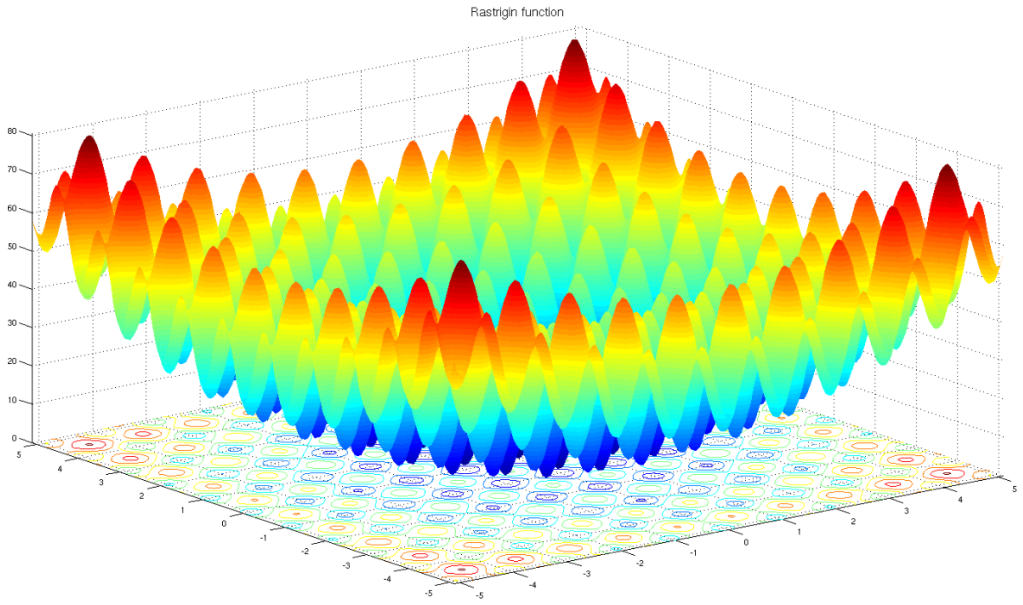
Projekt statystyka

Kacper Garus, Jakub Zawistowski

Poniższy projekt przedstawia analizę statystyczną porównania algorytmów minimalizacji stochastycznej. Porównywane przez nas algorytmy to algorytm Poszukiwania przypadkowego (PRS) oraz algorytm wielokrotnego startu (MS), natomiast wybrane funkcje to funkcja Rastrigina oraz funkcja Rosenbrocka

Funkcja Rastrigina

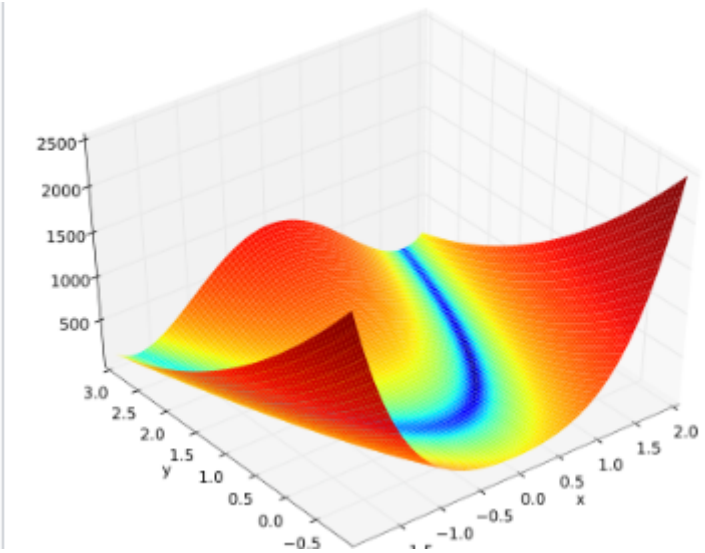
Funkcja Rastrigina jest funkcją matematyczną często używaną do oceny wydajności algorytmów optymalizacji . Przedstawia ciekawe pułapki w postaci wielu lokalnych minimów i maksimów. Został on zaproponowany w 1974 roku przez Rastrigina w dwóch wymiarach i uogólniony przez Mühlenbein.



$$f(\mathbf{x}) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i) \right]$$

Funkcja Rosenbrocka

Funkcja rosenbrocka jest niewypukła funkcja od dwóch zmiennych wykorzystywanych jako test matematycznych optymalizacji problemów. Została wprowadzona przez Howarda H. Rosenbrocka w 1960 roku. Znany jest również jako cecha bananowa .



$$f(x, y) = (a - x)^2 + b(y - x^2)^2$$

Algorytm Poszukiwania Przypadkowego (Pure Random Search)

Poniższy kod przedstawia implementację algorytmu PRS w języku R. Losujemy po kolei zadaną z góry liczbę punktów z rozkładem jednostajnym w danej dziedzinie. Dla każdego punktu obliczamy wartość danej funkcji i jeśli jest ona mniejsza niż najmniejsza do tej pory wyliczona, to zostaje ona nową najmniejszą wartością. Algorytm zwraca najmniejszą wartość.

```r
prsAlgorithm <- function(func, dimentions, nPoints){
  f <- func(dimentions)
  minVal <- NULL

  for(x in 1:nPoints){
    point <- runif(dimentions, getLowerBoxConstraints(f), getUpperBoxConstraints(f))
    value <- f(point)

    if(is.null(minVal) || value <  minVal){
      minVal <- value
    }
  }

  return(minVal)
}
```

Metoda wielokrotnego startu (Multi-Start,ms)

Poniższy kod przedstawia implementację algorytmu MS w języku R. Losujemy po kolei zadaną z góry liczbę punktów z rozkładem jednostajnym w danej dziedzinie. Używając funkcji optim() z metodą **L-BFGS-B** uzyskujemy wartość, i jeśli jest ona mniejsza od obecnie najmniejszej wartości, to zostaje ona nową najmniejszą wartością. Wynikiem algorytmu jest wartość zoptymalizowanej funkcji w tym punktów zwróconych przez uruchomienia metody lokalnej, w którym ta wartość jest najmniejsza.

```r
msAlgorithm <- function(func, dimentions, nPoints){
  f <- func(dimentions)
  counter <- 0
  minVal <- NULL

  for(x in 1:nPoints){
    point <- runif(dimentions, getLowerBoxConstraints(f), getUpperBoxConstraints(f))

    result <- optim(point, f, method = "L-BFGS-B", lower = getLowerBoxConstraints(f), upper = getUpperBoxConstraints(f))
    value <- as.numeric(result$value)
    counter <- counter + as.numeric(result$counts[1])

    if(is.null(minVal) || value < minVal){
      minVal <- value
    }
  }
  return(list(minVal, counter))
}
```

