Noname manuscript No.
(will be inserted by the editor)

Komputerowe modelowanie wielkości losowych

Karol Fijałkowski

Streszczenie Praca opisuje zagadnienia związane z modelowaniem wielkości losowych. Przedstawiono w niej także wybrane metody komputerowego modelowania liczb losowych.

Politechnika Wrocławska

Spis treści

1 Generatory liczb losowych 3
1.1 Praktyczne zastosowania 3
1.2 „Prawdziwie" a pseudolosowe liczby 4

2 Metody generowania 5
2.1 Metody fizyczne (sprzętowe) 5
2.2 Metody obliczeniowe 6
2.3 Generowanie wartości z rozkładu prawdopodobieństwa 7

3 Generowanie liczb psudolosowych za pomocą komputera 7
3.1 Nieliniowe sprzężenie zwrotne 7
3.2 Generowanie zmiennych losowych o określonej f. gęstości prawdopodobieństwa . 7
3.2.1 Metoda odwracania dystrybuanty (znana również jako inwersyjna) 7
3.2.2 Metoda odrzucania 8

4 Generowanie liczb losowych w praktyce 10
4.1 Przetwarzanie i kontrole statystyczne 10
4.2 Inne uwagi 10
4.3 Sekwencje o niskiej rozbieżności jako alternatywa 11

1 Generatory liczb losowych

Generator liczb losowych jest obliczeniowym lub fizycznym urządzeniem przeznaczonym do generowania sekwencji cyfr lub symboli, których nie można racjonalnie przewidzieć lepiej niż przez losowy przypadek.

Różne zastosowania losowości doprowadziły do rozwoju różnych sposobów generowania losowych danych, z których niektóre istniały już od czasów starożytnych. W ich szeregach są dobrze znane "klasyczne" przykłady, w tym rzut kośćmi, rzut monetą , tasowanie kart do gry , korzystanie z różdżek (do wróżenia), a także niezliczona ilość innych technik. Ze względu na mechaniczną naturę tych metod, generowanie dużych ilości wystarczająco losowych liczb (bardzo ważne w statystykach) wymaga wiele pracy i czasu. Tak więc, wyniki były czasem być gromadzone i rozpowszechniane jako tablice liczb losowych. Obecnie, po nadejściu obliczeniowych generatorów liczb losowych, rosnąca liczba loterii i gier loteryjnych zaczęła używać generatorów liczb losowych, zamiast bardziej tradycyjnych metod losowych. Generatory liczb losowych wykorzystywane są również w celu ustalenia odsetku wygranych w nowoczesnych automatach do gier losowych.

Istnieje kilka metod obliczeniowych do generowania liczb losowych. Wiele z nich jest blisko prawdziwej losowości, chociaż mogą one spełniać, z różnym powodzeniem, niektóre ze statystycznych testów na przypadkowość przeznaczonych do pomiaru, jak nieprzewidywalne są ich wyniki (to jest, do jakiego stopnia ich wzory są dostrzegalne). Jednak starannie zaprojektowane i kryptograficznie bezpieczne komputerowe metody generowania liczb losowych również istnieją, takie jak te oparte na algorytmie krwawnika (Yarrow) , Fortuna (Generator Liczb Pseudolosowych) i inne.[1]

1.1 Praktyczne zastosowania

Generatory liczb losowych mają zastosowanie w hazardzie , próbkowaniu statystycznym , symulacji komputerowej , kryptografii i innych obszarach, gdzie pożądane jest tworzenie nieprzewidywalnych efektów. Generalnie w aplikacjach, w których nieprzewidywalność jest najważniejsza, na przykład w aplikacjach bezpieczeństwa, generatory sprzętowe są zwykle korzystniejsze niż algorytmy pseudolosowe, gdy jest to możliwe.

Generatory liczb losowych są bardzo użyteczne w tworzeniu metod symulacji Monte Carlo, gdyż debugowanie jest tam ułatwione przez możliwość uruchomienia jeszcze raz tej samej sekwencji liczb losowych, przez zaczynanie z tych samych ziaren losowości . Są one również wykorzystywane w kryptografii - tak długo, jak ziarno jest tajne. Nadajnik i odbiornik mogą automatycznie generować ten sam zestaw numerów, do wykorzystywania jako klucze.

