


R. Wieczorkowski
R. Zieliński



**komputerowe
generatory
liczb
losowych**

Wydawnictwa
Naukowo-Techniczne

Spis treści

Przedmowa.....	4
Wykaz niektórych oznaczeń.....	5
1. „Liczby losowe”.....	6
2. Generatory liczb losowych o rozkładzie równomiernym.....	8
2.1. Wprowadzenie.....	8
2.2. Generatory liniowe.....	10
2.2.1. Opis.....	10
2.2.2. Okres generatora.....	11
2.2.3. Struktura przestrzenna.....	12
2.2.4. Ogólne generatory liniowe.....	13
2.2.5. Parametry statystyczne.....	13
2.2.6. Wybór parametrów dla generatorów liniowych.....	14
2.3. Generatory oparte na rejestrach przesuwnych.....	16
2.4. Generatory Fibonacciego.....	19
2.5. Kombinacje generatorów.....	20
2.6. Uniwersalny generator liczb losowych o rozkładzie równomiernym.....	21
2.7. Generatory oparte na odejmowaniu z pożyczką i generator ULTRA.....	22
2.8. Generatory oparte na mnożeniu z przeniesieniem.....	23
2.9. Generatory nieliniowe.....	24
2.10. Uwagi o implementacji numerycznej.....	26
2.11. Przykładowe implementacje w języku C.....	26
2.11.1. Inicjowanie generatorów.....	26
2.11.2. Ogólny generator liniowy.....	27
2.11.3. Generator Tauswortha z podrozdziału 2.3.....	27
2.11.4. Generator uniwersalny z podrozdziału 2.6.....	28
3. Generatory liczb losowych o dowolnych rozkładach prawdopodobieństwa.....	30
3.1. Ogólne metody konstrukcji generatorów liczb losowych o dowolnych rozkładach prawdopodobieństwa.....	30
3.1.1. Metoda odwracania dystrybuanty.....	30
3.1.2. Metoda eliminacji.....	32
3.1.3. Metoda superpozycji rozkładów.....	43
3.1.4. Metoda ROU.....	50
3.1.5. Rozkłady dyskretne.....	53
3.2. Metody konstrukcji generatorów dla podstawowych rozkładów prawdopodobieństwa.....	56
3.2.1. Rozkłady dyskretne.....	56
3.2.2. Rozkład wykładniczy.....	61
3.2.3. Rozkład normalny.....	63
3.2.4. Rozkład gamma.....	67
3.2.5. Rozkład beta.....	72
3.2.6. Rozkład Cauchy'ego.....	74
3.2.7. Rozkłady α -stabilne.....	76
3.3. Związki między rozkładami.....	77
4. Generatory liczb losowych o rozkładach wielowymiarowych.....	82
4.1. Przypadek ogólny.....	82
4.2. Rozkłady równomierne w R_m	84
4.2.1. Uwagi ogólne.....	84
4.2.2. Rozkład równomierny na sferze i na kuli w R_m	84
4.2.3. Rozkład równomierny na sympleksie i na powierzchni sympleksu.....	87
4.3. Wielowymiarowy rozkład normalny.....	88
5. Testowanie generatorów liczb losowych.....	89
5.1. Metodyka testowania generatorów.....	89
5.2. Testy zgodności z rozkładem $U(0,1)$	91
5.2.1. Test chi-kwadrat.....	91
5.2.2. Test zgodności z rozkładem wielowymiarowym.....	92
5.2.3. Test OPSO.....	93
5.3. Test Kołmogorowa.....	93
5.3.1. Wprowadzenie.....	95
5.3.2. Testy oparte na statystykach pozycyjnych.....	96
5.3.3. Test sum.....	96
5.3.4. Test d_2	97

5.3.5. Test urodzin dla spacji.....	97
5.3.6. Test najmniejszej odległości w parach.....	98
5.5. Testy serii.....	100
5.6. Testy kombinatoryczne.....	103
5.6.1. Test pokerowy.....	103
5.6.2. Test kolekcjonera.....	104
5.6.3. Test kolizji i test liczby pustych cel.....	104
5.6.4. Test permutacji.....	105
5.6.5. Test oparty na rzędzie losowych macierzy binarnych.....	105
5.7. Testowanie generatorów za pomocą zadań kontrolnych.....	106
6.Prace cytowane.....	107

Przedmowa

Losowanie prób w kontekście badań statystycznych (badania reprezentacyjne, symulacyjne badania estymatorów, testów i statystycznych reguł decyzyjnych) oraz w kontekście obliczeń numerycznych (metody Monte Carlo, klasyczne dla całek i równań z operatorami liniowymi i nowsze dla zadań optymalizacji), jak również symulacyjne badania modeli probabilistycznych w technice, ekonomii, naukach przyrodniczych i praktycznie we wszystkich dziedzinach wiedzy, wymagają wyposażenia współczesnego komputera w odpowiednie narzędzia. Takimi narzędziami są generatory liczb losowych.

Komputery trafiły pod strzechy i fala publikacji poświęconych różnym aspektom ich wykorzystania nie opada. Jesteśmy przekonani, że w czasie, jaki upłynie między postawieniem przez nas ostatniej kropki w komputeropisie tej książki a jej dotarciem do pierwszych Czytelników, pojawi się co najmniej kilkadziesiąt nowych publikacji i programów bezpośrednio związanych z tematyką generatorów liczb losowych. Wierzimy jednak, że to co proponujemy Czytelnikom, będzie jeszcze przez pewien czas aktualne, a przynajmniej ułatwi Im, i jeszcze długo będzie ułatwiało, poruszanie się w gąszczu coraz to nowych osiągnięć w tej dziedzinie.

Warszawa, maj 1997

Autorzy

Wykaz niektórych oznaczeń

a^T	transpozycja wektora a
$[a]$	największa liczba całkowita, mniejsza lub równa a
$\{a\}$	ułamkowa część liczby a
$\ a\ $	norma (długość) wektora a
$a \bmod b$	reszta z dzielenia liczby a przez b
$a \text{ xor } b$	operacja binarna $(a + b) \bmod 2$
A^c	dopełnienie zbioru A
$\text{Cov}(X, Y)$	kowariancja zmiennych losowych X i Y
$D^2(X)$	wariancja zmiennej losowej X
$E(X)$	wartość oczekiwana zmiennej losowej X
$\mathbf{1}_A$	funkcja charakterystyczna zbioru A
$l_m(A)$	miara (Lebesgue'a) zbioru A w R^m
$\ln a$	logarytm naturalny liczby a
$P\{A\}$	prawdopodobieństwo zdarzenia A
R	zbiór liczb rzeczywistych
R^n	przestrzeń euklidesowa n -wymiarowa
sign	funkcja znaku
$\Gamma(p)$	funkcja gamma Eulera
Φ	dystrybuanta rozkładu normalnego $N(0,1)$
ω	zdarzenie elementarne
Ω	przestrzeń zdarzeń elementarnych
\bullet	koniec przykładu
\square	koniec lematu lub twierdzenia

Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin. For, as has been pointed out several times, there is no such thing as a random number - there are only methods to produce random numbers, and a strict arithmetic procedure of course is not such a method.¹⁾

JOHN VON NEUMANN, 1951

1. „Liczby losowe”

Wykonajmy serię niezależnych rzutów monetą i zanotujmy obserwowane wyniki, pisząc, 0 - gdy wynikiem rzutu jest reszka, lub 1 - gdy wynikiem rzutu jest orzeł. Ponieważ prawdopodobieństwo zaobserwowania orla jest takie samo jak prawdopodobieństwo zaobserwowania reszki i równe $1/2$, więc zmienna losowa *wynik rzutu* ma rozkład dwupunktowy i przyjmuje wartości 0 lub 1 z jednakowym prawdopodobieństwem. Mówimy, że ta zmienna losowa ma *rozkład równomierny na zbiorze $\{0,1\}$* . W wyniku opisanego postępowania otrzymamy, więc np. następujący ciąg liczb: 1,0,0,1,1,... Taki ciąg przyjęto nazywać *ciągami liczb losowych o rozkładzie równomiernym na zbiorze $\{0,1\}$* . Monetę, za pomocą, której otrzymujemy tego typu ciągi, nazywamy *generatorem liczb losowych o rozkładzie równomiernym na zbiorze $\{0,1\}$* .

Przygotujmy 10000 jednakowych kartek i ponumerujmy je kolejnymi liczbami czterocyfrowymi: 0000,0001,0002,..., 9999. Wrzućmy wszystkie kartki do urny. Losujmy z urny po jednej kartce, tak żeby w każdym losowaniu każda z nich miała jednakową szansę na wyjęcie z urny. Zmienna losowa *wylosowany numer* ma *rozkład równomierny na zbiorze $\{0000,0001,0002,$*

1) Jawnie grzeszy, kto opowiada o arytmetycznych procedurach generowania liczb losowych. Nie ma bowiem, jak to już nieraz mówiono, czegoś takiego, jak liczba losowa. Istnieją metody losowego wytwarzania liczb, ale oczywiście żadna deterministyczna procedura arytmetyczna nie jest taką metodą. (Tłumaczenie autora)

..., 9999}. Jeżeli opisane losowanie powtórzymy wielokrotnie (po każdym z nich zwracając wylosowaną kartkę z powrotem do urny), to otrzymamy np. następujący ciąg liczb: 1722,4355, 0234,... Taki ciąg będziemy nazywać *ciągiem liczb losowych o rozkładzie równomiernym na zbiorze* $\{0000,0001,1\ 0002,\dots, 9999\}$, a urnę z ponumerowanymi kartkami - *generatorem liczb losowych o rozkładzie równomiernym na tym zbiorze*.

Weźmy pod uwagę ruletkę z kołem o obwodzie równym 1 i niech A będzie wyróżnionym punktem na obwodzie tarczy tej ruletki. Tarczę wprawiaj się w ruch obrotowy i rzuca na nią kulkę K , która, dzięki odpowiedniej konstrukcji ruletki, zatrzymuje się zawsze na brzegu tarczy. Niech U będzie długością łuku AK . Załóżmy, że zmienna losowa U ma rozkład równomierny na przedziale $(0,1)$ - rozkład ten będziemy oznaczali przez $U(0,1)$. W wyniku kilku eksperymentów z ruletką otrzymamy np. następujący ciąg liczb: 0.2217,1 0.5543, 0.3402,... Taki ciąg przyjęto nazywać *ciągiem liczb losowych o rozkładzie równomiernym na przedziale* $(0,1)$, a ruletkę - *generatorem liczb losowych o rozkładzie równomiernym $U(0,1)$* .

Zadania, w których do rozwiązywania używa się ciągów liczb losowych, można podzielić na trzy grupy.

Grupę pierwszą (również pierwszą historycznie) tworzą zadania związanej z badaniami reprezentacyjnymi. Problem opisu różnych zbiorów (populacji) za pomocą próbek losowanych z tych zbiorów jest typowym problemem statystycznym. Przykładami są tu badania różnych zjawisk społecznych przez szczegółowy opis jednostek wybranych losowo z populacji interesujących badacza obiektów (ludzi, zakładów pracy, środowisk, szkół, itp.) lub zadania ze statystycznej kontroli jakości, w których partie różnych towarów opisuje się na podstawie badania losowo wybranych próbek tych towarów. W praktyce stosuje się tu nie tylko takie schematy losowania, jak podany wyżej przykład losowania prostego z urny; obszerny przegląd różnych metod generowania prób losowych w takich sytuacjach można znaleźć np. w książce Zasępy (1962) i w książce Brachy (1996).

Grupę drugą stanowią zadania numeryczne rozwiązywane metodami Monte Carlo. Zadanie numeryczne (typowym przykładem jest zadanie obliczania wartości danej całki) zastępuje się wówczas zadaniem rachunku prawdopodobieństwa, które z kolei rozwiązuje się na drodze eksperymentu statystycznego. Podstawową częścią tego eksperymentu jest losowanie próbki z odpowiedniej populacji, a więc generowanie odpowiedniego ciągu liczb losowych. Obszerny wykład różnych sposobów postępowania w takich sytuacjach można znaleźć w książkach: Hammersleya i Handscomba (1961), Zielińskiego (1970), Jermakowa (1976), Niederreitera i Shiue (1995). Najnowsze metody tego typu dotyczą stochastycznych algorytmów szukania minimum globalnego danej funkcji, czemu poświęcona jest minimonografia Zielińskiego i Neumanna (1986) oraz liczne prace dotyczące *symulowanego wyżarzania* (ang. *simulated annealing*) z ostatniego dziesięciolecia; aktualne wyniki na ten temat można znaleźć w pracy Wieczorkowskiego (1995).

Grupę trzecią stanowią zadania związane z badaniem różnych zjawisk i procesów (technicznych, ekonomicznych, przyrodniczych) za pomocą ich komputerowej symulacji (modelowania). O przebiegu takich procesów de-dują najczęściej czynniki losowe, a modelowanie wpływu tych czynników sprowadza się do losowania próbek z odpowiednich rozkładów prawdopodobieństwa, czyli do generowania odpowiednich ciągów liczb losowych.

W sytuacjach realnych korzystanie z opisanych wyżej generatorów liczb losowych (moneta, urna lub ruletka) jest, oczywiście, najczęściej niemożliwe i w praktyce takie „prawdziwe” generatory są zastępowane pewnymi ich namiastkami. Jeszcze do niedawna powszechnie posługiwano się *tablicami liczb losowych*, obecnie stosuje się odpowiednie programy komputerowe. Wszędzie dalej w tej książce mówiąc o *generatorach liczb losowych* mamy właśnie na myśli takie programy. Podstawową rolę odgrywają *generatory liczb losowych o rozkładzie równomiernym $U(0,1)$* - omawiamy je w rozdz. 2. Liczby otrzymywane w wyniku obliczeń wykonywanych za pomocą programów komputerowych nie są oczywiście „tak losowe” jak liczby uzyskiwane przez rzuty monetą, losowanie z urny lub obracanie koła ruletki. W celu podkreślenia tego faktu używa się często nazw *liczby pseudolosowe* lub *liczby quasi-losowe*, ale nie będziemy

tutaj rygorystycznie trzymali się tych nazw; dalej mówimy po prostu o *programowych generatorach liczb losowych*.

