UNIWERSYTET JAGIELLOŃSKI
WYDZIAŁ MATEMATYKI I INFORMATYKI

Generatory liczb losowych z różnych rozkładów prawdopodobieństwa

Bartłomiej Szwaja
Informatyka, Rok II

Kraków 2021

Spis treści

1. Streszczenie

Projekt dotyczy implementacji generatorów liczb losowych z konkretnych rozkładów prawdopodobieństw oraz przetestowanie ich jakości.

2. Opis teoretyczny problemu

Celem projektu jest implementacja siedmiu generatorów liczb losowych:

- generator liczb całkowitych o rozkładzie jednostajnym (G),
- generator liczb z przedziału [0, 1) o rozkładzie jednostajnym (J),
- generator o rozkładzie Bernoullego (B),
- generator o rozkładzie dwumianowym (D),
- generator o rozkładzie Poissona (P),
- generator o rozkładzie wykładniczym (W),
- generator o rozkładzie normalnym (N).

Główny generator G będzie się opierał o arytmetykę modularną liczb. Skorzystamy z prostego, opracowanego w 1958 r. przez W. E. Thomson i A. Rotenberg zwanego *Liniowym Generatorem Kongruentnym*[1]. Polega on na obliczaniu kolejnych liczb pseudolosowych: x_1, x_2, …, x_n o zakresie 0, …, m-1 na podstawie poniższego wzoru:

$$x_i = (a \cdot x_{i-1} + c) \bmod m \qquad (1)$$

Wzór ten można podzielić na dwa przypadki:

- gdy $c = 0$ wtedy mamy generator mieszany,
- gdy $c \neq 0$ wtedy mamy generator multiplikatywny.

Zauważmy, że taki generator jest deterministyczny, gdyż zainicjowany tą samą wartością x_0 daje zawsze taki sam ciąg pseudolosowych liczb. Łatwo również zauważyć, iż taki generator w pewnym momencie zacznie losować wartości, które już wcześniej wylosował. Dochodzimy do wniosku, że nie może wylosować ciągu różnych liczb dłuższego niż m-1, ze względu na to, iż generowane liczby są resztą z dzielenia przez m, a takich reszt jest m, jednak x_0 jest zainicjowane stąd zostaje nam maksymalnie m-1 liczb. Dlatego też, uważa się, że m powinno być stosunkowo dużą liczbą. Ponadto z badań wynika, że najdłuższe ciągi są uzyskiwane gdy m jest potęga dwójki.

Przykłady kilku generatorów używanych w praktyce[2]:

Nazwa	**m**	**a**	**c**
Numerical Recipes	2^{32}	1664525	1013904223
Borland C/C++	2^{32}	22695477	1
GNU Compiler Collection	2^{32}	69069	5
ANSI C	2^{32}	1103515245	12345
Borland Delphi, Virtual Pascal	2^{32}	134775813	1
Microsoft Visual/Quick C/C++	2^{32}	214013	2531011
ANSIC	2^{31}	1103515245	12345

Od tego momentu w naszym projekcie wszelkie przykłady będą oparte na generatorze *Borland C/C++*.

3. Tworzenie generatorów

3.1 Generator G

Zadaniem generatora G jest zwracanie liczb całkowitych losowych z rozkładem jednostajnym. Stosujemy zatem liniowy generator kongruentny (1).

Łatwo zauważyć, że generator ten zwraca liczby z rozkładu jednostajnego gdyż każda liczba jest wylosowana tylko jeden raz. Zatem prawdopodobieństwo, że z wylosowanych liczb wybierzemy liczbę X wynosi 1/(liczbę wylosowanych liczb n):

$$P\{X\} = \frac{1}{n}$$

3.2 Generator J

Zadaniem generatora J jest zwracanie liczb z przedziału [0, 1) z rozkładu jednostajnego. Łatwo zauważyć, że możemy bezpośrednio użyć do tego celu generatora G, gdyż generuje on nam liczby z przedziału [0, m) zatem by uzyskać liczby z przedziału [0, 1) wystarczy po prostu podzielić każdą z liczb wylosowanych przez generator G przez liczbę m.

$$X_J = \frac{X_G}{m} \Rightarrow X_J \in [0, 1)$$

Z racji, że dokonujemy jedynie przekształceń na liczbach już wylosowanych to rozkład prawdopodobieństwa jest taki sam jak w przypadku generatora G.

3.3 Generator B

Zadaniem generatora B jest zwracanie liczb z rozkładu Bernoullego. Przypomnijmy definicję prawdopodobieństwa dla rozkładu Bernoullego:

$$\mathrm{P}\{\mathrm{X} = 0\} = 1 - \mathrm{p}$$

$$\mathrm{P}\{\mathrm{X} = 1\} = \mathrm{p}, 0 \leq \mathrm{p} \leq 1$$

Generator będzie zatem zwracał nam liczby 0 lub 1 z odpowiednim prawdopodobieństwem dla ustalonej liczby p. Łatwo możemy skonstruować taki generator korzystając z generatora J. Idea jest taka, że losujemy liczbę generatorem J a następnie porównujemy ją z liczbą p; jeśli jest mniejsza od p to zwracamy 1, zaś w przeciwnym wypadku zwracamy 0. Dzięki temu otrzymujemy rozkład Bernoullego.