Generowanie liczb pseudolosowych jest ważnym i pospolitym zadaniem w programowaniu komputerowym. Podczas gdy kryptografia i niektóre algorytmy numeryczne wymagają bardzo wysokiego stopnia pozornej przypadkowości, wiele innych

operacji wymaga tylko skromnej ilości nieprzewidywalności. Kilkoma prostymi przykładami mogą być: przedstawienie użytkownikowi "losowego cytatu dnia" lub określenie w jaki sposób może poruszyć się przeciwnik sterowany przez komputer w grze komputerowej. Słabsze formy losowości stosowane są w funkcjach skrótu oraz w tworzeniu zamortyzowanego wyszukiwania oraz algorytmów sortowania.

Niektóre aplikacje, które na pierwszy rzut oka wydają się być przystępne do randomizacji, w rzeczywistości nie są tak proste. Na przykład, system, który "przypadkowo" wybiera utwory muzyczne, musi tylko wydawać się losowy, a nawet może mieć sposoby na kontrolowanie wyboru muzyki. Prawdziwy losowy system nie miałby żadnych ograniczeń w pojawianiu się tej samej pozycji dwa lub trzy razy z rzędu.

1.2 „Prawdziwie" a pseudolosowe liczby

Istnieją dwie główne metody wykorzystywane do generowania liczb losowych. Pierwsza metoda polega na pomiarze jakiegoś stochastycznego zjawiska fizycznego, a następnie wyrównywaniu (kompensowaniu) potencjalnych odchyleń (bias) w procesie pomiarowym. Przykładowe źródła obejmują pomiar szumu atmosferycznego ,cieplnego i innych zewnętrznych zjawisk elektromagnetycznych i kwantowych. Na przykład kosmiczne promieniowanie lub rozpad promieniotwórczy mierzone w krótkich czasach stanowią źródła naturalnej entropii.[2]

Szybkość, z jaką entropia może być pobierana z naturalnych źródeł zależy od zasad mierzonego zjawiska fizycznego. Zatem o źródłach naturalnie występującej "prawdziwej" entropii mówi się, że blokują - są ograniczone dopóki zebrane jest na tyle entropii, aby spełnić dane wymogi. Na niektórych systemach uniksowych, w tym większości dystrybucji Linuksa , plik pseudourządzenia /dev/random będzie blokować aż do uzyskania wystarczającej entropii zebranej z otoczenia. W związku z tym zachowaniem blokującym, duże odczyty z /dev/random , takie jak wypełnianie dysku twardego losowymi bitami, często mogą być powolne w systemach wykorzystujących ten typ źródła entropii.

Druga metoda wykorzystuje algorytmy obliczeniowe , mogące wytwarzać długie sekwencje pozornie przypadkowych wyników, które są w rzeczywistości całkowicie określone przez krótszą wartość początkową, znaną jako wartość ziarna lub klucz . W wyniku tego, cała pozornie losowa sekwencja może być odtworzona jeśli wartość ziarna jest znana. Ten rodzaj generatora liczb losowych jest często nazywany generatorem liczb pseudolosowych. Ten typ generatora zazwyczaj nie opiera się na źródłach naturalnie występującej entropii, chociaż może być okresowo wspomagany ze źródeł naturalnych. Ten rodzaj generatora jest bez blokowania, więc nie są one ograniczone przez zdarzenia zewnętrzne, dzięki czemu odczyty wielkoilościowe stają się możliwe.

Niektóre systemy przyjmują podejście hybrydowe, zapewniając losowość pozyskaną ze źródeł naturalnych, jeśli są dostępne oraz wracanie do kryptograficznie bezpiecznych generatorów liczb pseudolosowych, opartych na oprogramowaniu o

okresowo zmiennym ziarnie losowości. Wracanie to następuje, gdy żądana szybkość odczytu losowości przekracza możliwości podejścia „pobierania naturalnego", aby nadążyć za danym żądaniem. Takie podejście pozwala uniknąć ograniczającego zachowania blokującego generatorów liczb losowych opartych na wolniejszych i czysto środowiskowych (fizycznych) metodach.