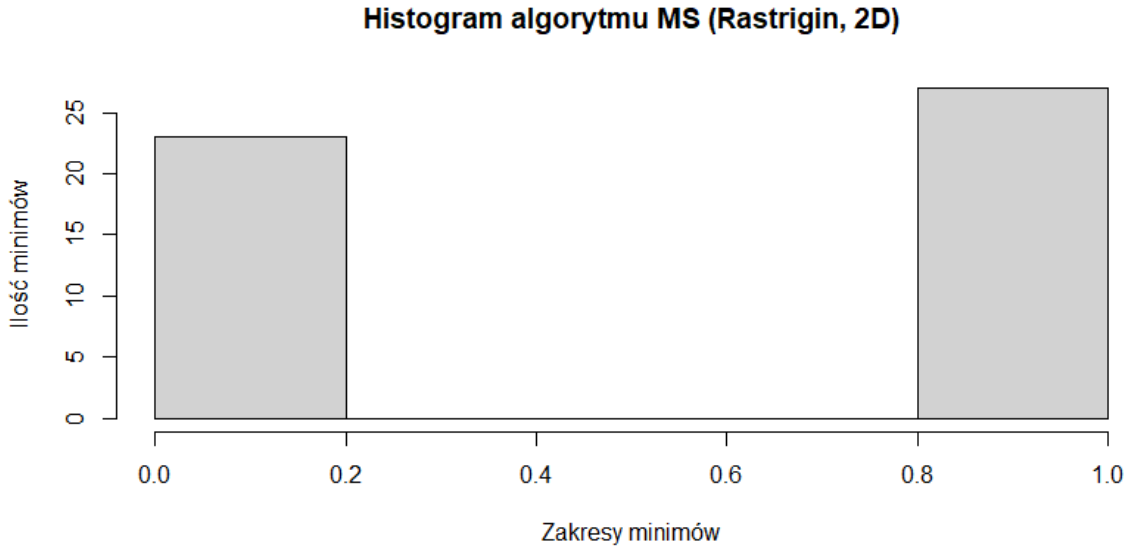
Algorytm porównujący wyniki algorytmów PRS i MS

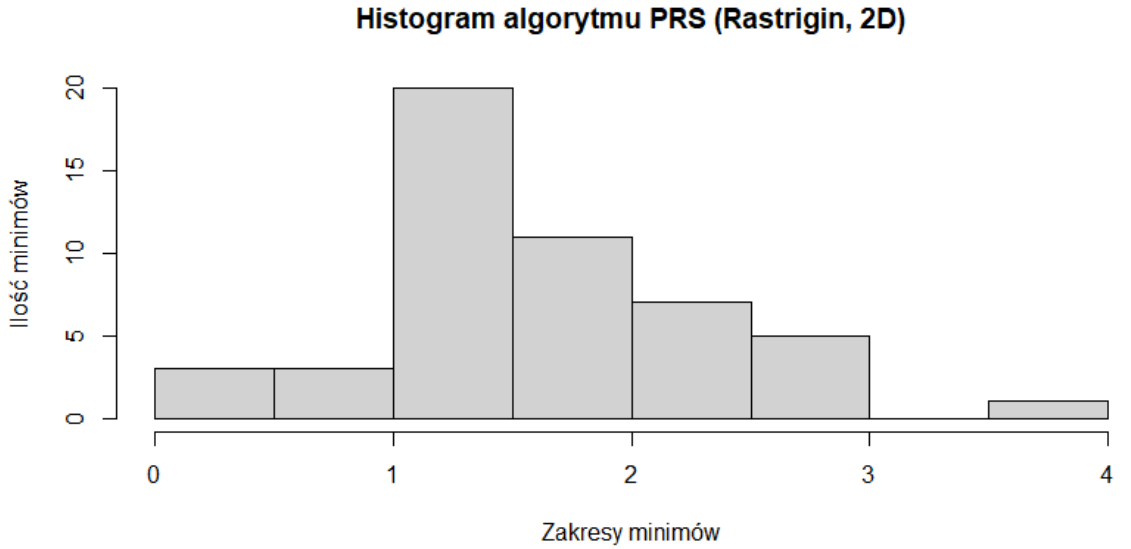
Algorytm wywołuje MS 50 razy przyjmując 100 punktów. Potem oblicza średnią z liczników uzyskanych przez MS, a następnie uruchamia PRS 50 razy z tą średnią jako parametrem. Funkcja zwraca listę zawierającą wyniki punktowe z MS oraz wyniki z PRS.

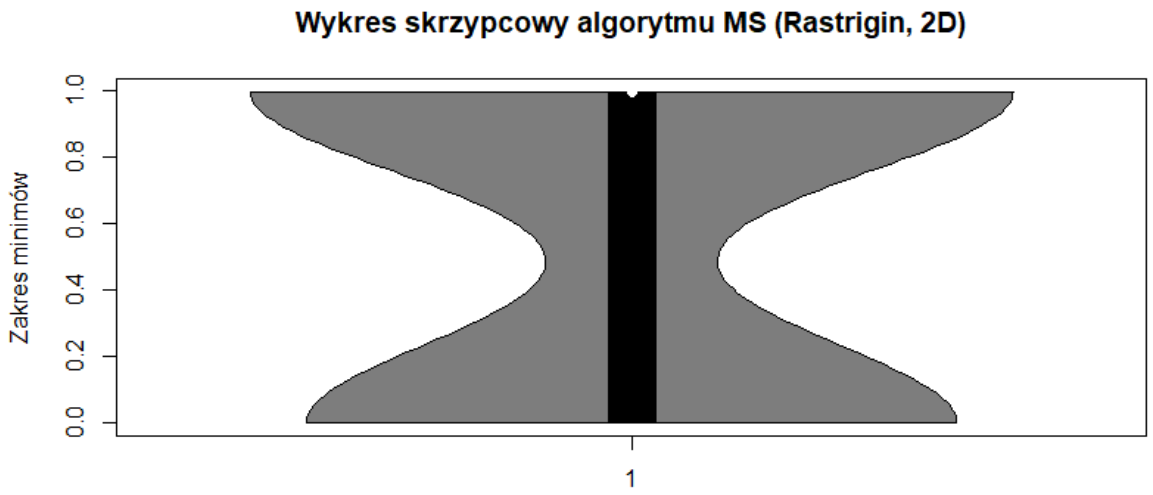
```
compareAlgorithms <- function(alg1, alg2, func, dimentions){
  alg1Result <- replicate(50, alg1(func, dimentions, 100))
  alg1Average <- as.numeric(alg1Result[2,])
  alg1Points <- as.numeric(alg1Result[1,])
  average <- mean(alg1Average)
  alg2Result <- replicate(50, alg2(func, dimentions, average))

  return(list(alg1Points, alg2Result))
}
```

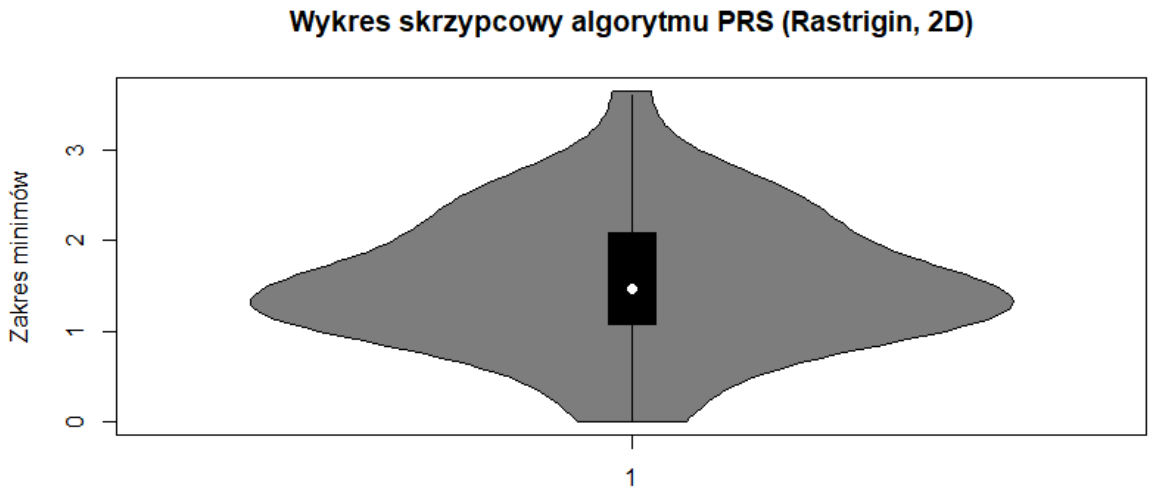
Opracowanie wyników

- Porównanie algorytmów dla funkcji Rastrigina i 2 wymiarów









Z wykresów wyników można zauważyć, że wyniki algorytmu MS były niższe od wyników algorytmu PRS.

➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 2 wymiarów dla algorytmu MS**

```
One Sample t-test

data:  rastrigin2Result[[1]]
t = 7.5843, df = 49, p-value = 8.299e-10
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.3949181 0.6796376
sample estimates:
mean of x
0.5372779
```

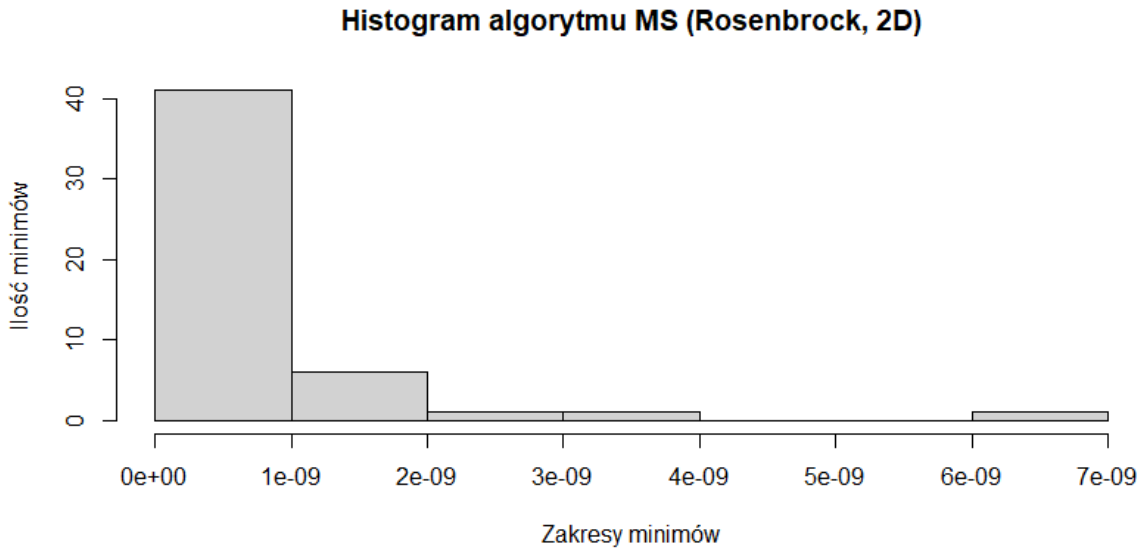
➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 2 wymiarów dla algorytmu PRS**

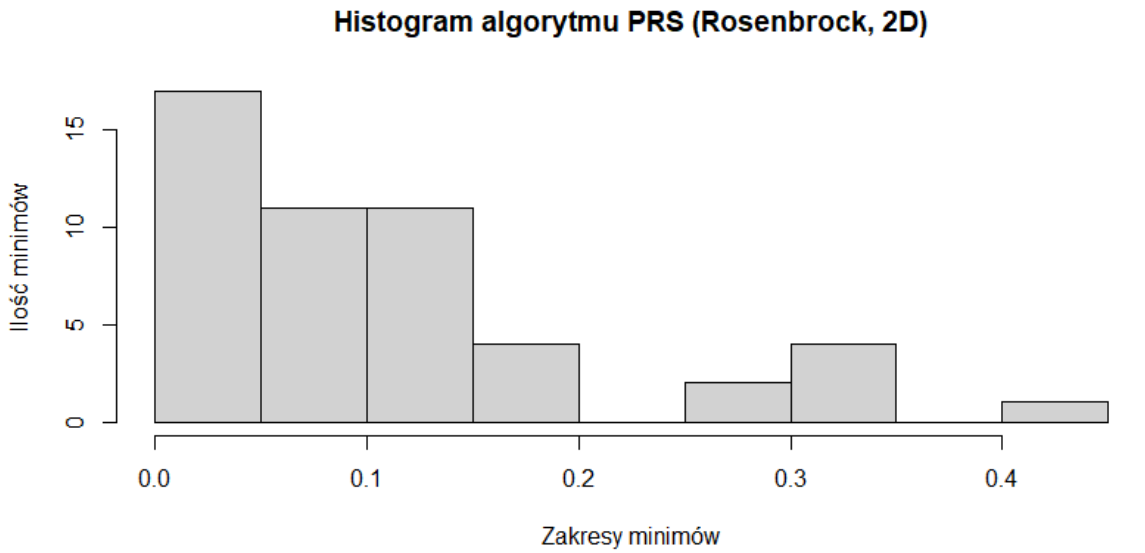
```
One Sample t-test

data:  rastrigin2Result[[2]]
t = 14.912, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.346910 1.766491
sample estimates:
mean of x
 1.556701
```

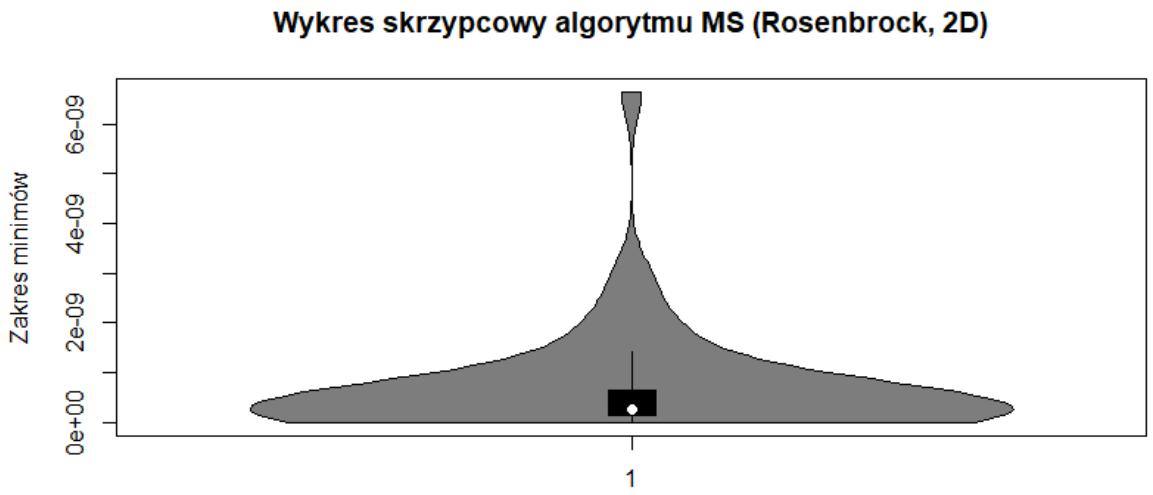
➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 2 wymiarów dla porównania algorytmów MS i PRS**

