Wprowadzenie do symulacji i metod Monte Carlo

Projekt 1: Generatory liczb pseudo-losowych

Klaudia Balcer

18. grudnia 2022

Wprowadzenie

Generator liczb pseudolosowych to program generujący deterministyczny ciąg liczb, który powinien być nieodróżnialny od ciągu prawdziwie losowego. Dobre właściwości generatorów można sprawdzać poprzez badanie zagadnień testowania losowości ciągów. Wiele analiz (zarówno w świecie nauki jak i biznesu) jest obecnie przeprowadzanych w oparciu o symulacje. Dobre generatory liczb pseudo-losowych są niezbędne by uzyskać wiarygodne wyniki.

Niniejszy dokument stanowi sprawozdanie z realizacji pierwszego projektu z przedmiotu *Wprowadzenie do symulacji i metod Monte Carlo*. Projekt obejmuje zagadnienia związane z generatorami liczb pseudo-losowych i testowaniem losowości. Rozdział pierwszy obejmuje opisy badancyh generatorów. Dalej następuje opis przeprowadzonych testów wraz z testowaniem drugiego rzędu. Kolejne dwa rozdziały obejmują odpowiednio wyniki symulacji oraz wnioski.

Liczby losowe będziemy rozpatrywać na trzy odpowiadające sobie sposoby:

- liczby naturalne X_i
- liczby $U_i \in [0, 1)$
- ciągi bitów $B_i = (b_1^{(i)}, b_2^{(i)}, \ldots, b_d^{(i)}) \in \{0, 1\}^d$

Implementacja symulacji dostępna jest online: https://github.com/kabalce/PRNG

1 Generatory

Generator liczb pseudolosowych to program operujący rekurencyjnie na pewnej skończonej przestrzeni stanów. W każdej iteracji aktualny stan przekształcamy na pseudolosową wartość zadanego typu. Liczby wygenerowane w ciągu determinowanę są poprzez ziarno - początkową wartość rekurencji.

Formalnie, generator liczb pseudolosowych to piątka (S, s_0, f, U, g_0):

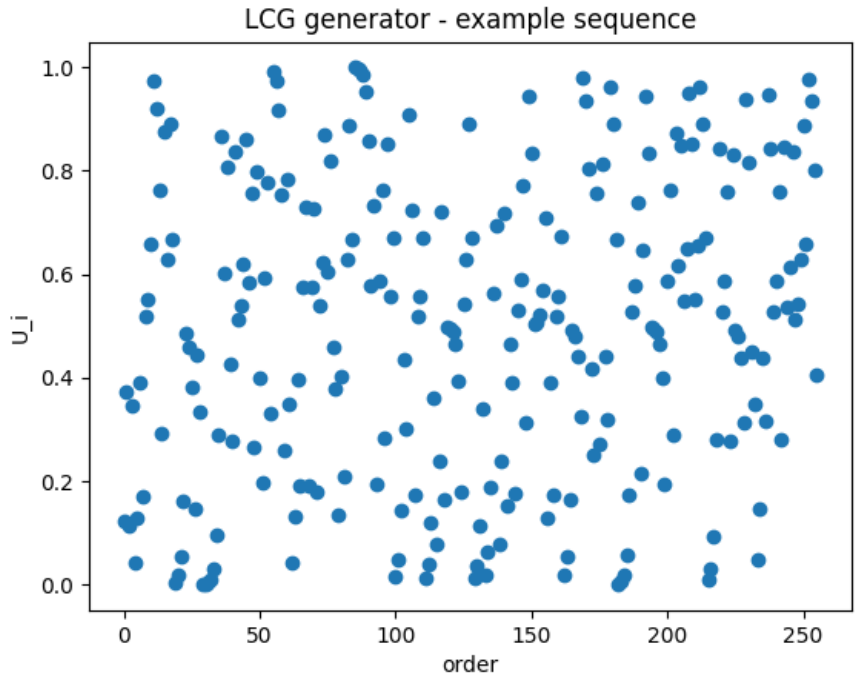
- S, taka że $|S| < \infty$ – przestrzeń stanów,
- $s_0 \in S$ – ziarno (początkowa wartość rekurencji),
- $f : S \to S, \quad f(s_i) = s_{i+1}$ – rekurencja na stanach,
- U, taka że $|U| < \infty$ – przestrzeń, z której pochodzą generowane wartości,
- $g : S \to U, \quad g(s_i) = u_i$ – funckja przekszałcająca stany w liczby pseudo losowe.

W kolejnych rozdziałach przedstawimy generatory użyte w badaniach.

1.1 LCG (a, c, M)

Linear **C**ongruential **G**enerator z paramterami a (mnożnik), c (przyrost), M (moduł):

- $S = \{0, 1, \ldots, M-1\}$,
- $s_0 \in S$ – ziarno,
- $f : S \to S, \quad f(s_i) = (a \cdot s_i + c) \bmod M$,
- $U = \{0, \frac{1}{M}, \ldots, \frac{M-1}{M}\}$,
- $g : S \to U, \quad g(s_i) = \frac{s_i}{M}$.

