



Porównanie Algorytmów Minimalizacji Stochastycznej Pure-Random-Search i Multi-Start

Radosław Rolka, Mateusz Kochelski
Rachunek Prawdopodobieństwa i Statystyka
Informatyka WI AGH, II rok

styczeń 2023

Spis treści

I. Wstęp	4
1 Wybrane Algorytmy Minimalizacji	4
1.1 Pure-Random-Search	4
1.2 Multi-Start	4
2 Wybrane Funkcje	4
2.1 Alpine01	4
2.2 funkcja Ackley'a	5
II. Metodologia	6
1 Sposób wykonania	6
2 Kod źródłowy	6
2.1 Pure-Random-Search	6
2.2 Multi-Start	6
2.3 Porównanie metod	7
III. Porównanie	8
1 dim = 2	8
1.1 Wykresy Alpine01	8
1.1.1 Pure-Random-Search	8
1.1.2 Multi-Start	9
1.1.3 Obserwacje	10
1.2 Wykresy funkcji Ackley'a	11
1.2.1 Pure-Random-Search	11
1.2.2 Multi-Start	12
1.2.3 Obserwacje	13
2 dim = 10	14
2.1 Wykresy Alpine01	14
2.1.1 Pure-Random-Search	14
2.1.2 Multi-Start	15
2.1.3 Obserwacje	16
2.2 Wykresy funkcji Ackley'a	17
2.2.1 Pure-Random-Search	17
2.2.2 Multi-Start	18
2.2.3 Obserwacje	19
3 dim = 20	20
3.1 Wykresy Alpine01	20
3.1.1 Pure-Random-Search	20
3.1.2 Multi-Start	21
3.1.3 Obserwacje	22
3.2 Wykresy funkcji Ackley'a	23
3.2.1 Pure-Random-Search	23
3.2.2 Multi-Start	24
3.2.3 Obserwacje	25

IV. Podsumowanie	26
1 Podsumowanie	26
2 Bibliografia	27

I. Wstęp

1 Wybrane Algorytmy Minimalizacji

1.1 Pure-Random-Search

Losujemy po kolei zadaną z góry liczbę punktów z rozkładem jednostajnym w zadanej dziedzinie. Każdy wylosowany punkt porównujemy z aktualnie zapamiętanym minimum i jeśli wartość minimalizowanej funkcji w tym punkcie jest mniejsza, to ten punkt zapamiętujemy jako aktualny punkt minimalny. Wartość funkcji w ostatnim zapamiętanym punkcie stanowi wynik algorytmu.

1.2 Multi-Start

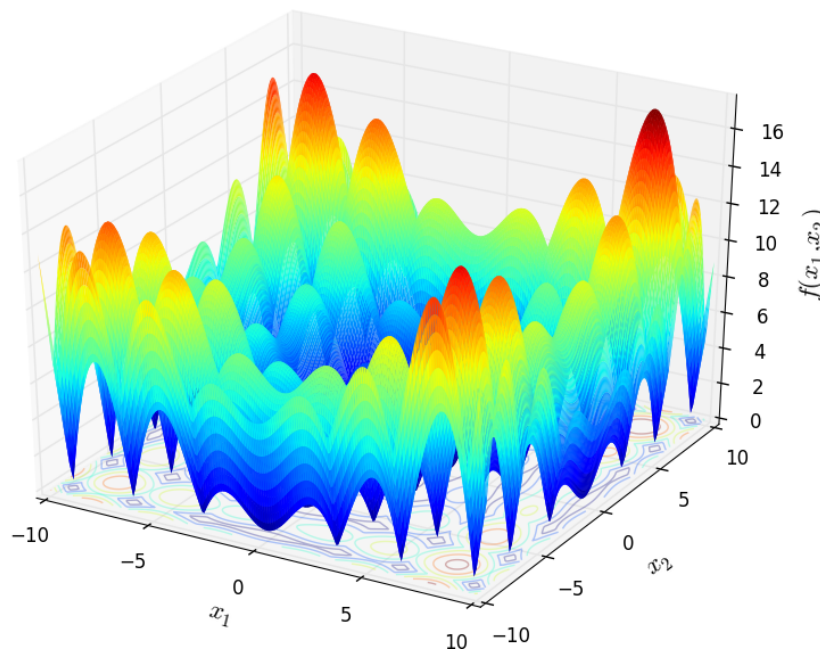
Losujemy (podobnie jak poprzednio) zadaną liczbę punktów z rozkładem jednostajnym w dziedzinie przeszukiwania, a następnie z każdego z wylosowanych punktów startujemy numeryczną metodą optymalizacji lokalnej, z wykorzystaniem metody L-BFGS-B. Wynikiem algorytmu jest wartość optymalizowanej funkcji w tym z punktów zwróconych przez uruchomienia metody lokalnej, w którym ta wartość jest najmniejsza.

2 Wybrane Funkcje

2.1 Alpine01

Alpine01 jest funkcją wielowymiarową, która dla n wymiarowego wektora $x = (x_1, \dots, x_n) \in [0; 10]^n$ zwraca wartość:

$$f(x) = \sum_{i=1}^n |x_i * \sin(x_i) + 0.1 * x_i|$$

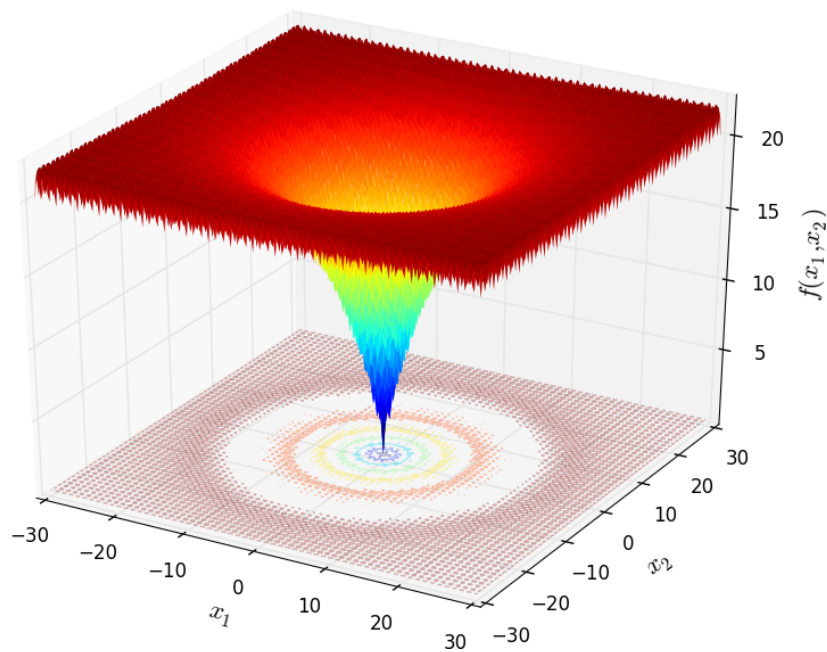


Rysunek 1: Wykres funkcji Alpine01 dla 2 wymiarów.

2.2 funkcja Ackley'a

Funkcja Ackley'a jest funkcją wielowymiarową, która dla n wymiarowego wektora $x = (x_1, \dots, x_n)$ zwraca wartość:

$$f(x) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)} + 20 + e$$



Rysunek 2: Wykres funkcji Ackley'a dla 2 wymiarów.

II. Metodologia

1 Sposób wykonania

Porównania zostały wykonane dla obu metod w przypadkach, gdy liczba wymiarów wynosiła:

- $dim = 2$
- $dim = 10$
- $dim = 20$

Dla każdego przypadku metoda została wykonana 50 razy. Ze względu na wielokrotne uruchamianie MS, wynikające z obliczania gradientu, Koszt wykonania będzie wyższy niż PRS. Aby zachować zrównoważony budżet wykorzystamy średni koszt MS wszystkich startów i proporcjonalnie zwiększymy koszt PRS.

Analizę statystycznej istotności różnicy między średnimi wynikami algorytmów dokonujemy poprzez test t Studenta dla dwóch prób. Robimy tak ze względu na dużą ilość prób, przez co możemy zakładać że rozkład średniej jest rozkładem normalnym.