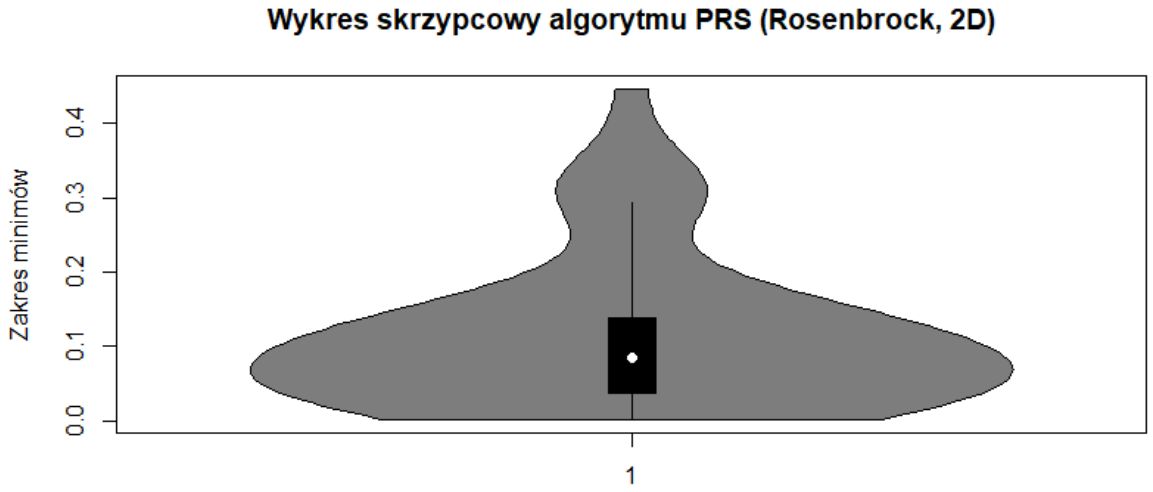
```
Welch Two Sample t-test

data:  rastrigin2Result[[2]] and rastrigin2Result[[1]]
t = 8.0803, df = 86.232, p-value = 3.582e-12
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.7686309 1.2702146
sample estimates:
mean of x mean of y
1.5567006 0.5372779
```

- Porównanie algorytmów dla funkcji Rosenbrocka i 2 wymiarów









Z wykresów wyników ponownie można zauważyć, przewagę działania algorytmu MS nad PRS, wyniki MS były niższe rzędu 10^(-8), dodatkowo w znacznie węższym obszarze.

➔ **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 2 wymiarów dla algorytmu MS**

```
One Sample t-test

data:  rosenbrock2Result[[1]]
t = 4.2014, df = 49, p-value = 0.0001118
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 3.404055e-10 9.646065e-10
sample estimates:
  mean of x
6.52506e-10
```

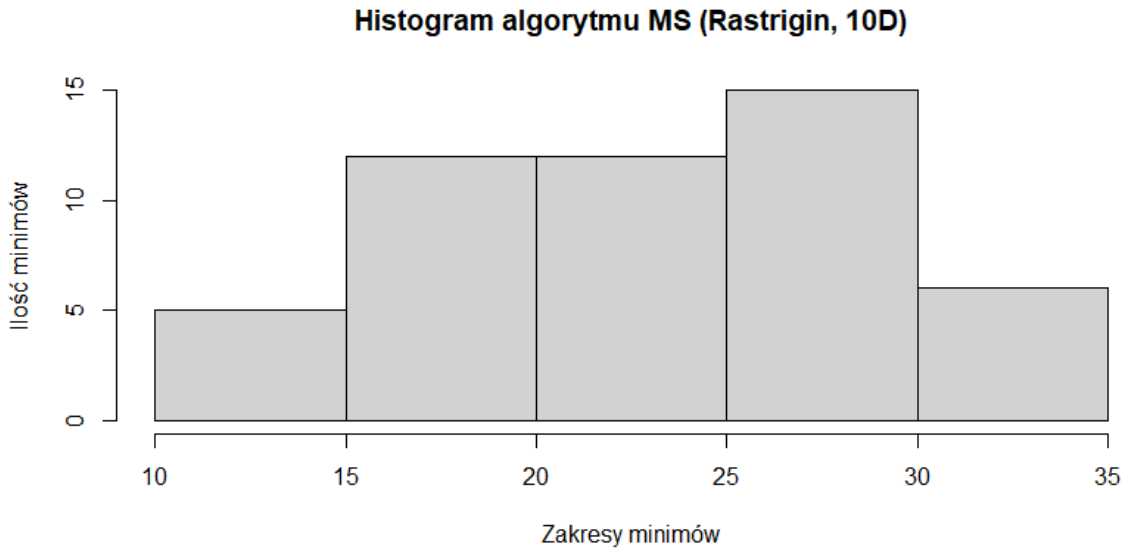
➔ **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 2 wymiarów dla algorytmu PRS**

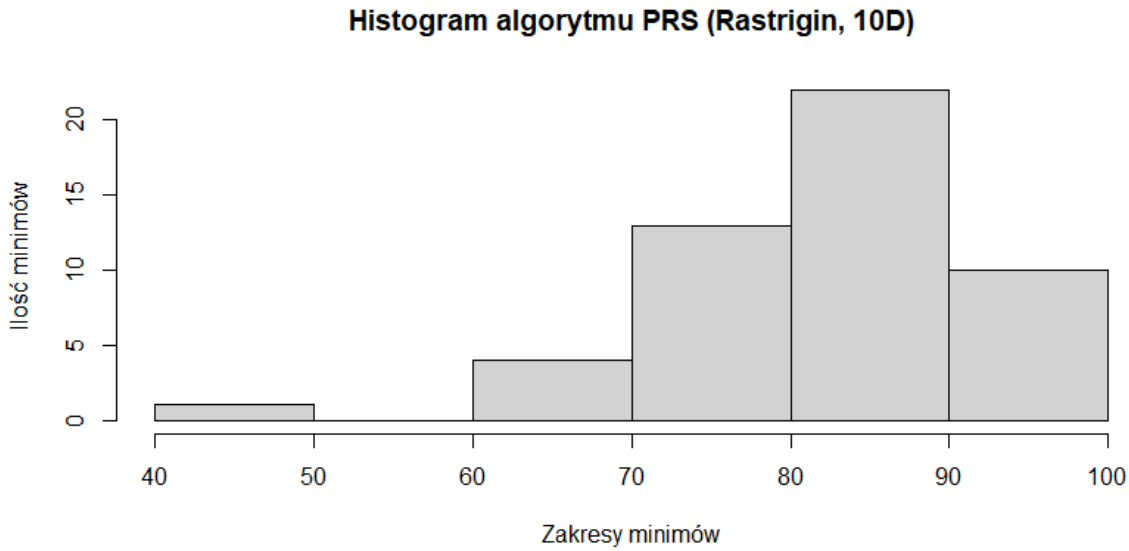
```
One Sample t-test

data:  rosenbrock2Result[[2]]
t = 7.769, df = 49, p-value = 4.316e-10
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.08335579 0.14152466
sample estimates:
mean of x
0.1124402
```

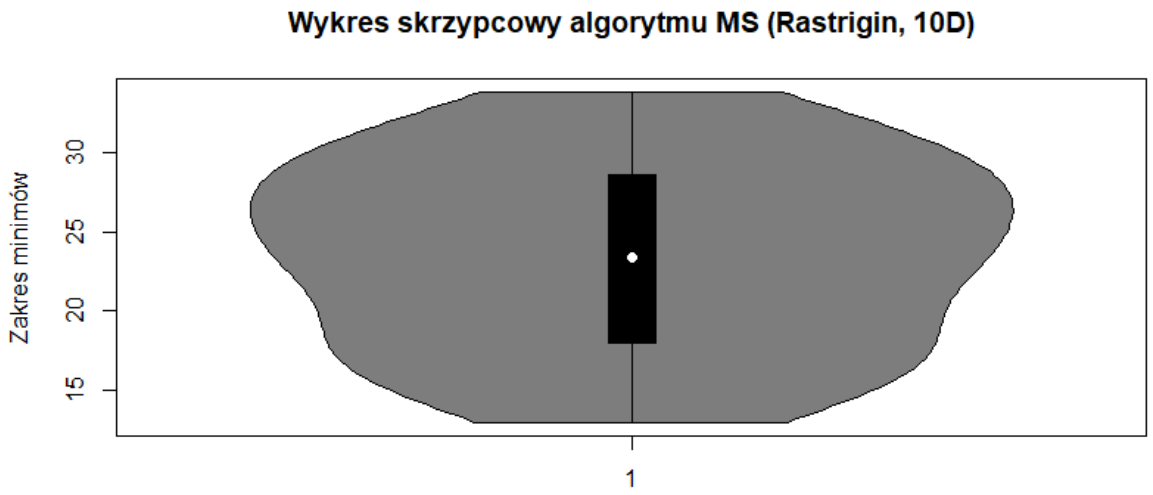
➔ **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 2 wymiarów dla porównania algorytmów MS i PRS**