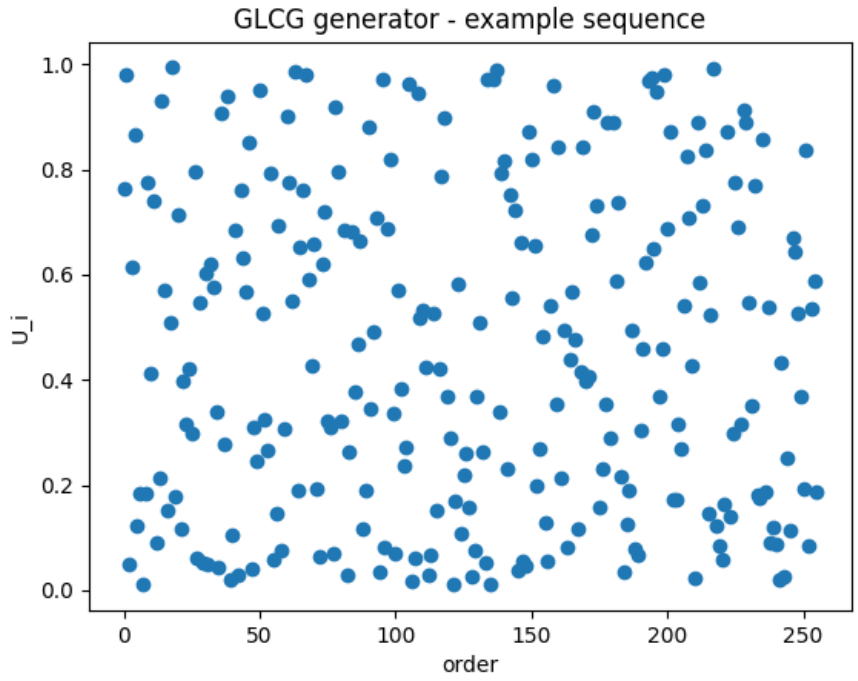


Rysunek 1: Fragment sekwencji z generatora LCG.

1.2 GLCG $(a_1, a_2, \ldots, a_k, \mathbf{M})$

Generalized **L**inear **C**ongruential **G**enerator z paramterami $a_1, a_2, \ldots, a_k$ (mnożniki), M (moduł):

- $S = \{0, 1, \ldots, M-1\}$,
- $s_0 \in S$ – ziarno,
- $f : S \to S, \quad f(s_i) = \left(\sum_{i=1}^{k} a_i \cdot s_{i-k+1}\right) \bmod M$,
- $U = \{0, \frac{1}{M}, \ldots, \frac{M-1}{M}\}$,
- $g : S \to U, \quad g(s_i) = \frac{s_i}{M}$.



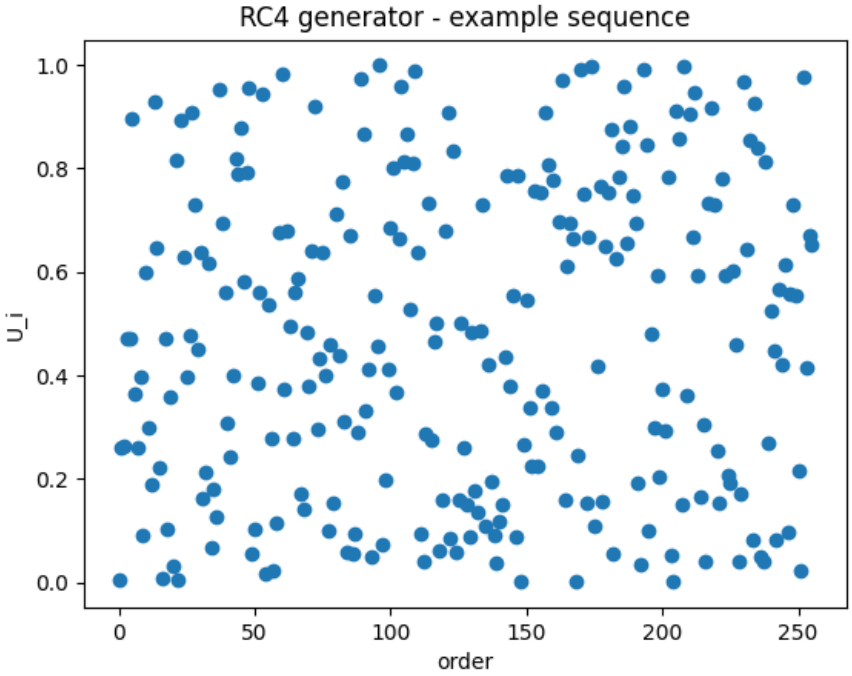
Rysunek 2: Fragment sekwencji z generatora GLCG.

1.3 RC4 (M)

Rivest **C**ipher z parametrem M (moduł):

- $S = S_M \times [M] \times [M]$, gdzie S_M to przestrzeń permutacji zbioru M-elementowego, a $[M]$ to zbiór M-elementowy $\{0, 1, \ldots, M-1\}$,
- $\sigma_0 \in S_M, i = 0, j = 0$ – ziarno,
- $f : S \to S, \quad f(\sigma, i, j) = \Big(swap(\sigma, i{+}1, j{+}\sigma[i{+}1]), \quad i{+}1, \quad j{+}\sigma[i{+}1]\Big)$, gdzie funckja *swap* robi tranzycję wskazanych elementów w podanej permutacji
- $U = \{0, \frac{1}{M}, \ldots, \frac{M-1}{M}\}$,
- $g : S \to U, \quad g(s_i) = \frac{s_i}{M}$.

Permutacja σ_0 wyznaczana jest z klucza $K = (k_l \in [M])_{l=1}^{L}$ o długości L poprzez algorytm KSA (**K**ey **S**chedule **A**lgorithm).