Podczas gdy generator liczb pseudolosowych oparty wyłącznie na logice deterministycznej nigdy nie może być uważany za "prawdziwe" źródło losowych liczb numer, w najczystszym tego słowa znaczeniu, w praktyce jest on na ogół wystarczający nawet dla wymagających aplikacji bezpieczeństwa o znaczeniu krytycznym. Rzeczywiście, starannie zaprojektowane i wdrożone generatory liczb pseudolosowych mogą być poświadczone dla celów kryptograficznych zabezpieczeń o krytycznym znaczeniu, jak to ma miejsce z algorytmem krwawnika (Yarrow) i Fortuna. Ten wcześniejszy to podstawa do źródła entropii /dev/random we FreeBSD , AIX , OS X , NetBSD i innych. OpenBSD wykorzystuje również algorytm liczb pseudolosowych w oparciu o szyfr strumieniowy ChaCha20 znany jako arc4random.

2 Metody generowania

2.1 Metody fizyczne (sprzętowe)

Najwcześniejsze metody generowania liczb losowych, takie jak kości, rzut monetą i ruletka, są stosowane do dziś, głównie w grach i hazardzie, ponieważ są zbyt powolne dla większości zastosowań w statystykach i kryptografii.

Fizyczny generator liczb losowych może być oparty na istotnie losowym atomowym lub subatomowym zjawisku fizycznym, którego nieprzewidywalność pochodzi od praw mechaniki kwantowej . Źródła entropii obejmują rozpad promieniotwórczy, szumy termiczne, szum lawinowy w diodach Zenera , dryf zegarowy , taktowanie rzeczywistych ruchów na głowicy odczytu/zapisu dysku twardego i szum radiowy. Jednakże zjawiska fizyczne i narzędzia użyte do ich zmierzenia, na ogół cechują się asymetrią i systematycznymi odchyłami , które sprawiają, że ich wyniki nie są jednolicie losowe. Ekstraktor losowości, taki jak kryptograficzna funkcja mieszania, może być użyty do zbliżania się do równomiernego rozkładu bitów z niejednorodnie losowego źródła, chociaż z mniejszą szybkością.

Zostały opracowane różne pomysłowe sposoby gromadzenia tego entropicznych informacji. Jedną z technik jest uruchomienie funkcji skrótu na ramce strumienia wideo z nieprzewidywalnego źródła. Generator sprzętowy „Lavarand" używał tej techniki z obrazami wielu lamp typu „lava" . Generator "HotBits" mierzy rozpadu promieniotwórczy za pomocą rurek Geigera-Mullera,[3] podczas gdy witryna „Random.org" wykorzystuje różnice w amplitudzie hałasu atmosferycznego nagranego zwykłym radiem.

Innym częstym źródłem entropii jest zachowanie człowieka jako użytkownika systemu. Podczas gdy ludzie nie są uważani za dobre generatory losowości na żądanie, generują oni losowe zachowanie całkiem dobrze w kontekście grania w gry strategii

mieszanej. Niektóre programy komputerowe związane z bezpieczeństwem wymagają od użytkownika, aby wykonał długą serię ruchów myszy lub wprowadzania znaków z klawiatury. Służy to do stworzenia wystarczającej entropii potrzebnej do generowania losowych kluczy lub zainicjowania generatora liczb pseudolosowych.

2.2 Metody obliczeniowe

Większość liczb losowych generowanych komputerowo używa generatorów liczb pseudolosowych, które są algorytmami mogącymi automatycznie tworzyć długie ciągi liczb losowych o dobrych właściwościach, ale w końcu sekwencja powtarza się (lub wykorzystanie pamięci rośnie bezgranicznie). Ten rodzaj liczb losowych jest wystarczający w wielu przypadkach, lecz nie są one tak losowe, jak liczby generowane z elektromagnetycznego szumu powietrza, stosowanego jako źródło entropii.[4] Seria wartości generowanych przez takie algorytmy jest zasadniczo określona przez ustaloną liczbę zwaną ziarnem losowości. Jednym z najbardziej powszechnych generatorów liczb pseudolosowych jest liniowy generator przystający, który wykorzystuje rekurencję

