Projekt 10 - GUS

Algorytmiczne Zastosowania Łańcuchów Markowa

Bruno Podbielski, Bartosz Pokora, Franciszek Saliński

11 czerwca 2025

Opis problemu

W badaniach przeprowadzanych przez Główny Urząd Statystyczny, często ma się do czynienia z sytuacją, gdy badana populacja (której jakaś cecha nas interesuje) posiada warstwy. Przedsiębiorstwa są podzielone na branże, państwo na województwa itp. Konstruuje się estymator interesującej nas cechy, który z definicji jest nieobciążony. W takiej sytuacji poszukuje się estymatora, którego wariancja będzie najmniejsza.

Przypadek jednostopniowy

Mamy dane:

- H - liczba warstw
- A_h - współczynnik wariancji w warstwie h

Zadaniem jest minimalizacja funkcji:

$$f(x_1, ..., x_H) = \sum_{h=1}^{H} \frac{A_h^2}{x_h},$$

gdzie x_h - liczba elementów z warstwy h brana do badania, przy warunkach:

- $\sum_{h \in H} x_h = n$ (ustalona liczność próby)
- $m_h \leqslant x_h \leqslant M_h$ (ograniczenia na liczbę elementów z warstwy)

Rozwiązanie

Praca J. Wesołowski, R. Wieczorkowski, W. Wójciak *Recursive Neyman algorithm for optimum sample allocation under box constraints on sample sizes in strata* rozwiązuje ten problem. Rozwiązanie to jest zaimplementowane w R jako funkcja RNABOX w bibliotece *stratallo*.

Naszym celem było wykorzystanie algorytmu symulowanego wyżarzania do rozwiązania tego zadania i porównanie go z RNABOX.

Algorytm symulowanego wyżarzania

Wejście:

- (A_h) - współczynniki wariancji
- (m_h) - ograniczenia dolne na x_h
- (M_h) - ograniczenia górne na x_h
- n - liczność próby
- β - współczynnik schładzania
- K - stała warunku stopu

Wyjście:

- (x_h) - liczby elementów branych do próby

Korzystamy z klasycznego algorytmu symulowanego wyżarzania z planem schładzania $T_0 = 1,\ T_n = \beta^n$. Warunkiem stopu jest pozostanie K razy pod rząd w tym samym stanie.

Algorytm symulowanego wyżarzania

Przestrzenią stanów jest zbiór przyporządkowań spełniających warunki:

$$S = \{(x_1, \ldots, x_H) : \sum_{h \in H} x_h = n,\ m_h \leqslant x_h \leqslant M_h\}.$$

Łańcuch Markowa, który "wkładamy" do algorytmu ma prawdopodobieństwa przejścia $\psi_{ij},\ i, j \in S$.

$$\psi_{ij} = \frac{\xi_{ij}}{\sum_{j \in S} \xi_{ij}}, \text{ gdzie}$$

$$\xi_{ij} = \begin{cases} 1, & \text{jeśli } i \text{ oraz } j \text{ różnią się o 1 na dokładnie 2 współrzędnych} \\ 0, & \text{w.p.p.} \end{cases}$$

W efekcie przechodząc między stanami przerzucamy element z jednej warstwy do drugiej.

Algorytm symulowanego wyżarzania

Prawdopodobieństwa akceptacji dane przez algorytm symulowanego wyżarzania w n-tym kroku mają postać:

$$a_{ij}^{(n)} = \begin{cases} 1, & \text{jeżeli } \Delta_{ij} < 0 \\ exp(-\frac{\Delta_{ij}}{T_n}), & w.p.p. \end{cases}$$

gdzie $\Delta_{ij} = f(j) - f(i)$.
Finalnie prawdopodobieństwa przejścia dane przez algorytm symulowanego wyżarzania w n-tym kroku mają postać:

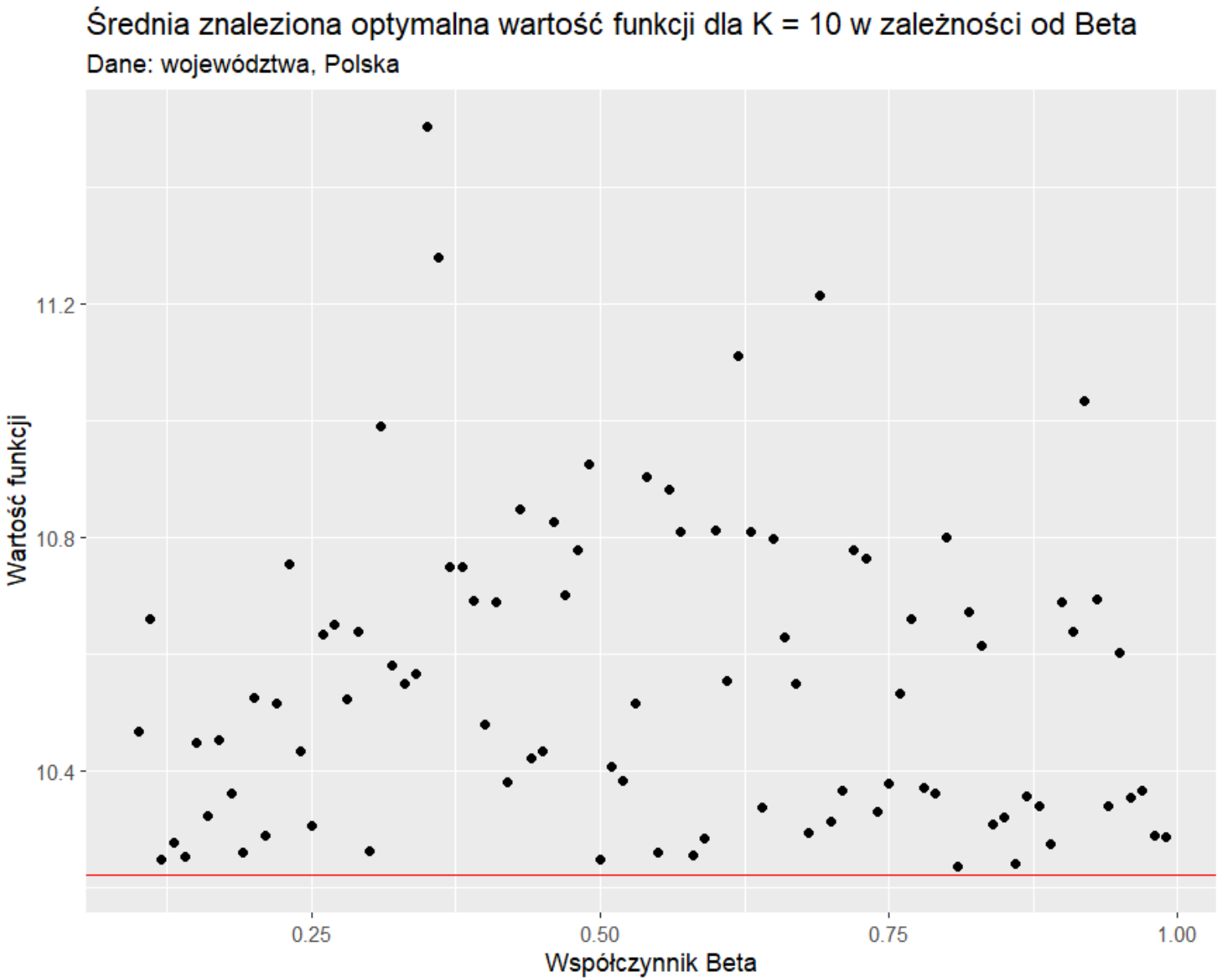
$$P_{ij}^{(n)} = \begin{cases} \psi_{ij}\ a_{ij}^{(n)}, & j \neq i \\ 1 - \sum_{k:k\neq i} \psi_{ik}\ a_{ik}^{(n)}, & j = i \end{cases}$$