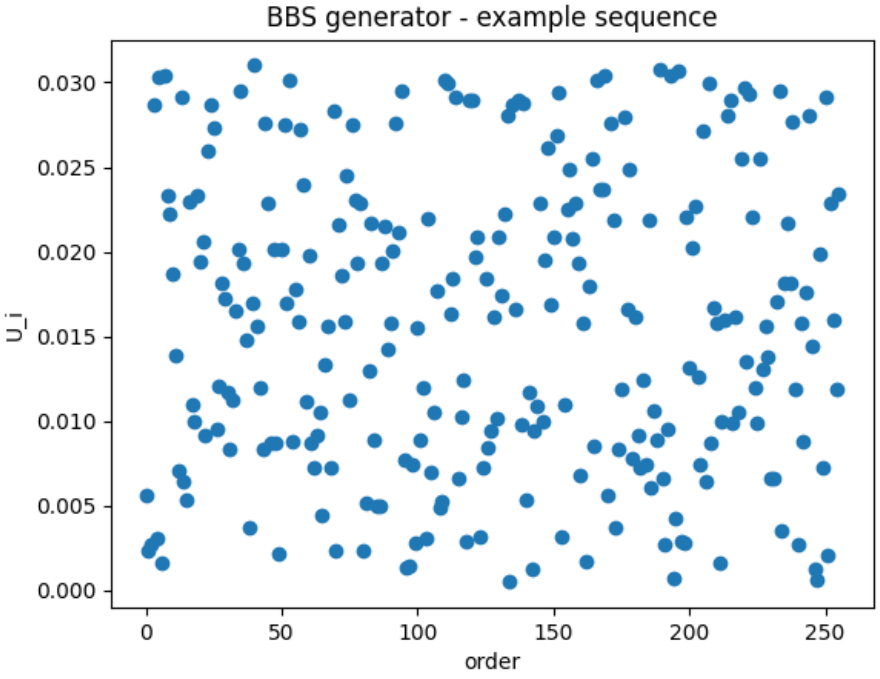
Rysunek 3: Fragment sekwencji z generatora RC4.

1.4 Blum Blum Shub

Generator **B**lum **B**lum **S**hub (BBS) z parametrem M (moduł):

- $S = \{0, 1, \ldots, M-1\}$,
- $s_0 \in S$ – ziarno,
- $f : S \to S, \quad f(s_i) = s_i^2 \bmod M$,
- $B^{(d)} = \{0,1\}^d$,
- $g : S \to B^{(d)}, \quad g(s_i) = (s_i \bmod 2^d)_2$.

Generator BBS oparty jest o dość prostą funckję rekurencji, jednakże wymaga dobrego dobranie parametrów. M to iloczyn dwóch dużych liczb pierwszych p i q, dających w dzieleniu przez 4 resztę 3. Dodatkowo, przekształcając stan s_i na zwracaną wartość patrzymy tylko na kilka (d) wybranych bitów wyniku.



Rysunek 4: Fragment sekwencji z generatora BBS.

2 Testy

Będziemy badać dobre własności generatorów testując następujące zagadnienie:

$\mathcal{H}_0$: badany ciąg jest losowy przeciwko $\mathcal{H}_1$: badany ciąg nie jest losowy.

Oczywiście, ciąg otrzymany z generatora nie jest ciągiem losowym, a jedynie może być dostatecznie zbliżony do ciągu losowego, by nie było podstaw odrzucać hipotezy zerowej.

Przeprowadzimy również testowanie drugiego rzędu (second level testing) badające zgodność empirycznego i teoretycznego (asymptotyczny) rozkładu statystyki testowej (lub p-wartości).

2.1 Test χ^2

Test χ^2 jest testem opartym na schemacie urnowym. Rozpatrzmy partycję hiperkostki $[0,1)^t$ powstałą poprzez brzegowy podział każdej współrzędnej na pewną liczbę równych odcinków. W przedstawioncyh testach będziemy rozpatrywać najprostszą partycję kostki jednowymiarowej. Dzielimy odcinek $[0,1)$ na k równych części i każdej liczbue U_i przypisujemy indeks *urny*:

$$X_i = \max_{j \in \mathbb{N}} \quad \frac{j}{k} \leqslant U_i < \frac{j+1}{k}$$

Następnie zliczamy liczbę obserwacji w każdym pudełku:

$$Y_j = \#\{X_i = j\}$$

Prawdopodobeińśtwo obserwacji zmiennej o rozkładzie jednostanym w urnie $[\frac{j}{k}, \frac{j+1}{k})$ oznaczamy poprzez p_j, wynosi ono $\frac{1}{k}$. Test χ^2 sprawdza zgodność zmiennyvh U_i z rozkładem $\mathcal{U}[0,1)$. Statystyka testowa dana jest wzorem:

$$\hat{\chi}^2 = \sum_{j=1}^{k} \frac{(Y_j - np_j)^2}{np_j} = \sum_{j=1}^{k} \frac{(Y_j - \frac{n}{k})^2}{\frac{n}{k}}$$

Rozkład statystyki $\hat{\chi}^2$ dla ustalonego k i $n \to \infty$ ma asymptotyczny rozkład χ^2_{k-1}. Odrzucamy hipotezę zerową dla dużych wartości statystyki.

Statystyka testowa ma asymptotycznie rozkład ciągły, toteż p-wartość ma asymptotycznie rozkład $\mathcal{U}[0,1)$. Testowanie drugiego rzędu mozemy przeprowadzić poprzez przeprowadzenie testu ponownie na ciągu otrzymanych p-wartości.