$$X_{n+1} = (aX_n + b) mod\ m \tag{1}$$

do generowania liczb, gdzie a, b i m są liczbami całkowitymi, a $X_n + 1$ jest następny w X jako ciąg liczb pseudolosowych. Maksymalna ilość liczb, którą formuła może wytworzyć jest moduł z „m" . W celu uniknięcia pewnych nieprzypadkowych właściwości pojedynczego liniowego generatora przystającego, można użyć równolegle kilka takich generatorów liczb losowych o nieznacznie różnych wartościach współczynnika mnożącego „a", z nadrzędnym generatorem liczb losowych, który wybiera spośród kilku różnych generatorów.

Prostą metodą generowania liczb losowych jest tzw. metoda środkowego kwadratu zaproponowana przez Johna von Neumanna . Choć prosta do wykonania, jej efekt jest słabej jakości. Ma bardzo krótki okres i poważne niedociągnięcia, takie jak to, że sekwencja wyjściowa jest prawie zawsze zbieżna do zera.

Większość języków programowania zawiera funkcje lub procedury biblioteczne, które dostarczają losowych generatorów liczb. Często są one zaprojektowane tak, aby zapewnić losowy bajt, słowo, lub liczbę zmiennoprzecinkową o rozkładzie jednostajnym ciągłym między 0 i 1.

Jakość czyli przypadkowość takich funkcji bibliotecznych bardzo się różni, od całkowicie przewidywalnego wyniku, do kryptograficznie bezpiecznego. Domyślny generator liczb losowych w wielu językach, w tym Python, Ruby, R, IDL i PHP bazuje na algorytmie Mersenne Twister i jest nie wystarczający dla celów kryptograficznych, co jest wyraźnie podane w dokumentacji języka. Takie funkcje biblioteczne często mają słabe właściwości statystyczne, a niektóre będą powtarzać wzorce po zaledwie kilkudziesięciu tysiącach prób. Często są one inicjowane przy użyciu komputerowego zegara czasu rzeczywistego jako ziarna losowości, ponieważ taki zegar zwykle mierzy się w milisekundach, dużo poza precyzją człowieka. Funkcje te mogą zapewniać wystarczającą losowość dla określonych zadań (na przykład

gier wideo), ale nie nadaje się, gdy wymagana jest wysoka jakość przypadkowości, na przykład w zastosowaniach kryptografii, statystyce czy analizie numerycznej.

Źródła liczb losowych o znacznie wyższej jakości przypadkowości są dostępne na większości systemów operacyjnych; na przykład /dev/random na różnych wersjach BSD, Linux, Mac OS X, IRIX i Solaris oraz CryptGenRandom dla Microsoft Windows. Większość języków programowania, w tym te wymienione powyżej, zapewniają dostęp do tych źródeł o wyższej jakości.

2.3 Generowanie wartości z rozkładu prawdopodobieństwa

Istnieje kilka metod generowania losowej liczby na podstawie funkcji gęstości prawdopodobieństwa . Metody te obejmują przekształcanie w różny sposób liczby losowej z rozkładu jednostajnego. Z tego powodu, metody te działają równie dobrze w wytwarzaniu zarówno pseudolosowych i rzeczywiście losowych liczb. Jedna z metod, zwana metodą inwersji , polega na integrowaniu się z obszarem większym niż lub równym liczbie losowej (która powinna być generowana pomiędzy 0 i 1 dla dobrego rozkładu). Druga metoda, zwana metodą akceptacji-odrzucenia, polega na wyborze wartości x i y oraz testowania, czy funkcja z x jest większa od wartości y. Jeżeli tak, to wartość x jest akceptowana. W przeciwnym razie wartość x jest odrzucana, a algorytm wykonuje kolejną próbę.[5]