3.4 Generator D

Zadaniem generatora D jest zwracanie liczb z rozkładu dwumianowego. Przypomnijmy definicję prawdopodobieństwa dla rozkładu dwumianowego:

$$p(i) = \binom{n}{i} p^i (1-p)^{n-i}, \quad i = 0, 1, \ldots, n \text{ (dla ustalonych } n, p)$$

W skrócie, rozkład ten opisuje liczbę sukcesów w **n** niezależnych próbach, z których każda ma stałe prawdopodobieństwo sukcesu równe **p**.

Zatem by stworzyć generator o takim rozkładzie prawdopodobieństwa posłużymy się generatorem B z poprzedniego punktu.

Idea jest taka by uruchomić generator B **n** razy i zliczyć liczbę sukcesów, czyli liczb będących mniejszymi od **p**. Liczba sukcesów będzie naszą liczbą losową.

3.5 Generator P

Zadaniem generatora P jest zwracanie liczb z rozkładu Poissona. Przypomnijmy definicję prawdopodobieństwa dla rozkładu Poissona:

$$p(i) = P\{X = i\} = \frac{e^{-\lambda}\lambda^{i}}{i!}, \quad i = 0, 1, 2, \ldots (\text{dla } \lambda > 0)$$

Możemy nie mieć pomysłu w jaki sposób utworzyć taki generator. Jednak z pomocą przyjdzie nam *metoda odwracania dystrybuanty*[3].

Wiemy, że dystrybuanta określonego rozkładu prawdopodobieństwa jest funkcją:

$$F : R \to R$$

Niemalejącą i prawostronnie ciągłą

$$\lim_{x \to -\infty} F(X) = 0$$

$$\lim_{x \to \infty} F(X) = 1$$

Dystrybuanta jednoznacznie definiuje rozkład prawdopodobieństwa.

Związek pomiędzy dystrybuantą a gęstością prawdopodobieństwa f(x):

$$F(x) = \int_{-\infty}^{x} f(y)\, dy$$

Jeśli uda nam się znaleźć F^{-1} to:

$$U = F(x) \to x = F^{-1}(U)$$

Zmienna losowa x ma rozkład o dystrybuancie F.

U jest zmienną losową o rozkładzie równomiernym w przedziale [0, 1).

Nową zmienną losową będzie:

$$X = F^{-1}(U) \qquad (2)$$

Dowód:

$$P\{X \leq x\} = P\{F^{-1}(U) \leq x\} = P\{U \leq F(x)\} = F(x)$$

Generujemy więc ciąg liczb pseudolosowych:

$$U_1, U_2, \dots, U_n \in [0, 1)$$

Który przekształcamy w ciąg:

$$X_1, X_2, \dots, X_n \in (-\infty, \infty)$$

Wówczas liczby X_i mają rozkład prawdopodobieństwa o dystrybuancie F.

Odwracanie dystrybuanty można wykorzystać także w przypadku rozkładów dyskretnych (w generatorze P właśnie o to nam chodzi, gdyż rozkład Poissona jest rozkładem dyskretnym).

Na przykład ciąg zmiennych:

$$X_1, X_2, \dots, X_n$$

o rozkładzie

$$p_k = P\{X = k\}, \quad k = 0, 1, 2, \dots$$

Można wygenerować przy użyciu ciągu

$$U_1, U_2, \dots, U_n \in [0, 1)$$

Korzystając ze wzoru

$$X_n = \min\left\{k: U_n \leq \sum_{i=0}^{k} p_i\right\}, \quad n = 1, 2, \dots. \tag{3}$$

W ten właśnie sposób utworzymy nasz generator P korzystając z metody odwrotnej dystrybuanty dla rozkładu dyskretnego. Losujemy **n** liczb generatorem J i stosujemy wzór (3) by wyznaczyć X_i.

3.6 Generator W

Zadaniem generatora W jest zwracanie liczb z rozkładu wykładniczego. Przypomnijmy funkcję skumulowaną (dystrybuantę) dla rozkładu wykładniczego:

$$F(x) = \int_0^x \lambda e^{-\lambda t}\, dt = 1 - e^{-\lambda x}, \quad x > 0$$

Skorzystamy z *metody odwrotnej dystrybuanty* (2), by wyznaczyć zmienną losową o rozkładzie wykładniczym.

$$U = F(X)$$

$$U = 1 - e^{-\lambda x}$$

$$1 - U = e^{-\lambda x}$$

$$-\lambda x = \ln(1 - U)$$

$$\boldsymbol{x = -\frac{1}{\lambda}\ln(1 - U)}$$

Ponieważ U ∈ [0, 1) to bezpiecznie możemy wykonać to działanie. Zatem idea jest taka, że najpierw losujemy liczbę rzeczywistą generatorem J, a następnie *przekształcamy* ją do rozkładu wykładniczego dzięki równaniu powyżej.

3.7 Generator N

Zadaniem generatora N jest zwracanie liczb z rozkładu normalnego. Przypomnijmy funkcję gęstości prawdopodobieństwa dla rozkładu normalnego:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right), \quad -\infty < x < \infty$$

Nietrudno zauważyć, że jest to dosyć skomplikowany wzór, zatem należy się spodziewać, że i dystrybuanta (całka z tego wzoru) również nie będzie należała do prostych wzorów. Z tego powodu, *metoda odwrotnej dystrybuanty* może okazać się zawodna. Dlatego w tym wypadku, skorzystamy z *transformacji Boxa-Mullera*[4]. Jest to metoda generowania liczb losowych o rozkładzie normalnym standaryzowanym, na podstawie dwóch wartości zmiennej o rozkładzie jednostajnym na przedziale (0, 1].