2.2 Test odstępów dni urodzin

Test odstępów dni urodzin również jest oparty na schemacie urnowym, tym razem będziemy rozpatrywać odstępy między statystykami pozycyjnumi $Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)}$ (gdzie Y_j jest zdefiniowane w rozdziale 2.1):

$$S_i = Y_{(i+1)} - Y_{(i)}$$

Statystyka testowa T zlicza liczbę powtórzeń odstępów:

$$T = \sum_{i=1}^{n-1} \mathbb{1}(S_{(i+1)} = S_{(i)})$$

Dla dużych n i $\lambda = \frac{n^3}{4k}$ statystyka T przy $\mathcal{H}_0$ ma w przybliżeniu rozkład $Poi(\lambda)$.

Zaleca się, żeby $n^3 \leqslant k^{\frac{5}{4}}$. Hipotezę zerową odrzucamy dla dużych wartości statystyki.

Statystyka testowa ma asymptotyczny rozkład Poissona (dyskretny), toteż testowanie drugiego rzędu przeprowadzamy poprzez test χ^2 z urnami zawierającymi podzbiory liczb naturalnych i prawdopodobieństwami z rozkładu Poissona.

Test pominięto w symulacjach.

2.3 Test Kołmogorowa-Smirnowa

Test Kołmogorowa-Smirnowa (zwany również testem λ Kołmogorowa) bada zgodność z zadanym rozkładem, w naszym przypadku $\mathcal{U}[0,1)$. Do sformułowania testu potrzebna jest nam definicji dystrybuanty empirycznej:

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(X_i \leqslant x)$$

Statystyka testowa bada rozbieżność między dystrybuantą zadanego rozkładu a dystrybuantą empiryczną:

$$\hat{D}_n = \sqrt{n} \sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F_n(x)|$$

W przypadku badania zgodność z rozkładem jednostajnym $\mathcal{U}[0,1)$ statystyka testowa ma postać:

$$\hat{D}_n = \sqrt{n} \sup_{x \in [0,1)} |\hat{F}_n(x) - x|$$

Statystyka testowa $\hat{D}_n$ ma asymptotyczny rozkład λ Kołmogorowa:

$$\mathbb{P}(\hat{D}_n > t) \longrightarrow \sum_{i=1}^{\infty} (-1)^{i-1} e^{-2i^2 t}$$

Hipotezę zerową odrzucamy dla dużych wartości statystyki.

Statystyka testowa ma asymptotycznie rozkład ciągły, toteż p-wartość ma asymptotycznie rozkład $\mathcal{U}[0,1)$. Testowanie drugiego rzędu mozemy przeprowadzić poprzez przeprowadzenie testu ponownie na ciągu otrzymanych p-wartości.

2.4 Frequency (Monobit) Test

Test częstotliwości bada równowagę między wartościami bitów. Statystyka testowa ma postać:

$$Z = \frac{1}{\sqrt{2N}} \left| \sum_{i=1}^{N} 2b_i - 1 \right|$$

gdzie N oznacza liczbę bitów w sekwencji. Statystyka T ma rozkład half-normal. Hipotezę zerową odrzucamy dla dużych wartości statystyki.

Statystyka testowa ma asymptotycznie rozkład ciągły, toteż p-wartość ma asymptotycznie rozkład $\mathcal{U}[0,1)$. Testowanie drugiego rzędu mozemy przeprowadzić poprzez przeprowadzenie testu opisanego w sekcji 2.1 wariantuχ^2 na ciągu otrzymanych p-wartości.

2.5 Runs Test

Niniejszy test bada długości serii bitów o tej samej wartości.

Pierwszą fazą testu jest sprawdzenie, czy proporcja między bitami 0 i 1 jest odpowiednio zbilansowana, by miało sens przeprowadzanie bardziej szczegółowego testu:

$$\left|\frac{\sum_{j=1}^{N} b_j}{N} - \frac{1}{2}\right| \geqslant \frac{2}{\sqrt{N}}$$

Co jest równoznaczne ze sprawdzeniem czy statystyka testowa z monobitu jest większa niż $2\sqrt{2}$, czyli przeprowadzeniem testu opisanego w rozdziale 2.4 na poziomie istotności około 0.0047.

Oznaczmy: $\pi \coloneqq \frac{\sum_{j=1}^{\dot{N}} b_j}{N}$. Oznaczmy liczbę zmnian wartości biru w sekwencji:

$$V_N = \sum_{j=1}^{N-1} \mathbb{1}(b_j \neq b_{j+1}) + 1$$

Wówczas możemy zdefiniować statystykę testową o rozkładzie half-normal:

$$T = \frac{|V_N - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}$$

Hipotezę zerową odrzucamy dla dużych wartości statystyki.

Statystyka testowa ma asymptotycznie rozkład ciągły. Biorąc pod uwagę pretestowanie, p-wartość ma asymptotycznie rozkład będący mieszanką $\mathcal{U}[0,1)$ z atomem (poza $[0,1)$) o prawdopodobieństwie o odrzuceniu hipotezy zerowej już na etapie wstępnym (oznaczmy p_0) równym około 0.0047. Oznaczmy liczbę odrzuceń hipotezy zerowej na etapie pretestowanie przez Y_0. Testowanie drugiego rzędu możemy przeprowadzić poprzez przeprowadzenie testu χ^2 z dodatkowym pudełkiem dla atomu:

$$\hat{\chi}^2 = \sum_{j=1}^{k} \frac{(Y_j - \frac{n}{k} \cdot (1-p_0))^2}{\frac{n}{k} \cdot (1-p_0)} + \frac{(Y_0 - p_0)^2}{p_0} \sim \chi^2_k$$