3 Generowanie liczb psudolosowych za pomocą komputera

3.1 Nieliniowe sprzężenie zwrotne

Jedną z metod generowania liczby pseudolosowej za pomocą komputera jest zastosowanie nieliniowego sprzężenia zwrotnego[6]

$$x_k = F(x_{k-1}, x_{k-2}, ..., x_{k-q}) \tag{2}$$

- Funkcja F powinna zapewniać jak nawiększy okres ciągu $\{x_k\}$ (można wykorzystać na przykład operację wyznaczania reszty z dzielenia)
- Funkcja F powinna zapewnić odpowiednio regularny rozkład wartości x_k

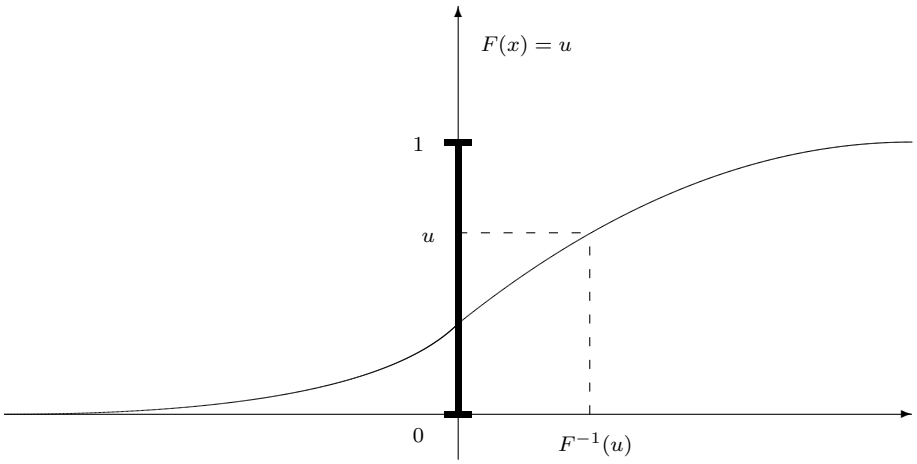
3.2 Generowanie zmiennych losowych o określonej f. gęstości prawdopodobieństwa

3.2.1 Metoda odwracania dystrybuanty (znana również jako inwersyjna)

W tej metodzie ważne jest posiadanie dobrego generatora zmiennych losowych o rozkładzie jednostajnym $U[0, 1]$. Zrealizować można generator zmiennej losowej X, o funkcji gętości prawdopodobieństwa $f(x)$, której dystrybuanta $F(x)$ jest ściśle

rosnąca.[6]

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)dx \tag{3}$$



Rysunek 1 Generowanie zmiennych losowych o określonej funkcji gęstości prawdopodobieństwa metodą odwracania dystrybuanty

Gdy dystrybuanta $F(x)$ jest ściśle rosnąca, istnieje jej odwrotność $F^{-1}()$ i jest dobrze określona. Odwzorowanie dystrybuanty odwrotnej jest wzajemnie jednoznaczne.

$$u = F(x), \qquad u \in [0,1] \tag{4}$$

$$x = F^{-1}(u), \qquad x \in R \tag{5}$$

Wartości od 0 do 1 są uprzywilejowane ze względu na definicję dystrybuanty.

Aby stworzyć generator liczb losowych dla określonej funkcji gęstości prawdopodobieństwa należy:

- wyznaczyć dystrybuantę $F(x)$ dla danej funkcji gęstości prawdopodobieństwa,
- wyznaczyć wzór na odwrotność dystrybuanty $F^{-1}(u)$,
- wylosować wartości u z rozkładu jednostajnego $U[0,1]$,
- obliczyć $F^{-1}(u)$.