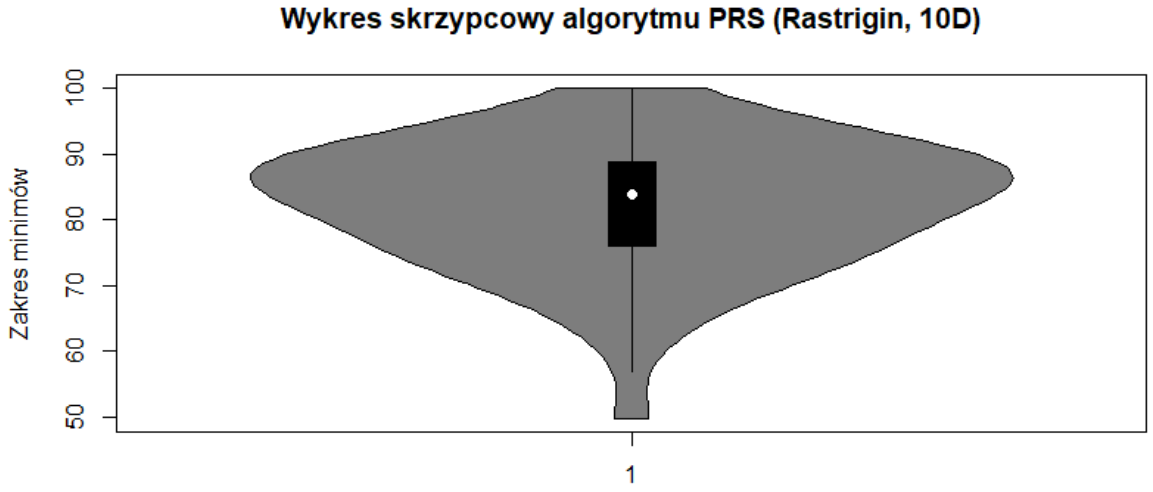
```
Welch Two Sample t-test

data:  rosenbrock2Result[[2]] and rosenbrock2Result[[1]]
t = 7.769, df = 49, p-value = 4.316e-10
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.08335579 0.14152466
sample estimates:
   mean of x    mean of y
1.124402e-01 6.525060e-10
```

- Porównanie algorytmów dla funkcji Rastrigina i 10 wymiarów









Ponownie algorytm MS sprawdził się lepiej w znajdowaniu minimów od algorytmu PRS.

➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 10 wymiarów dla algorytmu MS**

```
One Sample t-test

data:  rastrigin10Result[[1]]
t = 28.218, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 21.84573 25.19583
sample estimates:
mean of x
 23.52078
```

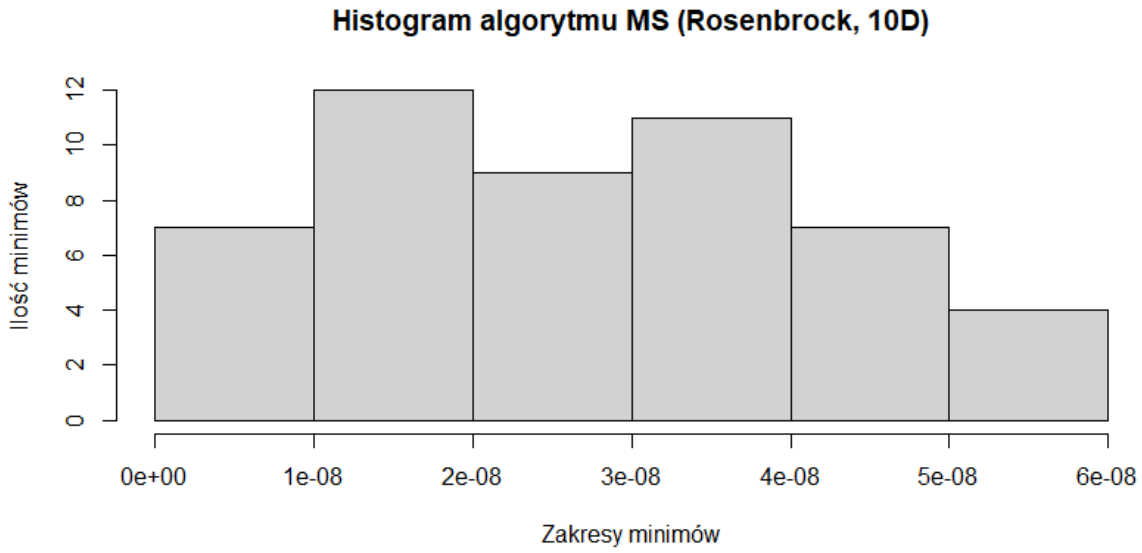
➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 10 wymiarów dla algorytmu PRS**

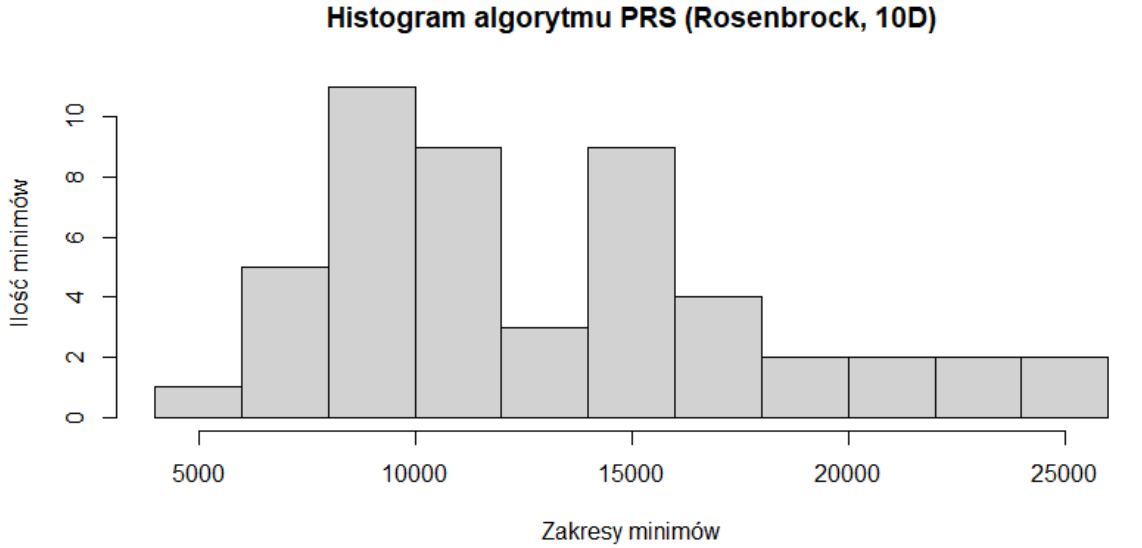
```
One Sample t-test

data:  rastrigin10Result[[2]]
t = 59.878, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 79.46475 84.98383
sample estimates:
mean of x
 82.22429
```

