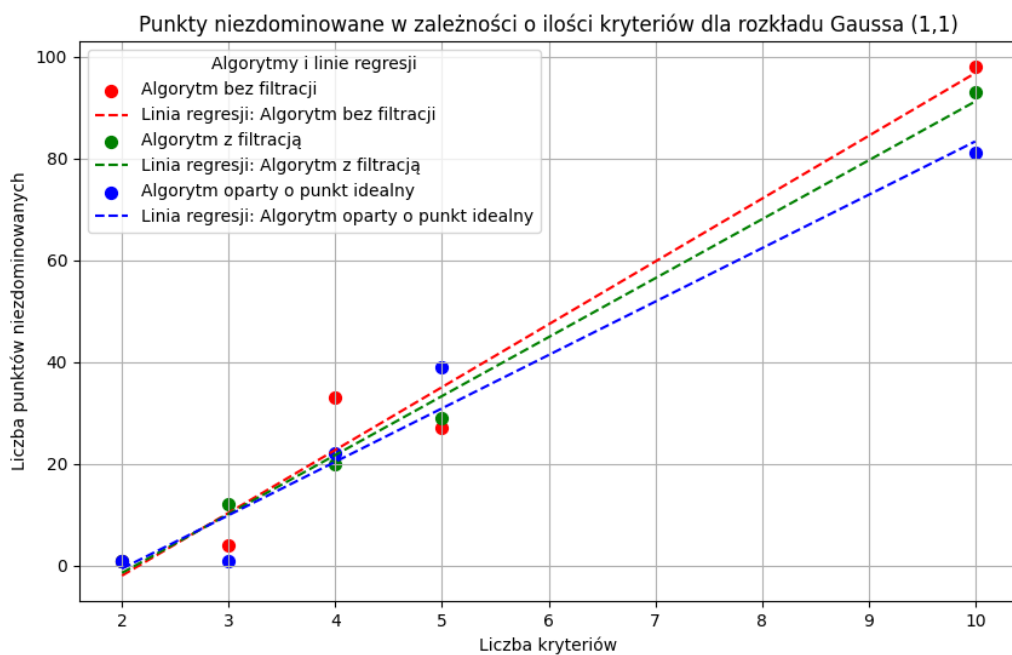
Porównanie ilości punktów niezdominowanych w zależności od liczby kryteriów:

a) rozkład Jednostajny na $[0, 2]$



Rys. 17. Porównanie ilości punktów niezdominowanych w zależności od liczby kryteriów dla rozkładu Jednostajnego na $[0, 2]$

b) rozkład Gaussa $(1,1)$



Rys. 18. Porównanie ilości punktów niezdominowanych w zależności od liczby kryteriów dla rozkładu Gaussa $(1,1)$

b) Przeprowadzono symulacje, dla $K2 \geq 25$ zbiorów 1000 elementowych o liczbie kryteriów 2,3,4,5,10. Zastosowano dwa rozkłady: jednostajny na $[0, 2]$ oraz Gaussa(1,1).

1. Liczba kryteriów – 2, Rozkład jednostajny na $[0, 2]$:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	9	2,739.12	0.0008
1	algorytm z filtracją	6	2,317.92	0.0009
2	algorytm punkt idealny	19	9,080	0.0028

Rys. 19. Liczba kryteriów – 2, Rozkład jednostajny na $[0, 2]$:

2. Liczba kryteriów – 3, Rozkład jednostajny na $[0, 2]$:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	20	13,192.8	0.0027
1	algorytm z filtracją	22	6,367.68	0.0025
2	algorytm punkt idealny	30	31,840	0.0062

Rys. 20. Liczba kryteriów – 3, Rozkład jednostajny na $[0, 2]$:

3. Liczba kryteriów – 4, Rozkład jednostajny na $[0, 2]$:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	73	65,030.08	0.0111
1	algorytm z filtracją	82	27,457.6	0.0095
2	algorytm punkt idealny	56	79,440	0.0129

Rys. 21. Liczba kryteriów – 4, Rozkład jednostajny na $[0, 2]$:

4. Liczba kryteriów – 5, Rozkład jednostajny na $[0, 2]$:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	105	220,524	0.035
1	algorytm z filtracją	124	93,702.4	0.0274
2	algorytm punkt idealny	157	160,080	0.0276

Rys. 22. Liczba kryteriów – 5, Rozkład jednostajny na $[0, 2]$:

5. Liczba kryteriów – 10, Rozkład jednostajny na $[0, 2]$:

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	737	3,687,506.4	0.4717
1	algorytm z filtracją	715	3,107,984.8	0.6361
2	algorytm punkt idealny	741	769,520	0.3308

Rys. 23. Liczba kryteriów – 10, Rozkład jednostajny na $[0, 2]$:

6. Liczba kryteriów – 2, Rozkład Gaussa(1,1):

Wyniki benchmarku				
	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	1	2,033.12	0.0009
1	algorytm z filtracją	1	1,998	0.0006
2	algorytm punkt idealny	1	2,000	0.0026