2 Kod źródłowy

2.1 Pure-Random-Search

```

1 PRS = function(func,dim,random_trials) {
2   f = func(dim)
3   result = Inf
4   RandomPoints = replicate(random_trials,
5                             runif(dim,
6                                   getLowerBoxConstraints(f),
7                                   getUpperBoxConstraints(f)))
8   for(i in 1:random_trials) {
9     result=min(result,f(RandomPoints[,i]))
10  }
11  return (result)
12 }
```

Kod źródłowy 1: Funkcja wykorzystująca metodę PRS.

2.2 Multi-Start

```

1 MS_Minimalize <- function(func, dim, points_quantity) {
2   f <- func(dim)
3   values <- c(1:points_quantity)
4   counter <- c(1:points_quantity)
5   MS <- replicate(n = points_quantity, optim(runif(dim,
6                                                 getLowerBoxConstraints(f),
7                                                 getUpperBoxConstraints(f)),
8                                             f,
9                                             method = "L-BFGS-B",
10                                            lower = getLowerBoxConstraints(f),
11                                            upper = getUpperBoxConstraints(f))
12 )
13 for (i in 1:points_quantity) {
14   values[i] = MS[[2, i]]
15 }
```

```
15     counter[i] = MS[[3, i]][2]
16   }
17   return (list(min(values), mean(counter)))
18 }
```

Kod źródłowy 2: Funkcja wykorzystująca metodę MS.

2.3 Porównanie metod

```
1 compare_MS_to_PRS = function(dim,func,numberOfRuns,NumberOfPoints) {
2   MS_resultsCounter = replicate(numberOfRuns,MS_Minimalize(func,dim,NumberOfPoints))
3   MS_Counter = mean(as.numeric(MS_resultsCounter[2,]))
4   MS_results = as.numeric(MS_resultsCounter[1,])
5   PRS_results = replicate(numberOfRuns,PRS(func,dim,NumberOfPoints*MS_Counter))
6   print(t.test(MS_results,PRS_results,conf.level=0.95))
7 }
```

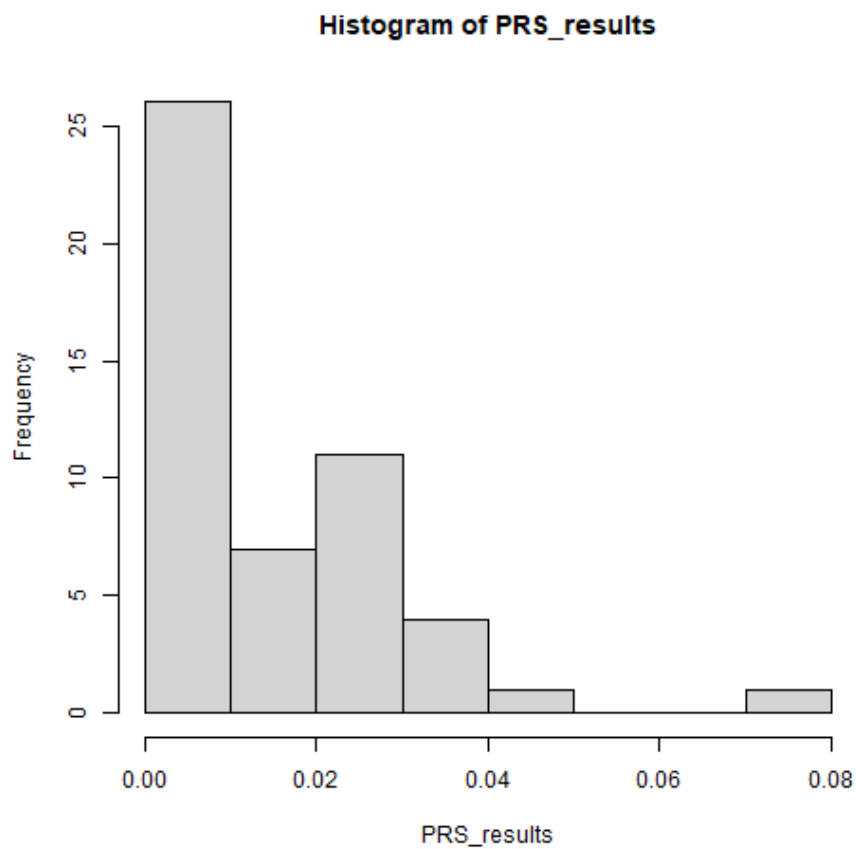
Kod źródłowy 3: Funkcja wykorzystująca t-testy.

III. Porównanie

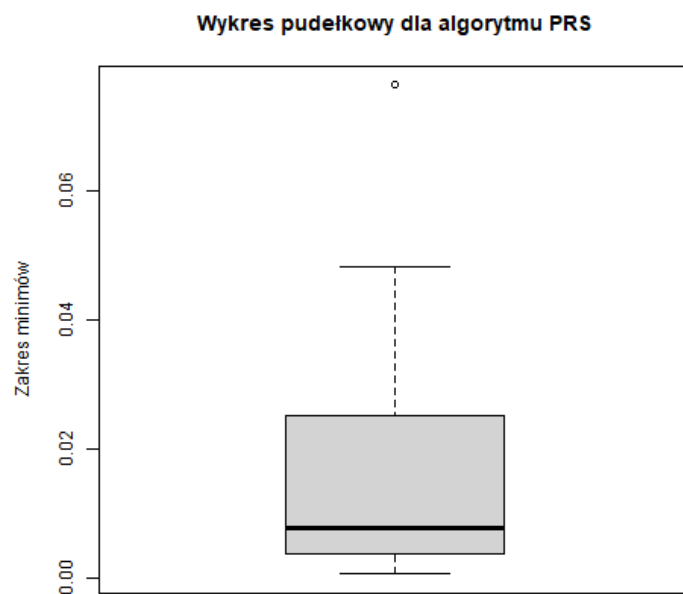
1 $\dim = 2$

1.1 Wykresy Alpine01

1.1.1 Pure-Random-Search

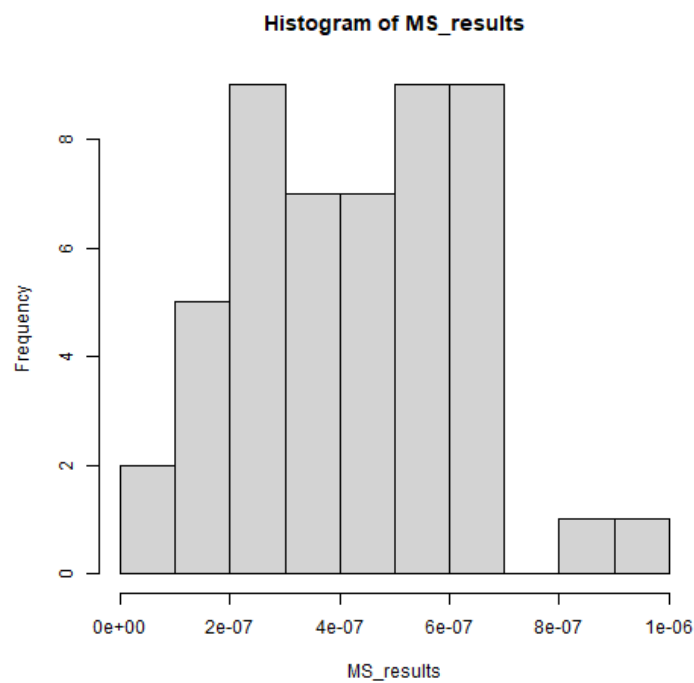


Rysunek 3: Histogram funkcji Alpine01, metody PRS, dla 2 wymiarów.