Niech U_1 oraz U_2 będą niezależnymi zmiennymi losowymi o rozkładzie jednostajnym na przedziale (0, 1]. Niech zmienne R oraz Θ dane w odpowiednim układzie współrzędnych polarnych spełniają równania:

$$R^2 = -2\ln(U_1)$$

$$\Theta = 2\pi U_2$$

Wówczas zmienne losowe Z_1 oraz Z_2 są niezależne i o rozkładzie normalnym standaryzowanym:

$$Z_1 = R\cos\Theta$$

$$Z_2 = R\sin\Theta$$

Jednakże naszym celem jest by wygenerowane liczby były o rozkładzie normalnym, czyli dowolne σ oraz μ. Dlatego korzystamy z równości[5], która opisuje zależność między rozkładem normalnym a rozkładem normalnym standaryzowanym:

$$Z = \frac{(X - \mu)}{\sigma}$$

Gdzie Z to zmienna losowa o rozkładzie normalnym standaryzowanym, natomiast X to zmienna o rozkładzie normalnym z parametrami σ oraz μ. My chcemy wyznaczyć zmienną X.

$$X - \mu = Z \cdot \sigma$$

$$\boldsymbol{X = Z \cdot \sigma + \mu}$$

Zatem nasz generator N na początku wylosuje dwie liczby generatorem J. Następnie obliczy R oraz Θ by wyznaczyć Z_1 lub Z_2. Po czym obliczy zmienną X korzystając ze wzoru powyżej. Przyjmujemy, że nasz generator zwróci jedną liczbę - dlatego będziemy obliczać tylko zmienną Z_1.

4. Implementacja generatorów

Generatory zostały zaimplementowane w języku C++. Ich implementacje znajdują się w pliku *generatory.cpp*. Zostały również zaimplementowane funkcje, które umożliwiają wyeksportowanie wylosowanych liczb do pliku o rozszerzeniu txt.

5. Testy statystyczne

Testy statystyczne[6] pozwalają na stwierdzenie czy dana hipoteza badawcza, zwana hipotezą zerową, może zachodzić. Podczas sprawdzania hipotez statystycznych wyróżnia się dwa rodzaje błędów:

- błąd I rodzaju – odrzucenie hipotezy zerowej w przypadku gdy jest ona prawdziwa – błąd ten określa się tzw. *poziomem istotności α*, czyli prawdopodobieństwem wystąpienia błędu I rodzaju.

- błąd II rodzaju – przyjęcie hipotezy zerowej w przypadku gdy jest ona fałszywa – błąd ten określa się symbolem β i jest on równy prawdopodobieństwu wystąpienia błędu II rodzaju.

Wyróżniamy dwa rodzaje testów statystycznych:

- testy parametryczne[7] - najczęściej weryfikują sądy o takich parametrach populacji jak średnia arytmetyczna, wskaźnik struktury i wariancja. Testy te konstruowane są przy założeniu znajomości postaci ogólnej dystrybuanty w populacji.

- testy nieparametryczne[7] - służą do weryfikacji różnorodnych hipotez, dotyczących m.in. zgodności rozkładu cechy w populacji z określonym rozkładem teoretycznym, zgodności rozkładów w dwóch populacjach, a także losowości doboru próby.

W naszych badaniach skorzystamy z testów nieparametrycznych; gdyż będziemy chcieli zbadać czy wylosowane liczby, przez kolejne generatory, są zgodne z rozkładem jaki te generatory są zobligowane generować; oraz będziemy chcieli sprawdzić losowość liczb generowanych przez te generatory. Do tego celu posłużymy się testem chi-kwadrat oraz testem serii.

5.1 Test chi-kwadrat

Test chi-kwadrat jest testem pokazującym relację np. pomiędzy dwoma rozkładami, badanym i teoretycznym. Rezultatem testu będzie liczba mówiąca o tym jak bardzo badany rozkład różni się od teoretycznego. Niska wartość tego testu będzie świadczyła o tym, że istnieje duża korelacja pomiędzy naszymi badanymi wartościami (otrzymanymi, czyli wylosowanymi liczbami) a wartościami spodziewanymi z danego rozkładu.

By w pełni stwierdzić czy dana wartość testu (*wartość empiryczna*) pozwala na zaakceptowanie hipotezy badawczej, musimy ją porównać z tzw. *wartością krytyczną* znajdującą się w tabeli rozkładu chi-kwadrat. Jeśli wartość empiryczna będzie mniejsza od wartości krytycznej, wtedy

będziemy mogli zaakceptować naszą hipotezę badawczą, w przeciwnym wypadku będzie należało ją odrzucić.

Jak zatem należy wyliczać wartość empiryczną? Do tego celu posłuży nam wzór:

$$\chi_c^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

gdzie

c – liczba stopni swobody[8] - liczba niezależnych wyników obserwacji pomniejszona o liczbę związków, które łączą te wyniki ze sobą; wyliczana według wzoru: c = n – d – 1 (d oznacza liczbę oszacowań).

n – liczba kategorii - u nas będą to zwykle przedziały liczbowe lub unikalne liczby.