Rys. 24. Liczba kryteriów – 2, Rozkład Gaussa(1,1):

7. Liczba kryteriów – 3, Rozkład Gaussa(1,1):

Wyniki benchmarku

	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	1	3,516.6	0.0008
1	algorytm z filtracją	1	2,997	0.0008
2	algorytm punkt idealny	1	3,000	0.0024

Rys. 25. Liczba kryteriów – 3, Rozkład Gaussa(1,1):

8. Liczba kryteriów – 4, Rozkład Gaussa(1,1):

Wyniki benchmarku

	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	1	12,643.84	0.0023
1	algorytm z filtracją	15	5,914.72	0.0016
2	algorytm punkt idealny	1	8,720	0.0035

Rys. 26. Liczba kryteriów – 4, Rozkład Gaussa(1,1):

9. Liczba kryteriów – 5, Rozkład Gaussa(1,1):

Wyniki benchmarku

	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	48	56,024	0.0086
1	algorytm z filtracją	1	16,873.4	0.0047
2	algorytm punkt idealny	45	39,160	0.0078

Rys. 27. Liczba kryteriów – 5, Rozkład Gaussa(1,1):

10. Liczba kryteriów – 10, Rozkład Gaussa(1,1):

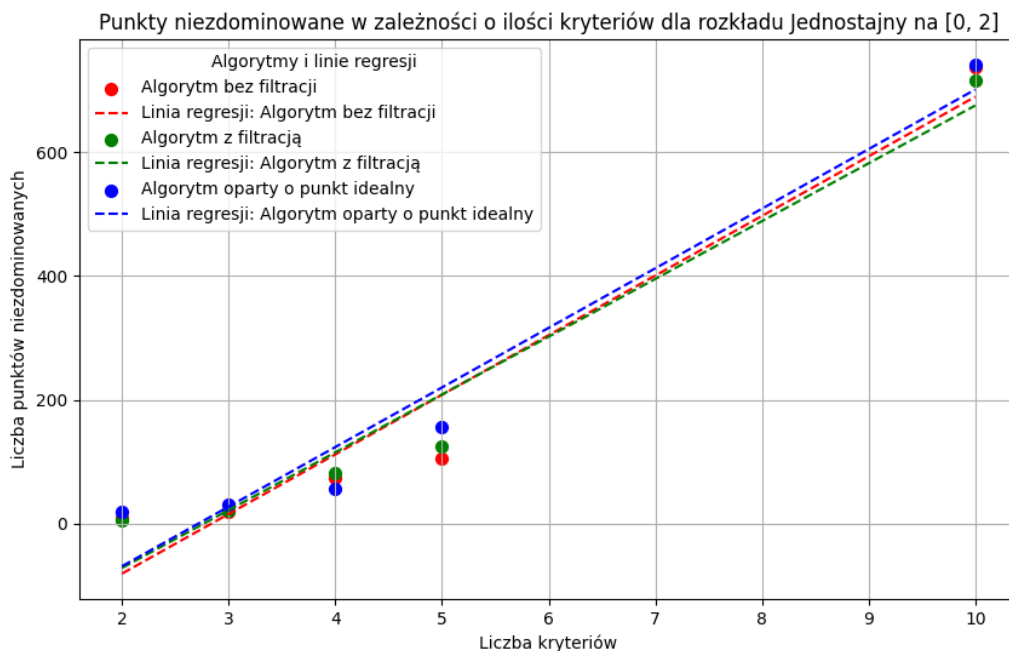
Wyniki benchmarku

	Algorytm	Punkty niezdominowane	Liczba porównań (średnia)	Czas wykonania (s)
0	algorytm bez filtracji	449	2,909,608.8	0.3603
1	algorytm z filtracją	605	2,211,847.6	0.4722
2	algorytm punkt idealny	622	638,520	0.2427

Rys. 28. Liczba kryteriów – 10, Rozkład Gaussa(1,1):