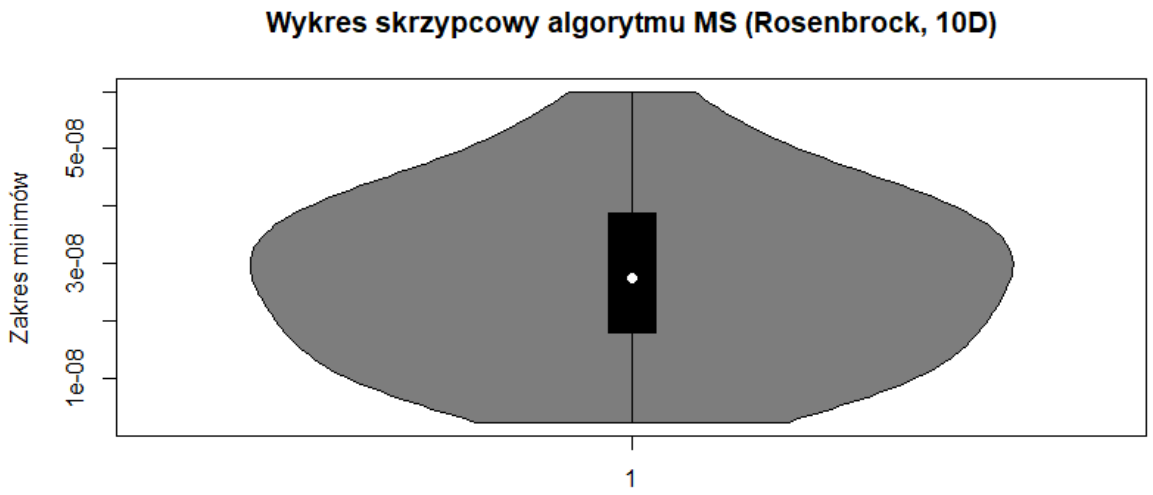
➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 10 wymiarów dla porównania algorytmów MS i PRS**

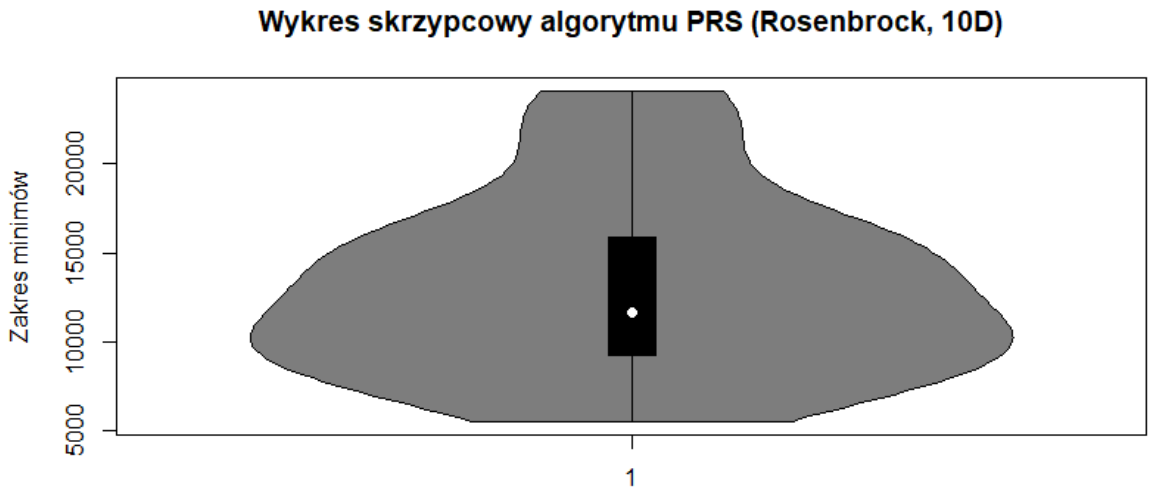
```
Welch Two Sample t-test

data:  rastrigin10Result[[2]] and rastrigin10Result[[1]]
t = 36.544, df = 80.792, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 55.50720 61.89982
sample estimates:
mean of x mean of y
 82.22429  23.52078
```

- Porównanie algorytmów dla funkcji Rosenbrocka i 10 wymiarów









PRS miał problem ze znalezieniem minimów dla funkcji Rosenbrocka 10D w przeciwieństwie do MS, który znalazł ich więcej na mniejszym zakresie.

- **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 10 wymiarów dla algorytmu MS**

```
One Sample t-test

data:  rosenbrock10Result[[1]]
t = 13.446, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 2.336840e-08 3.158097e-08
sample estimates:
   mean of x
2.747468e-08
```

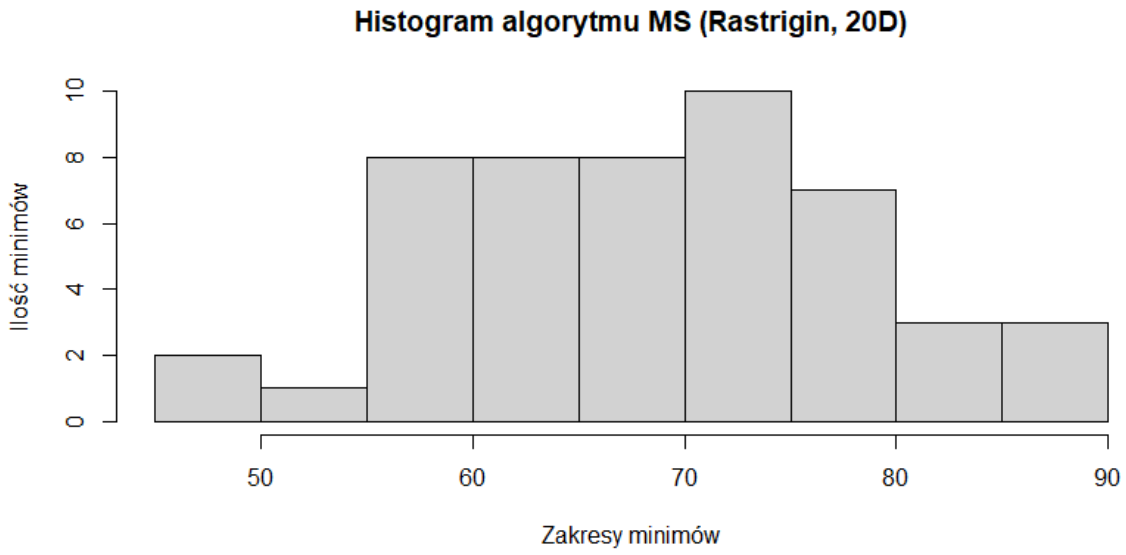
- **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 10 wymiarów dla algorytmu PRS**

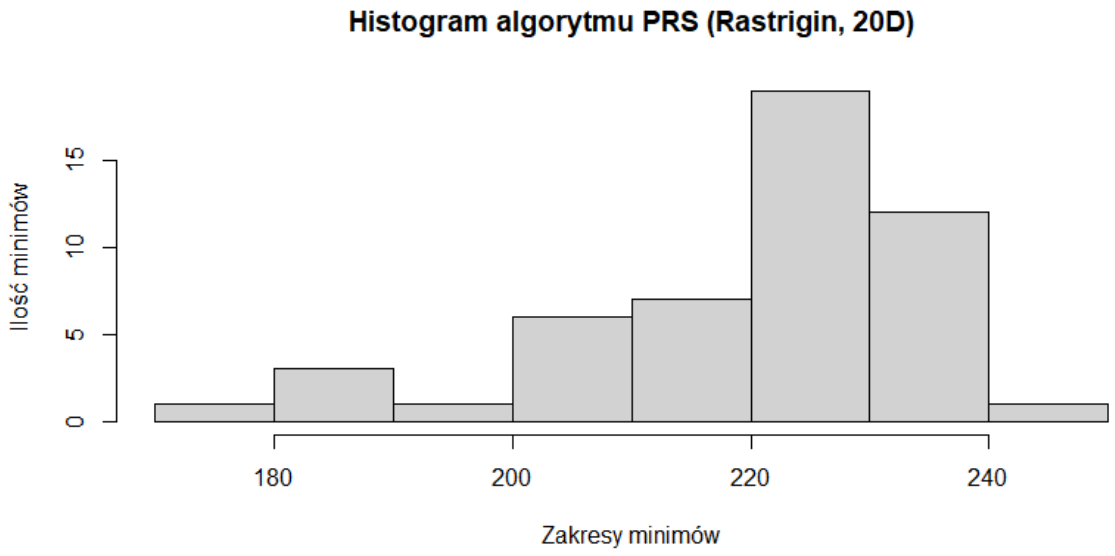
```
One Sample t-test

data:  rosenbrock10Result[[2]]
t = 18.67, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 11753.29 14588.67
sample estimates:
mean of x
 13170.98
```