Jak już wspomnieliśmy, w zastosowaniach potrzebne są liczby losowe o różnych rozkładach prawdopodobieństwa, np. liczby o rozkładzie normalnym lub liczby opisujące realizacje procesu Poissona. Wszystkie takie liczby losowe można otrzymać przez odpowiednie manipulacje liczbami z generatora liczb losowych o rozkładzie równomiernym $U(0,1)$. Mówimy o tym dokładnie w rozdz. 3. (przypadek rozkładów jednowymiarowych) i w rozdz. 4. (rozkłady wielowymiarowe).

W związku ze stosowaniem różnych generatorów liczb losowych powstaje problem *testowania* tych generatorów. Ogólnie mówiąc, sprowadza się on do testowania odpowiednich hipotez statystycznych o generatorze, co omówimy w rozdz. 5.

Komputerowe generatory liczb losowych są w dzisiejszych czasach jednym z najczęściej używanych narzędzi każdego badacza: matematyka, lekarza, inżyniera, ekonomisty, socjologa, chemika i fizyka - do symulacji procesów losowych. Użytkownik tego komputerowego narzędzia zwykle bardzo szybko zaczyna doceniać jego potężne możliwości, a zarazem, posługując się nim, odczuwa przyjemność i satysfakcję. Życzymy tego naszym Czytelnikom.

2. Generatory liczb losowych o rozkładzie równomiernym

2.1. Wprowadzenie

Najprostszymi generatorami liczb losowych są oczywiście generatory fizyczne, jak np. wymienione w rozdz. 1. moneta, urna lub ruletka. Są to w ścisłym znaczeniu tego słowa urządzenia losowe. Generatory takie mają jednak niewielkie zastosowanie praktyczne i mogą być przydatne tylko do losowania niedużych próbek do badań reprezentacyjnych. Można zbudować tego typu urządzenia współpracujące z komputerem, np. w przeszłości wielokrotnie konstruowano urządzenia wykorzystujące zjawisko promieniotwórczości lub zjawisko szumów elementów elektronicznych. Istotnym problemem jest tu jednak problem stabilności takich generatorów: niewielkie zmiany własności fizycznych źródła lub zmiany warunków otoczenia mogą pociągnąć za sobą istotne zmiany własności probabilistycznych otrzymywanych ciągów liczb losowych. W związku z tym generatory fizyczne wymagają dodatkowych urządzeń testujących i ewentualnie korygujących, co znacznie komplikuje ich budowę. Pojawia się trudny problem synchronizacji okresów testowania i okresów eksploatacji takich generatorów. Wszystkie te kłopoty z jednej strony i łatwość eksploatacji generatorów programowych z drugiej strony spowodowały, że współcześnie te ostatnie całkowicie wyparły generatory fizyczne, a pewne namiastki generatorów fizycznych (np. zegar systemowy) używane są tylko do inicjowania generatora programowego.

Wszystkie generatory programowe (ponieważ dalej mówimy tylko o takich generatorach, więc przymiotnik „programowe” będziemy opuszczali), jakie prezentujemy w tym rozdziale, produkują dodatnie liczby całkowite lub bity (liczby 0 lub 1). Nieujemne całkowite liczby losowe produkowane przez rozważany generator będziemy oznaczali w zasadzie dużymi literami $X, X_1, X_2, \dots, X_n, \dots$, ale czasami użyjemy innej litery, np. Y lub Z . Te liczby będą zawsze mniejsze od pewnej ustalonej dodatniej liczby całkowitej m co oczywiście jest związane z arytmetyką komputera: np. w komputerze 32-bitowym mamy $m = 2^{32}$. Losowe bity będziemy zwykle oznaczali przez b_1, b_2, \dots

Liczby z przedziału (0,1), które mają reprezentować liczby losowe o rozkładzie równomiernym na przedziale (0,1), będziemy oznaczali literą U (lub V), ewentualnie z odpowiednimi indeksami. Otrzymujemy je zawsze w wyniku operacji dzielenia $U = X/m$ lub operacji składania bitów losowych w liczbę ułamkową: $U = 0. b_1, b_2, \dots$

Za najwcześniejszy algorytm programowego generowania liczb losowych jest uważany tzw. *algorytm kwadratowy von Neumanna* (Hammer(1951)). Podstawowa idea generatora von Neumanna polega na generowaniu kolejnych N -cyfrowych (N - parzyste) nieujemnych całkowitych liczb losowych X_n za pomocą prostej formuły $X_n = f(X_{n-1})$, gdzie funkcja f jest określona w następujący sposób: oblicza się kwadrat liczby X_{n-1} i, ewentualnie dopisując odpowiednią liczbę zer na początku, otrzymuje się wynik będący liczbą $2N$ -cyfrową. Za kolejną liczbę X_n przyjmuje się liczbę utworzoną z N środkowych cyfr tego wyniku. Okazało się jednak, że generator von Neumanna produkuje zbyt krótkie tablice liczb losowych (patrz np. Gajewski i Zieliński (1965)) i z tego powodu został zaniechany.

Idea von Neumanna znajduje zastosowanie i rozwinięcie we współcześnie stosowanych generatorach: obecnie używane generatory programowe liczb losowych produkują ciągi liczb X_0, X_1, \dots , przy czym każdy element takiego ciągu jest obliczany za pomocą ściśle określonej formuły matematycznej, zastosowanej do pewnej liczby poprzednich elementów. Odnotujmy od razu, że tak tworzone ciągi liczb muszą być ciągami okresowymi, co rażąco koliduje z losowością. Oznacza to, że istnieją liczby naturalne v i P takie, że dla $i \geq v$ mamy $X_i = X_{i+P}, j = 1, 2, \dots$. Fragment ciągu $X_0, X_1, \dots, X_{v+P-1}$ nazywamy *okresem aperiodyczności ciągu*, natomiast liczbę P *okresem ciągu*. Wprowadzone pojęcia można zilustrować następująco:

okres ciągu

$$X_0, X_1, \dots, X_v, X_{v+1}, \dots, X_{v+P-1}, X_{v+P}, \dots$$

Okres ciągu zwykle daje się wyznaczyć teoretycznie, chociaż w niektórych Przypadkach może to być trudne. Mówimy o tym dokładniej przy szczegółowej prezentacji wybranych generatorów.

Obecnie najczęściej stosowanymi generatorami są: generatory liniowe, generatory oparte na rejestrach przesuwanych, uogólnione generatory Fibonacciego, generatory oparte na odejmowaniu z pożyczką, na mnożeniu 2 przeniesieniem oraz generatory nieliniowe. Przedstawimy dokładniej wymienione klasy generatorów. W tej prezentacji położymy nacisk na podstawowe idee konstrukcji, które zilustrujemy przykładami pochodzącymi z najnowszej literatury. Liczba publikacji z tej dziedziny rośnie lawinowo. Wśród ostatnio wydanych pozycji przeglądowych poświęconych generatorom liczb losowych o rozkładzie równomiernym chcielibyśmy wyróżnić książkę Tezuki (1995). Najnowsze informacje na temat generatorów (w postaci opisów nowych technik, pakietów programów w różnych językach programowania oraz odnośników do literatury) zawiera światowa sieć komputerowa Internet (poprzez serwery WWW, *ftp*, listy dyskusyjne itp.). Poruszanie się w tym gąszczu informacji ułatwiają specjalne serwery wyszukujące; jeżeli chodzi o interesującą nas tematykę generatorów liczb losowych, mogą to być np. takie słowa kluczowe jak: *mndom number generators, simulation, Monte--Carlo*.

2.2. Generatory liniowe

2.2.1. Opis

Generatory liniowe to generatory postaci

$$X_{n+1} = (a_1 X_n + a_2 X_{n-1} + \dots + a_k X_{n-k+1} + c) \bmod m \quad (2.1)$$

gdzie do, a_0, a_1, \dots, a_k, c i m są ustalonymi liczbami całkowitymi (parametrami generatora), natomiast $a \bmod b$ oznacza resztę z dzielenia liczby a przez liczbę b . Inicjując działanie generatora, użytkownik dostarcza dane początkowe: X_0, X_1, \dots, X_k . W typowych implementacjach (np. Pascal, C i C++) przyjmuje się $k = 1$, co prowadzi do generatora

$$X_{n+1} = (aX_n + c) \bmod m \quad (2.2)$$

po raz pierwszy zaproponowanego przez Lehmera (1951). Jeśli $c = 0$, to otrzymujemy tzw. generator moltiplicatywny, a jeśli $c \neq 0$, mówimy o generatorze mieszanym. W dalszym ciągu dokładniej omówimy generatory liniowe postaci (2.2), gdyż z obszernej klasy generatorów liniowych one właśnie są standardowo używane we współczesnych komputerach. Informacje na temat ogólnego generatora liniowego (2.1) przedstawimy w p. 2.2.4.

Zauważmy przede wszystkim, że ciągi (2.2) są dość prostymi ciągami deterministycznymi i wobec tego raczej tylko w wyjątkowych przypadkach możemy „udawać”, że mamy tu do czynienia z ciągami liczb losowych. Okazuje się, że na drodze czysto arytmetycznych rozważań niektóre z takich ciągów można od razu zdyskwalifikować.

Po pierwsze, ciągi produkowane przez generatory liniowe są ciągami okresowymi. Na przykład, jeżeli w ciągu (2.2) pewna liczba pojawi się po raz drugi (a musi to nastąpić wcześniej lub później, bo istnieje tylko m różnych reszt z dzielenia przez m), to od tej pory cały ciąg będzie już tylko reprodukcją swojego poprzedniego odcinka. Jeżeli okres ciągu jest zbyt mały, to liczby $C_i = X_i/m, i = 1, 2, \dots$, będą zbyt rzadko wypełniały przedział $(0,1)$ i ciąg takich liczb nie będzie mógł być zaakceptowany jako ciąg symulowanych realizacji zmiennej losowej o rozkładzie równomiernym $U(0,1)$. O tym jak wybierać parametry generatora, żeby zagwarantować dostatecznie duży jego okres, powiemy w p. 2.2.2.

Po drugie, okazuje się, że przy ustalonej liczbie $d \geq 1$ punkty

$$(U_1, U_2, \dots, U_d), (U_2, U_3, \dots, U_{d+1}), \dots \quad (2.3)$$

jak również punkty

$$(U_1, U_2, \dots, U_d), (U_{d+1}, U_{d+2}, \dots, U_{2d}), \dots \quad (2.4)$$

„bardzo nielosowo” wypełniają kostkę jednostkową T^d (tzn. przedział $[0,1]^d$ w d -wymiarowej przestrzeni R^d). Tu znowu można przez odpowiednie manipulacje parametrami generatora uzyskać mniej lub bardziej zadowalające wypełnienia; mówimy o tym w p. 2.2.3.

Po trzecie, jeżeli wynikowy ciąg U_1, U_2, \dots ma udawać realizację ciągu niezależnych zmiennych losowych o rozkładzie równomiernym $U(0,1)$, to średnia produkowanych przez generator liczb powinna być równa $1/2$, ich wariancja $1/12$, a współczynniki autokorelacji w ciągu tych liczb powinny być równe zero. Dla ciągów (2.2) znane są teoretyczne wzory opisujące te wielkości - zależą one od parametrów generatora. Więcej informacji na ten temat podamy w p. 2.2.5.

Po czwarte, jeżeli wynikowy ciąg U_1, U_2, \dots ma udawać realizację ciągu niezależnych zmiennych losowych o rozkładzie równomiernym $U(0,1)$, to ciągi liczb produkowane przez generator powinny spełniać różne testy statystyczne. Sprawie testowania generatorów poświęcamy cały odrębny rozdział 5. W punkcie 2.2.6 podamy przykładowo kilka generatorów, które pozytywnie przeszły różne kryteria teoretyczne i testy statystyczne.

2.2.2. Okres generatora

Okres generatora liniowego (2.2) jest równy $P = \min\{i: X_i = X_{o,i} > 0\}$ (zauważmy, że parametr aperiodyczności $v = 0$). Okres ten nie może oczywiście przekraczać liczby m . Taki okres przy odpowiednim wyborze parametrów może osiągnąć generator mieszany, ale nie może go osiągnąć generator multiplikatywny. Jeżeli np. $m = 2^L$, to okres generatora multiplikatywnego nie przekracza liczby 2^{L-2} , a jeżeli m jest liczbą pierwszą, to maksymalny okres generatora multiplikatywnego jest równy $m - 1$.

Generator multiplikatywny ze stałą $m = 2^L$, $L \geq 4$, osiąga maksymalny okres tylko wtedy, gdy X_0 jest liczbą nieparzystą oraz $a = 3 \pmod 8$ lub $a = 5 \pmod 8$ (zauważmy, że fakt osiągnięcia maksymalnego okresu nie zależy od liczby L , która tylko decyduje o tym, jak długi jest ten maksymalny okres). Przykładem generatora o maksymalnym okresie jest generator RANDU z parametrami $a = 2^{16} + 3$ i $m = 2^{31}$, który był standardowo używany w komputerach IBM360/370 i PDP11. Ma on jednak bardzo krótki okres, a ponadto nie spełnia niektórych testów statystycznych. Innym przykładem jest generator RNB z parametrami $a = 2^2 \cdot 23^7 + 1$ i $m = 2^{31}$, opisany, uzasadniony i statystycznie przetestowany w pracy Zielińskiego (1966).