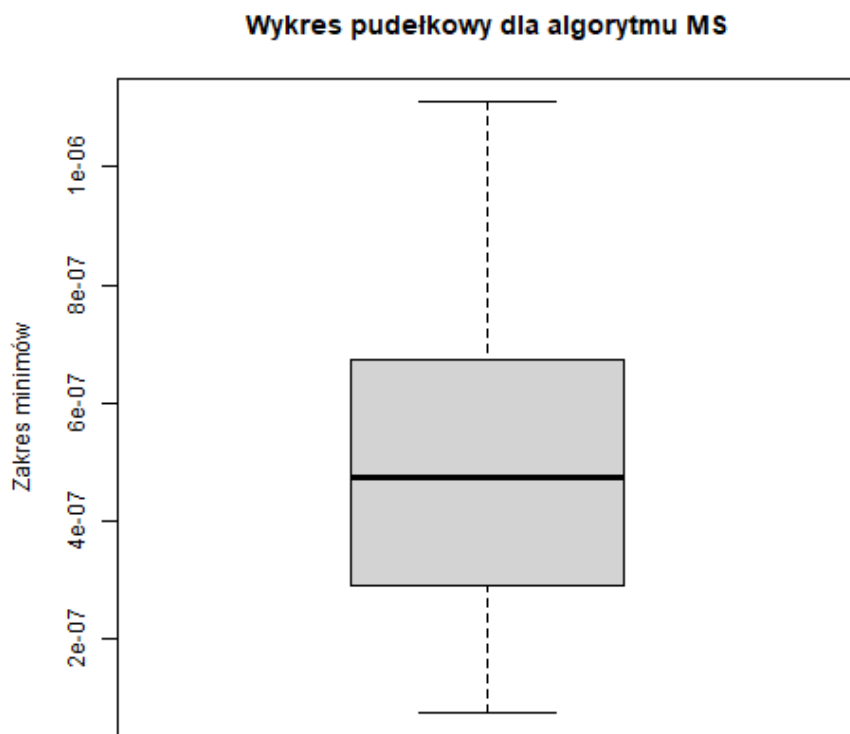


Rysunek 4: Wykres pudełkowy funkcji Alpine01, metody PRS, dla 2 wymiarów.

1.1.2 Multi-Start



Rysunek 5: Histogram funkcji Alpine01, metody MS, dla 2 wymiarów.



Rysunek 6: Wykres pudełkowy funkcji Alpine01, metody MS, dla 2 wymiarów.

1.1.3 Obserwacje

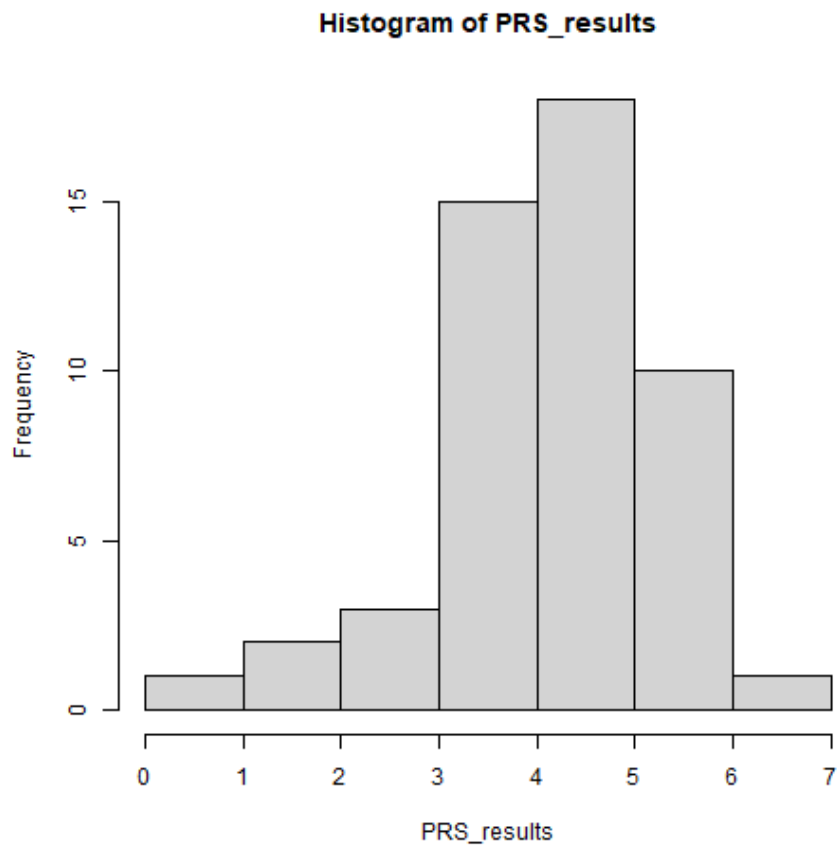
```
Welch Two Sample t-test
data: MS_results and PRS_results
t = -7.22, df = 49, p-value = 3.029e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.01900637 -0.01072978
sample estimates:
 mean of x    mean of y 
4.266703e-07 1.486850e-02
```

Rysunek 7: Wynik funkcji t.test.

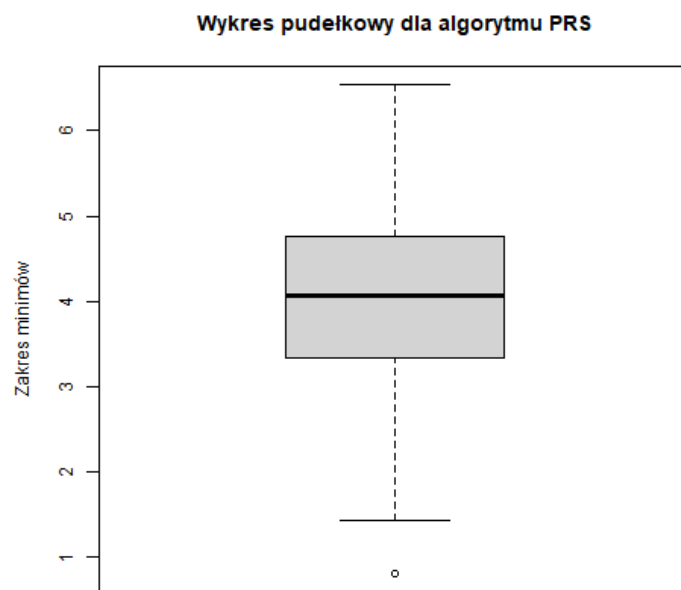
Średnia wyników dla metody PRS wynosi $1.486850 \cdot 10^{-2}$, natomiast dla metody MS wynosi $4.266703 \cdot 10^{-7}$. Jest to różnica 5 rzędów wielkości na korzyść metody MS. P-value wynosi $3.029 \cdot 10^{-9}$, a przedział ufności nie zawiera zera. Z tego wynika że odrzucamy hipotezę zerową, czyli średnie wyników obu algorytmów nie są sobie równe.

1.2 Wykresy funkcji Ackley'a

1.2.1 Pure-Random-Search

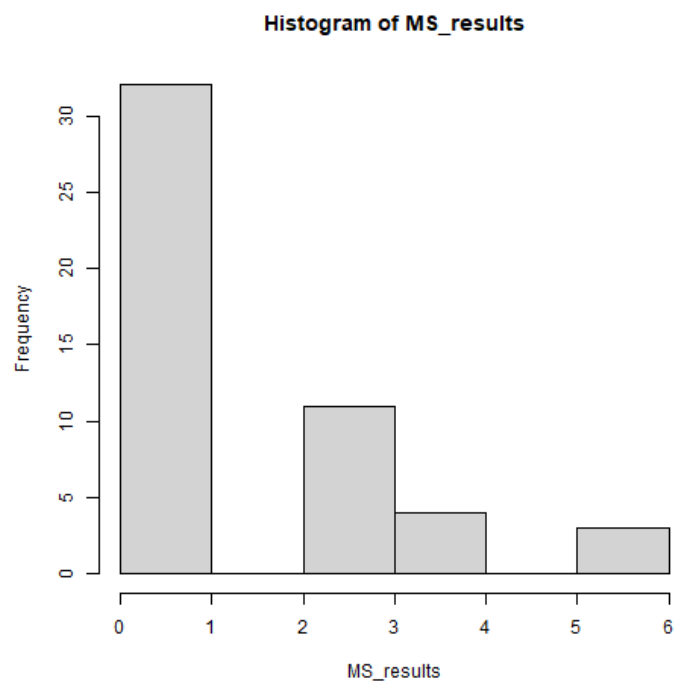


Rysunek 8: Histogram funkcji Ackley'a, metody PRS, dla 2 wymiarów.