Jako rozkład początkowy przyjmujemy rozkład jednostajny na S.

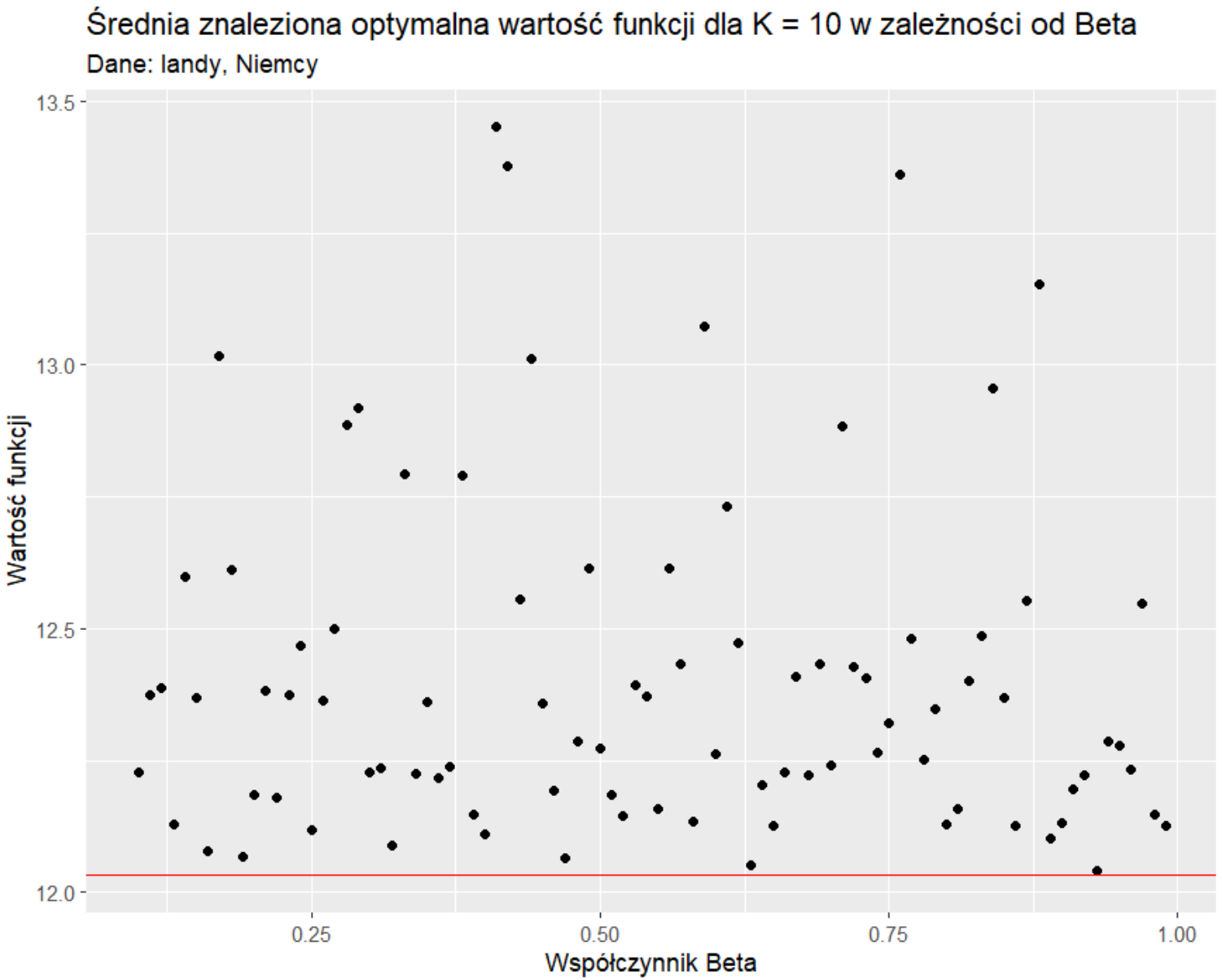
Implementacja algorytmu

- Losujemy stan początkowy $(x_1, x_2, \ldots, x_H)$
- $T \leftarrow 1$
- Dopóki nie pozostaniemy w tym samym stanie po raz K-ty z rzędu:
  - Losujemy parę warstw h_1, h_2 dopóki zamiana $x_{h_1} \leftarrow x_{h_1} - 1,\ x_{h_2} \leftarrow x_{h_2} + 1$ nie spełni ograniczeń
  - Obliczamy Δ
  - Wykonujemy krok zgodnie z $(P_{ij}^{(n)})$
  - $T \leftarrow \beta T$