3 Wyniki testowania generatorów

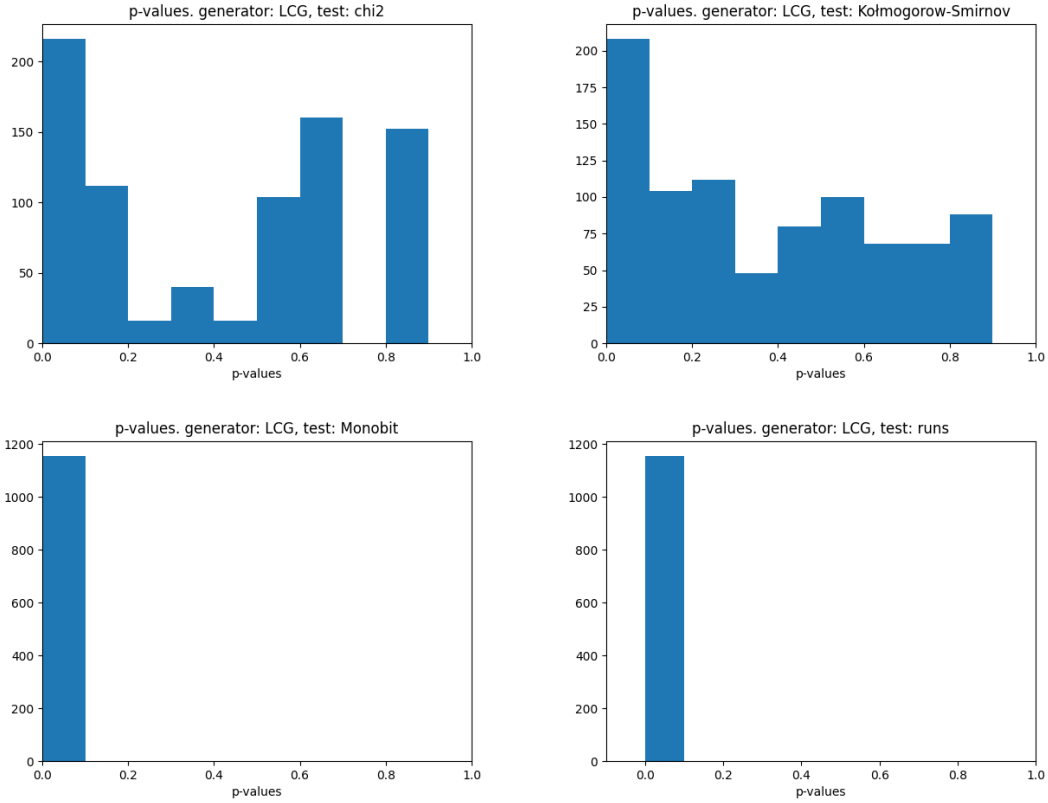
		chi2	runs	freq_mono	ks
LCG	1st_level	1.0	0.0	0.0	1.000000e+00
	2nd_level	0.0	0.0	0.0	9.074578e-16
GLCG	1st_level	2.411988e-10	0.0	0.0	0.885537
	2nd_level	0.000000e+00	0.0	0.0	0.914646
RC4	1st_level	0.873511	0.0	0.0	0.532086
	2nd_level	0.000000	0.0	0.0	0.006885
BBS	1st_level	0.777779	0.0	0.0	0.034524
	2nd_level	0.000000	0.0	0.0	0.666535

Tabela 1: Wyniki testowania losowości dla przedstawionych generatorów.

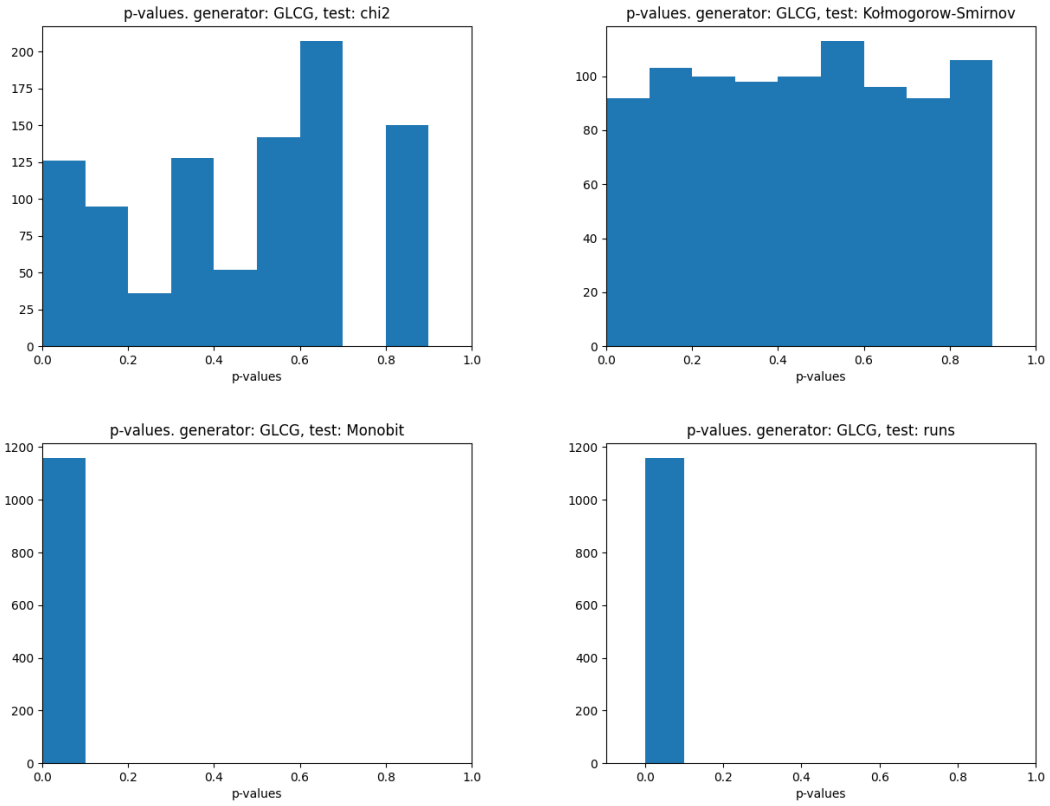
Zaproponowane generatory nie mają dobrych właściwości. Dla wszystkich generatorów, któryś z testów na poziomie istotności $\alpha = 0.01$ odrzuca hipotezę zerową o losowości ciągu. W przypadku generatorów LCG i RC4, wszystkie testy drugiego rzędu odrzucają hipotezę zerową. W przypadku generatorów GLCG hipoteza zerowa jest odrzucona w przypadku testowania pierwszego rzędu w teście chi^2 (jest to jedyne odrzucenie w tym teście w testowaniu pierwszego rzędu). Najlepiej poradził sobie generator BBS, dla którego hipoteza o losowości sekwencji została przyjęta trzy razy: w testach KS oraz w teście χ^2 pierwszego rzędu.