O_i – ang. *observed*, zaobserwowana liczba danych należących do danej kategorii i.

E_i – ang. *expected*, oczekiwana liczba danych należących do danej kategorii i wyliczona dla konkretnego rozkładu prawdopodobieństwa. Obliczamy ją w sposób następujący: liczba danych * prawdopodobieństwo rozkładu.

Tabelka rozkładu chi-kwadrat dla pierwszych 10 stopni swobody:

Liczba stopni swobody	χ^2										
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.63	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.61	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.81	11.34	16.27
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88
10	3.94	4.87	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59
α	0.95	0.90	0.80	0.70	0.50	0.30	0.20	0.10	0.05	0.01	0.001

Zwykle przyjmuje się wartość α jako 0.05. Na tej liczbie również będziemy opierać nasze badania.

Przeprowadzimy teraz testy dla konkretnych generatorów. Dla każdego testu będziemy losować 1000 liczb. Testy będą przeprowadzone z danymi początkowymi generatora:

$x_0 = 3$

$m = 2^{32}$

$a = 22695477$

$c = 1$

Dla generatora G zaprezentujemy tabelkę dzięki której obliczymy wartość empiryczną testu chi-kwadrat. Nie będziemy prezentować tej tabelki dla pozostałych generatorów ze względu na jej dużą wielkość na stronie. Taką tabelkę można wygenerować korzystając z implementacji testu chi-kwadrat znajdującego się w pliku *test_chiKwadrat.cpp* napisanego w języku C++. Przy czym dane do testu są wczytywane z pliku tekstowego. Dlatego też, w pliku *generatory.cpp* jest możliwość wyeksportowania wylosowanych liczb do pliku tekstowego.

5.1.1 Generator G

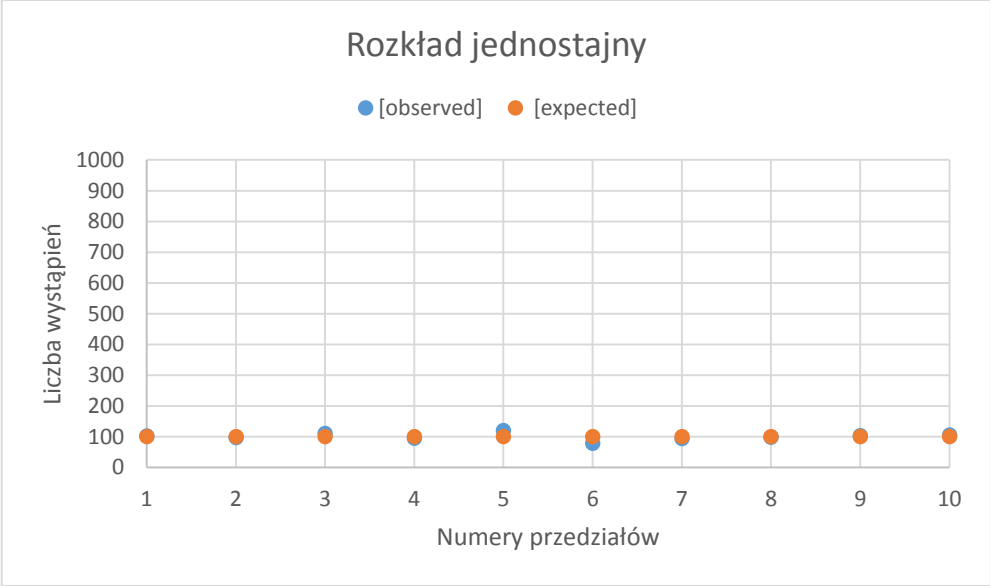
Tabelka do obliczenia wartości testu chi-kwadrat:

Nr przedziału	Przedział	Observed	Expected	(Obs-Exp)^2	(Obs-Exp)^2/Exp
1	[0.0, 429496729.5]	102	100	4	0,04
2	(429496729.5, 858993459.0]	97	100	9	0,09
3	(858993459.0, 1288490188.5]	110	100	100	1
4	(1288490188.5, 1717986918.0]	95	100	25	0,25
5	(1717986918.0, 2147483647.5]	119	100	361	3,61
6	(2147483647.5, 2576980377.0]	78	100	484	4,84
7	(2576980377.0, 3006477106.5]	93	100	49	0,49
8	(3006477106.5, 3435973836.0]	98	100	4	0,04
9	(3435973836.0, 3865470565.5]	103	100	9	0,09
10	(3865470565.5, 4294967295.0]	105	100	25	0,25

Wartość empiryczna = 10.7.

Odczytujemy wartość krytyczną z tabelki dla c = 10-0-1 = 9. Wynosi ona 16.92. Widzimy, że nasza otrzymana wartość jest od niej mniejsza zatem możemy zaakceptować naszą hipotezę iż wylosowane liczby są o rozkładzie jednostajnym.

Poglądowy rezultat testu chi-kwadrat możemy zilustrować graficznie poprzez wykres liczby wystąpień od numeru przedziału. Pierwszy wykres będzie przyjmował wartości zaobserwowane, zaś drugi wartości oczekiwane. Im bliżej te wartości będą siebie tym mniejsza będzie wartość empiryczna, zatem tym bardziej prawdopodobne, że hipoteza będzie prawdziwa.