Generator multiplikatywny z liczbą pierwszą m osiąga maksymalny okres tylko wtedy, gdy $a^{(m-1)/p} \not\equiv 1 \pmod m$ dla każdego czynnika pierwszego p liczby $m - 1$. Przykładem takiego generatora jest generator z parametrami $a = 7^5$ oraz $m = 2^{31} - 1$ (m jest przykładem tzw. *liczby Mersenne'a*, czyli liczby postaci $2^p - 1$, gdzie p jest liczbą pierwszą).

Generator mieszany osiąga pełny okres m wtedy, gdy jednocześnie są spełnione trzy następujące warunki:

- a) liczby c i m nie mają wspólnych dzielników,
- b) $a = 1 \pmod p$ dla każdego czynnika pierwszego liczby m ,
- c) $a = 1 \pmod 4$, jeżeli 4 jest dzielnikiem liczby m .

Przykład takiego generatora uzyskujemy przyjmując: $a = 69069$, $c = 1$, $m = 2^{32}$.

Dowody odpowiednich twierdzeń można znaleźć w monografiach Janssona (1966), Knutha (1981) i Ripleya (1987).

Zwracamy uwagę jeszcze na jedną własność związaną z okresowością ciągów liczb produkowanych przez generatory liniowe w przypadku parametru m nie będącego liczbą pierwszą. Mianowicie, jeżeli liczby X otrzymywane z takiego generatora zapiszemy w pewnym systemie pozycyjnym, w szczególności w systemie binarnym, w postaci $X = b_1 b_2 \dots b_L$ i następnie przez obcięcie początkowych (najstarszych) bitów utworzymy nowe liczby, np. $X' = b_1 b_{l+1} \dots b_L$, to te nowe liczby także utworzą ciąg okresowy, a okres nowego ciągu będzie krótszy od okresu ciągu wyjściowego. W szczególności dla $c = 0$, $m = 2^L$, końcowe (najmłodsze) bity tworzą ciąg o okresie równym 1; np. jeśli $a = 8k+5$, $X_0 = 4s+1$ (gdzie k, s są pewnymi liczbami całkowitymi) oraz $m = 2^L$, to trzy końcowe (najmłodsze) bity tworzą ciąg o wartościach postaci 001 i 101 (Zieliński (1979), Andersen (1990)). Wynika stąd, że liczby losowe z omawianego generatora liniowego mogą być używane tylko w takich obliczeniach, w których końcowe bity liczb nie odgrywają istotnej roli; w szczególności poszczególne cyfry liczb nie mogą być traktowane jak cyfry losowe, chociaż w przypadku „prawdziwych” ciągów niezależnych zmiennych losowych o rozkładzie

równomiernym $U(0,1)$ takie postępowanie jest w pełni uzasadnione.

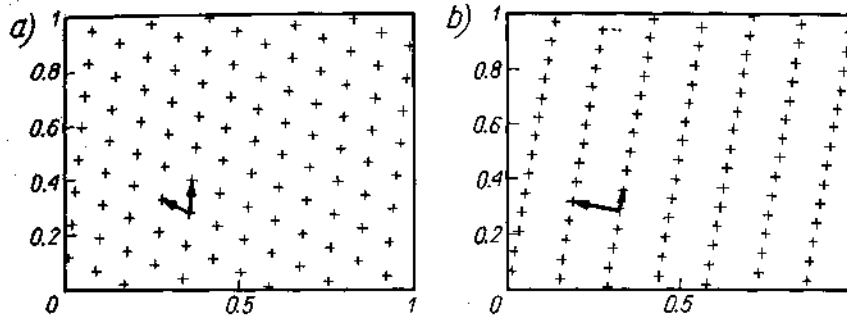
2.2.3. Struktura przestrzenna

Jeżeli U_1, U_2, \dots jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym $U(0,1)$, to punkty losowe (2.3) i (2.4) mają rozkład równomierny na kostkach jednostkowych T^d w d -wymiarowej przestrzeni R^d . Jeżeli natomiast punkty (2.3) i (2.4) są utworzone przez liczby U_1, U_2, \dots produkowane przez generator liniowy, to po pierwsze nie wypełniają one tych kostek dostatecznie gęsto i po drugie - układają się w tych kostkach w regularne struktury geometryczne.

Zbiór punktów (2.3) jest zawarty w zbiorze postaci $L^d \cap (x + \Lambda)$ gdzie

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_d), \bar{x}_1 = 0, \bar{x}_2 = c/m, \bar{x}_i = a\bar{x}_{i-1} + c/m, i = 3, \dots, d$$

oraz Λ jest kratą, czyli zbiorem wektorów postaci $t_1 e_1 + \dots + t_d e_d$ dla pewnych liniowo niezależnych wektorów bazowych e_1, e_2, \dots, e_d i liczb t_1, t_2, \dots, t_d przebiegających zbiór liczb całkowitych Z . Przykładowe zbiory $L^d \cap (x + \Lambda)$ wraz z wektorami bazowymi, przedstawiono na rys. 2.1.



Rys. 2.1. Wykres zbioru punktów $(X_i/m, X_{i+1}/m)$, $i = 0, 1, 2, \dots$ z generatora multiplikatywnego dla: a) $m = 101, a = 12$; b) $m = 101, a = 7$

Zbiór punktów (2.4) jest oczywiście pewnym podzbiorem zbioru (2.3).

Szczególne rolę w opisie rozkładu punktów (2.3) w kostce T^d odgrywa tzw. *baza fizyczna* $\{e_1, e_2, \dots, e_d\}$, charakteryzująca się tym, że e_1 jest najkrótszym wektorem w Λ oraz dla $i \geq 2$ e_i jest najkrótszym wektorem w podprzestrzeni ortogonalnej do podprzestrzeni rozpiętej na wektorach e_1, e_2, \dots, e_{i-1} (por. rys. 2.1). Jedną z miar zagęszczenia punktów (2.3) w T^d jest wtedy długość l_d najdłuższego wektora bazy fizycznej, a jedną z miar równomierności rozkładu tych punktów jest iloraz l_d/l_1 , gdzie l_1 jest długością najkrótszego wektora tej bazy. Inną miarą gęstości wypełnienia kostki T^d punktami (2.3) jest maksymalna odległość D_d między hiperpłaszczyznami, na których leżą te punkty. Pomysł obliczania wielkości D_d pochodzi z pracy Coveyou i MacPhersona (1967), gdzie zastosowano metodę analizy Fouriera, stąd w literaturze problem wyznaczania D_d jest nazywany *testem spektralnym*.

Znane są następujące nierówności wiążące wprowadzone wielkości (Ripley (1987), Tezuka (1995)):

$$1 \leq l_d / D_d \leq \gamma_d^d, \quad m D_d^d \gamma_d \geq 1 \quad \text{dla} \quad d \geq 2$$

gdzie γ_d jest tzw. *stałą Hermite'a*; a dokładne wartości tej stałej są znane tylko

dla $d \leq 8$ ($D_d^{2d} = 1,4/3, 2,4, 8, 64/3, 64, 256, d = 1, 2, \dots, 8$) (Knuth (1981)).

Odległość D_d można równoważnie zdefiniować jako długość najkrótszego niezerowego wektora tzw. *bazy dualnej* do bazy $\{e_1, e_2, \dots, e_d\}$. Bazę dualną $\{e_1^*, e_2^*, \dots, e_d^*\}$ określa się za

pomocą zależności: $e_i^t e_j^t = \delta_{ij}$ (gdzie δ_{ij} jest deltą Kroneckera: $\delta_{ij} = 1$ dla $i = j$, $\delta_{ij} = 0$ dla $i \neq j$). Podana definicja sprowadza obliczanie D_d do problemu minimalizacji całkowitoliczbowej z kwadratową funkcją celu.

Więcej szczegółów oraz wybrane algorytmy numeryczne związane z badaniem struktury geometrycznej generatorów liniowych można znaleźć w obszernej literaturze (Coveyou i Mac Pherson (1967), Knuth (1981), ASerbach i Grothe (1985), Fincke i Pohst (1985)).

Pewne oszacowania z góry wielkości l_1 i l_d można uzyskać przez obliczenie długości odpowiednich wektorów dowolnej bazy. Taką bazą może być np. baza zawierająca wektor $e_1 = (1, a, a^2, \dots, a^{d-1})/m$ i wektory $e_i, i \geq 2$, o i -tej składowej równej jedności i pozostałych składowych równych zeru.

Lepsze przybliżenie otrzymuje się poprawiając tę bazę w jeden z następujących sposobów:

(1) W każdej parze wektorów e_i, e_j wektor dłuższy, np. e_j , zastępujemy wektorem $e_j - s e_i$, gdzie s jest zaokrągleniem liczby $e_i^t e_j / (e_i^t e_i)$ do najbliższej liczby całkowitej, która jednocześnie bliższa jest zeru. Kontynuujemy tę procedurę dopóty, dopóki można jeszcze uzyskać redukcję długości rozważanych wektorów (Marsaglia (1972)).

(2) Wektory e_i porządkujemy według długości i każdy z nich zastępujemy najkrótszym wektorem postaci $e_i + \sum_{j < i} c_j e_j$, gdzie $c_j \in \{0, +1, -1\}$ (Ripley (1983)).

Stosując powyższe algorytmy, zawsze po skończonej liczbie kroków osiągamy bazę, której już nie można poprawić, gdyż współrzędne wektorów bazowych są pewnymi wielokrotnościami wartości $1/m$. Na rysunku 2.1 dla dwóch przykładowych generatorów liniowych zaznaczono wektory bazowe zbioru Λ . Wektory te uzyskano wykonując jeden krok zgodnie z algorytmem (1), przy czym baza początkowa miała postać: $e_1 = (1/m, a/m)$, $e_2 = (0, 1)$.

2.2.4. Ogólne generatory liniowe

Jako naturalne uogólnienie generatorów (2.1) rozważa się również generatory postaci

$$X_{n+1} = AX_n \text{ mod } m \quad (2.5)$$

gdzie X_1, X_2, \dots są wektorami w R^k , $k > 1$, A jest macierzą. Operacja mod jest wykonywana „po współrzędnych”. Te ogólniejsze generatory umożliwiają symulowanie wielowymiarowych zmiennych losowych o rozkładzie równomiernym na kostkach T^k , a możliwość wyboru macierzy A pozwala na manipulowanie zależnościami między składowymi generowanych wektorów. Nie będziemy tutaj rozwijać tej problematyki; zainteresowanego Czytelnika odsyłamy do literatury (L'Ecuyer (1990, 1996a), Eichenauer-Hermann, Grothe i Lehn (1989)).

2.2.5. Parametry statystyczne

Jak już podkreślaliśmy, ciągi liczb z generatora liniowego (2.1) są ciągami deterministycznymi i tylko pewne ich własności „nieuporządkowania” usprawiedliwiają stosowanie ich do symulacji ciągów losowych: jeżeli wynikowy ciąg U_1, U_2, \dots ma udawać realizację ciągu niezależnych zmiennych losowych o rozkładzie równomiernym $U(0,1)$, to średnia produkowanych przez generator liczb powinna być równa $1/2$, ich wariancja $1/12$, a współczynniki autokorelacji w ciągu otrzymanych liczb powinny być równe zeru. Dla ciągów produkowanych przez generatory liniowe (2.2) znane są wzory teoretyczne dla tych wielkości (Jansson 1966); zależą one od parametrów generatora. Podamy kilka takich wzorów dla konkretnej klasy generatorów multiplikatywnych w celu zilustrowania zagadnienia.

Weźmy pod uwagę dowolny multiplikatywny generator liczb L -bitowych o maksymalnym okresie,

tzn. o okresie 2^{L-2} . Generator taki produkuje tylko cztery rodzaje ciągów:

- (1) jeśli $c = 3 \pmod 8$ oraz $X_0 = 1, 3, 9$ lub $11 \pmod{16}$, to ciąg X_0, X_1, X_2, \dots jest permutacją liczb postaci $8j + 1$ i $8j + 3, j = 0, 1, 2, \dots, 2^{L-3} - 1$;
- (2) jeśli $c = 1 \pmod 8$ oraz $X_0 = 5, 7, 13$ lub $15 \pmod{16}$, to ciąg X_0, X_1, X_2, \dots jest permutacją liczb postaci $8j + 5$ i $8j + 7, j = 0, 1, 2, \dots, 2^{L-3} - 1$;
- (3) jeśli $c = 5 \pmod 8$ oraz $X_0 = 1 \pmod 4$, to ciąg X_0, X_1, X_2, \dots jest permutacją liczb postaci $4j + 1, j = 0, 1, 2, \dots, 2^{L-2} - 1$;
- (4) jeśli $c = 5 \pmod 8$ oraz $X_0 = 3 \pmod 4$, to ciąg X_0, X_1, X_2, \dots jest permutacją liczb postaci $4j + 3, j = 0, 1, 2, \dots, 2^{L-2} - 1$.

Rozważmy np. ciąg postaci (1). Średnia wszystkich liczb $U_i = X_i/2^L$ produkowanych przez generator jest równa $1/2 - (1/2)^{L-1}$, a ich wariancja $1/12 - 13/(3 \cdot 2^{2m})$. Wynika stąd, że taki ciąg jako całość wypełnia, przy odpowiednio dużym L , przedział $(0,1)$ prawie tak dobrze jak ciąg „prawdziwych” liczb losowych.