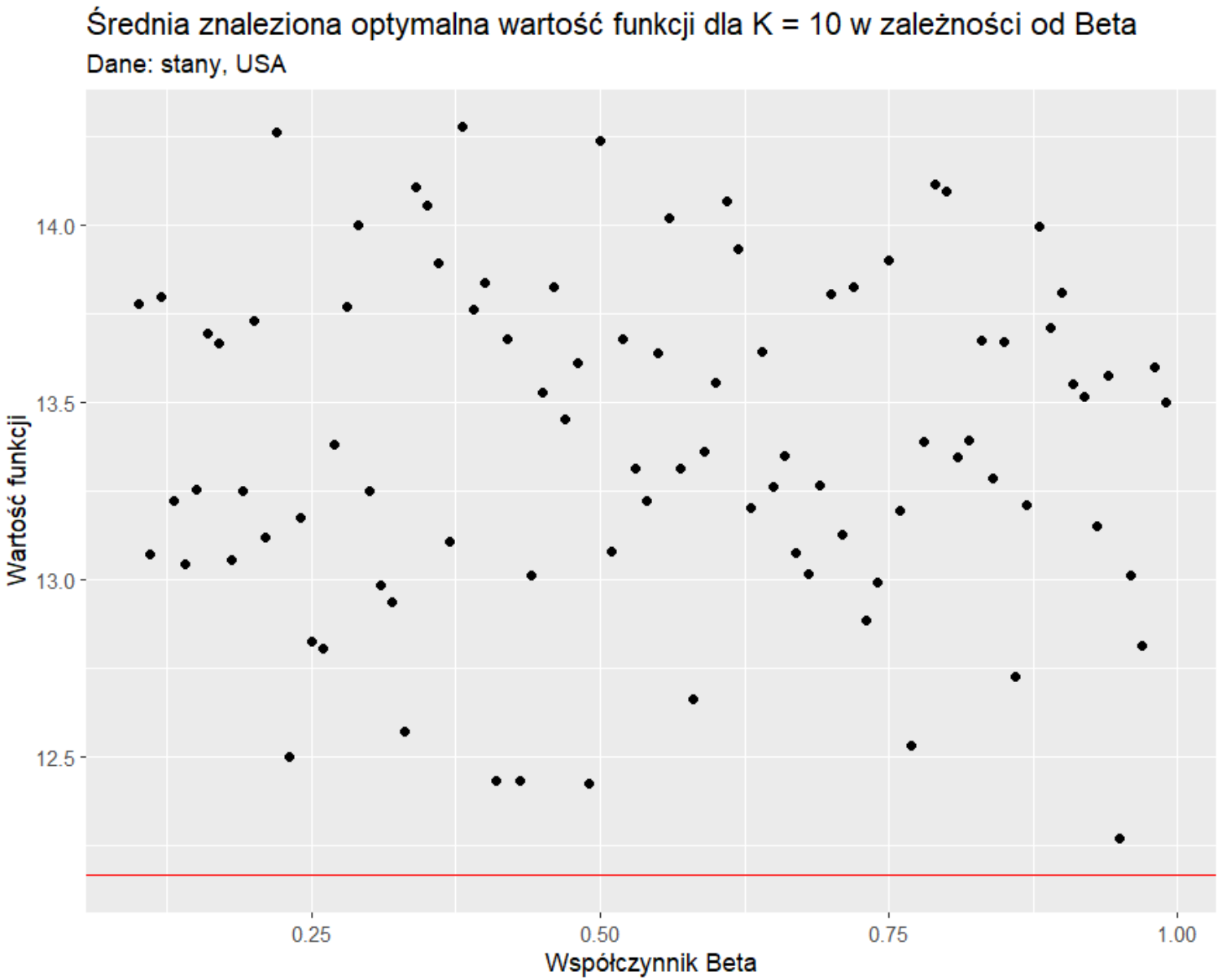
Wyniki eksperymentów



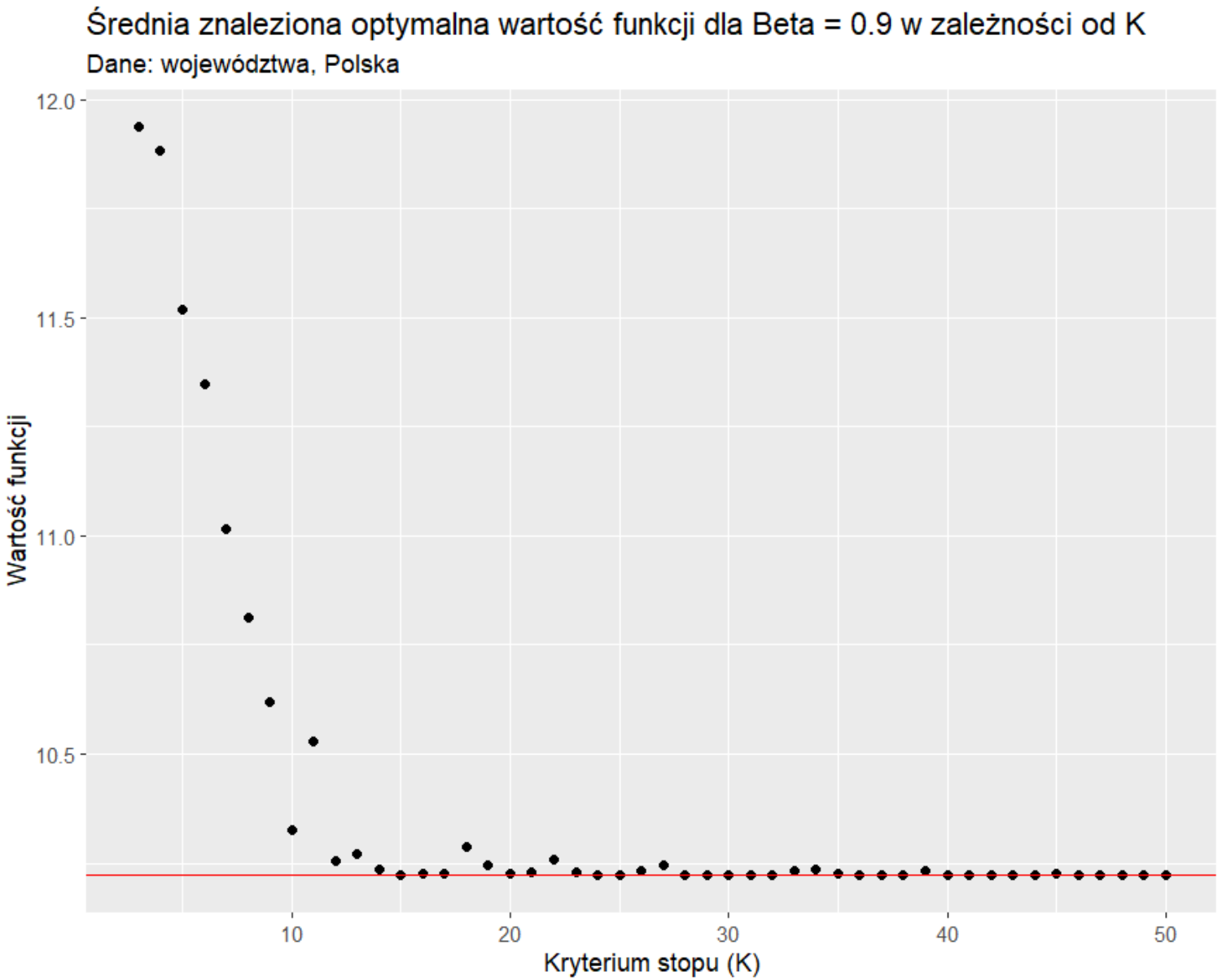
Wyniki eksperymentów



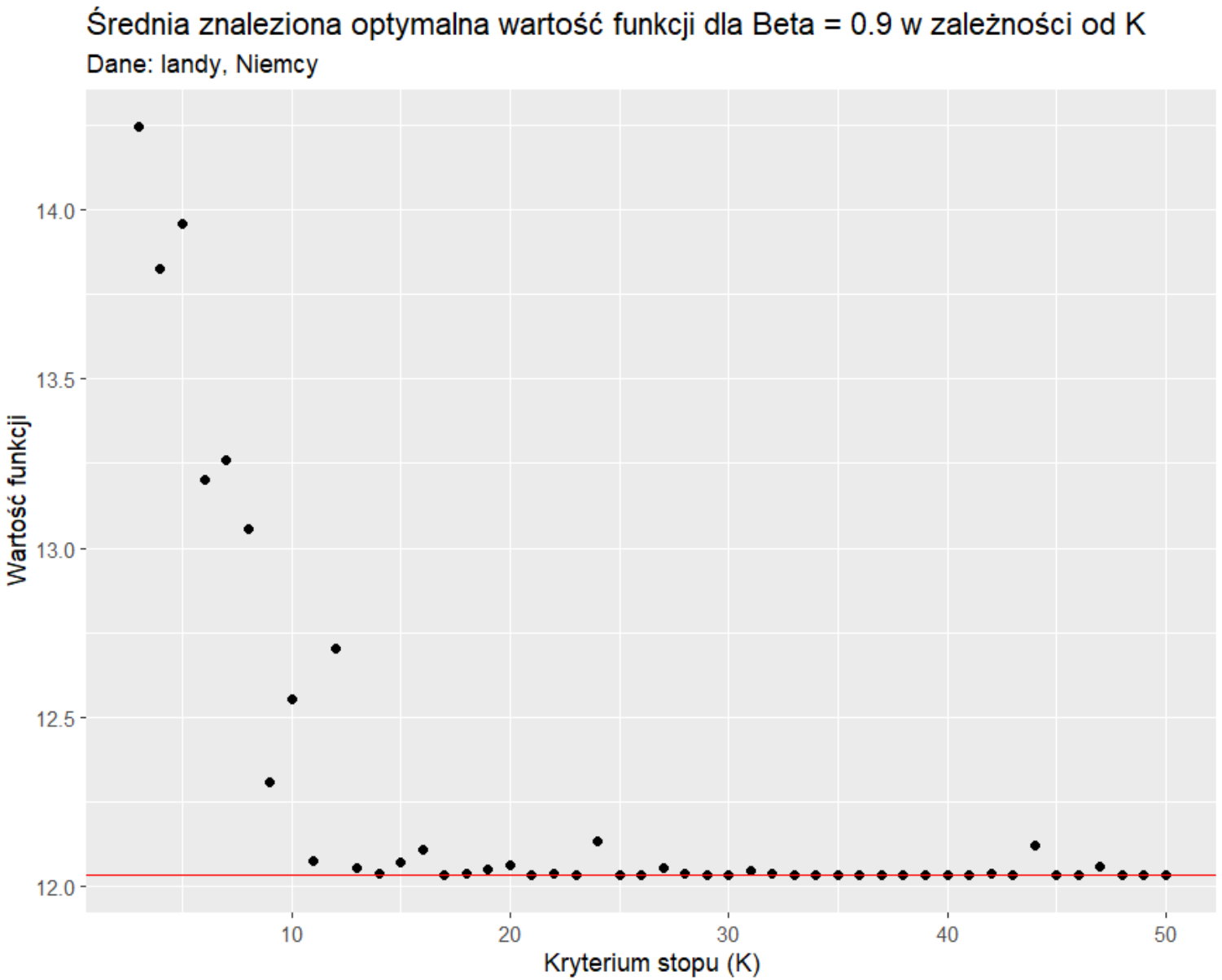
Wyniki eksperymentów



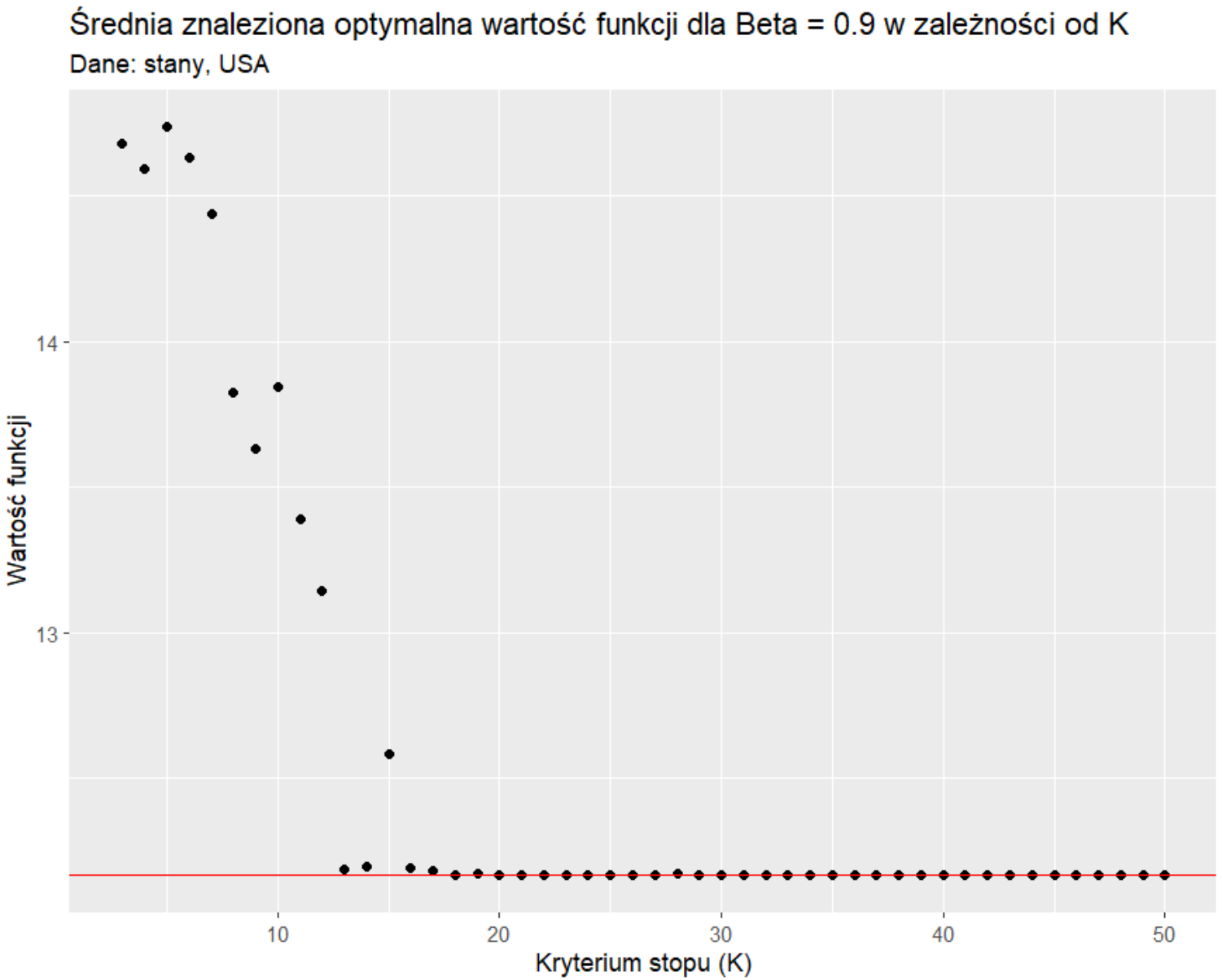
Wyniki eksperymentów



Wyniki eksperymentów



Wyniki eksperymentów



Przypadek dwustopniowy

W przypadku losowania dwustopniowego z warstwami na pierwszym stopniu (na przykład losujemy szkoły, gdzie warstwami są województwa, a następnie ze szkół losujemy uczniów) wariancja ma bardziej skomplikowaną postać:

$$f\Big((m_h),(n_{hj})\Big)=\sum_{h=1}^{H}\left(\frac{1}{m_h}-\frac{1}{M_h}\right)M_h^2D_h^2+\sum_{h=1}^{H}\frac{M_h}{m_h}\sum_{j=1}^{M_h}\left(\frac{1}{n_{h,j}}-\frac{1}{N_{h,j}}\right)N_{h,j}^2$$

- M_h to znana liczba szkół w województwie numer h oraz D_h^2 to znany współczynnik wariancji dla szkół w tym województwie.
- $N_{h,j}$ to znana liczba uczniów w j-tej szkole w h-tym województwie oraz $S_{h,j}$ to znany współczynnik wariancji dla uczniów tej szkoły.