Testy bitowe nie wskazują żadnego zróżnicowania między generatoreami.

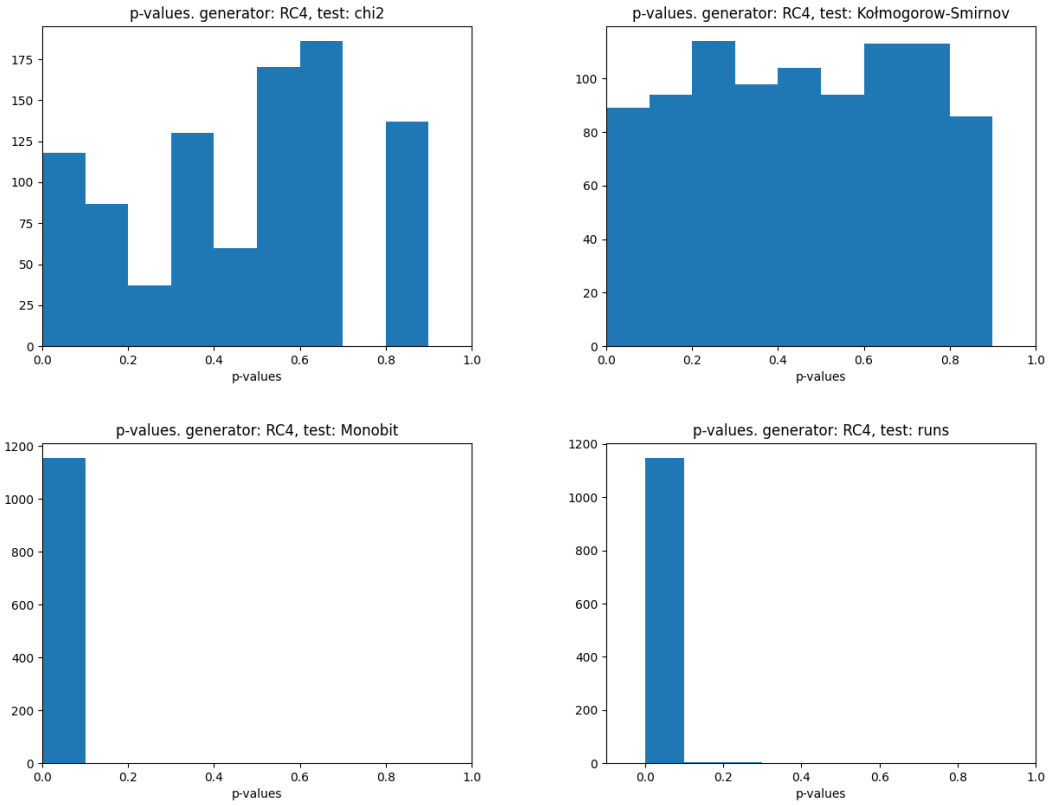
3.1 Hitogramy w testowaniu drugiego rzędu



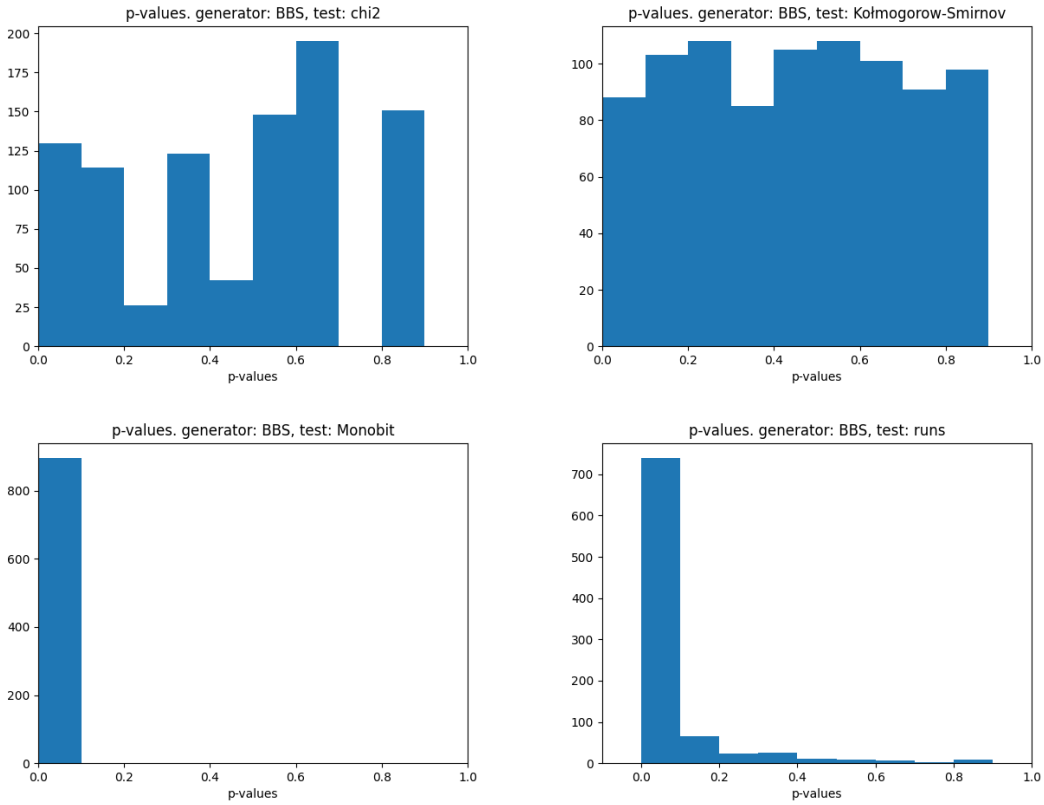
Rysunek 5: Testowanie drugiego rzędu generatora LCG.