Widzimy, że wartości się wręcz pokrywają, zatem nasze wnioski zgadzają się z tymi otrzymanymi z tabelki. Czyli wylosowane liczby mają rozkład jednostajny.

Łatwo również zauważyć, że wykres przedstawia rozkład prawdopodobieństwa, zatem nie wykonując testu chi-kwadrat moglibyśmy już na podstawie samego wykresu stwierdzić, że liczby mają rozkład jednostajny. Będąc bardziej dokładnym, nasz wykres jest jedynie poglądowy, gdyż przy wykresie rozkładu prawdopodobieństwa na osi pionowej byłyby wartości nieprzekraczające 1, gdyż prawdopodobieństwo nie może być większe niż 1. Jednakże wykres taki wyglądałby tak samo jak nasz, ponieważ wystarczyłoby jedynie podzielić liczbę wystąpień przez liczbę wszystkich danych. Nastąpiłoby więc skalowanie, które nie zmieniłoby kształtu wykresu.

5.1.2 Generator J

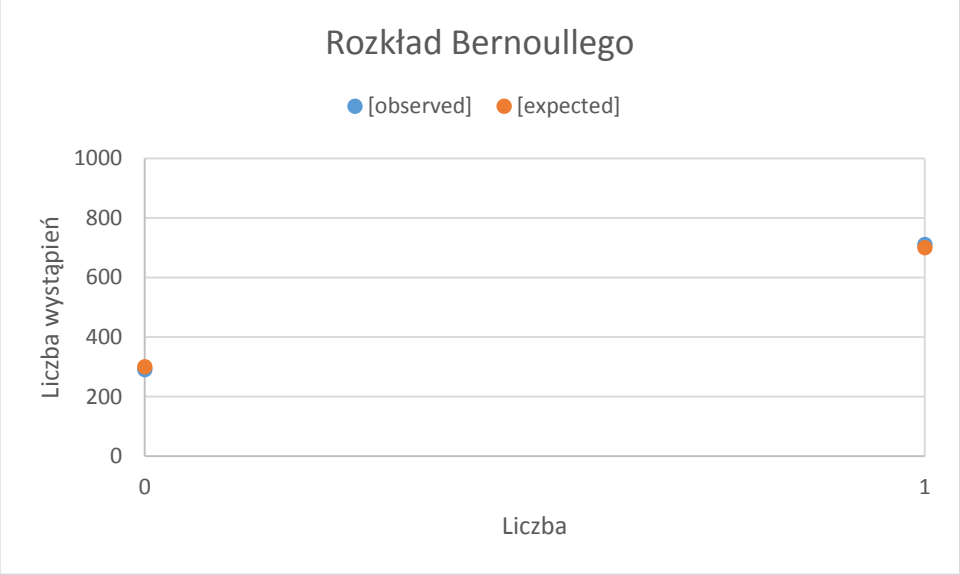
Liczba stopni swobody: 10. Wartość empiryczna z testu chi-kwadrat: 10.7. Tabelka obliczeniowa dla testu jest dokładnie taka sama za wyjątkiem przedziałów. Gdyż generowane liczby były takie same jak przez generator G. Jedynie podzieliliśmy każdą z wylosowanych liczb przez **m**, by uzyskać liczbę z przedziału [0, 1).

Zatem generator J również generuje liczby o rozkładzie jednostajnym.

5.1.3 Generator B

Test przeprowadzimy dla p = 0.7.

Liczba stopni swobody: 1. Wartość empiryczna z testu chi-kwadrat: 0.4. Wartość krytyczna: 3.84. Zatem hipoteza, że liczby mają rozkład Bernoullego jest prawdziwa.

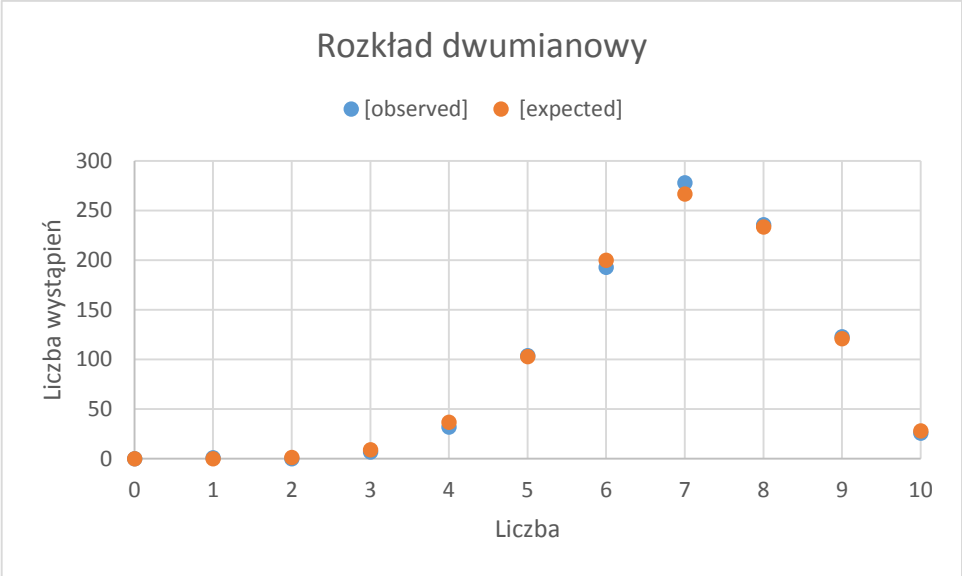


Z wykresu również widać, że wartości spodziewane jak i zaobserwowane są niemalże równe.