Wzory dla współczynników autokorelacji są nieco bardziej zawile, ale wykonanie obliczeń według tych wzorów nie nastęrcza większych trudności; np. dla generatora multiplikatywnego

$$X_{n+1} = aX_n \pmod{2^L} \text{ mamy}$$

$$\begin{aligned} \frac{1}{2^{L-2}} \sum_{j=0}^{2^{L-3}-1} X_j X_{j+r} &= \frac{a_r}{12} (2^{2m} - 6 \cdot 2^m - 1) - \\ &- 32 \sum_{j=0}^{2^{L-3}-1} j \left(\frac{a_r j + a_r / 8}{2^{L-3}} \right) - 4 \sum_{j=0}^{2^{L-3}-1} j \left(\frac{a_r j + a_r / 8}{2^{L-3}} \right) - \\ &- 32 \sum_{j=0}^{2^{L-3}-1} j \left(\frac{a_r j + 3a_r / 8}{2^{L-3}} \right) - 12 \sum_{j=0}^{2^{L-3}-1} j \left(\frac{a_r j + 3a_r / 8}{2^{L-3}} \right) \end{aligned}$$

gdzie $a_r = a^r \pmod{2^L}$. Odpowiednie algorytmy obliczeniowe, a także tablice już obliczonych współczynników autokorelacji w standardowych generatorach liniowych, można znaleźć w monografii Janssona (1966).

Znane są również mniej dokładne, ale wygodniejsze w projektowaniu generatorów wzory, np. wartość współczynnika autokorelacji między dwiema kolejnymi liczbami z generatora mieszanego (2.2) znajduje się w przedziale $\frac{1}{6c} \left| 1 - \frac{1}{c} \right| \pm \frac{1}{c}$ (Greenberger (1961, 1962)).

2.2.6. Wybór parametrów dla generatorów liniowych

Ostatecznym kryterium zaakceptowania generatora jest to, że nie został on zakwestionowany przez żaden z zastosowanych testów statystycznych. Metody testowania omówimy dokładnie w rozdz. 5., gdzie również opiszemy liczne testy statystyczne. Tutaj, mówiąc o generatorach liniowych, zwracamy uwagę, że jeżeli wartości pewnych parametrów, jak np. średnia i wariancja produkowanych liczb, okres ciągu lub współczynniki autokorelacji w ciągu

liczb produkowanych przez generator, mogą dla danego generatora być wyznaczone teoretycznie w sposób ścisły, to oczywiście nie ma żadnego sensu testowanie hipotez o takich parametrach za pomocą testów statystycznych.

W obszernej, liczącej już ponad 40 lat, literaturze poświęconej generatorom multiplikatywnym i mieszanym, można znaleźć liczne raporty z wykonanych analiz teoretycznych oraz testów statystycznych i wyróżnić generatory, które najlepiej spełniały przyjęte kryteria porównawcze. W tabeli 2.1 przedstawiamy kilka takich generatorów postaci (2.2); odnośniki do wielu wcześniejszych prac znaleźć można w książce Zielińskiego (1979).

Wszystkie podane w tabeli generatory osiągają maksymalne okresy. Popularność parametrów m postaci 2^{32} lub $2^{31} - 1$ wynika z łatwości implementacji takich generatorów we współczesnych systemach komputerowych. Wiele generatorów z tabeli nadal stanowi standardowe wyposażenie szeroko używanych systemów operacyjnych, języków programowania, a nawet specjalistycznych pakietów oprogramowania do obliczeń naukowych. Należy tutaj podkreślić, że w obecnie realizowanych obliczeniach symulacyjnych z użyciem liczb pseudolosowych okres generatora rzędu 2^{32} jest zbyt mały, gdyż zużycie wszystkich liczb z takiego generatora następuje za szybko. Wielu autorów sugeruje, aby liczba N liczb z generatora używanych w symulacji była dużo mniejsza niż okres generatora P . W pracy Maclarena (1992) uzasadniano, że liczba N nie powinna przekraczać liczby $P^{2/3}$; odpowiednie ograniczenie w przypadku generatorów liniowych, podane przez Ripleya (1987), wynosi $P^{1/2}$.

Generatory liniowe postaci (2.2) nie spełniają pewnych nowszych testów statystycznych, np. testu OPSO (patrz rozdz. 5.). Eksperymenty numeryczne potwierdziły również (np. Marsaglia (1984,1995)) lepsze własności statystyczne generatorów liniowych konstruowanych z parametrem m będącym liczbą pierwszą (jednak komplikuje to implementację generatora oraz wpływa na jego szybkość).

Tabela 2.1

a	c	m	Źródło
$2^2 \cdot 23^7 + 1$	0	2^{35}	Zieliński (1966)
69069	1	2^{32}	Marsaglia (1972)
16807	0	$2^{31}-1$	Park, Miller (1980) Carta (1990)
630360016 397204094	0 0	$2^{31}-1$ $2^{31}-1$	Fishman, Moore (1982)
410092949	0	2^{32}	Borosh, Niederreiter (1983)
742938285	0	$2^{31}-1$	Fishman, Moore (1986)
40692	0	$2^{31} - 249$	L'Ecuyer (1988)
1099087573	0	2^{32}	Fishman (1990)
68909602460261	0	2^{48}	Fishman (1990)

Z ogólniejszych generatorów liniowych (2.1) dobrą ocenę statystyczną (w tym we wspomnianych nowszych testach) uzyskały m.in. następujące generatory (Marsaglia (1995)):

- 1) $X_n = (1176X_{n-1} + 1476X_{n-2} + 1776X_{n-3}) \bmod (2^{32} - 5)$
- 2) $X_n = 2^{13}(X_{n-1} + X_{n-2} + X_{n-3}) \bmod (2^{32} - 5)$
- 3) $X_n = (1995X_{n-1} + 1998X_{n-2} + 2001X_{n-3}) \bmod (2^{35} - 849)$
- 4) $X_n = 2^{19}(X_{n-1} + X_{n-2} + X_{n-3}) \bmod (2^{35} - 1629)$

Generatory te osiągają maksymalne okresy równe $m^3 - 1$, gdzie liczba m jest odpowiednim modułem: dla pierwszego i drugiego z powyższych generatorów równym $2^{32}-5$, dla trzeciego $2^{35}-849$ oraz dla czwartego $2^{35}-1629$. Przedstawione generatory mogą być łatwo zaimplementowane na współczesnych komputerach w każdym języku programowania. Przykładową implementację ogólnego generatora liniowego zamieszczamy w podrozdz. 2.11 (patrz również uwagi w podrozdz. 2.10).

2.3. Generatory oparte na rejestrach przesuwnych

Przyjmijmy, że k jest ustaloną liczbą naturalną i weźmy pod uwagę ciąg bitów zdefiniowany wzorem rekurencyjnym

$$b_i = (a_1 b_{i-1} + \dots + a_k b_{i-k}) \bmod 2, \quad i = k + 1, k + 2, \dots \quad (2.6)$$

gdzie współczynniki a_1, a_2, \dots, a_k są stałymi binarnymi, tzn. liczbami 0 lub 1, oraz b_1, b_2, \dots, b_k jest ustalonym ciągiem inicjującym.

Zależność (2.6) można opisać za pomocą operatora logicznego xor, zwanego *różnicą symetryczną* lub *alternatywą wyłączającą*, o następującej tabelce działań:

a	b	$a \text{ xor } b$
0	0	0
0	1	1
1	0	1
1	1	0

Inaczej to ujmując, dla zmiennych boolowskich mamy po prostu
 $a \text{ xor } b = (a + b) \bmod 2$

Dalej będziemy używać tego operatora również w odniesieniu do liczb całkowitych; w takim przypadku wynik jest liczbą całkowitą złożoną z bitów powstałych w wyniku działania operatora na poszczególnych pozycjach reprezentacji binarnej argumentów.

Wzór (2.6) przyjmuje wtedy równoważną postać: jeśli $a_{j1} = \dots = a_{jk} = 1$, a pozostałe współczynniki są równe zeru, to

$$b_i = b_{i-j_1} \text{ xor } b_{i-j_2} \text{ xor } \dots \text{ xor } b_{i-j_k}$$

Ciąg bitów (2.6) jest oczywiście ciągiem okresowym. Ponieważ istnieje 2^k różnych układów k-elementowych (b_1, b_2, \dots, b_k) , okres ciągu (2.6) nie może być większy od 2^k . Faktycznie może on być równy co najwyżej $2^k - 1$, bo gdyby pojawiło się w nim k kolejnych zer, to cały ciąg musiałby składać się z samych zer. Mówiąc dalej o ciągu o maksymalnym okresie, mamy na myśli ciąg o okresie $2^k - 1$.

Metodami algebraicznymi można badać, dla jakich współczynników a_1, a_2, \dots, a_k ciąg (2.6) ma maksymalny okres. Nie będziemy tutaj rozwijać tego technicznego wątku (odsyłamy Czytelnika np. do pracy Golomba (1967)), ograniczymy się natomiast do prostszego, ale dla nas atrakcyjnego przypadku, gdy w formule rekurencyjnej (2.6) tylko dwa współczynniki a_i są różne od zera. Schemat iteracyjny (2.6) ma wtedy postać

$$b_i = b_{i-p} \text{ xor } b_{i-q} \quad (2.7)$$

dla pewnych ustalonych liczb naturalnych p oraz q . Bez zmniejszania ogólności rozważań przyjmijmy, że $p > q$.

W tabeli 2.2 podajemy przykładowe wartości parametrów p i q , dla których ciąg (2.7) ma maksymalny okres. Tabelę tę utworzono na podstawie tabeli z książki Ripleya (1987), gdzie podano wartości (p,q) zapewniające maksymalny okres dla $p < 36$ oraz tabeli z Berdnikova, Turtii i Compagnera (1996), w której z kolei zawarto znane wartości (p,q) , dające maksymalny

Tabela 2.2

P	q	P	q
2	1	33	13
3	1	35	2
4	1	36	11
5	2	89	38
6	1	127	1, 7, 15, 30, 63
7	1,3	521	32, 48, 158, 168
9	4	607	105, 147, 273
10	3	1279	216, 418
11	2	2281	715, 915, 1029
15	1,4,7	3217	67, 576
17	3,5,6	4423	271, 369, 370, 649,
18	7		1393, 1419, 2098
20	3	9689	84, 471, 1836, 2444, 4187
21	2	19937	881, 7083, 9842
22	1	23209	1530, 6619, 9739
23	5,9	44497	8575, 1034
25	3,7	110503	25230, 53719
28	3, 9, 13	132049	7000, 33912, 41469,
29	2		52549, 54454
31	3, 6, 7, 13		

okres ciągu (2.7) dla liczb $p < 132049$, gdzie $2^p - 1$ są liczbami pierwszymi (założenie to znacznie upraszcza obliczenia). Dodatkowo można rozważać pary współczynników $(p, p - q)$, gdyż jeśli ciąg (2.7) ma maksymalny okres dla pewnych p i q , to własność tę ma również schemat iteracyjny z parametrami p i $p - q$.

Istnieje wiele sposobów uzyskiwania L -bitowych liczb losowych o wartościach w przedziale $(0,1)$ na podstawie ciągu bitów $(b_i, i = 1, 2, \dots)$. Najprostszy polega na konstruowaniu ich za pomocą wzoru

2. Generatory o rozkładzie równomiernym

$$U_i = \sum_{j=1}^L 2^{-j} b_{is+j} = 0.b_{is+1} \dots b_{is+L}, \quad i=0,1,2,\dots \quad (2.8)$$

gdzie s jest ustaloną dodatnią liczbą całkowitą oraz $s \leq L$. Jeżeli $s < L$, to kolejne liczby U_i wykorzystują te same podciągi bitów, a jeżeli $s = L$, to liczby U_i i U_{i+1} utworzone są z rozłącznych fragmentów ciągu ($b_i = 1, 2, \dots$). Liczby U_i otrzymuje się łatwo za pomocą rejestrów przesuwnych oraz bramek logicznych, realizujących operator xor (ten fakt uzasadnia nazwę rozważanej klasy generatorów). Generator (2.8) jest znany w literaturze jako *generator Tauswortha*, gdyż po raz pierwszy był analizowany w pracy Tauswortha (1965). Jeżeli liczba s jest tak wybrana, że nie ma wspólnych dzielników z liczbą $2^k - 1$, to ciąg (2.8) jest ciągiem o maksymalnym okresie, który jest równy $2^k - 1$.

Efektywny algorytm generowania tak zdefiniowanych ciągów ($U_i, i = 1, 2, \dots$) za pomocą ciągu bitów (2.7), w przypadku, gdy $q < p/2$ oraz $0 < s < p - q$, podał w swojej monografii Tezuka (1995). W formalnym zapisie tego algorytmu użyjemy następujących symboli: $A \ll k$ oznacza przesunięcie bitów reprezentacji binarnej liczby A o k pozycji w lewo, natomiast $A \gg k$ oznacza takie przesunięcie w prawo. Przy przesuwaniu w lewo młodsze (zwalniane) bity są zapełniane zerami. Przy przesuwaniu w prawo zwalniane starsze bity są również zapełniane zerami. Zwracamy jednak uwagę na to, że w konkretnej implementacji działanie operatora przesunięcia bitowego w prawo może zależeć od zadeklarowanego typu liczby całkowitej.

W poniższym algorytmie A jest L -bitową liczbą całkowitą o bitach początkowych $b_i, \dots, \&l$ (są to bity składające się na liczbę C/o), natomiast B jest L -bitową zmienną pomocniczą.

ALGORYTM T:IMPLEMENTACJA SCHEMATU TAUSWORTHY (2.8)

DLA $0 < s \leq p - q$

1: $B = ((A \ll q) \text{ xor } A) \ll (L - p)$

2: $A = (A \ll s) \text{ xor } (B \gg (L - s))$

3: *Return A; goto 1*

Algorytm ten jest bardzo łatwy w realizacji np. w języku C, który zawiera operatory przesuwania bitowego oraz operator xor. W podrozdziale 2.11 zamieszczamy przykładową implementację w tym języku. Podana tam realizacja używa dodatkowo kombinacji trzech generatorów omawianego typu za pomocą operatora xor (patrz podrozdz. 2.6). W książce Tezuki (1995) jest zawarta analiza struktury punktów w kostce $(0,1)^d$, $d = 2, 3, \dots, 20$, tworzonych z kolejnych liczb z generatora.