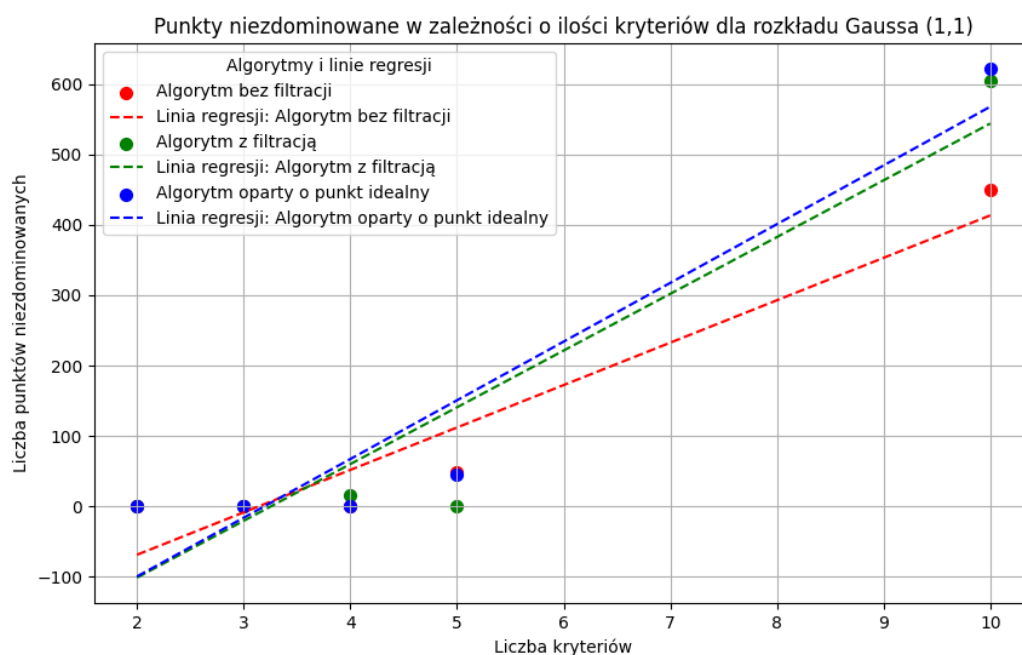
Porównanie ilości punktów niezdominowanych w zależności od liczby kryteriów:

a) rozkład Jednostajny na $[0, 2]$



Rys. 29. Porównanie ilości punktów niezdominowanych w zależności od liczby kryteriów dla rozkładu Jednostajnego na $[0, 2]$

b) rozkład Gaussa $(1,1)$



Rys. 30. Porównanie ilości punktów niezdominowanych w zależności od liczby kryteriów dla rozkładu Gaussa $(1,1)$

6. Podsumowanie oraz wnioski

W trakcie eksperymentów przeprowadzono symulacje dla zbiorów o różnych rozkładach punktów (jednostajny i Gaussa), różnej liczbie kryteriów (od 2 do 10) oraz na danych pochodzących z dwóch serii eksperymentalnych (50 i 25 symulacji zbiorów po 100 punktów każda). Wyniki przedstawiają liczbę punktów niezdominowanych w zależności od liczby kryteriów i zastosowanego algorytmu.

Wyniki i obserwacje

1. Wpływ liczby kryteriów na liczbę punktów niezdominowanych:

- Wraz ze wzrostem liczby kryteriów liczba punktów niezdominowanych rośnie w obu rozkładach (jednostajnym i Gaussa). Jest to zgodne z intuicją – większa liczba wymiarów powoduje, że trudniej jest znaleźć punkty zdominowane, co zwiększa liczbę punktów należących do Pareto-frontu.
- Efekt ten jest szczególnie wyraźny w przypadku rozkładu jednostajnego, gdzie dla wysokiej liczby kryteriów (np. 10) praktycznie wszystkie punkty stają się niezdominowane.

2. Porównanie algorytmów:

- **Algorytm bez filtracji punktów zdominowanych:** Jego prostota przekłada się na większe obciążenie obliczeniowe, zwłaszcza przy dużych zbiorach i liczbie kryteriów. Wyniki są jednak poprawne, co potwierdza jego przydatność w sytuacjach, gdzie zasoby obliczeniowe nie są krytyczne.
- **Algorytm z filtracją punktów zdominowanych:** Jest bardziej efektywny od wersji bez filtracji, eliminując punkty zdominowane w trakcie działania. Dzięki temu redukuje liczbę porównań w późniejszych iteracjach, co czyni go bardziej skalowalnym przy dużych zbiorach danych.
- **Algorytm oparty o punkt idealny:** Sprawdził się najlepiej w obu rozkładach, szczególnie w rozkładzie Gaussa. Wykorzystanie punktu idealnego pozwala szybko odrzucić punkty zdominowane, co czyni go efektywnym zarówno pod względem liczby operacji, jak i czasu obliczeń.

3. Wpływ rozkładu punktów:

- W rozkładzie jednostajnym punkty są równomiernie rozmieszczone, co powoduje większą liczbę punktów niezdominowanych przy większej liczbie kryteriów.
- W rozkładzie Gaussa (bardziej skupionym) liczba punktów niezdominowanych jest znacznie mniejsza przy niskiej liczbie kryteriów, ale rośnie stopniowo wraz z ich wzrostem.

Wnioski

1. Wybór algorytmu powinien być dostosowany do problemu:

- Algorytm bez filtracji jest dobry w prostych przypadkach (np. mała liczba kryteriów i punktów).
 - Algorytm z filtracją jest bardziej uniwersalny i lepszy przy większej liczbie danych.
 - Algorytm oparty o punkt idealny jest najbardziej efektywny i powinien być stosowany w sytuacjach wymagających dużej skalowalności.
2. Liczba kryteriów ma kluczowy wpływ na wynik – w przypadku wysokiej liczby kryteriów liczba punktów niezdominowanych może stanowić większość analizowanego zbioru.