Przypadek dwustopniowy

Naszym zadaniem jest minimalizacja powyższej wariancji (jako funkcji $m_1, \ldots, m_h$ i $n_{1,1}, \ldots, n_{h,1}, \ldots$) przy warunkach:

$$\sum_{h=1}^{H} m_h = m, \quad \text{(liczba szkół w ostatecznej próbce)},$$

$$\sum_{h=1}^{H} \frac{m_h}{M_h} \sum_{j=1}^{M_h} n_{h,j} = n \quad \text{(oczekiwana liczba uczniów w ostatecznej próbce)}.$$

Algorytm rozwiązujący powyższy problem jest znany tylko bez uwzględniania naturalnych ograniczeń górnych $m_h \leqslant M_h$ oraz $n_{h,j} \leqslant N_{h,j}$. (W. Niemiro, J. Wesołowski *FIXED PRECISION OPTIMAL ALLOCATION IN TWO-STAGE SAMPLING*)

Rozwiązanie

Postępujemy analogicznie do przypadku jednostopniowego. Tym razem przestrzenią stanów jest zbiór par $\Big((m_h),(n_{hj})\Big)$ spełniających warunki i ograniczenia. Dalej jednym z proponowanych kroków jest przeniesienie szkoły między województwami, ale dochodzi także możliwość przeniesienia ucznia między szkołami. Chcemy z pewnym p-stwem p proponować pierwszy rodzaj przejścia, a drugi z p-stwem $1-p$.

Problem 1

Problemem okazała się niekompatybilność całkowitoliczbowości z postacią drugiego warunku. Okazuje się, że przy proponowanych przejściach, takich jak zamiana szkoły między województwami lub zamiana uczniów między szkołami z dwóch różnych województw prawie zawsze powoduje zmianę wartości drugiego warunku:

$$\sum_{h=1}^{H} \frac{m_h}{M_h} \sum_{j=1}^{M_h} n_{h,j}.$$

Relaksacja

Zdecydowaliśmy się "poluzować" drugi warunek. Zamiast konkretnej liczby, wymagamy wartości z przedziału dookoła niej:

$$\sum_{h=1}^{H} \frac{m_h}{M_h} \sum_{j=1}^{M_h} n_{h,j} \in \Big(n - \varepsilon, n + \varepsilon\Big)$$

Testowaliśmy wyniki algorytmu dla różnych wartości ε. Finalnie zdecydowaliśmy się na $\varepsilon = n \cdot 0.001$. (Oczywiście dla $n < 1000$ powinno być większe, minimalnie 1.)