Lewis i Payne (1973) zaproponowali inny schemat generowania L -bitowych liczb całkowitych Y_i na podstawie ciągu (2.6), według zależności

$$Y_i = b_i b_{i-l_2} \dots b_{i-l_l}$$

gdzie l_2, \dots, l_l są ustalonymi parametrami przesunięcia. Dla ciągu (2.7) otrzymujemy stąd schemat $Y_i = Y_{i-p} \text{ xor } Y_{i-q}$. Generatory realizujące ten schemat są nazywane *uogólnionymi generatorami opartymi na rejestrach przesuwnych*. Wymagają one odpowiedniego wyboru punktów startowych (Y_0, Y_i, \dots, Y_p) (Lewis i Payne (1973), Collings i Hembree (1986)). Interesujące przykłady efektywnej implementacji w omawianej klasie generatorów zawierają artykuły: Kirkpatricka i Stolla (1981) (dla $p = 250, q = 147$) oraz Ripleya (1990) (dla $p = 521, q = 32$).

W pracy Berdnikova, Compagnera i Turtii (1996) zaproponowano kombinacje (patrz podrozdz. 2.5) kilku generatorów z omawianej klasy, dla odpowiednio dobranych parametrów p i q . Gwarantuje to bardzo duży okres generatora (np. dla konkretnej kombinacji czterech generatorów składowych uzyskano okres rzędu 10^{16378}). Ponadto wykazano, że taka konstrukcja zapewnia bardzo

dobrze własności dotyczące niezależności kolejnych długich (rzędu kilkunastu tysięcy) ciągów liczb z generatora (patrz również Com-pagner i Wang (1993), Compagner (1995)). Kody źródłowe w języku C omawianych generatorów można znaleźć w Internecie (np. <http://www.can.nl> lub <ftp://can.nl>).

2.4. Generatory Fibonacciego

Ciąg rekurencyjny

$$f_n = f_{n-2} + f_{n-1}, \quad n \geq 2, \quad f_0 = f_1 = 1$$

badał już Fibonacci (Leonardo z Pizy) i wyniki swoich badań opublikował w pracy *Liber abaci* w 1202 roku. Ciąg reszt, przy ustalonej dodatniej liczbie całkowitej m , zdefiniowany wzorem

$$X_n = X_{n-2} + X_{n-1} \bmod m, \quad n \geq 2, \quad (2.9)$$

zachowuje się na tyle bezładnie, że już dawno zainteresował matematyków poszukujących prostych modeli dla procesów losowych. Wydaje się, że pierwsze wyniki sprawdzania tego ciągu za pomocą testów statystycznych zostały opublikowane w pracy Taussky i Todd (1956).

Okazało się, że ciągi (2.9) spełniają testy równomierności rozkładu, ale nie spełniają testów niezależności, a więc także wielu innych testów (np. testów serii), gdzie niezależność odgrywa kluczową rolę. Tej wady można było się pozbyć, uogólniając ciąg (2.9):

$$X_n = X_{n-r} + X_{n-s} \bmod m, \quad n \geq r, \quad r > s > 1 \quad (2.10)$$

ale odbywało się to kosztem czasu, co czyniło takie generatory mało konkurencyjnymi dla rozpowszechnionych generatorów multiplikatywnych. Argument, że takie uogólnione generatory miały dużo dłuższy okres od generatorów multiplikatywnych, nie był przekonujący, bo na stosunkowo wolnych komputerach rzadko dochodziło do wyczerpania okresu ciągów multiplikatywnych. Trudno się natomiast dziwić, że w dzisiejszej dobie szybkich komputerów generatory Fibonacciego, w różnych wersjach uogólnień, przeżywają prawdziwy renesans.

Następny krok w tych uogólnieniach polega na zastąpieniu dodawania w (2.10) jakąś inną operacją. Ogólnie, wybraną operację oznaczamy symbolem \diamond i zakładamy, że jest ona wykonywana modulo m . Uogólniony generator Fibonacciego przyjmuje wtedy postać

$$X_n = X_{n-r} \diamond X_{n-s}, \quad n \geq r, \quad r > s \geq 1 \quad (2.11)$$

i jest oznaczany przez $F(r, s, \diamond)$.

Jeżeli $m = 2^l$, to maksymalny okres generatorów $F(r, s, +)$ oraz $F(r, s, -)$ jest równy $(2^r - 1)2^{l-1}$; dla $F(r, s, *)$, gdzie $*$ oznacza tu (i w całej książce) mnożenie, wynosi on $(2^r - 1)2^{n-3}$, natomiast dla generatora $F(r, s, \text{xor})$ wynosi $2^r - 1$. Dowodzi się tego, korzystając z pewnych własności odpowiednich macierzy; szczegóły można znaleźć w pracach Marsaglii (1984) oraz Marsaglii i Tsaya (1985). Generatory $F(r, s, \text{xor})$ dla $m = 2^l$ opisaliśmy już w poprzednim podrozdziale, w klasie generatorów opartych na rejestrach przesuwanych.

Poniżej podajemy tabelkę z przykładowymi parametrami, zapewniającymi maksymalny okres generatora (2.11):

r	s
17	5
31	13
55	24
68	33
97	33
607	273
1279	418

Na przykład, generatory F(17,5,+) oraz F(17,5,-) dla $m = 2^{32}$ dają ciągi o okresie $(2^{17} - 1)2^{31}$, generator F(17,5,*) osiąga okres $(2^{17} - 1)2^{29}$, natomiast F(17,5, xor) osiąga okres $2^{17}-1 = 131071$. Omawiane generatory są łatwe w implementacji.

2.5. Kombinacje generatorów

Doświadczenia z użyciem generatorów skonstruowanych przez łączenie dwóch lub większej liczby prostszych generatorów wykazały, że generatory takie mają lepsze własności statystyczne niż generatory wyjściowe. Przytoczymy wyniki, które potwierdzają te obserwacje.

Założmy, że mamy zmienne losowe X i Y , określone na zbiorze $S = \{1,2,\dots, n\}$, o rozkładach prawdopodobieństwa

$$P\{X=i\}=p_i, \quad P\{Y=i\}=q_i, \quad i=1,2,\dots,n$$

Niech $p = 1, \dots, \infty$ będzie dowolną ustaloną liczbą i niech $t = (t_1, t_2, \dots, t_n)$ będzie ustalonym wektorem. Weźmy pod uwagę p -normę tego wektora, zdefiniowaną wzorem

$$\|t\| = \left(\sum_{i=1}^n t_i^p \right)^{1/p}$$

Dla zdefiniowanej wyżej zmiennej losowej X wprowadźmy miarę $\delta(X)$ „bliskości” rozkładu tej zmiennej do rozkładu równomiernego na zbiorze S

$$\delta(X) = \|(p_1, p_2, \dots, p_n) - (1/n, 1/n, \dots, 1/n)\|$$

Rozpatrzmy dwuargumentowe działanie \circ na zbiorze S takie, którego tabelka tworzy kwadrat łaciński (tzn. każdy jej wiersz i kolumna jest pewną permutacją elementów zbioru S). Można udowodnić (np. Brown i Solomon (1979)), że rozkład zmiennej losowej $X \circ Y$ jest bliższy rozkładowi równomiernemu na zbiorze S w tym sensie, że

$$\delta(X \circ Y) \leq \min \{\delta(X), \delta(Y)\}$$

W odniesieniu do ciągów produkowanych przez generatory liczb losowych ten wynik teoretyczny można interpretować w następujący sposób: jeśli tablica działań operatora \circ jest kwadratem łacińskim, to nowy ciąg $(X_1 \circ Y_1, X_2 \circ Y_2, \dots)$ powinien być bardziej równomiernie (a przynajmniej nie mniej równomiernie) rozłożony niż każdy z ciągów X_1, X_2, \dots i Y_1, Y_2, \dots . Najczęściej za operator \circ przyjmuje się rozważane wcześniej operatory $+$, $-$, $*$, xor. Ponadto okazuje się, że kombinacje

generatorów produkują ciągi, które są nie tylko „bardziej równomierne”, ale również „bardziej niezależne”. Podaną konstrukcję dla dwóch generatorów w naturalny sposób uogólnia się na większą liczbę generatorów składowych. Ponadto kombinacje generatorów produkują ciągi o większym okresie niż okresy ciągów składowych. W szczególności wiadomo, że jeśli ciąg X_1, X_2, \dots ma okres P_1 oraz ciąg Y_1, Y_2, \dots ma okres P_2 , gdzie P_1 i P_2 są liczbami względnie pierwszymi, to okres ciągu $X_1 \circ Y_1, X_2 \circ Y_2, \dots$ wynosi $P_1 P_2$. Fakt ten jest prostym wnioskiem z tzw. *chińskiego twierdzenia o resztach* (patrz np. Graham, Knuth, Patashnik (1996)).

Idea kombinowania generatorów w celu zwiększenia okresu i polepszenia własności statystycznych ma już swoją historię za sobą, ale ciągle jest bardzo często stosowana w najnowszych konstrukcjach generatorów. Pierwsze pomysły dotyczące kombinacji generatorów pojawiły się w pracach Mac Larena i Marsaglii (1965) oraz Marsaglii i Braya (1968). Dużą popularnością wśród użytkowników cieszył się pomysł kombinowania generatorów za pomocą dodawania modulo 1 w dziedzinie liczb rzeczywistych, zaproponowany w pracy Wichmanna i Hilla (1982). Wyniki teoretyczne związane z kombinacjami generatorów można znaleźć również w pracach: Brown i Solomon (1979), Marsaglia (1984), Deng (1990), L'Ecuyer i Tezuka (1991). Najnowsze rezultaty zawiera np. praca L'Ecuyera (1996b).

2.6. Uniwersalny generator liczb losowych o rozkładzie równomiernym

Przez *generator uniwersalny* rozumiemy generator dający identyczne wyniki na komputerach, w których liczby całkowite są reprezentowane, przez co najmniej 16 bitów, a liczby w arytmetyce zmiennopozycyjnej mają przynajmniej 24-bitową reprezentację mantysy. Przedstawimy tutaj szczegółowo generator liczb losowych o rozkładzie równomiernym na przedziale $[0,1)$, opublikowany w pracy Marsaglii, Zamana i Tsanga (1990). Będziemy go nazywali *generatorem MZT*. Spełnia on wszystkie znane testy statystyczne i ma duży okres, równy 2^{144} . Generator ten jest kombinacją dwóch prostszych generatorów.

Pierwszy z nich jest generatorem typu $F(97, 33, 0)$ (patrz podrozdz. 2.4) i produkuje liczby V_n z przedziału $[0,1)$ według wzoru rekurencyjnego:

$$V_n = V_{n-97} \circ V_{n-33}$$

gdzie $x \circ y = x - y$ dla $x \geq y$ oraz $x \circ y = x - y + 1$ dla $x < y$. Zainicjowanie generatora polega na wyznaczeniu liczb V_1, V_2, \dots, V_{97} za pomocą ciągu bitów (b_n) w taki sposób, że $V_1 = 0.b_1 b_2 \dots b_{24}$, $V_2 = 0.b_{25} b_{26} \dots b_{48}$, ... itd. Z kolei ciąg bitów jest generowany za pomocą kombinacji dwóch różnych i łatwych w implementacji generatorów zadanych ciągami liczb całkowitych (y_n) i (z_n) :

$$y_n = (y_{n-3} * y_{n-2} * y_{n-1}) \bmod 179$$

$$z_n = (52z_{n-1} + 1) \bmod 169$$

$$b_n = \begin{cases} 0, & \text{jeśli } y_n z_n \bmod 64 < 32 \\ b_n = 1 & \text{w przeciwnym wypadku} \end{cases}$$

Wybór małych wartości 179 i 169 zapewnia uniwersalność generatora. Użytkownik musi dostarczyć procedurze inicjowania czterech wartości całkowitych:

$y_1 y_2 y_3 \in \{1, 2, \dots, 178\}$ (nie wszystkie równe 1), $z_1 \in \{0, 1, \dots, 168\}$

Okres tego generatora jest równy 2^{120} .

Drugim generatorem jest generator liczb losowych z przedziału (0,1):

$$c_n = c_{n-1} \circ (7654321/16777216), \quad n \geq 2, \quad c_1 = 362436/16777216$$

gdzie $c \circ d = c - d$ dla $c \geq d$ oraz $c \circ d = c - d + (16777213/16777216)$ dla $c < d$, $c, d \in [0, 1)$. Okres tego generatora jest równy $2^{24} - 3$.

Ostatecznie generator MZT przyjmuje postać

$$U_n = V_n \circ c_n$$

Przykładową implementację (w języku C) generatora MZT podamy w podrozdz. 2.11.

2.7. Generatory oparte na odejmowaniu z pożyczką i generator ULTRA

W pracy Marsaglii i Zamana (1991) wprowadzono nową klasę generatorów wykorzystującą operację tzw. *odejmowania z pożyczką* (SWB - ang. *subtract with borrow*). W tej operacji (oznaczymy ją przez Θ) bierze dodatkowo udział parametr c , przyjmujący wartości 0 lub 1, zwany *bitem przeniesienia*. Wynikiem operacji $x \Theta y \bmod m$ są liczby

$$x - y - c + m \quad \text{oraz} \quad c = 1, \quad \text{gdy } x - y - c < 0$$

$$\text{lub } x - y - c \quad \text{oraz} \quad c = 0 \quad \text{w przeciwnym przypadku,}$$

a początkową wartością bitu przeniesienia c jest 0.