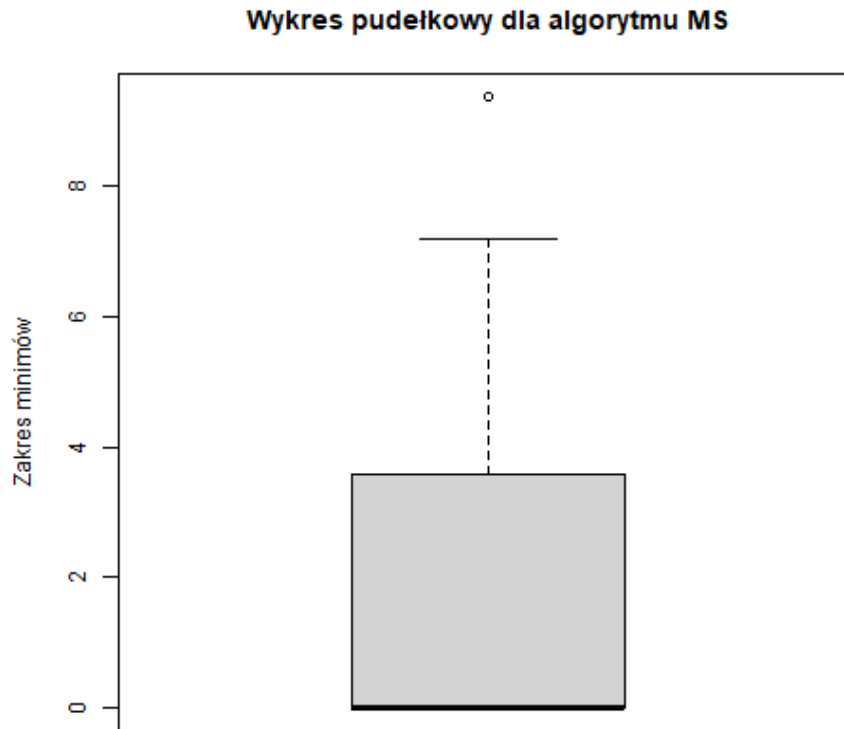


Rysunek 9: Wykres pudełkowy funkcji Ackley'a, metody PRS, dla 2 wymiarów.

1.2.2 Multi-Start



Rysunek 10: Histogram funkcji Ackley'a, metody MS, dla 2 wymiarów.



Rysunek 11: Wykres pudełkowy funkcji Ackley'a, metody MS, dla 2 wymiarów.

1.2.3 Obserwacje

```
Welch Two Sample t-test

data: MS_results and PRS_results
t = -10.01, df = 87.781, p-value = 3.521e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.527195 -2.358619
sample estimates:
mean of x mean of y
 1.176452  4.119359
```

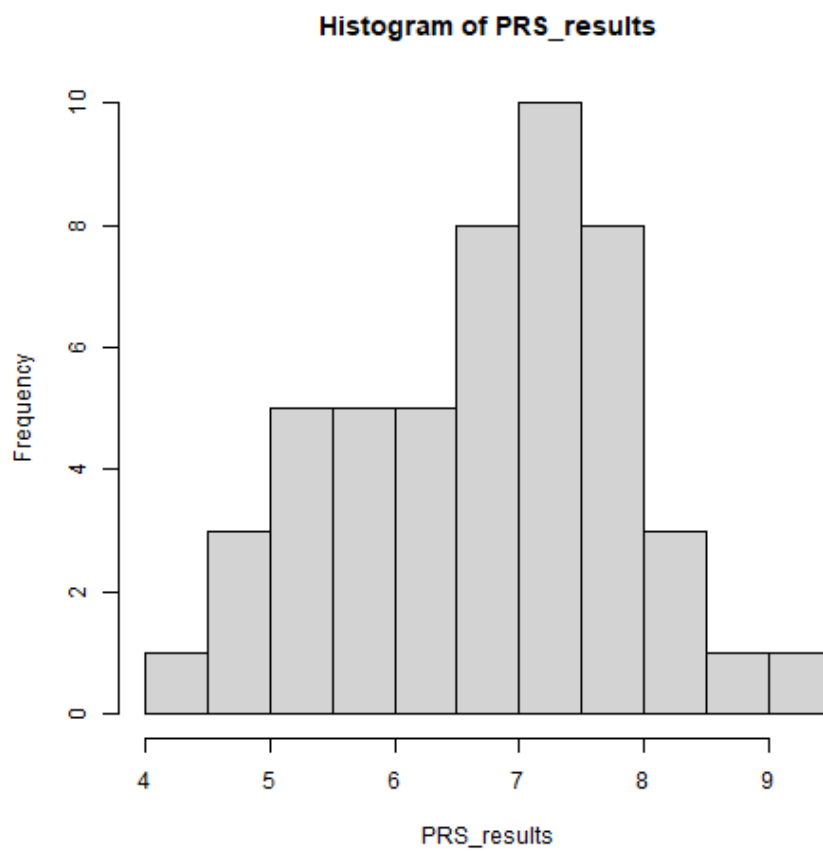
Rysunek 12: Wynik funkcji t.test.

Średnia wyników dla metody PRS wynosi 4.119359, natomiast dla metody MS wynosi 1.176452. Metoda MS jest dokładniejsza. P-value wynosi $3.521 \cdot 10^{-16}$, a przedział ufności nie zawiera zera. Z tego wynika że odrzucamy hipotezę zerową, czyli średnie wyników obu algorytmów nie są sobie równe.

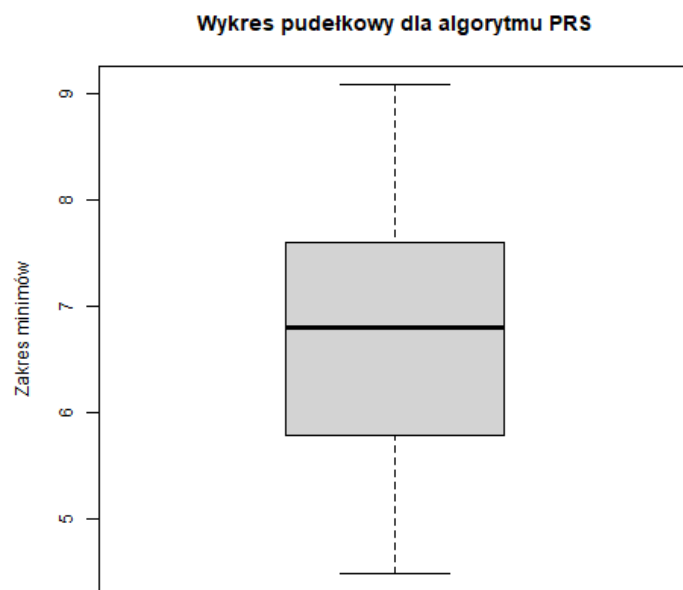
2 $\dim = 10$

2.1 Wykresy Alpine01

2.1.1 Pure-Random-Search

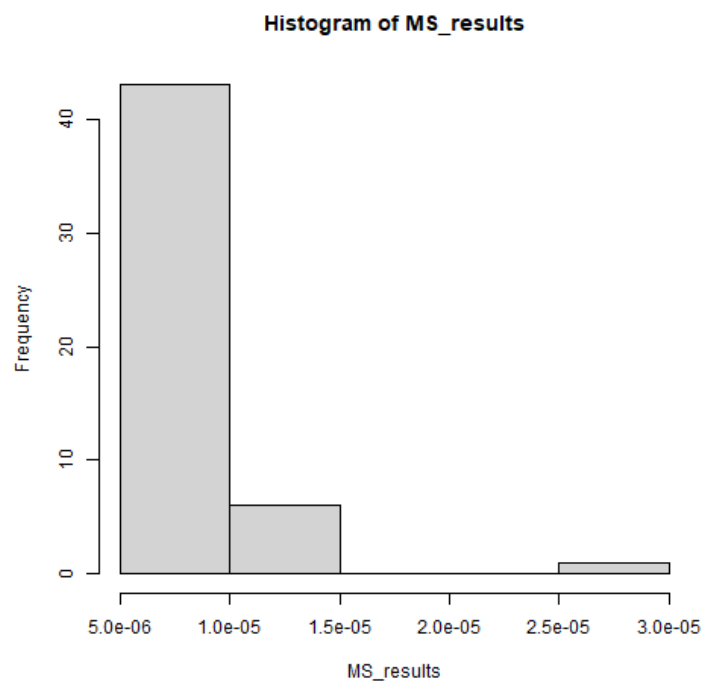


Rysunek 13: Histogram funkcji Alpine01, metody PRS, dla 10 wymiarów.