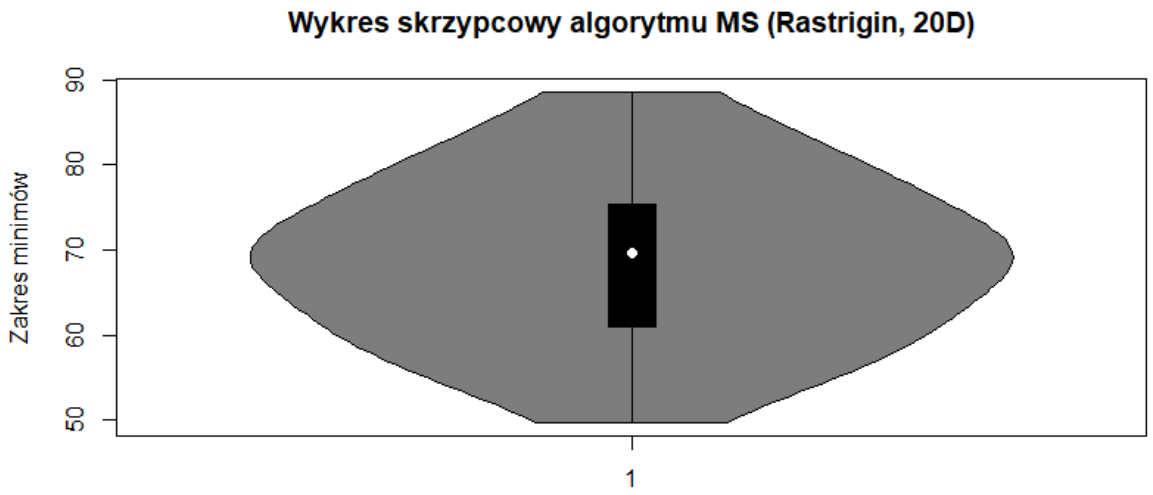
- **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 10 wymiarów dla porównania algorytmów MS i PRS**

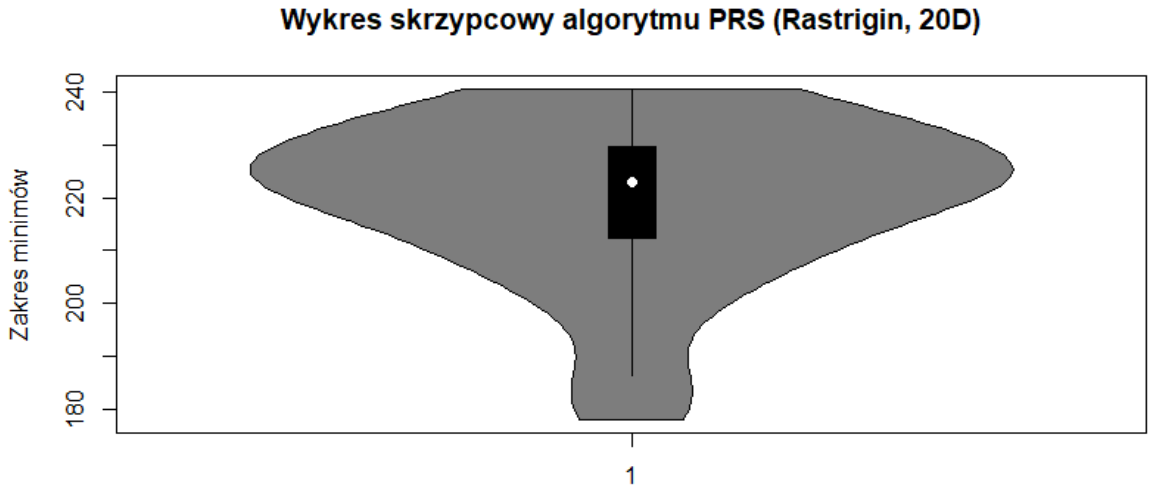
```
Welch Two Sample t-test

data:  rosenbrock10Result[[2]] and rosenbrock10Result[[1]]
t = 18.67, df = 49, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 11753.29 14588.67
sample estimates:
   mean of x    mean of y
1.317098e+04 2.747468e-08
```

- Porównanie algorytmów dla funkcji Rastrigina i 20 wymiarów









MS znajduje minima bliższe 0 i na mniejszym zakresie niż PRS dla dla funkcji Rastrigina dla 20 wymiarów.

➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 20 wymiarów dla algorytmu MS**

```
One Sample t-test

data:  rastrigin20Result[[1]]
t = 49.824, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 65.97844 71.52443
sample estimates:
mean of x
 68.75143
```

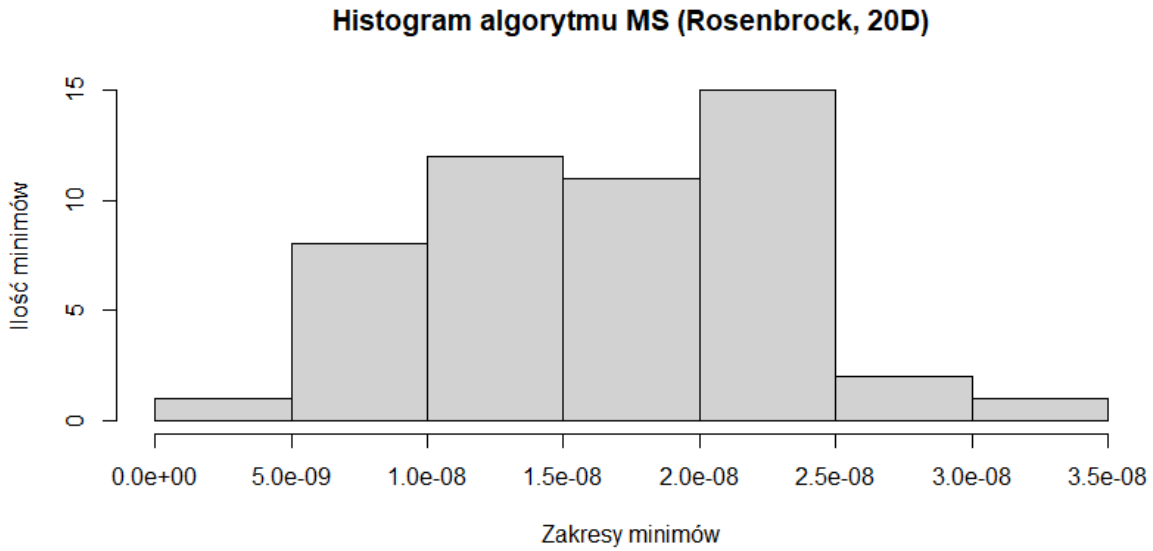
➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 20 wymiarów dla algorytmu PRS**