3.2.2 Metoda odrzucania

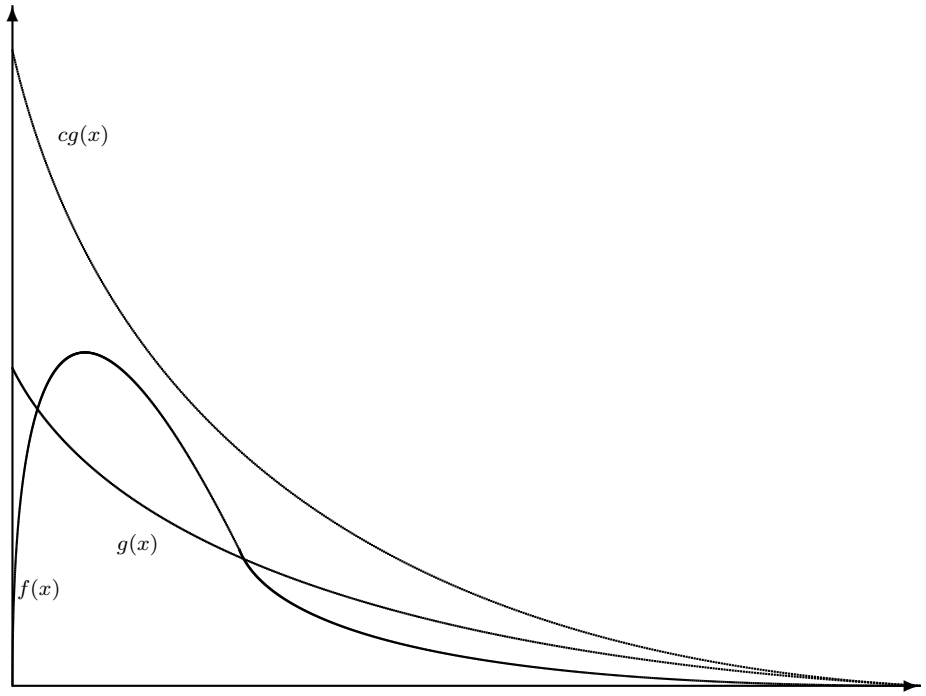
Metoda odrzucania wykorzystuje rozkład jednostajny z przedziału $[0,1]$ do generowania innych rozkładów. Polega ona na losowaniu punktów z płaszczyzny i

brania pod uwagę jedynie tych, które znajdą się pod wykresem funkcji gęstości, a odrzucaniem pozostałych. Dzięki temu otrzymuje się punkty wylosowane zgonie z rozkładem, który chcieliśmy osiągnąć za pomocą funkcji f.[6]

Za $f(x)$ przyjęto gęstość prawdopodobieństwa zmiennej X, którą chcemy zastosować, a za $g(x)$ pomocniczą gęstość innej pomocniczej (prostej do wygenerowania) zmiennej losowej. Zakłada się że istnieje $c > 0$, takie że $f(x) \leq cg(x)$ dla każdego $x \in R$.

Aby stworzyć generator liczb losowych dla określonej funkcji gęstości prawdopodobieństwa należy:

- wygenerować liczby losowe x ze zbioru X o gęstości $g(x)$ (np. metodą odwracania dystrybuanty),
- wygenerować liczby losowe u ze zbioru U o gęstości rozkładu jednostajnego $U[0,1]$,
- stworzyć pary $(x,y) = (x, cg(x)u)$ o rozkładzie równomiernym na zbiorze $A_g = \{(x,y) : x \in R, y \in [0, cg(x)]\}$,
- następnie należy odrzucać wcześniej wygenerowane pary (x,y), takie które NIE należą do zbioru $A_f = \{(x,y) : x \in R, y \in [0, f(x)]\}$.



Rysunek 2 Generowanie zmiennych losowych o określonej funkcji gęstości prawdopodobieństwa metodą odrzucania

Pozostaną tylko te pary, które będą mieć rozkład jednostajny na zbiorze A_f oraz spełniać będą warunek:

$$cg(x)u \leq f(x) \tag{6}$$

Warunki jakie powinna spełniać metoda aby była nietrudna w zastosowaniu:

- dla funkcji $f(x)$ powinno istnieć odpowiednie $g(x)$,
- funkcja $g(x)$ powinna być prosta w generacji,
- funkcja $g(x)$ powinna możliwie blisko "przebiegać" nad funkcją $f(x)$, w celu zagwarantowania dużego odsetku przyjętych par.