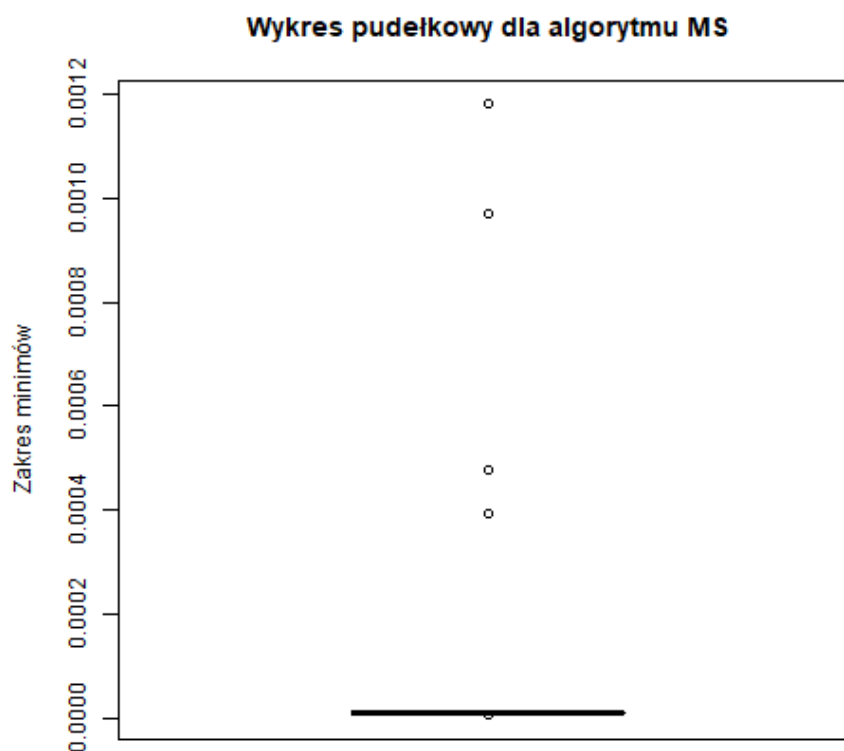


Rysunek 14: Wykres pudełkowy funkcji Alpine01, metody PRS, dla 10 wymiarów.

2.1.2 Multi-Start



Rysunek 15: Histogram funkcji Alpine01, metody MS, dla 10 wymiarów.



Rysunek 16: Wykres pudełkowy funkcji Alpine01, metody MS, dla 10 wymiarów.

2.1.3 Obserwacje

```
Welch Two Sample t-test

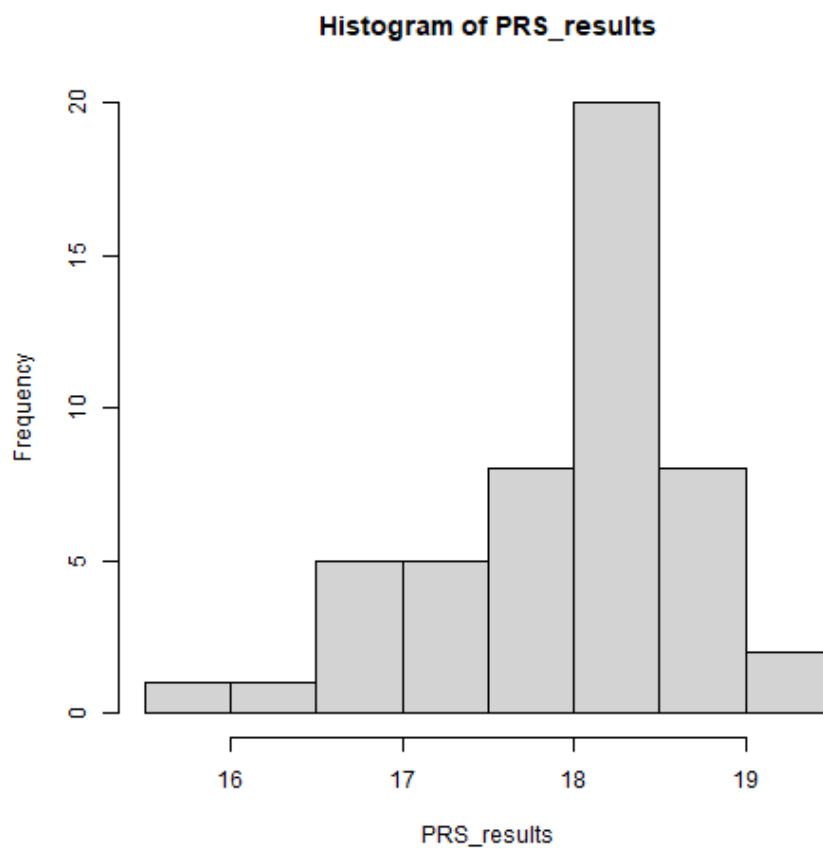
data: MS_results and PRS_results
t = -43.003, df = 49, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.017572 -6.390977
sample estimates:
 mean of x      mean of y 
9.158494e-06 6.704284e+00
```

Rysunek 17: Wynik funkcji t.test.

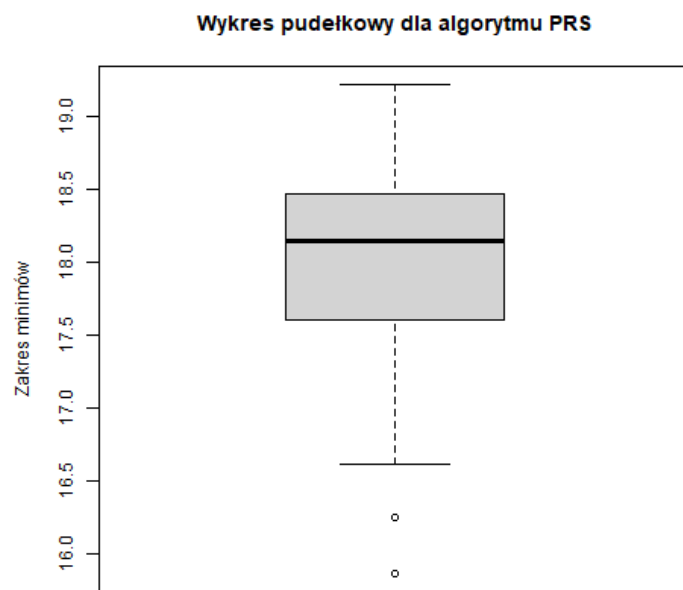
Średnia wyników dla metody PRS wynosi 6.704284, natomiast dla metody MS wynosi $9.158494 * 10^{-6}$. Jest to różnica 6 rzędów wielkości na korzyść metody MS. P-value wynosi $2.2 * 10^{-16}$, a przedział ufności nie zawiera zera. Z tego wynika że odrzucamy hipotezę zerową, czyli średnie wyników obu algorytmów nie są sobie równe.