5.1.4 Generator D

Test przeprowadzimy dla n = 10, p = 0.7.

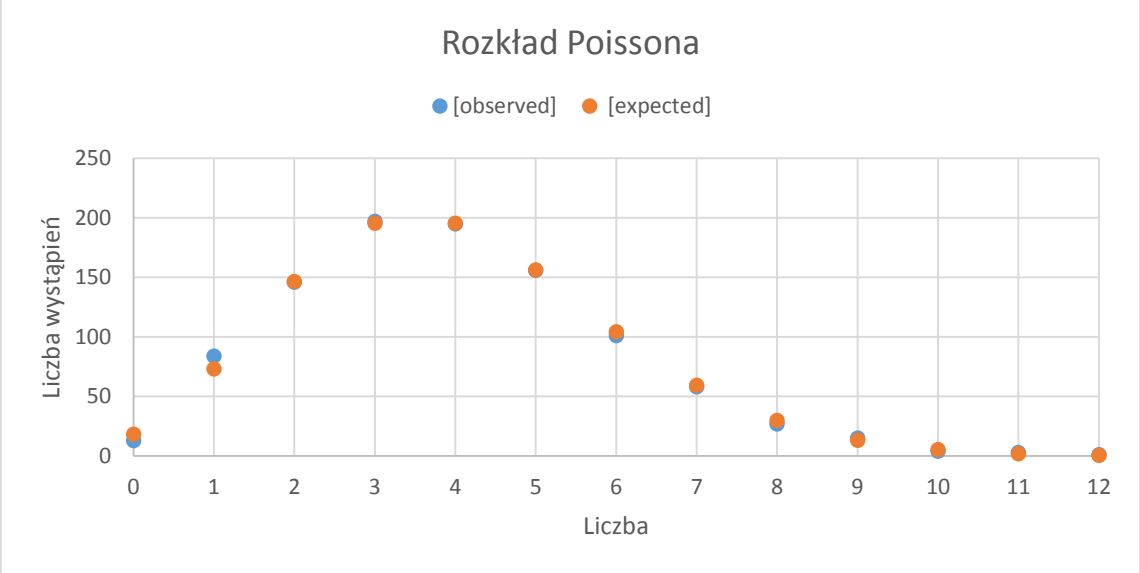
Liczba stopni swobody: 10. Wartość empiryczna wyniosła 8.88. Wartość krytyczna odczytana z tabeli: 18.31. Zatem hipoteza, że liczby mają rozkład dwumianowy jest prawdziwa.



5.1.5 Generator P

Test przeprowadzimy dla $\lambda = 4$.

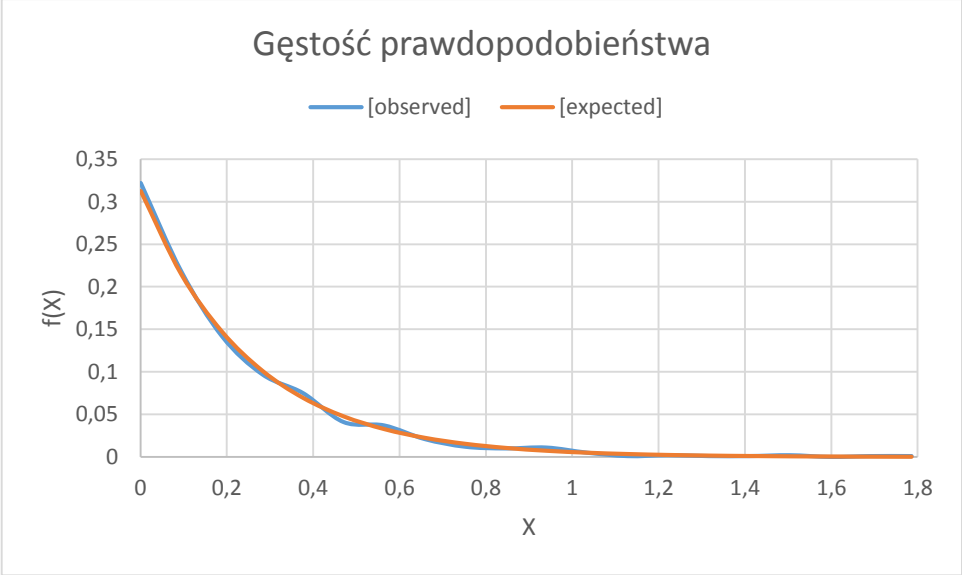
Liczba stopni swobody: 12. Wartość empiryczna wyniosła 4.88. Wartość krytyczna (w naszej tabeli nie ma takiego zakresu, jednakże w Internecie znajduje się wiele tabel[9] czy też kalkulatorów umożliwiających sprawdzenie szukanej wartości) wynosi 21.03. Zatem tezę możemy uznać za poprawną, czyli wylosowane liczby mają rozkład Poisson.



5.1.6 Generator W

Test przeprowadzimy dla $\lambda = 4$.