Problem 2

Kolejnym napotkanym problemem, było to, że algorytm zaczął nam działać w nieskończoność. Okazało się, że natrafia na skupiska sąsiadujących stanów z identyczną wartością optymalizowanej funkcji f. Wobec tego w kółko akceptował on kolejne stany z tego skupiska, nie uruchamiając w ten sposób nigdy warunku stopu.

Zdecydowaliśmy się patrzeć w warunku stopu na brak zmiany wartości funkcji celu (zerowanie się Δ), zamiast na pozostawanie w tym samym stanie.

Implementacja algorytmu

- Losujemy stan początkowy $(m_1, \ldots, m_h, n_{1,1}, \ldots, n_{h,1}, \ldots)$
- $T \leftarrow 1$
- Dopóki Δ nie będzie równa 0 po raz K-ty z rzędu:
  - Losujemy jeden z dwóch typów zamiany według p-stwa p.
  - W zależności od wyniku, losujemy parę województw lub szkół, dopóki zamiana między nimi nie spełni drugiego warunku i ograniczeń
  - Obliczamy Δ
  - Jeśli $\Delta < 0$, to wykonujemy krok
  - W przeciwnym wypadku akceptujemy proponowany krok z prawdopodobieństwem $exp(-\frac{\Delta}{T})$
  - $T \leftarrow \beta T$

Im więcej tym lepiej

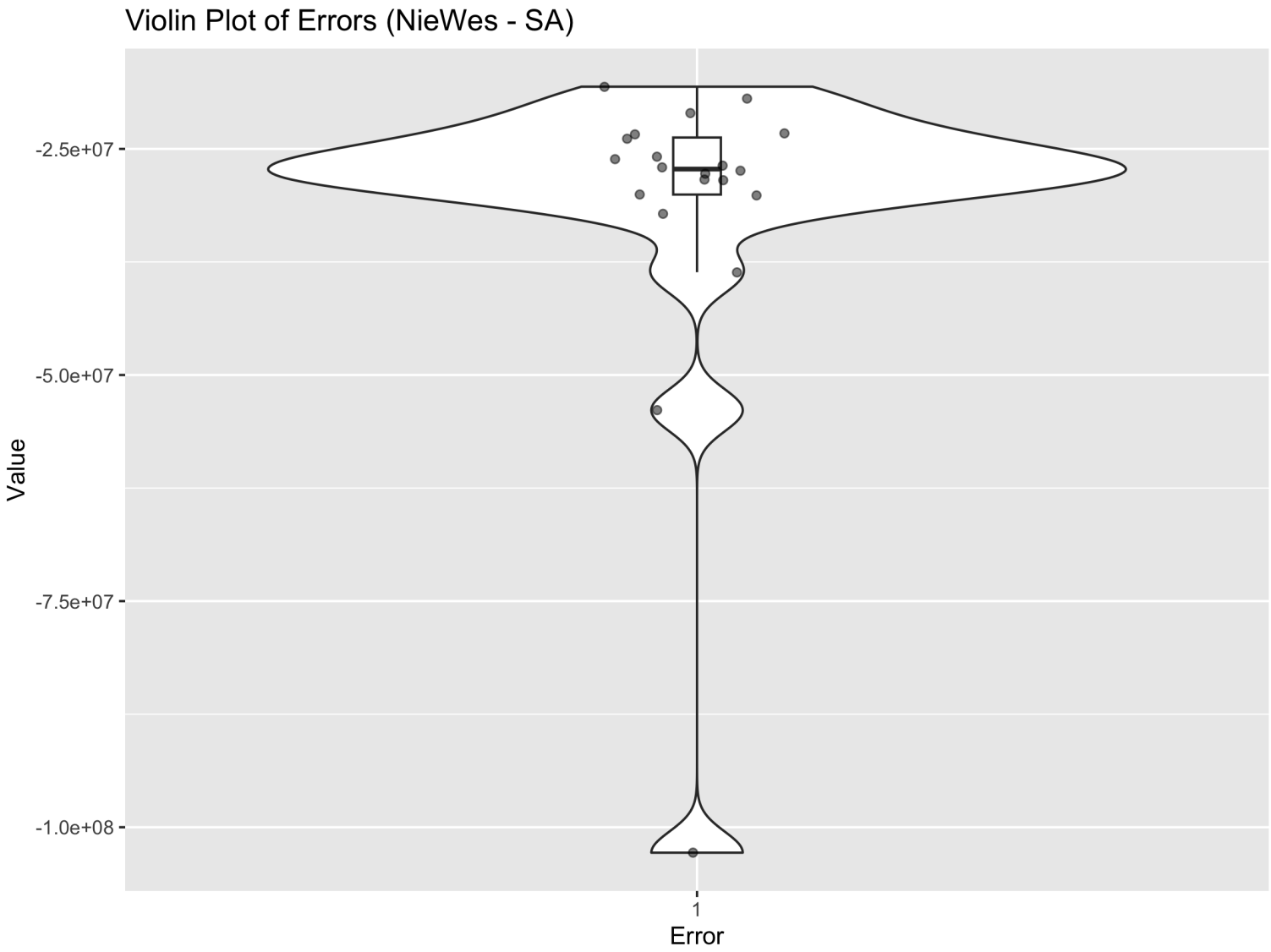
Algorytm symulowanego wyżarzania warto odpalić parę razy (tak, żeby zaczynał z różnych punktów startowych) i na koniec wziąć minimum z uzyskanych wyników. Daje to większe prawdopodobieństwo znalezienia globalnego optimum. (Algorytm często mimo wszystko kończy w minimach lokalnych).