Omawiana klasa generatorów została wykorzystana przez Marsaglię i Zamana do konstrukcji generatora ULTRA. Parametrami generatora ULTRA są dodatnie liczby całkowite $L \geq 32$, s , r ($r > s$). Wygodnie jest używać oznaczenia $m = 2^L$. W etapie inicjowania generatora tworzy się ciąg liczb całkowitych $X_1, \dots, X_r \in (0, m)$ oraz ustala się $c = 0$. W etapie roboczym wyznacza się kolejne liczby X_n według wzoru

$$X_n = X_{n-r} \Theta X_{n-s} \bmod m \quad (2.12)$$

Etap inicjowania polega na utworzeniu L-bitowych liczb X_1, X_2, \dots, X_r :

$$X_l = b_L b_{L-1} \dots b_1$$

$$X_2 = b_{2L}b_{2L-1} \dots b_{L+1}$$

Gdzie b_1, b_2, \dots jest ciągiem utworzonym z bitów znaków liczb w_1, w_2, \dots , otrzymywanych w następujący sposób: użytkownik generatora podaje dwie liczby $u_0, v_0 \in (0, m)$, a następnie są obliczane kolejno liczby:

$$u_i = \lambda u_{i-1} \bmod m$$

$$v_i = (v_i \gg k_1) \text{ xor } v_i$$

$$v_i = (v_i \ll k_2) \text{ xor } v_i$$

$$w_i = u_i \text{ xor } v_i$$

gdzie k_1, k_2 są ustalonymi dodatnimi liczbami całkowitymi, a operacje: xor, \ll , \gg zdefiniowano w podrozdz. 2.3.

Generator ULTRA jest bardzo szybki, gdy operację (2.12) programuje się z użyciem języka maszynowego danego procesora. Tak skonstruowany generator ma dobre własności statystyczne oraz długi okres, który można obliczyć teoretycznie, korzystając z następującego twierdzenia:

Jeśli $M = m^r - m^s + 1$ jest liczbą pierwszą, to okres generatora SWB jest równy najmniejszej liczbie naturalnej k , takiej że $m^k \bmod M = 1$.

W stworzonym przez Marsaglię i Zamana pakiecie ULTRA przyjęto $L = 32, r = 37, s = 24$, a do inicjowania $\lambda = 69069$. Tak skonstruowany generator ma okres rzędu 10^{346} . Odpowiednie wywołania generatora ULTRA pozwalają na bezpośrednie otrzymywanie dodatnich całkowitych liczb losowych: 32-, 31-, 8-, 7-, a nawet 1-bitowych (w tym ostatnim przypadku -bitów losowych), jak również liczb losowych z przedziału $(0, 1)$ lub $(-1, 1)$, o pojedynczej lub podwójnej precyzji.

2.8. Generatory oparte na mnożeniu z przeniesieniem

Marsaglia rozwinął ideę konstrukcji generatorów klasy omówionej w podrozdz. 2.7, wprowadzając nową rodzinę generatorów opartych na tzw. *mnożeniu z przeniesieniem* (MWC - ang. *multiply with carry*). Klasa ta daje możliwości łatwej implementacji szybkich generatorów o dużych okresach, przy czym wszystkie grupy bitów uzyskiwanych ciągów liczb całkowitych spełniają wiele znanych testów losowości. Dokładniejszy opis i przykłady implementacji znaleźć można w Internecie, w 16024 artykule grupy dyskusyjnej sci.math.num-analysis oraz na wydanym przez Marsaglię CD-ROMie (1995). Poniżej przedstawimy główne idee tej nowej metody generowania liczb pseudolosowych.

Algorytm typu mnożenia z przeniesieniem jest oparty na zależności

$$X_n = (a_1 X_{n-1} + a_2 X_{n-2} + \dots + a_r X_{n-r} + c) \bmod m$$

gdzie a_1, \dots, a_r są ustalonymi parametrami oraz X_1, \dots, X_r i c inicjujemy dowolnymi wartościami początkowymi; nowa wartość zmiennej c (tzw. wartość przeniesienia) jest liczbą całkowitą, określoną wzorem

$$[(a_1 X_{n-1} + a_2 X_{n-2} + \dots + a_r X_{n-r} + c) / m]$$

(gdzie $[\cdot]$ oznacza część całkowitą).

Wprowadźmy oznaczenie $M = a_r m^r + \dots + a_1 m - 1$. Okres generatora MWC jest równy najmniejszej liczbie naturalnej k , takiej że $m^k \bmod M = 1$. W praktyce przyjmuje się $m = 2^{16}$ lub $m = 2^{32}$, wtedy nowa wartość X_n i nowe przeniesienie c są po prostu dolną i górną częścią odpowiednio 32- lub 64-bitowej liniowej kombinacji 16- lub 32-bitowych liczb całkowitych. Jeśli dobierze się liczbę m tak, aby M i $(M - 1)/2$ były liczbami pierwszymi, to okres będzie równy $(M - 1)/2$ (szczegółowe wyniki dotyczące okresu generatora MWC można znaleźć w pracy Koca (1995)).

Przykładem efektywnej realizacji omawianego generatora jest następująca propozycja Marsaglii:

$$X_n = (12013X_{n-8} + 10666X_{n-7} + 1215X_{n-6} + 1492X_{n-5} + 1776X_{n-4} + 1812X_{n-3} + 1860X_{n-2} + 1941X_{n-1} + c) \bmod 2^{16}$$

Otrzymuje się stąd liczby całkowite 16-bitowe. Aby uzyskać zakres 32 bitów i okres rzędu 2^{250} , dokonuje się połączenia bitów z innego generatora:

$$X_n = (9272X_{n-8} + 7777X_{n-7} + 6666X_{n-6} + 5555X_{n-5} + 4444X_{n-4} + 3333X_{n-3} + 2222X_{n-2} + 1111X_{n-1} + c) \bmod 2^{16}$$

Opisany generator wymaga zainicjowania w postaci 16 liczb całkowitych 16-bitowych, co łatwo zrealizować, np. za pomocą jakiegoś klasycznego generatora liniowego.

Innym przykładem połączenia dwóch generatorów typu MWC jest konstrukcja ciągu liczb całkowitych 32-bitowych za pomocą dwóch ciągów 16-bitowych

$$\begin{aligned} X_n &= 18000X_{n-1} + c_1 \bmod 2^{16} \\ Y_n &= 30903 Y_{n-1} + c_2 \bmod 2^{16} \end{aligned}$$

Okres takiego generatora jest rzędu 2^{60} . W etapie inicjowania użytkownik podaje dwie liczby całkowite 32-bitowe, z których formuje się 16-bitowe wartości X_0, Y_0 oraz c_1, c_2 . Generator ten jest bardzo szybki i łatwo go zaprogramować, używając języka maszynowego. Przykładowa implementacja w języku C może być podana w szczególnie prostej postaci:

```
x = 18000 * (x&65535) + (x >> 16); y = 30903 * (y&65535) + (y >> 16);
return ((x << 16) + (y&65535));
```

gdzie zmienne x oraz y oznaczają liczby 32-bitowe, zawierające w swych starszych i młodszych 16-bitowych częściach odpowiednio wartości X_n, c_1 oraz Y_n, c_2 , natomiast $\&$ jest standardowym operatorem języka C.

2.9. Generatory nieliniowe

Przedstawione w poprzednich podrozdziałach klasy generatorów są oparte na liniowych wzorach rekurencyjnych. Niepożądaną konsekwencją tej liniowości jest fakt, że odpowiednie punkty (2.3) i (2.4) w przestrzeni wielowymiarowej skupiają się tylko na pewnej liczbie hiperpłaszczyzn, co rażąco odbiega od naszych oczekiwań wobec punktów losowych (por. p. 2.2.3).

Na przewyższenie przeszkód naturalnym pomysłem wydaje się zatem rozważenie ciągów opartych na formułach, które nie są liniowe. Ten kierunek badań nad generatorami rozwija się dopiero od niedawna i nadal jest otwarty. Przedstawimy tutaj pewne wyniki, które dotyczą generatorów opartych na obliczaniu odwrotności oraz kwadratów.

W pracy Eichenauera i Lehna (1986) zaproponowano generator

$$X_{n+1} = (aX_n^{-1} + b) \bmod m, \quad n = 0, 1, \dots \quad (2.13)$$

gdzie m jest liczbą pierwszą. Odwrotność modulo m jest definiowana następująco: jeśli $c = 0$, to $c^{-1} \bmod m = 0$, w przeciwnym przypadku $c^{-1} \bmod m$ jest taką liczbą całkowitą, że $c \cdot c^{-1} \bmod m = 1$, czyli $c^{-1} = c^{m-2} \bmod m$. W ten sposób uzyskujemy ciąg o wartościach w zbiorze $\{0, 1, \dots, m-1\}$, który można w zwykły sposób przekształcić na ciąg liczb w przedziale $[0, 1)$ za pomocą wzoru $U_n = X_n/m$.

Innym wariantem generatora opartego na odwrotności jest zaproponowany w pracy Eichenauera-Hermannna (1993a) generator

$$X_n = (a(n + n_0) + b)^{-1} \bmod m, \quad n = 0, 1, \dots \quad (2.14)$$

który wyróżnia się tym, że kolejna wartość X_n może być uzyskana niezależnie od innych elementów produkowanego ciągu. Taki generator może być szczególnie użyteczny w obliczeniach prowadzonych na komputerach równoległych. Okazuje się, że dla każdej liczby $a \in \{1, 2, \dots, m\}$ jego okres jest równy m , a więc jest to generator o okresie maksymalnym. Warunek na to, aby generator (2.13) osiągał maksymalny okres równy m , jest nieco bardziej skomplikowany: tak się dzieje wtedy, gdy $m^2 - 1$ jest najmniejszą liczbą całkowitą taką, że $z^{m^2-1} - 1 \bmod (z^2 - bz - a)$.

Struktury przestrzenne tworzone przez odpowiednie ciągi w przestrzeniach wielowymiarowych dla omawianych generatorów są opisane przez Eichenauera-Hermannna (1991) oraz Niederreitera (1994)). Ponadto wiele eksperymentów obliczeniowych potwierdziło dobre własności statystyczne tej nowej klasy generatorów.

W pracach Eichenauera- Hermannna (1993b, 1994, 1995) analizuje się kombinacje kilku generatorów postaci (2.13) lub (2.14) (patrz również podrozdz. 2.5). Przypuśćmy, że mamy $r \geq 5$ takich generatorów

$$X_n^{(j)}, \quad j = 1, \dots, r$$

z parametrami $m_j, j = 1, \dots, r$, które są liczbami pierwszymi. Zdefiniujmy nowy ciąg

$$U_n = (U_n^{(1)} + \dots + U_n^{(r)}) \bmod 1, \quad n = 0, 1, 2, \dots$$

gdzie $U_n^{(j)} = X_n^{(j)}/m_j, j = 1, 2, \dots, r$. Okres tego ciągu jest równy $m = m_1 \cdot \dots \cdot m_r$. Pozwala to na efektywną konstrukcję generatora o długim okresie, przy czym - jak się okazuje - ten wynikowy generator zachowuje dobre własności strukturalne generatorów składowych.

Innym przykładem generatora nieliniowego jest rozważany w pracy Bluma (L.), Bluma (M.) i Shuba (1986) generator liczb losowych postaci

$$X_{n+1} = X_n^2 \bmod m, \quad n = 0, 1, 2, \dots$$

W cytowanej pracy pokazano zastosowania takiego generatora w kryptologii (jest to dziedzina badań zajmująca się metodami szyfrowania informacji). Bardziej szczegółowe informacje

dotyczące związków między teorią szyfrowania a generatorami liczb losowych, Czytelnik może znaleźć na przykład w książkach Tezuki (1995) oraz Schneiera (1995).

2.10. Uwagi o implementacji numerycznej

Jak już mówiliśmy (p. 2.2.1), generatory produkują ostatecznie liczby U_1, U_2, \dots z przedziału $(0, 1)$. W niektórych generatorach mogą pojawić się kłopoty, gdy w ciągu kolejno otrzymywanych liczb pojawi się 0 lub 1.

Te kłopoty mogą mieć charakter wewnętrzny w tym sensie, że po wyprodukowaniu zera generator w dalszym ciągu produkuje już tylko same zera, albo zewnętrzny w tym sensie, że niektóre działania na kolejnych liczbach losowych mogą okazać się niewykonalne (dzielenie przez zero lub liczbę bliską zero, logarytmowanie takiej liczby, itp.). Dodatkowa trudność polega na tym, że faktycznie liczby U_1, U_2, \dots są uzyskiwane z liczb całkowitych X_1, X_2, \dots za pomocą standardowej operacji dzielenia $U_n = X_n/m$ i wówczas mała liczba całkowita X_n może zamienić się w maszynowe zero. Zwracamy uwagę na ten rodzaj trudności, chociaż nie potrafimy podać tutaj żadnej uniwersalnej recepty na poradzenie sobie z nimi.

Inny problem jest związany z tym, że typowe generatory są oparte na arytmetyce reszt względem ustalonej dodatniej liczby całkowitej m . Przed obliczeniem takiej reszty musimy jednak wykonać pewne inne operacje, np. dodawanie lub mnożenie, na liczbach całkowitych ze zbioru $\{1, 2, \dots, m\}$, a otrzymany wynik może wyprowadzać i z reguły wyprowadza poza ten zbiór. Istnieją specjalne algorytmy pozwalające operować liczbami całkowitymi w taki sposób, żeby wyniki działań pośrednich nie przekraczały liczby m . Nie będziemy rozwijać dalej tego wątku; zainteresowanego Czytelnika odsyłamy do prac: L'Ecuyer (1990), L'Ecuyer i Cote (1991), Dwyer (1995), Knuth (1981).