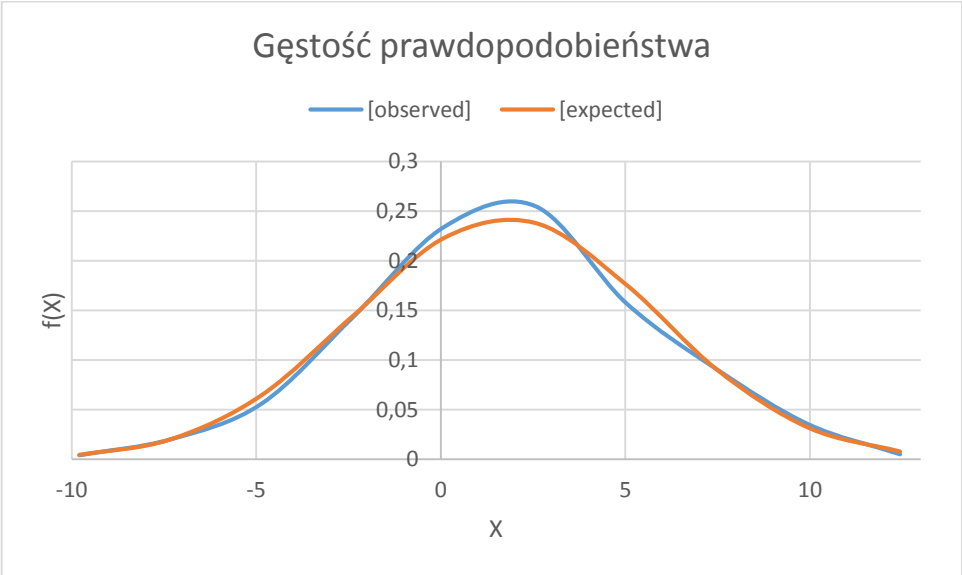
Liczba stopni swobody: 19. Wartość empiryczna wyniosła 13.42. Wartość krytyczna: 30.14. Zatem teza jest prawdziwa. Liczby mają rozkład wykładniczy.



5.1.7 Generator N

Test przeprowadzimy dla $\sigma = 4$, $\mu = 3$.

Liczba stopni swobody: 9. Wartość empiryczna wyniosła 6.20. Wartość krytyczna jest równa 16.92. Zatem możemy przyjąć hipotezę, że liczby mają rozkład normalny.



5.2 Test serii

Test serii[10] jest testem losowości próby, zwany również *testem serii Stevensena*. Mamy dwie hipotezy, zerową i alternatywną:

- H_0 – dobór jednostek do próby jest losowy,

- H_1 – dobór jednostek do próby nie jest losowy.

Algorytm testu jest następujący. Wybieramy medianę z liczb. Następnie zamieniamy liczby mniejsze od mediany na literki A, zaś liczby większe od mediany zamieniamy na literki B. Liczby będące medianą pomijamy. W wyniku tego działania otrzymujemy ciąg liter A oraz B. Zliczamy serie liter **k**; serią nazywamy ciąg tych samych liter. Następnie zliczamy litery A $\mathbf{n_1}$ oraz B $\mathbf{n_2}$. Mając te dane zaglądamy do tablic rozkładu serii dla poziomu istotności $\alpha = 0.05$ oraz $\alpha = (1-0.05) = 0.95$. Wybieramy z nich liczby $\mathbf{k_1}$ oraz $\mathbf{k_2}$ dla znanych nam $\mathbf{n_1}$ oraz $\mathbf{n_2}$. W ostatnim kroku sprawdzamy czy nasza liczba **k** należy do przedziału (k_1, k_2). Jeśli tak jest, wtedy próbę możemy uznać za losową. W przeciwnym wypadku nie ma mowy o losowości.

Tabelka[11] rozkładu serii dla $\alpha = 0.05$ oraz $\alpha = 0.95$:

α=0,05																			
$n_2 \setminus n_1$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2																			
3																			
4			2																
5		2	2	3															
6		2	3	3	3														
7		2	3	3	4	4													
8	2	2	3	3	4	4	5												
9	2	2	3	4	4	5	5	6											
10	2	3	3	4	5	5	6	6	6										
11	2	3	3	4	5	5	6	6	7	7									
12	2	3	4	4	5	6	6	7	7	8	8								
13	2	3	4	4	5	6	6	7	8	8	9	9							
14	2	3	4	5	5	6	7	7	8	8	9	9	10						
15	2	3	4	5	6	6	7	8	8	9	9	10	10	11					
16	2	3	4	5	6	6	7	8	8	9	10	10	11	11	11				
17	2	3	4	5	6	7	7	8	9	9	10	10	11	11	12	12			
18	2	3	4	5	6	7	8	8	9	10	10	11	11	12	12	13	13		
19	2	3	4	5	6	7	8	8	9	10	10	11	12	12	13	13	14	14	
20	2	3	4	5	6	7	8	9	9	10	11	11	12	12	13	13	14	14	15