Eksperyment

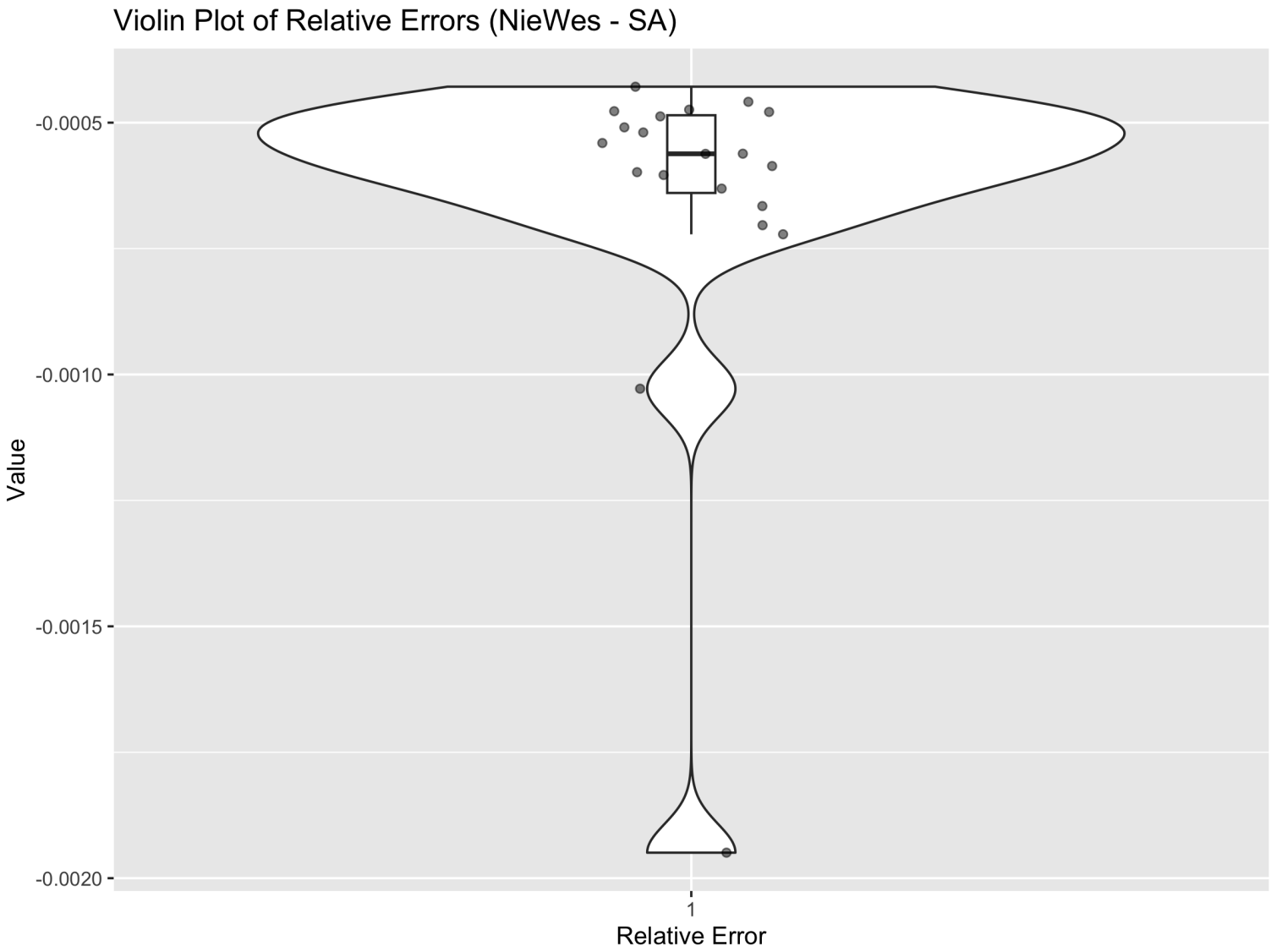
Przeprowadziliśmy eksperyment, w którym dla 20 losowo wygenerowanych zbiorów danych (o tych samych parametrach) odpaliliśmy algorytm SA po 20 razy i wzięliśmy wynik minimalizujący f. Przez SA oznaczamy wartość f uzyskaną w ten sposób, a przez NieWes wartość f dla rozwiązania analitycznego ze wspomnianej pracy. Porównujemy po kolei:

- różnicę f dla obu rozwiązań (error),
- procentową różnicę f dla obu rozwiązań (relative error),
- różnicę między f dla NieWes zaokrąglonego do liczb całkowitych, a f dla naszego SA,
- szacowaną odległość od siebie rozwiązania SA od NieWes (minimalna liczba kroków algorytmu, potrzebna, żeby przejść z jednego stanu do drugiego).