2.2 Wykresy funkcji Ackley'a

2.2.1 Pure-Random-Search

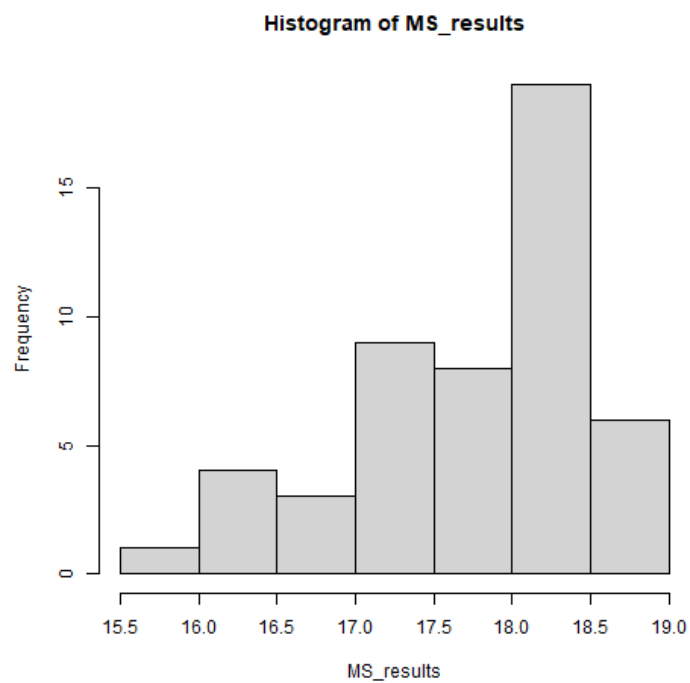


Rysunek 18: Histogram funkcji Ackley'a, metody PRS, dla 10 wymiarów.

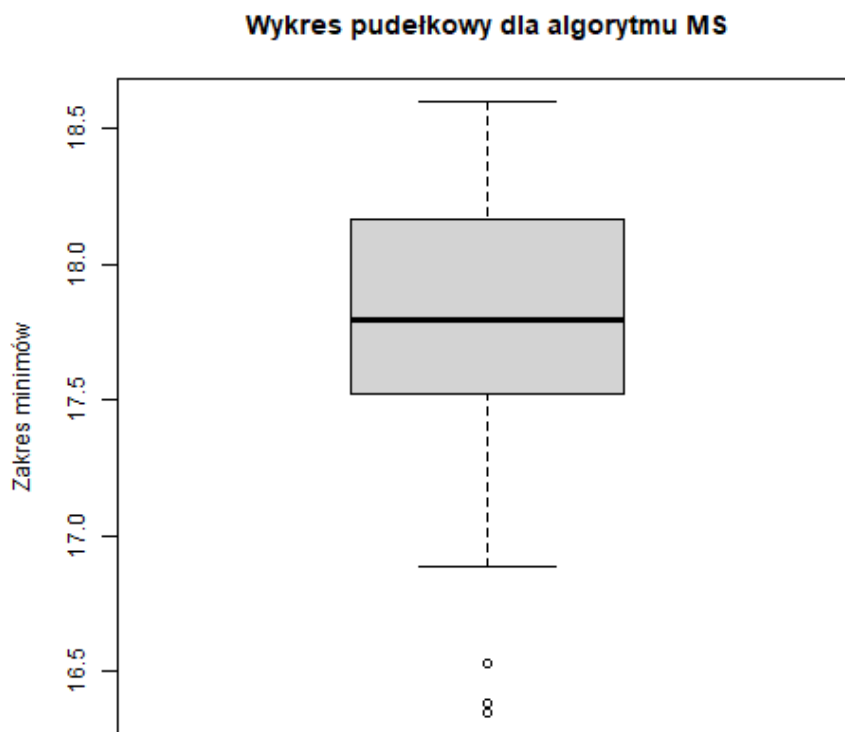


Rysunek 19: Wykres pudełkowy funkcji Ackley'a, metody PRS, dla 10 wymiarów.

2.2.2 Multi-Start



Rysunek 20: Histogram funkcji Ackley'a, metody MS, dla 10 wymiarów.



Rysunek 21: Wykres pudełkowy funkcji Ackley'a, metody MS, dla 10 wymiarów.

2.2.3 Obserwacje

```
welch Two Sample t-test  
  
data: MS_results and PRS_results  
t = -1.6377, df = 97.973, p-value = 0.1047  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
-0.54073809 0.05176934  
sample estimates:  
mean of x mean of y  
17.74146 17.98594
```

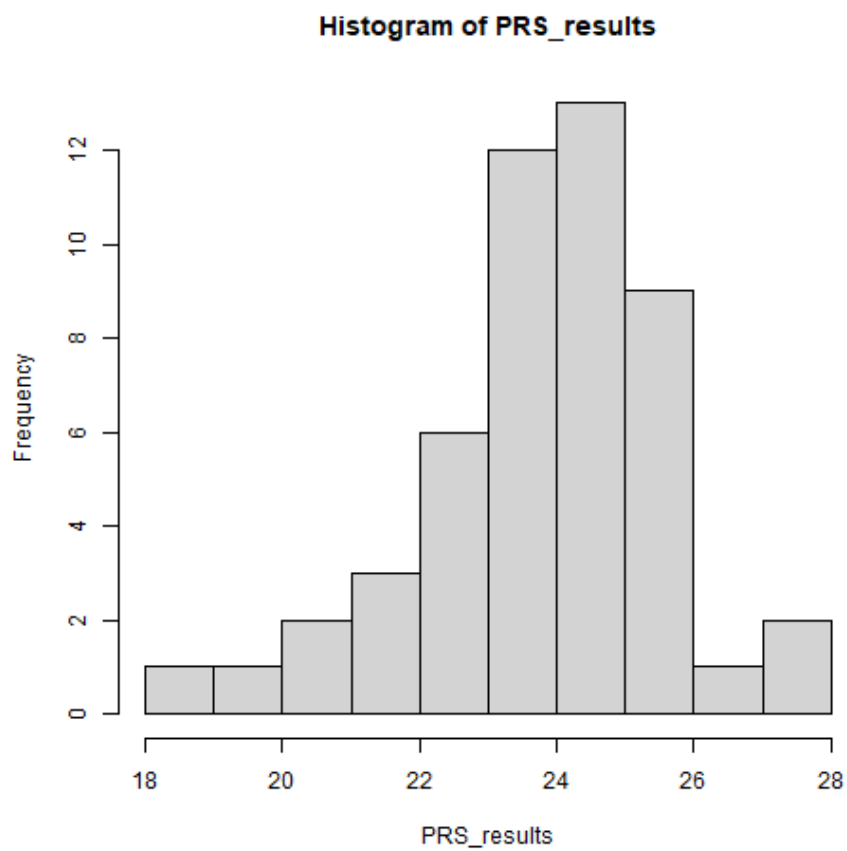
Rysunek 22: Wynik funkcji t.test.

Średnia wyników dla metody PRS wynosi 17.98594, natomiast dla metody MS wynosi 17.7414. Dokładniejsza była metoda MS. P-value wynosi 0.1047, a przedział ufności zawiera 0. Z tego wynika że nie odrzucamy hipotezy zerowej, więc nie możemy nic powiedzieć o równości średnich.