4 Generowanie liczb losowych w praktyce

4.1 Przetwarzanie i kontrole statystyczne

Nawet używając źródła wiarygodnych liczb losowych (na przykład generatora sprzętowego), uzyskiwanie liczb, które są całkowicie nieobciążone wymaga ostrożności. Ponadto zachowanie tych generatorów często zmienia się wraz z temperaturą, napięciem zasilania, wiekiem urządzenia lub innej ingerencji z zewnątrz. Błąd oprogramowania w procedurze generowania liczby pseudolosowej lub błąd sprzętowy na używanym sprzęcie, może być podobnie trudne do wykrycia.

Generowane liczby losowe są często poddawane testom statystycznym przed ich użyciem, aby upewnić się, że podstawowe źródło generatora nadal działa, a następnie przetworzone w celu poprawy ich właściwości statystycznych. Przykładem może być sprzętowy generator liczb losowych TRNG9803, który wykorzystuje pomiar entropii jako test sprzętowy, a następnie przetwarza losowe sekwencje za pomocą rejestru przesuwnego szyfru strumieniowego. Używanie testów statystycznych do sprawdzania poprawności generowanych liczb losowych jest na ogół trudne. Wang i Nicol zaproponowali technikę badania statystycznego opartą na odległości, która używana jest do określenia słabych stron kilku generatorów losowych.

4.2 Inne uwagi

Liczby losowe o rozkładzie jednostajnym pomiędzy 0 i 1 mogą być używane do generowania liczb losowych o pożądanym rozkładzie poprzez przepuszczenie ich przez odwrotną dystrybuantę pożądanego rozkładu. Aby wygenerować parę statystycznie niezależnych liczb losowych (x, y) o rozkładzie normalnym, można wygenerować najpierw współrzędne biegunowe (r, θ), gdzie $r \sim \mathrm{X}_2{}^2$, a $\theta \sim U(0, 2\pi)$.

Niektóre generatory liczb losowych zawierają 0, a nie zawierają 1, podczas gdy inne zawierają lub nie zawierają obu.

Wyjścia wielu niezależnych generatorów liczb losowych mogą być połączone (na przykład za pomocą operacji logicznej XOR), aby uzyskać łączny generator liczb

losowych przynajmniej tak dobry, jak najlepszy użyty w nim generator. Określane jest to jako programowe wybielanie.

Obliczeniowe oraz sprzętowe generatory liczb losowych są czasami, aby oddać zalety obu tych rodzajów. Obliczeniowe generatory liczb losowych zazwyczaj mogą generować liczby pseudolosowe znacznie szybciej niż generatory sprzętowe, natomiast generatory sprzętowe mogą generować "prawdziwą losowość".

4.3 Sekwencje o niskiej rozbieżności jako alternatywa

Niektóre obliczenia wykorzystujące generator liczb losowych, można określić jako obliczenie całkowitej lub średniej wartości, jak na przykład obliczenie całki metodą Monte Carlo. Do takich zadań, możliwe jest znalezienie bardziej dokładnego rozwiązania poprzez zastosowanie tak zwanych sekwencji niskiej rozbieżności, zwanych również jako quasi-losowe liczby. Takie sekwencje mają określony wzór, który wypełnia luki równomiernie. Prawdziwie losowa sekwencja może, i zazwyczaj nie pozostawia większych luk.

Literatura

1. William Feller, "Wstęp do rachunku prawdopodobieństwa Cz. 1.", PWN, Warszawa (2006)
2. Marek Leśniewicz, "Sprzętowa generacja losowych ciągów binarnych", WAT, Warszawa (2009)
3. Walker, John, "HotBits: Genuine Random Numbers", http://www.fourmilab.ch/hotbits/ (dostęp 10.01.2017)
4. "RANDOM.ORG - True Random Number Service", https://www.random.org/ (dostęp 10.01.2017)
5. The Numerical Algorithms Group. "G05 – Random Number Generators" http://www.nag.co.uk/numeric/fl/nagdoc_fl23/pdf/G05/g05intro.pdf (dostęp 10.01.2017)
6. Grzegorz Mzyk, "Generacja liczb losowych o róznych rozkładach", http://staff.iiar.pwr.wroc.pl/grzegorz.mzyk/kmi/kmi03.pdf (dostęp 10.01.2017)