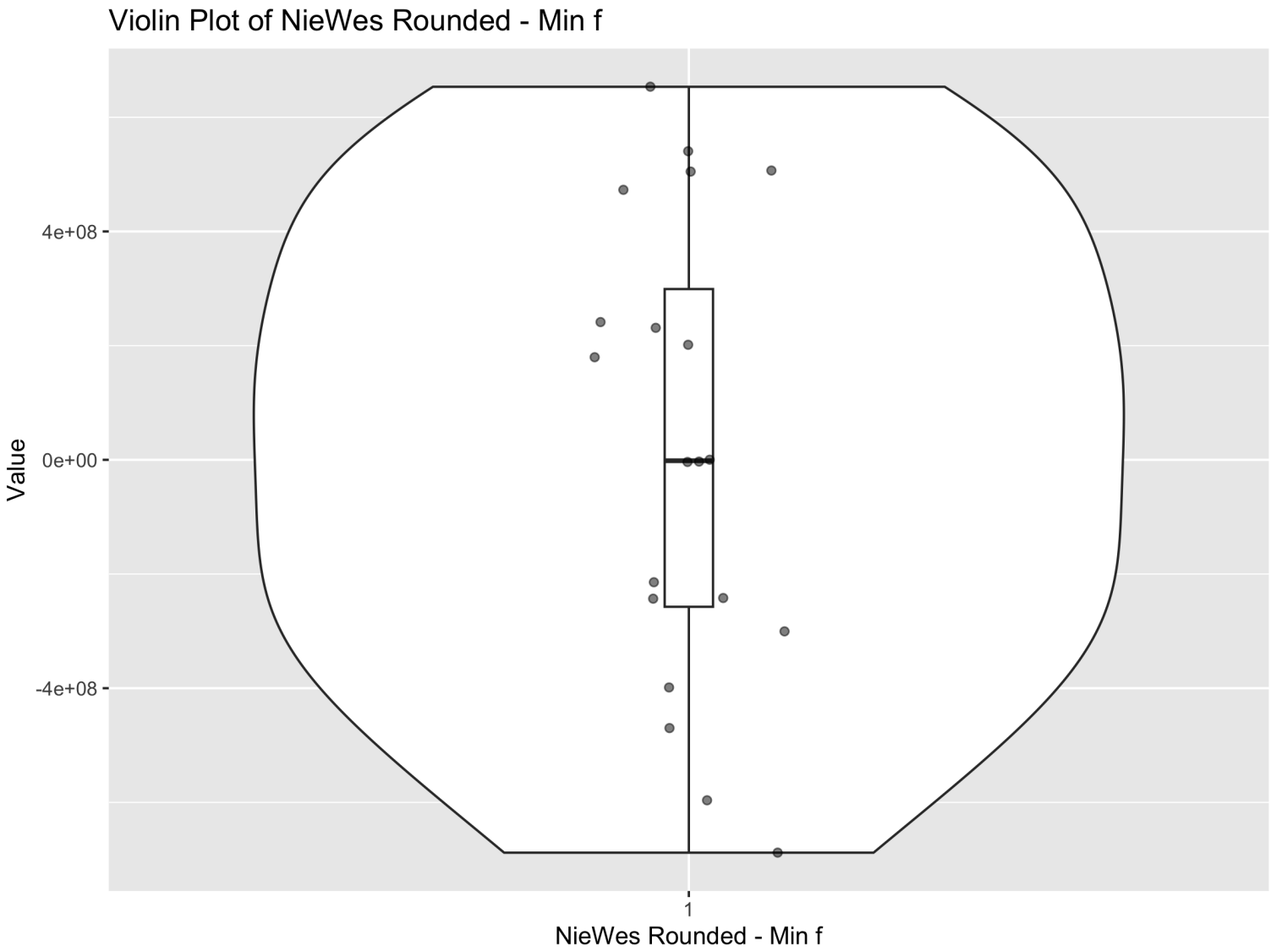
Wyniki eksperymentu



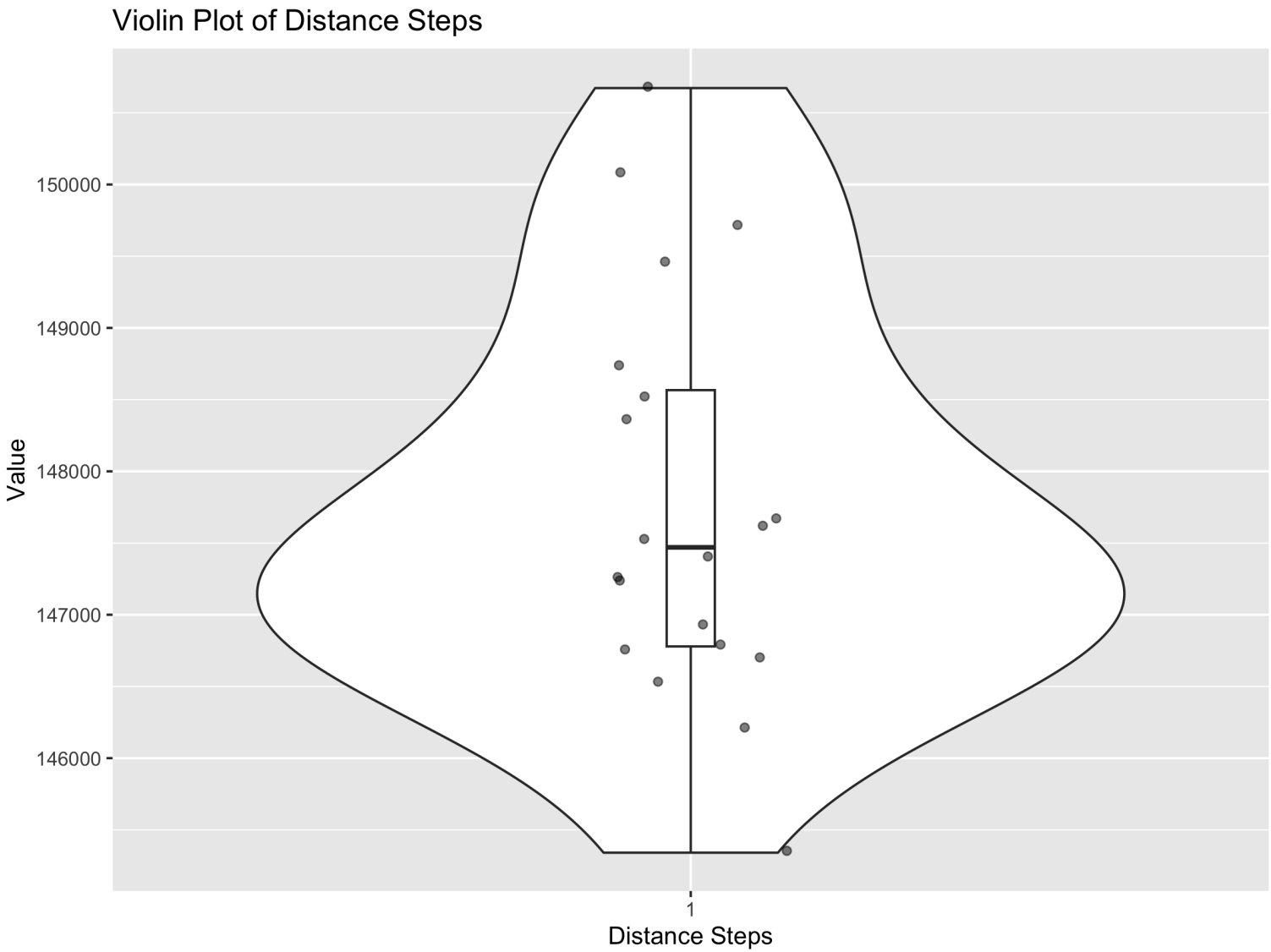
Wyniki eksperymentu



Wyniki eksperymentu



Wyniki eksperymentu



Wnioski z eksperymentu

- SA prawie zawsze zwraca istotnie różne rozwiązanie od NieWes (duży dystans na poziomie szkół).
- f dla SA jest prawie zawsze gorsze niż dla NieWes (ale nadal niewiele gorsze w skali problemu)
- za to kiedy zaokrąglimy NieWes (co jest dość sensowne z punktu praktycznego zastosowania), to już okazuje się, że SA radzi sobie średnio tak samo dobrze jak NieWes, często lepiej.