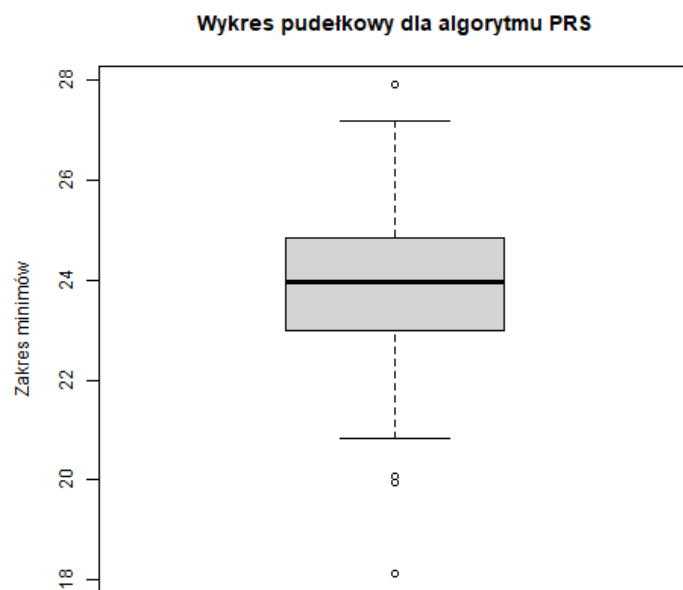
3 $\dim = 20$

3.1 Wykresy Alpine01

3.1.1 Pure-Random-Search

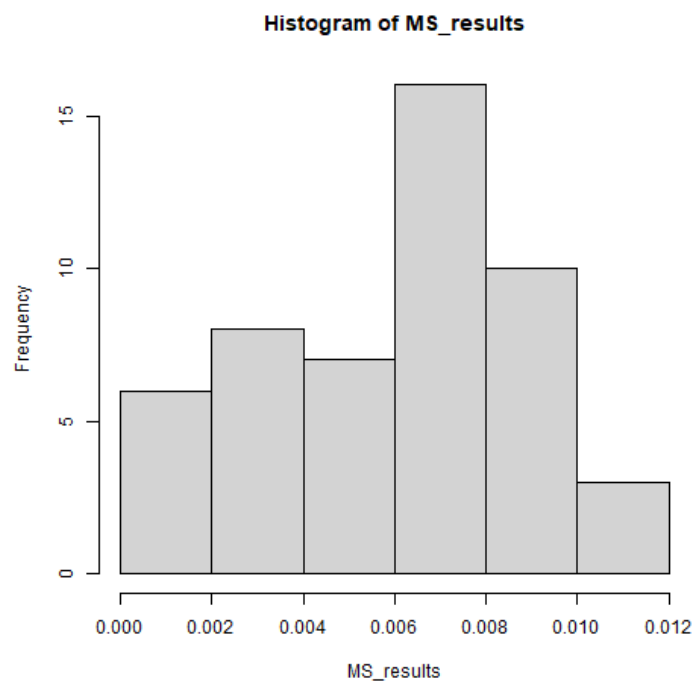


Rysunek 23: Histogram funkcji Alpine01, metody PRS, dla 20 wymiarów.

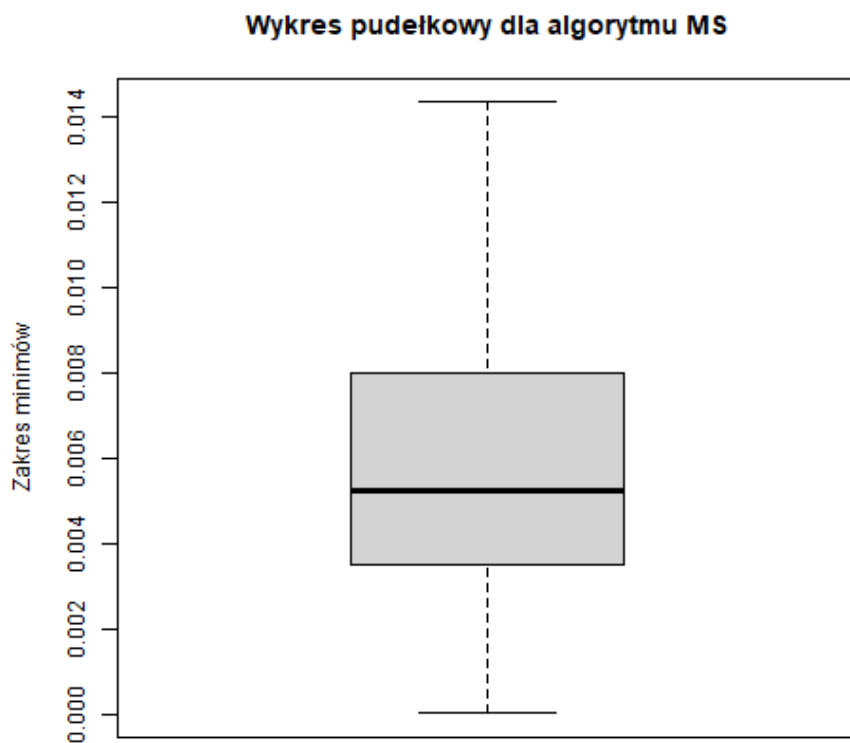


Rysunek 24: Wykres pudełkowy funkcji Alpine01, metody PRS, dla 20 wymiarów.

3.1.2 Multi-Start



Rysunek 25: Histogram funkcji Alpine01, metody MS, dla 20 wymiarów.



Rysunek 26: Wykres pudełkowy funkcji Alpine01, metody MS, dla 20 wymiarów.

3.1.3 Obserwacje

```
welch Two sample t-test

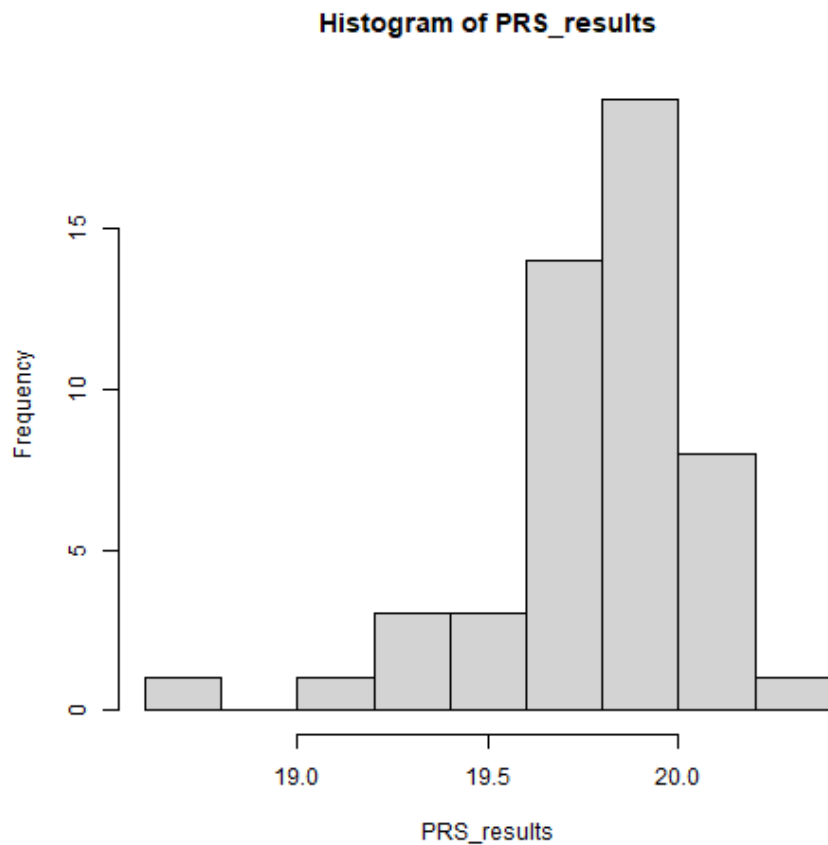
data: MS_results and PRS_results
t = -94.673, df = 49, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.28683 -23.27720
sample estimates:
 mean of x      mean of y 
0.005877498 23.787893129
```

Rysunek 27: Wynik funkcji t.test.

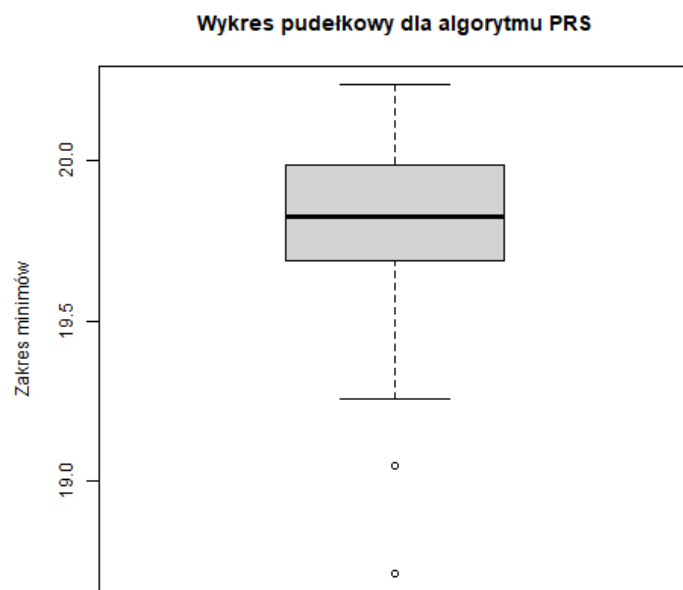
Średnia wyników dla metody PRS wynosi 23.787893, natomiast dla metody MS wynosi $5.8774 \cdot 10^{-3}$. Jest to różnica 4 rzędów wielkości na korzyść metody MS. P-value wynosi $2.2 \cdot 10^{-16}$, a przedział ufności nie zawiera 0. Z tego wynika że odrzucamy hipotezę zerową, z tego wynika że średnie nie są sobie równe.

