Generator liczb pseudolosowaych

Konrad Nowak

14 czerwca 2021

Spis treści

1 Wstęp

1.1 Cel projektu

Głównym celem projektu jest zaimplementowanie standardowego generatora liczb pseudolosowych bez korzystania z wbudowanych bibliotek losujących liczby. Z głównego generatora będą korzystały generatory m.in. różnych rozkładów w tym:

- Generator liczb zmiennoprzecinkowych w przedziale $(0, 1)$
- Generator Bernoulliego
- Generator rozkładu dwumianowego
- Generator rozkładu Poissona
- Generator rozkładu wykładniczego
- Generator rozkładu normalnego

Kolejnym z celów jest analiza pseudolosowości oraz poprawności każdego z generatorów. Jakie możliwe poprawki można by było nanieść w implementacji i sprawdzenie, czy wygenerowany ciąg liczb sprawia wrażenie losowego.

1.2 Implementacja

Implementacja zawiera 3 pliki:

- *Generator.py* - klasa zawierająca implementację wszystkich generatorów, w konstruktorze przyjmująca wartość ziarna. Dodatkowo posiada metody zmiany i resetu ziarna.
- *Test.py* - klasa zawierająca metody pozwalające testować generatory za pomocą testów serii oraz pozwalająca wypisać histogram generatora podanego w konstruktorze.
- *main.py* - główny plik do kompilowania, w którym są zapisane przykładowe wywołania generatorów i testów.

2 Rodzaje generatorów

2.1 Generator G

Generator addytywny postaci:

$$X_n = (a \cdot X_{n-1} + c) \mod m$$

Gdzie:

- X_n - nasza wygenerowana liczba
- X_{n-1} - poprzednio wygenerowana liczba lub początkowa
- a - stała, mnożnik
- c - stała, przyrost, spełniająca warunek $NWD(c, m) = 1$
- m - moduł

W naszym przypadku:

$$X_n = (5^{10} \cdot X_{n-1} + 7151) \mod 2^{31} - 1$$

Generator może wylosować liczby z przedział do maksymalnej wartości *int*, co pozwala na znaczne zróżnicowanie wyników. Stała c oraz m muszą być liczbami względnie pierwszymi, by nie doszło do sytuacji, gdzie pojawia się ciąg zer.
Generator będzie używany jako baza dla reszty przedstawionych generatorów.

2.2 Generator J

Generator liczb zmiennoprzecinkowych w przedziale $(0, 1)$ przy wykorzystaniu generatora G. Jego działanie jest bardzo proste - biorąc wynik z generatora G dzielimy go przez Wykorzystując generator G, otrzymujemy wynik przez wzór $\frac{G()+1}{m+1}$, co zagwarantuje, że wyniki będą w przedziale $(0, 1)$.

2.3 Generator B

Generator dwupunktowy, zwracający 1 w przypadku sukcesu, bądź 0 w przypadku porażki. Sprawdzamy wynik z wykorzystanego generatora J, czy wynik jest poniżej lub równy podanego w argumencie p (prawdopodobieństwa), co uznajemy za sukces. W przeciwnym wypadku - porażkę.

2.4 Generator D

Generator rozkładu dwumianowego dla podanych w argumencie p (prawdopodobieństwa) oraz n (ilości prób). Dla każdego coraz to kolejnych wyników z generatora B zliczamy łączną liczbą sukcesów.

2.5 Generator P

Jako argument przyjmuje λ (wartość oczekiwaną). Generator korzysta z generatora J, mnoży kolejne prawdopodobieństwa wydarzenia się danej liczby zdarzeń w jednej jednostce czasu aż wartość nie przekroczy $q = e^{-\lambda}$ (wylosowana na początku wartość). Rozkład Poissona jest szczególnym przypadkiem rozkładu dwumianowego, w którym nasze zdarzenia rozpatrujemy w coraz to mniejszych przedziałach czasowych, stąd $q = e^{-\lambda}$, gdyż $\lim_{n\to\infty}(\frac{\lambda}{n})^k \cdot (1 - \frac{\lambda}{n})^{n-k} = e^{-\lambda}$.