2.11. Przykładowe implementacje w języku C

2.11.1. Inicjowanie generatorów

W konkretnych realizacjach algorytmów generujących liczby losowe, pojawia się problem wyboru odpowiednich wartości inicjujących generator. Jeżeli użytkownik chce zautomatyzować również ten początkowy etap obliczeń, może wykorzystać jakiś generator fizyczny np. zegar systemowy. W standardowym języku C wbudowany generator liniowy inicjuje się np. wywołując funkcję `srand((unsigned)time(NULL))`; liczba początkowa dla generatora jest wtedy równa liczbie sekund, które upłynęły od 1 stycznia 1970 roku. „Bardziej losowe” liczby startowe dla generatora można uzyskiwać za pomocą dostępnych w systemie parametrów związanych z czasem oraz datą. Anderson (1990) zaproponował następującą formułę dla liczby początkowej X_0 :

$$X_0 = r + 100(m - 1 + 12(d - 1 + 31(g + 24 * (min + 60 * s))))$$

gdzie r oznacza dwie ostatnie cyfry roku, m - miesiąc (od 1 do 12), d - dzień (od 1 do 31), g - godzinę (od 0 do 23), min - minutę (od 0 do 59) oraz s - sekundę (od 0 do 59). Pewną odmianą tej formuły jest podana w opracowaniu *Random numbers* (1991-1995) propozycja

$$X_0 = s + 60(min + 60(g + 24(d - 1 + 31(m - 1 + 12r))))$$

Dodatkowo, w celu zapewnienia nieparzystości liczby X_0 , zaleca się zamianę ostatniego bitu tej

liczby na 1.

Powyższe metody są z reguły stosowane do prostych generatorów liniowych, natomiast w przypadku innych klas generatorów, wymagających dużej liczby punktów startowych, zaleca się inicjowanie za pomocą prostszego generatora, np. liniowego, który zainicjowano opisanymi wyżej metodami.

2.11.2. Ogólny generator liniowy

Omówiony tu generator jest implementacją schematu liniowego (2.1). Okres takiego generatora jest rzędu 2^{96} . Ponadto generator ten spełnia znane testy statystyczne, m.in. baterię testów DIEHARD (Marsaglia (1995)). W proponowanej implementacji plik nosi nazwę **ecng.c**. Do inicjowania generatora służy procedura **init_ecng()**, która wykorzystuje trzy nieujemne liczby całkowite i, j, k podane przez użytkownika. Generator jest realizowany przez funkcję **ecng()**, która przed pierwszym użyciem wymaga inicjowania za pomocą procedury **init.ecng()**, a której wynikiem jest liczba losowa *typu double* (jest to przekształcona na typ double i przedział (0,1) liczba całkowita 32-bitowa).

Odpowiednia implementacja przyjmuje następującą postać:

```
/* PLIK: ecng.c */
#include <stdlib.h>
/* zmienne wykorzystywane przez generator */
static double a,b,c;
void init_ecng(int ia, int ib, int ic)
{
  a = ia; b=ib; c=ic;
}
double ecng()
{
  static int n;
  static double d,x;
  d = (a + b + c) * 8192;
  x =fmod(d,4294967291.0);
  a = b; b=c; c=x;
  if (x < (float)2147483648.) n=(int) x;
  else n = (int) (x - 4294967296.);
  return (n*2.3283064365e-10) ;
}
```

2.11.3. Generator Tauswortha z podrozdziału 2.3

Omawiany tu generator jest implementacją algorytmu opisanego w podrozdz. 2.3, a dokładniej jest to kombinacja za pomocą operatora *xor* trzech generatorów opartych na rejestrach przesuwanych. Okres generatora wynosi $(2^{28} - 1) (2^{29} - 1) (2^{31} - 1)$, czyli jest rzędu $3 \cdot 10^{26}$. Pewne własności teoretyczne takiej konstrukcji zostały podane w książce Tezuka (1995). Proponowany generator spełnia znane testy statystyczne. W prezentowanej implementacji plik nosi nazwę **tezuka.c**. Do inicjowania generatora służy procedura **init()**, która wykorzystuje trzy nielocalne nieujemne liczby całkowite podawane przez użytkownika: s_1, s_2, s_3 . Liczby te powinny spełniać nierówności: $s_1 < 2^{28}, s_2 < 2^{29}, s_3 < 2^{31}$. Generator realizowany przez funkcję **combT()**, która przed pierwszym użyciem wymaga zainicjowania za pomocą procedury **init()**, a której wynikiem jest liczba losowa typu double (dokładniej mówiąc precyzja wyniku wynosi 32 bity), przyjmuje postać:

```

/* PLIK: tezuka.c */
/* zmienne wykorzystywane przez generator */
static unsigned int s1, s2, s3;
void init(unsigned int i, unsigned int j, unsigned int k)
{
    unsigned int b;
    s1=i; s2=j; s3=k;
    b = ((s1 « 9) – s1) « 4;
    s1 = (s1 « 4) - (b » 28);
    b = ((s2 « 2) - s2) « 3;
    s2 = (s2 « 3) - (b » 29);
    b = ((s3 « 6) - s3) « 1;
    s3 = (s3 « 1) - (b » 31);
}
double combT()
{
    unsigned int    b;
    b = ((s1 << 9) - s1) << 4;
    s1 = (s1 << 13) - (b >> 19);
    b = ((s2 << 2) - s2) << 3;
    s2 = (s2 << 20) - (b >> 12);
    b = ((s3 << 6) - s3) << 1;
    s3 = (s3 << 17) - (b >> 15);
    return ((s1 - s2 - s3)*2.3283064365e-10);
}

```

2.11.4. Generator uniwersalny z podrozdziału 2.6

Generator uniwersalny opisany w podrozdz. 2.6 nie został dotychczas zakwestionowany przez żaden test statystyczny. Jego okres jest równy 2^{144} . Liczby produkowane przez ten generator są identyczne we wszystkich komputerach, w których liczby całkowite mają co najmniej 16 bitów i mantysa liczb zmiennopozycyjnych ma co najmniej 24 bity. W proponowanej implementacji plik nosi nazwę **uni.c**. Procedura **rstart** służy do inicjowania generatora. Danymi wejściowymi tej procedury są liczby całkowite i, j, k z przedziału $[1, 178]$ oraz liczba całkowita l z przedziału $[0, 168]$, przy czym liczby i, j, k nie powinny być jednocześnie równe 1. Wynikiem procesu inicjowania jest wypełniona tablica pomocnicza $uu[0], \dots, uu[96]$. Generator jest realizowany przez funkcję **uni()**, która przed pierwszym użyciem wymaga inicjowania za pomocą procedury **rstart**, a której wynikiem jest liczba losowa typu double. Zmiennymi nielokalnymi są $ip, jp, uu[97], cc, cd, cm$.

Odpowiedni program przyjmuje postać:

```

/* PLIK: uni.c */
/* zmienne wykorzystywane przez generator */
static double uu[97];
static int ip=97;
static int jp=33;
static double cc=362436. 0/16777216.0;
static double cd=7 654321. 0/16777216.0;
static double cm=16777213. 0/16777216.0;

void rstart(int i, int j, int k, int l)
{
    int ii, jj,m,wi,wj,wk,wl; double s, t;
    wi=i; wj=j; wk=k; wl=l;
    for(ii=0;ii<97;ii++)
    {
        s=0;t=0.5;
        for(jj=1;jj<=24;jj++)

```

```

    {
        m= ( ( wi*w j ) %179) *wk) %179 ;
        wi =wj; wj=wk; wk=m ;
        wl=(53*wl+1)%169;
        if ( (wl*m)%64>=32 ) s+=t;
        t*=0.5;
    }
    uu[ii]=s;
}
}
double uni(void)
{
    double pom;
    pom=uu[ip-1] -uu[jp-1] ;
    if (pom < 0.0) pom+=1;
    uu[ip-1]=pom;
    ip -- ;
    if (ip==0) ip=97;
    jp -- ;
    if (jp==0) jp=97;
    cc -=cd;
    if ( cc < 0.0) cc+=cm;
    pom -=cc;
    if (pom < 0.0) pom+=1;
    return(pom);
}

```

Jako test poprawności zaprogramowania generatora służy niżej podany program tuni.c. W wyniku powinniśmy otrzymać tablicę:

6	3	11	3	0	4	0
13	8	15	11	11	14	0
6	15	0	2	3	11	0
5	14	2	14	4	8	0
7	15	7	10	12	2	0

```

/* PLIK: tuni.c. */
/* Test uniwersalnego generatora liczb losowych uni(). */
#include <stdio.h>
#include <math.h>
#include "uni.c"
void main(void)
{
    double x;
    int i,j,ii;
    rstart(12,34,56,78); /* inicjowanie generatora */
    for (ii=1;ii<=20005;ii++)
    {
        x=uni();
        if (ii>20000)
        {
            for(i=1;i<=7;i++)
                printf("%5.0f",fmod(floor(pow(16,i)*x),16));
            printf("\n");
        }
    }
}

```

3. Generatory liczb losowych o dowolnych rozkładach prawdopodobieństwa

3.1. Ogólne metody konstrukcji generatorów liczb losowych o dowolnych rozkładach prawdopodobieństwa

3.1.1. Metoda odwracania dystrybuanty

Załóżmy, że U jest zmienną losową o rozkładzie równomiernym $C/(0, 1)$. Niech F będzie ciągłą i ściśle rosnącą dystrybuantą pewnego rozkładu prawdopodobieństwa. Zdefiniujmy nową zmienną losową

$$X = X^I(U) \tag{3.1}$$

Ponieważ

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

więc zmienna losowa X ma rozkład prawdopodobieństwa o dystrybuancie F . Jeżeli zatem wygenerujemy ciąg liczb losowych U_1, \dots, U_n o rozkładzie równomiernym $U(0, 1)$ i na tej podstawie wyznaczymy nowy ciąg $X_i = F^{-1}(U_i)$, $i = 1, 2, \dots, n$, to ten nowy ciąg X_1, \dots, X_n będzie ciągiem liczb losowych o rozkładzie z dystrybuantą F .

Opisane postępowanie można uogólnić na przypadek dowolnych, niekoniecznie ciągłych i ściśle rosnących dystrybuant F . W tym celu wystarczy zdefiniować

$$F^{-1}(t) = \inf\{x: t \leq F(x)\} \tag{3.2}$$

i postępować w sposób wyżej opisany. W szczególności liczby losowe X_1, X_2, X_3, \dots o rozkładzie dyskretnym $p_k = P\{X = k\}$, $k = 0, 1, 2, \dots$, mogą być generowane przez przekształcenie liczb losowych U_1, U_2, \dots o rozkładzie równomiernym $U(0, 1)$ za pomocą wzoru

$$X_n = \min\left\{k : U_n \leq \sum_{i=0}^k p_i\right\}, \quad n = 1, 2, \dots \tag{3.3}$$

Twierdzenie 3.1. Niech F będzie dystrybuantą pewnego rozkładu prawdopodobieństwa. Jeżeli U jest zmienną losową o rozkładzie równomiernym $U(0,1)$, to zmienna losowa $X = \inf\{x: U \leq F(x)\}$ ma rozkład o dystrybuancie F .

Dowód. Udowodnimy, że zdarzenie losowe $\{X \leq t\}$ zachodzi wtedy i tylko wtedy, gdy zachodzi zdarzenie $\{U \leq F(t)\}$.

Przypuśćmy, że zachodzi zdarzenie $\{X \leq t\}$.

Niech $X = t$. Z definicji zmiennej losowej X (infimum) wynika, że dla każdego $m = 1, 2, \dots$ istnieje punkt x_m spełniający warunki $t \leq x_m < t + 1/m$ oraz $U \leq F(x_m)$. Gdy $m \rightarrow +\infty$, wtedy $t \leq x_m \rightarrow t$ (z prawej strony), a stąd, wobec prawostronnej ciągłości dystrybuanty, otrzymujemy nierówność $U \leq F(t)$.

Niech $X < t$. W zbiorze $\{x: U \leq F(x)\}$ istnieje zatem punkt $y < t$. Wtedy oczywiście $F(y) \leq F(t)$ oraz, z racji przynależności punktu y do zbioru $\{x: U \leq F(x)\}$, mamy $U \leq F(y)$.

Czyli $U \leq F(t)$.

Przypuśćmy, że $U \leq F(t)$; zatem t należy do zbioru $\{x: U \leq F(x)\}$, czyli $t \geq \inf\{x: t \leq F(x)\} = X$, co kończy dowód twierdzenia.

Być może najbardziej spektakularnym efektem zastosowania metody odwracania dystrybuanty jest generator liczb losowych o rozkładzie wykładniczym z gęstością $f(x) = e^{-x}$, $x \geq 0$: jeżeli U_1, U_2, \dots jest ciągiem liczb losowych o rozkładzie równomiernym $U(0,1)$, to X_1, X_2, \dots , gdzie $X_n = -\ln U_n$, jest ciągiem liczb losowych o rozkładzie wykładniczym z gęstością $f(x)$, a w przypadku gdy $X_n = \theta - \lambda \ln U_n$ - ciągiem liczb losowych o rozkładzie wykładniczym $E(\theta, \lambda)$ z gęstością $(1/\lambda)\exp(-(x - \theta)/\lambda)$, $x \geq \theta$.

Odwracanie dystrybuanty jest jednak zwykle zabiegiem skomplikowanym numerycznie i dlatego opisaną wyżej metodę rzadko się stosuje. Na przykład, dla funkcji odwrotnej do dystrybuanty rozkładu normalnego $N(0,1)$ podano (patrz Odeh i Evans (1974)) następujące przybliżenie w praktycznie wystarczającym przedziale:

$$\Phi^{-1}(u) = \begin{cases} g(u), & \text{gdy } 10^{-20} < u < 0.5 \\ -g(1-u), & \text{gdy } 0.5 \leq u < 1 - 10^{-20} \end{cases}$$

gdzie:

$$g(u) = t - \frac{L(t)}{M(t)}, \quad t = \sqrt{-2 \ln u}$$

3.1. Metody ogólne

$$L(t) = 0.322232431088 + t + 0.342242088547t^2 + \\ + 0.0204231210245t^3 + 0.0000453642210148t^4,$$

$$M(t) = 0.0993484626060 + 0.588581570495t + 0.531103462366t^2 + \\ + 0.103537752850t^3 + 0.0038560700634t^4$$

Metodą odwracania dystrybuanty można oczywiście generować wszelkie zmienne losowe, chociaż czasami może to być bardzo uciążliwe numerycznie.