3.2 Wykresy funkcji Ackley'a

3.2.1 Pure-Random-Search

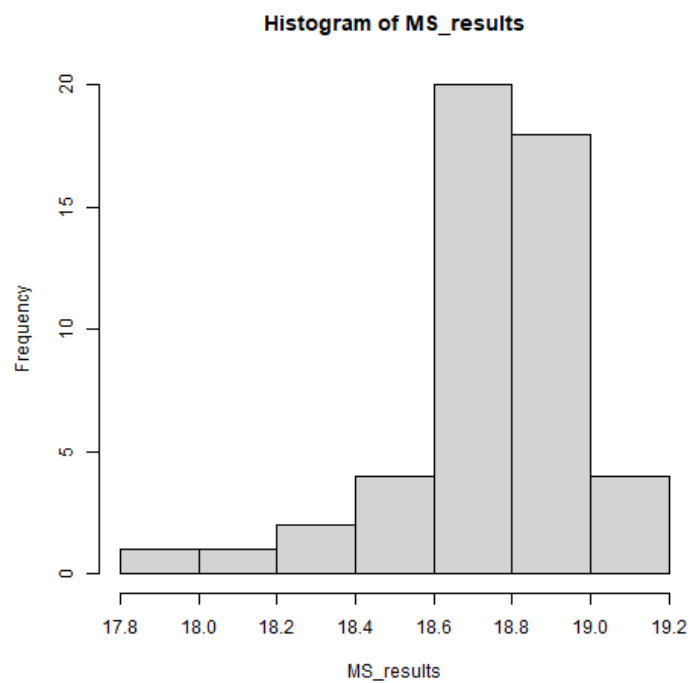


Rysunek 28: Histogram funkcji Ackley'a, metody PRS, dla 20 wymiarów.

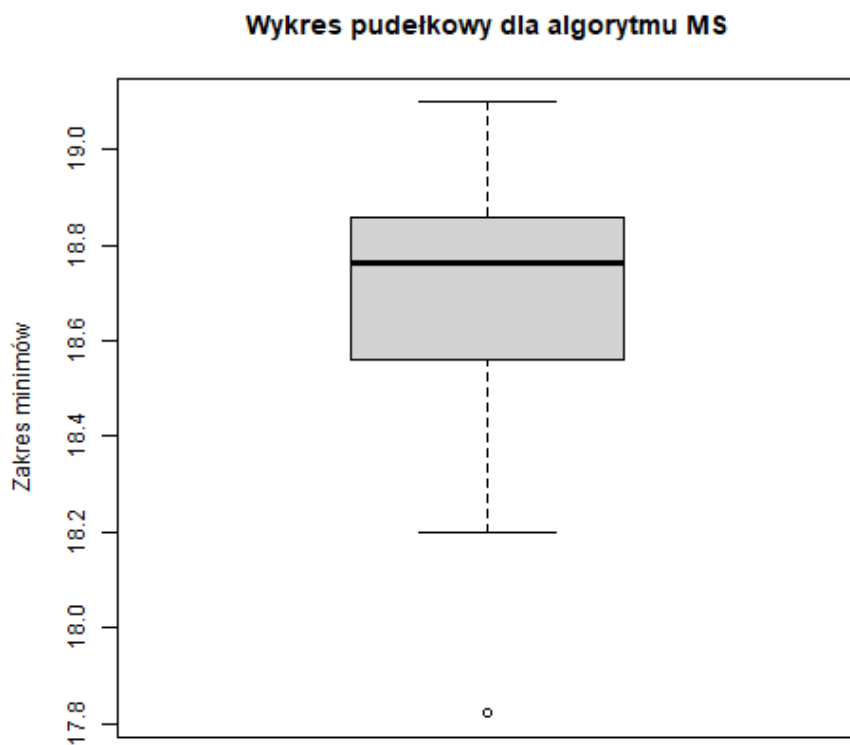


Rysunek 29: Wykres pudełkowy funkcji Ackley'a, metody PRS, dla 20 wymiarów.

3.2.2 Multi-Start



Rysunek 30: Histogram funkcji Ackley'a, metody MS, dla 20 wymiarów.



Rysunek 31: Wykres pudełkowy funkcji Ackley'a, metody MS, dla 20 wymiarów.

3.2.3 Obserwacje

```
welch Two Sample t-test  
  
data: MS_results and PRS_results  
t = -20.308, df = 93.216, p-value < 2.2e-16  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
-1.1549982 -0.9492407  
sample estimates:  
mean of x mean of y  
18.73841 19.79053
```

Rysunek 32: Wynik funkcji t.test.

Średnia wyników dla metody PRS wynosi 19.7905, natomiast dla metody MS wynosi 18.7384. Metoda MS jest nieznacznie dokładniejsza. P-value wynosi $2.2 \cdot 10^{-16}$, a przedział ufności nie zawiera 0. Z tego wynika że odrzucamy hipotezę zerową, z tego wynika że średnie nie są sobie równe.

iv. Podsumowanie

1 Podsumowanie

Na podstawie przeprowadzonych porównań można zauważyć że metoda MS jest skuteczniejszym algorytmem znajdowania minimum funkcji. Zazwyczaj wyniki były zauważalnie lepsze dla metody MS. Dzieje się tak, ponieważ metoda PRS opiera się na losowym wyborze punktów pomiarowych, w odróżnieniu od metody MS która wykorzystuje informacje o gradiencie funkcji do wyznaczenia kierunku optymalizacji.

Wraz ze wzrostem liczby wymiarów wyniki obu metod pogarszają się. Przyczyną tego stanu rzeczy jest wzrastające skomplikowanie problemu przy stałej liczbie losowanych punktów.

Metoda PRS ze względu na swoją losowość nie zależy tak bardzo od charakterystyki testowanej funkcji. W odróżnieniu od metody MS, która jest wrażliwa na ukształtowanie gradientu, co jest szczególnie zauważalne przy badaniu funkcji Ackley'a dla której obie metody dawały zbliżone wyniki.

Analiza funkcji Alpine01 wykazała, że metoda MS była zdecydowanie skuteczniejsza ze względu na optymalizację wyboru minimum na podstawie gradientu funkcji.

W pięciu na sześć porównań z analizy istotności statystycznej różnicy między średnim wynikiem obu algorytmów, opartej na 95-procentowym przedziale ufności odrzuciliśmy hipotezę zerową zakładającą równość obu średnich na rzecz hipotezy alternatywnej zakładającej to że średnie są od siebie różne.

W jednym przypadku hipoteza zerowa nie została odrzucona, wyniki obu algorytmów w tamtym przypadku mocno odbiegają od realnego minimum lokalnego i prawdopodobnie przypadkowo są do siebie bardzo zbliżone co jest przyczyną tego że nie można jednoznacznie określić czy te średnie różnią się od siebie przy zadanym przedziale ufności.

2 Bibliografia

Źródła i inspiracje wykorzystane przy tworzeniu projektu:

- Wykłady z Rachunku prawdopodobieństwa i statystyki, prowadzone przez dr hab. Macieja Smółkę, na 3 semestrze Informatyki AGH WI.
- <https://www.agh.edu.pl/o-agh/multimedia/znak-graficzny-agh/>
- http://infinity77.net/global_optimization/test_functions_nd_A.html
- <https://pogotowiestatystyczne.pl/aploud/2023/04/2023-02-Test-t-Studenta-dla-prob-niezaleznych.pdf>