Rysunek 6: Testowanie drugiego rzędu generatora GLCG.



Rysunek 7: Testowanie drugiego rzędu generatora RC4.



Rysunek 8: Testowanie drugiego rzędu generatora BBS.

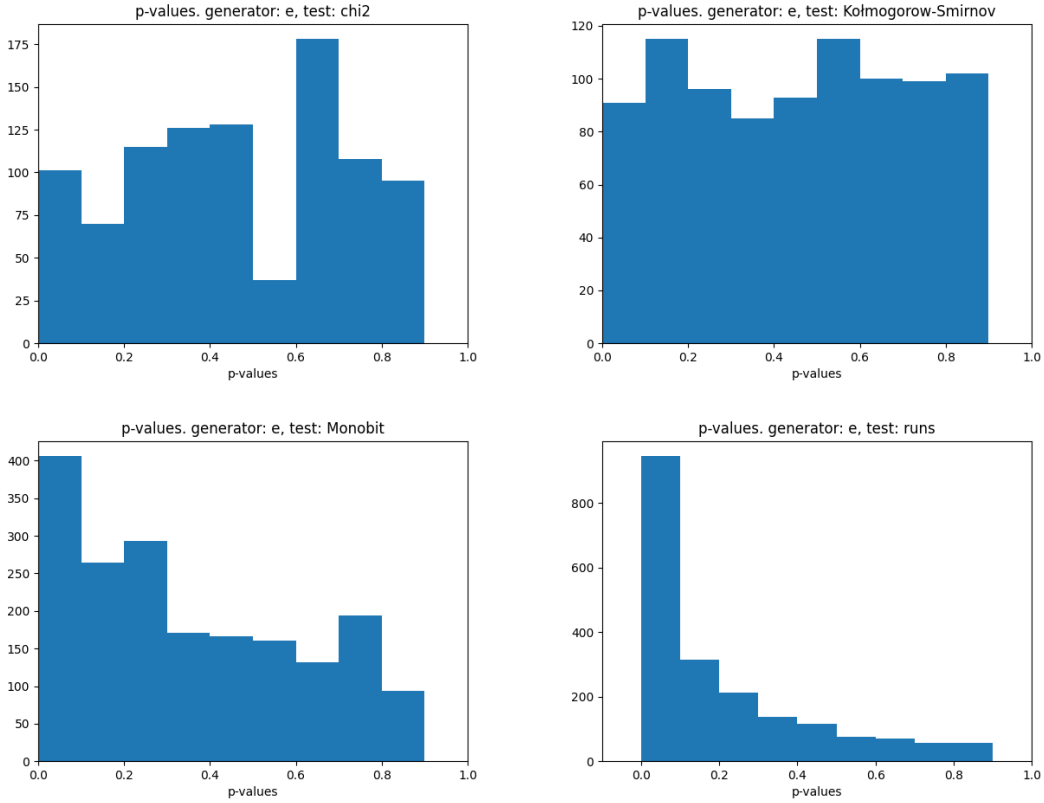
4 Wyniki testowania losowości rozwinięć dziesiętnych liczb niewymiernych

		chi2	runs	freq_mono	ks
e	1st_level	0.216982	0.0	0.0	0.38762
	2nd_level	0.000000	0.0	0.0	0.76765
π	1st_level	0.630165	0.0	0.0	0.469611
	2nd_level	0.000000	0.0	0.0	0.975881
$\sqrt{2}$	1st_level	0.642409	0.0	0.0	0.555865
	2nd_level	0.000000	0.0	0.0	0.238767