2.6 Generator W

Jako argument przyjmuje λ (wartość oczekiwaną). Generator korzysta z obliczonej dystrybuanty dla $f(x) = \lambda e^{-\lambda x}$ $F(x) = 1 - e^{-\lambda x}$ i jej odwrotności $F^{-1}(x) = \frac{-\log(U)}{\lambda}$, gdzie U to liczba pomiędzy 0 a 1 z naszego generatora J (we wzorze jest $\log(1 - U)$, ale skoro $U \in (0, 1)$, to możemy to po prostu zapisać jako U. Rozkład wykładniczy opisuje odstępy czasu między kolejnymi wydarzeniami.

2.7 Generator N

Korzysta z metody Boxa-Mullera, która wyznacza parę dwóch zmiennych (x, y) wyznaczonych wzorami $x = \sqrt{-2\log(1-U_1)} \cdot \cos(2U_2)i$ oraz $y = \sqrt{-2\log(1-U_1)} \cdot \sin(2U_2)$, gdzie U_1 i U_2 to wartości wygenerowane przez generator J. Następnie przeskalowuje wyliczony x o podaną w argumencie zmienną o oraz przesuniecie u (zamiana standardowego rozkładu na parametryzowany).

3 Testy

3.1 Eksperymenty

Dla każdego z generatorów zostały przeprowadzone cztery serie eksperymentów: dla ziaren: 454565756, 246457877, 897654345 oraz 986168. Dla każdego z generatorów zostały przeprowadzone takie same wywołania generatorów (z odpowiednimi zmiennymi wejściowymi):

```
main_test.G_histogram(10000)
main_test.J_histogram(10000)
main_test.B_histogram(10000, 0.75)
main_test.B_histogram(10000, 0.15)
main_test.D_histogram(10000, 0.80, 30)
main_test.D_histogram(10000, 0.20, 100)
main_test.P_histogram(10000, 1)
main_test.P_histogram(10000, 5)
main_test.P_histogram(10000, 10)
main_test.W_histogram(10000, 10)
main_test.N_histogram(10000, 9765625, 7151)
```

3.2 Testy serii

Testy serii zostały wykonane na takich samych ziarnach na generatorze G oraz J (powinny dawać dokładnie takie same wyniki) i próbkach 10000 elementów. Test serii polega na wyliczeniu mediany wygenerowanych elementów i podzielenie ich na dwa zbiory A i B - wartości poniżej mediany i powyżej mediany (wartości równe medianie pomijamy). Tworzymy dodatkową tablicę i sprawdzamy, ile jest serii powtarzających się elementów z jednego i drugiego zbioru. Liczba serii będzie podlegała rozkładowi naturalnemu, więc obliczamy wartość średnią i wariancję, a z nich statystykę Z. Sprawdzamy, czy Z mieści się między -1.96 a 1.96 ($2,5\%$ z obu stron rozkładu odcinamy, by sprawdzić, czy wynik mieści się w 95% wyników).

4 Wnioski

Wynikowe histogramy oraz testy serii pokazały, że generator G jest w stanie generować liczby w miarę pseudolosowe i poprawne z nich rozkłady. Jednakże nie dla każdego ziarna jest to tak dokładne. Do poprawy mogłyby być na pewno współczynniki w generatorze G dobrane znacznie lepiej oraz jego znaczna modyfikacja zwiększająca poczucie losowości danych.

5 Źródła

- $http://home.agh.edu.pl/\ chwiej/mn/generatory_16.pdf$
- $https://pl.wikipedia.org/wiki/Test_serii$
- $https://eduinf.waw.pl/inf/alg/001_search/0020.php$
- $https://pl.wikipedia.org/wiki/Rozkad_Poissona$
- $Wieczorkowski_R.-Komputerowe_generatory_liczb_losowych$