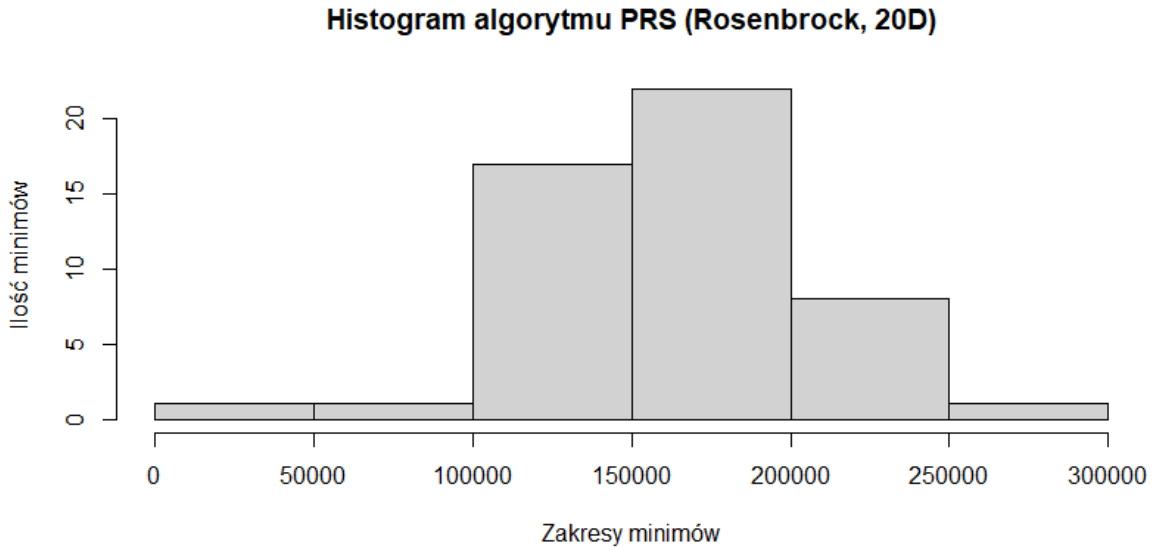
```
One Sample t-test

data:  rastrigin20Result[[2]]
t = 100.3, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 214.9658 223.7558
sample estimates:
mean of x
 219.3608
```

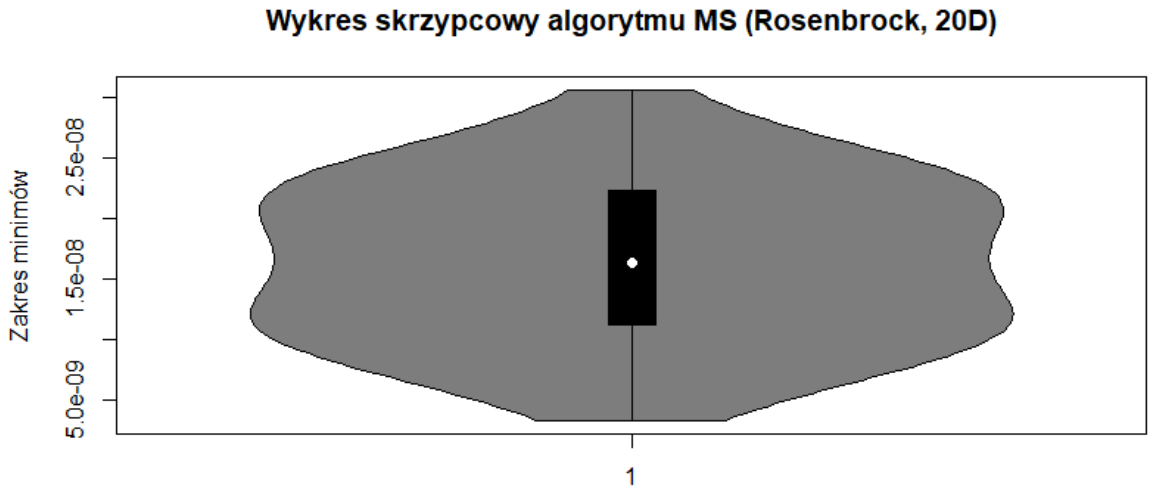
➔ **Przedział ufności i hipoteza zerowa funkcji Rastrigina 20 wymiarów dla porównania algorytmów MS i PRS**

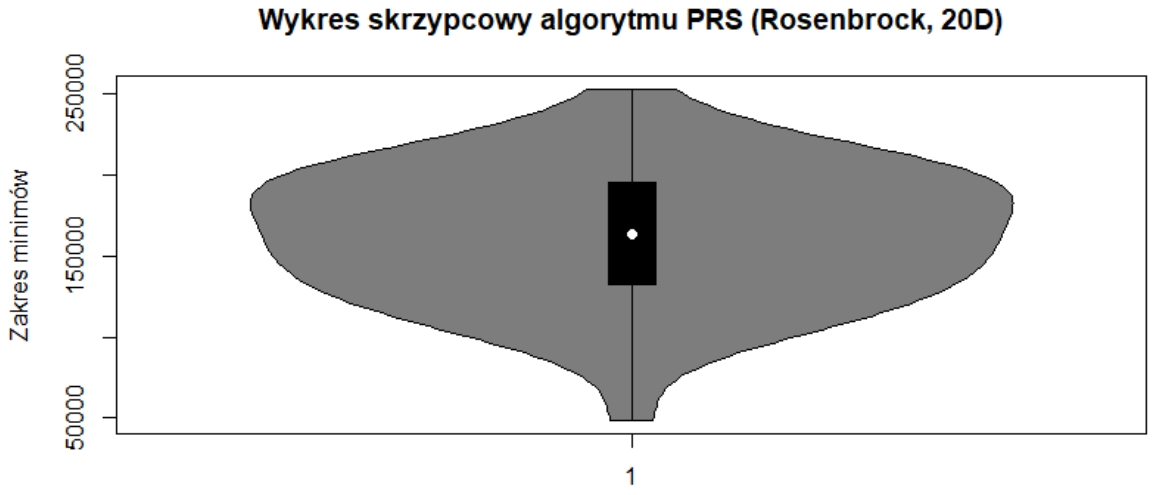
```
Welch Two Sample t-test

data:  rastrigin20Result[[2]] and rastrigin20Result[[1]]
t = 58.241, df = 82.676, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 145.4657 155.7530
sample estimates:
mean of x mean of y
219.36078  68.75143
```

- Porównanie algorytmów dla funkcji Rosenbrocka i 20 wymiarów









MS znajduje minima bliższe 0 i na mniejszym zakresie i znajduje ich więcej niż algorytm PRS dla dla funkcji Rosenbrocka dla 20 wymiarów

➔ **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 20 wymiarów dla algorytmu MS**

```
One Sample t-test

data:  rosenbrock20Result[[1]]
t = 17.645, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.446783e-08 1.818683e-08
sample estimates:
   mean of x
1.632733e-08
```

➔ **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 20 wymiarów dla algorytmu PRS**

```
One Sample t-test

data:  rosenbrock20Result[[2]]
t = 28.298, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 151555.9 174726.6
sample estimates:
mean of x
 163141.2
```

➔ **Przedział ufności i hipoteza zerowa funkcji Rosenbrocka 20 wymiarów dla porównania algorytmów MS i PRS**

```
Welch Two Sample t-test

data:  rosenbrock20Result[[2]] and rosenbrock20Result[[1]]
t = 28.298, df = 49, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 151555.9 174726.6
sample estimates:
   mean of x    mean of y
1.631412e+05 1.632733e-08
```

Podsumowanie

Algorytm MS wypadł lepiej niż algorytm PRS, znajdując większą liczbę minimów na mniejszym zakresie.