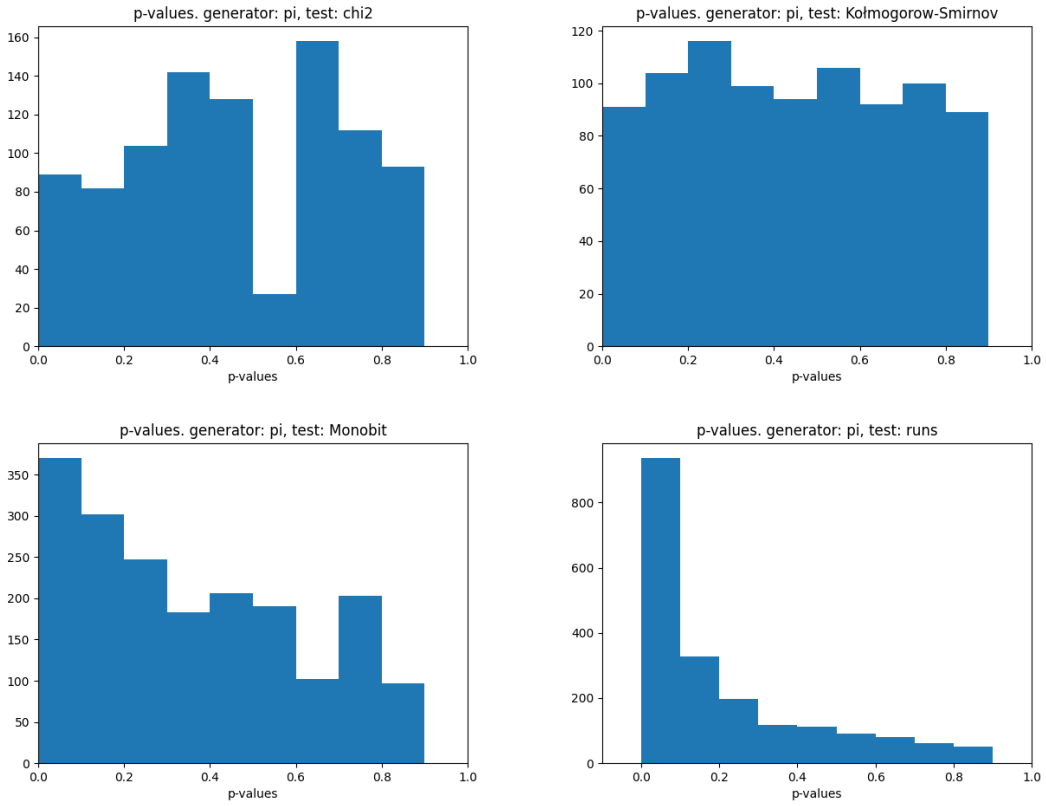
Tabela 2: Wyniki testowania lsoowości dla rozwinięć dziesiętnych liczb niewymiernych.

Rozwinięcia liczb niewymiernych dały ogólnie lepsze wyniki niż proponowane generatory. Dla wszystkich rozwinięć przyjmujemy hipotezę zerową dla trzech testów, podobnie jak dla najlepszego generatora. Ponadto, na poniższych wykresach widać znaczącą poprawę rozkładu p-wartości w testach bitowych.

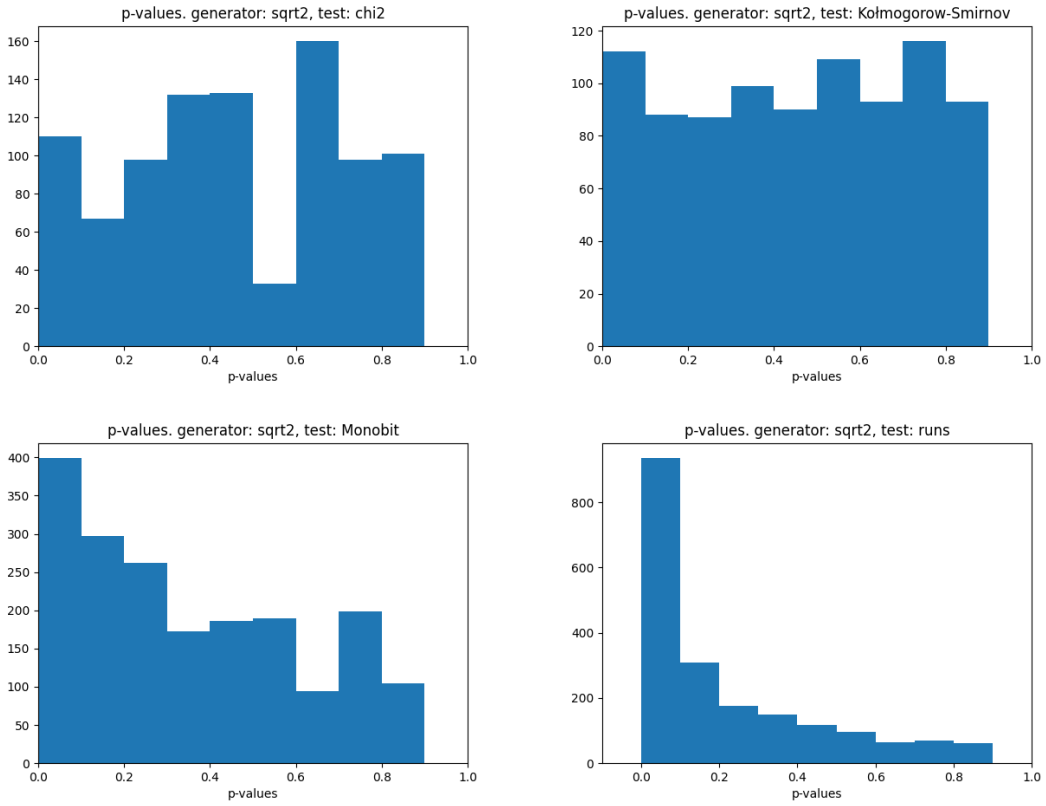
4.1 Hitogramy w testowaniu drugiego rzędu



Rysunek 9: Testowanie drugiego rzędu rozwin ięcia dziesiętnego liczby e.



Rysunek 10: Testowanie drugiego rzędu rozwin ięcia dziesiętnego liczby π.



Rysunek 11: Testowanie drugiego rzędu rozwin ięcia dziesiętnego liczby $\sqrt{2}$.