α=0,95																			
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$n_1 \setminus n_2$
4																			2
5	6																		3
5	6	7																	4
5	7	8	8																5
5	7	8	9	10															6
5	7	8	9	10	11														7
5	7	9	10	11	12	12													8
5	7	9	10	11	12	13	13												9
5	7	9	10	11	12	13	14	15											10
5	7	9	11	12	13	14	14	15	16										11
5	7	9	11	12	13	14	15	16	16	17									12
5	7	9	11	12	13	14	15	16	17	17	18								13
5	7	9	11	12	13	15	16	16	17	18	19	19							14
5	7	9	11	13	14	15	16	17	18	18	19	20	20						15
5	7	9	11	13	14	15	16	17	18	19	20	20	21	22					16
5	7	9	11	13	14	15	16	17	18	19	20	21	21	22	23				17
5	7	9	11	13	14	15	17	18	19	20	20	21	22	23	23	24			18
5	7	9	11	13	14	15	17	18	19	20	21	22	22	23	24	24	25		19
5	7	9	11	13	14	16	17	18	19	20	21	22	23	24	24	25	26	26	20

Teraz przeprowadzimy test serii dla każdego generatora jaki zaimplementowaliśmy. Do tego celu używamy implementacji testu z pliku *test_serii.cpp*. Dane początkowe generatora są takie same jak dla testu chi-kwadrat. Będziemy losowali 20 liczb. Pliki z wylosowanymi liczbami zawarte są w załącznikach.

5.2.1 Generator G

Otrzymaliśmy:

Liczba serii: 11
Liczba wystąpień A: 10
Liczba wystąpień B: 10

Z tabelki odczytujemy przedział (6, 15). Naturalnie liczba 11 należy do tego przedziału więc próba jest losowa.

5.2.2 Generator J

Otrzymaliśmy dokładnie takie same wyniki jak dla generatora G co nie powinno dziwić (wytłumaczenie w poprzednich rozdziałach):

Liczba serii: 11
Liczba wystąpień A: 10
Liczba wystąpień B: 10

Zatem próba jest losowa.

5.2.3 Generator B

Zauważmy, że nie ma zbytnio sensu wykonywać testu dla tego generatora gdyż działamy na liczbach już wylosowanych przez generator J, który generuje liczby losowe co zaobserwowaliśmy powyżej. Poza tym w przypadku gdy mamy nieparzystą liczbę wylosowanych liczb wtedy 0 lub 1 będzie medianą, a to oznacza, że literka A lub B nie wystąpi ani razu. Zauważmy, że w tabelce rozkładu serii nie mamy danych dla takiej sytuacji.

5.2.4 Generator D

Otrzymaliśmy:

Liczba serii: 9
Liczba wystąpień A: 10
Liczba wystąpień B: 10

Z tabelki odczytujemy przedział (6, 15). Naturalnie liczba 9 należy do tego przedziału więc próba jest losowa.

5.2.5 Generator P

Otrzymaliśmy:

Liczba serii: 9
Liczba wystąpień A: 9
Liczba wystąpień B: 5

Z tabelki odczytujemy przedział (4, 10). Naturalnie liczba 9 należy do tego przedziału więc próba jest losowa.

5.2.6 Generator W

Otrzymaliśmy:

Liczba serii: 13
Liczba wystąpień A: 10
Liczba wystąpień B: 10

Z tabelki odczytujemy przedział (6, 15). Naturalnie liczba 13 należy do tego przedziału więc próba jest losowa.

5.2.7 Generator N

Otrzymaliśmy:

Liczba serii: 14
Liczba wystąpień A: 10
Liczba wystąpień B: 10

Z tabelki odczytujemy przedział (6, 15). Naturalnie liczba 14 należy do tego przedziału więc próba jest losowa.

6. Wnioski

Celem naszej pracy było utworzenie kilku generatorów liczb losowych o określonych rozkładach prawdopodobieństwa. Początkowo mogłoby się wydawać, iż zaimplementowanie generatora liczb losowych może być rzeczą trudną. Jednakże jak mogliśmy zauważyć, nawet proste operacje modularne mogą stanowić podstawy takiego generatora. Z racji dużych wartości **m** ciężko jest przewidzieć jaką resztę otrzymamy, stąd może pojawiać się losowość naszych danych. Ponadto im większa wartość **m** tym może być większy okres naszego generatora. Badania wykazały, że najdłuższe okresy otrzymamy gdy **m** będzie potęgą liczby 2.

Poznaliśmy również testy statystyczne sprawdzające zgodność danych z określonym rozkładem prawdopodobieństwa (test chi-kwadrat) oraz testy sprawdzające losowość próby (test serii) i zbadaliśmy na ich podstawie rozkład i losowość naszych danych.

Bibliografia

[1] https://en.wikipedia.org/wiki/Linear_congruential_generator

[2] http://www.algorytm.org/liczby-pseudolosowe/generator-lcg-liniowy-generator-kongruentny.html

[3] http://home.agh.edu.pl/~chwiej/mn/generatory_1819.pdf

[4] https://pl.wikipedia.org/wiki/Transformacja_Boxa-Mullera

[5] https://www.statsdirect.com/help/distributions/normal.htm

[6] https://pracownik.kul.pl/files/12167/public/zmsm/konwers6.pdf

[7] https://pl.wikipedia.org/wiki/Test_statystyczny

[8] https://pl.wikipedia.org/wiki/Liczba_stopni_swobody_(statystyka)

[9] https://people.richland.edu/james/lecture/m170/tbl-chi.html

[10] https://pl.wikipedia.org/wiki/Test_serii

[11] http://home.agh.edu.pl/~kca/stat/WYK%A3ADY_pdf/WYK%A3AD%208.pdf