Przykład 1. Jeżeli zmienna losowa U ma rozkład równomierny $U(0, 1)$, to zmienna losowa $X = bU^{1/a}$ ma rozkład Pareto o dystrybuancie $F_{a,b}(x) = 1 - (b/x)^a$, $x \geq b$. •

Przykład 2. Jeżeli zmienna losowa U ma rozkład równomierny $U(0,1)$, to zmienna losowa $X = \ln(U/(1-U))$ ma rozkład logistyczny o dystrybuancie $F(x) = 1/(1 + e^{-x})$. •

Przykład 3. Jeżeli zmienna losowa U ma rozkład równomierny $U(0,1)$, to zmienna losowa $X = F^{-1}(F(a) + (F(b) - F(a))U)$ ma rozkład o dystrybuancie F uciętej do przedziału $[a, b]$. •

Przykład 4. Jeżeli $U_{k:n}$ jest k -tą statystyką pozycyjną z ciągu U_1, \dots, U_n niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym $U(0,1)$, to $F^{-1}(U_{k:n})$ ma taki sam rozkład jak k -ta statystyka pozycyjna z ciągu X_1, \dots, X_n niezależnych zmiennych losowych o jednakowym rozkładzie z dystrybuantą F . •

Podstawowa zaleta metody odwracania dystrybuanty polega na tym, że do wygenerowania zmiennej losowej o danym rozkładzie potrzebna jest tylko jedna zmienna losowa o rozkładzie równomiernym $U(0,1)$. Fakt, że wyróżniamy to jako zaletę metody, jest związany z niedoskonałością generatorów liczb losowych o rozkładzie równomiernym, polegającą na tym, że kolejne liczby z takiego generatora mogą nie być niezależne, a więc użycie kilku kolejnych liczb $U(0,1)$ może deformować wynikowy rozkład prawdopodobieństwa. Obserwuje się to szczególnie wtedy, gdy używa się standardowych procedur typu *Random* wmontowanych w kompilatory typowych języków programowania lub gdy niektóre rozpowszechnione pakiety programów są używane bezkrytycznie.

3.1.2. Metoda eliminacji

Metodę eliminacji zaproponował John von Neumann (von Neumann (1951)). Najpierw przedstawimy tę metodę dla najprostszego przypadku, gdy gęstość prawdopodobieństwa $f(x)$ interesującego nas rozkładu jest dodatnia na pewnym ograniczonym przedziale (a, b) , ograniczona przez pewną stałą $d > 0$ i równa zero poza tym przedziałem. Następnie podamy ogólną postać tej metody, jej zastosowanie do bardzo ważnego w praktyce generatora liczb losowych z ogona rozkładu normalnego oraz zaprezentujemy metodę eliminacji dla przypadku, gdy gęstość rozkładu prawdopodobieństwa generowanej zmiennej losowej można przedstawić w pewnej szczególnej postaci. Powiemy także o pewnych możliwościach przyspieszania algorytmów konstruowanych tą metodą.

Wariant podstawowy metody eliminacji

Niech f będzie gęstością prawdopodobieństwa interesującego nas rozkładu, dodatnią na pewnym ograniczonym przedziale (a, b) , równą zero poza tym przedziałem i ograniczoną przez pewną stałą $d > 0$. Następujący algorytm generuje liczby losowe o rozkładzie z taką gęstością $f(x)$:

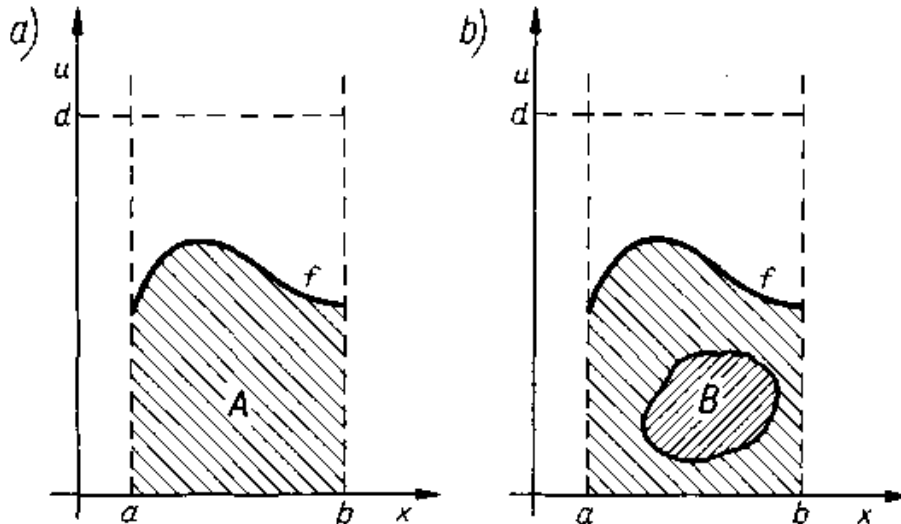
1. Wygenerować dwie niezależne zmienne losowe U_1 i U_2 o rozkładach równomiernych $U(a, b)$ i $U(0, d)$.
2. Jeżeli $U_2 \leq f(U_1)$, to przyjąć $X = U_1$; w przeciwnym przypadku parę (U_1, U_2) wyeliminować i powtórzyć obliczenia od wygenerowania nowej pary zgodnie z punktem 1.

Otrzymana w ten sposób zmienna losowa X ma rozkład o gęstości f ; mówią o tym dwa podane niżej twierdzenia. Zanim je sformułujemy, wprowadzimy niezbędne oznaczenia.

Jeżeli C jest danym zbiorem na płaszczyźnie, to przez $l_2(C)$ oznaczamy pole powierzchni (miarę Lebesgue'a na płaszczyźnie) tego zbioru. Przez A oznaczamy zbiór (rys. 3.1a)

$$A = \{(x, u): a \leq x \leq b, 0 \leq u \leq f(x)\}$$

Niech $(a, b) \times (0, d)$ oznacza prostokąt o wysokości $d > 0$, zbudowany na odcinku (a, b) . Mamy oczywiście $l_2((a, b) \times (0, d)) = (b - a)d$, a ponieważ f jest gęstością pewnego rozkładu prawdopodobieństwa, więc $l_2(A) = 1$.



Rys. 3.1

Mówimy, że punkt losowy ω ma rozkład równomierny na zbiorze $D \subset \mathbb{R}^2$, jeżeli dla podzbiorów C tego zbioru zachodzi równość $P\{\omega \in C\} = l_2(C)/l_2(D)$

TWIERDZENIE 3.2. Niech $(X_1, U_1), (X_2, U_2), \dots$ będzie ciągiem punktów losowych o rozkładzie równomiernym na prostokącie $(a, b) \times (0, d)$ i niech (X, U) będzie pierwszym punktem tego ciągu, który wpada do zbioru A . Wtedy punkt losowy (X, U) ma rozkład równomierny na zbiorze A .

Dowód. Rozważmy podzbiór B zbioru A (rys. 3.1b) i niech $l_2(B)$ będzie polem jego powierzchni. Mamy udowodnić, że $P\{(X, U) \in B\} = l_2(B)/l_2(A)$. Wynika to łatwo z następujących rachunków:

$$\begin{aligned} P\{(X, U) \in B\} &= \sum_{i=1}^{\infty} P\{(X_1, U_1) \notin A, \dots, (X_{i-1}, U_{i-1}) \notin A, (X_i, U_i) \in B\} = \\ &= \sum_{i=1}^{\infty} \left(1 - \frac{l_2(A)}{(b-a)d} \right)^{i-1} \frac{l_2(B)}{(b-a)d} = \frac{l_2(B)}{l_2(A)} \end{aligned}$$

TWIERDZENIE 3.3. (a) Jeżeli U ma rozkład równomierny $U(0, 1)$, X ma rozkład o gęstości $f(x)$ oraz X i U są niezależne, to punkt losowy $(X, Uf(X))$ ma rozkład równomierny na zbiorze A .

(b) Jeżeli punkt losowy (X, U) ma rozkład równomierny na zbiorze A , to zmienna losowa X ma rozkład o gęstości $f(x)$.

D o w ó d. (a) Jeżeli U ma rozkład równomierny $U(0, 1)$, to dla każdego ustalonego x zmienna

losowa $V = U f(x)$ ma rozkład równomierny $U(0, f(x))$. Dla ustalonego x i dla danego zbioru $B \subset C$ oznaczamy $B_x = \{u: (x, u) \in B\}$.

Wtedy

$$P\{(X, Uf(X)) \in B\} = \int \left(\int_{B_x} \frac{dv}{f(x)} \right) f(x) dx = \iint_B dv dx = l_2(B) = \frac{l_2(B)}{l_2(A)}$$

czyli $(X, Uf(X))$ ma rozkład równomierny na zbiorze A .

b) Oznaczamy $A_t = \{(x, u): a \leq x \leq t, 0 \leq u \leq f(x)\}$. Wtedy

$$P\{X \leq t\} = P\{(X, U) \in A_t\} = \int_a^t \int_0^{f(x)} \frac{dudx}{l_2(A)} = \int_a^t f(x) dx$$

czyli $f(x)$ jest gęstością rozkładu prawdopodobieństwa zmiennej losowej X .

Wariant ogólny metody eliminacji

W rozważanej najprostszej wersji algorytmu metody eliminacji wyjściowy punkt losowy (U_1, U_2) miał dwuwymiarowy rozkład równomierny na prostokącie o podstawie (a, b) i wysokości d . Wykres funkcji gęstości całkowicie mieścił się w tym prostokącie. Punkt (U_1, U_2) był akceptowany, jeżeli wpadł pod wykres funkcji gęstości lub eliminowany w przeciwnym przypadku.

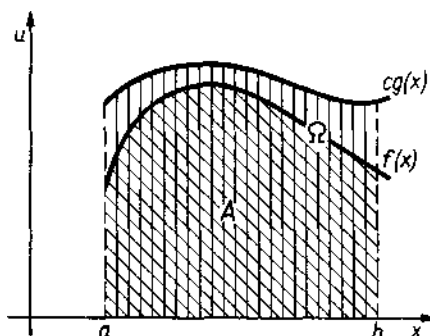
Najogólniejszy wariant metody eliminacji, w którym generowany element losowy X może być wielowymiarową zmienną losową (wektorem losowym), a obszarem zawierającym wykres gęstości może być dowolny obszar w przestrzeni o odpowiedniej liczbie wymiarów, jest teoretycznie uzasadniony przez dwa następujące twierdzenia.

Twierdzenie 3.2 A. Przypuśćmy, że ξ_1, ξ_2, \dots jest ciągiem niezależnych elementów losowych o jednakowym rozkładzie w R^k . Niech A będzie zbiorem w R^k , takim że $P\{\xi_1 \in A\} > 0$ oraz niech η oznacza element losowy równy pierwszemu elementowi ξ_n przyjmującemu wartość w zbiorze A . Wtedy dla $B \subset R^k$ zachodzi równość

$$P\{\eta \in B\} = \frac{1}{2} P\{\xi_1 \in A \cap B\}$$

gdzie $p = P\{\xi_1 \in A\}$. W szczególności, jeżeli ξ_1, ξ_2, \dots są niezależnymi punktami losowymi o rozkładzie równomiernym na pewnym zbiorze $\Omega \supset A$, to η ma rozkład równomierny na zbiorze A .
D

Twierdzenie 3.3A. (a) Jeżeli X jest punktem losowym o rozkładzie z gęstością g w k -wymiarowej przestrzeni R^k , U jest zmienną losową o rozkładzie równomiernym $U(0, 1)$ i X oraz U są niezależne, to punkt losowy $(X, Cug(X))$ ma rozkład równomierny na zbiorze $\Omega = \{(x, u): x \in R^k, 0 \leq u \leq cg(x)\} \subset R^{k+1}$, gdzie $c > 0$ jest dowolną stałą, (b) Odwrotnie: jeżeli punkt losowy (X, U) ma rozkład równomierny na zbiorze Ω , to X ma rozkład o gęstości g .



Rys.3.2.

Dowody tych twierdzeń są łatwymi modyfikacjami dowodów twierdzeń 3.2 i 3.3. Prowadzą one do następującej konstrukcji generatora liczb losowych o rozkładzie z gęstością f , określoną na niekoniecznie ograniczonym zbiorze w przestrzeni k -wymiarowej R^k dla $k > 1$ (rys. 3.2).

1. Wybrać taką gęstość prawdopodobieństwa g , żeby generowanie liczb losowych o tej gęstości było łatwe i szybkie oraz wyznaczyć stałą $c > 0$, taką żeby

$$f(x) \leq cg(x) \quad \text{dla wszystkich } x \quad (3.4)$$

Ze względu na ten warunek gęstość $g(x)$ będziemy nazywali gęstością dominującą. Za obszar Ω w twierdzeniu 3.2A przyjąć

$$\Omega = \{(x, u): x \in R^k, 0 \leq u \leq cg(x)\}$$

2. Wygenerować punkt losowy X o rozkładzie z gęstością g oraz liczbę losową U o rozkładzie równomiernym $U(0,1)$. Wtedy, na mocy pierwszej tezy tw. 3.3A, punkt losowy $(X, cUg(X)) \in R^{k+1}$ ma rozkład równomierny na zbiorze Ω .

3. Powtarzać generowanie według p. 2 dopóty, dopóki kolejno wygenerowany punkt nie wpadnie do zbioru $A = \{(x, u): x \in R^k, 0 \leq u \leq f(x)\}$, tzn. dopóki nie zostanie spełniony warunek akceptacji

$$cUg(X) \leq f(X) \quad (3.5)$$

Na mocy tw. 3.2A tak uzyskany punkt ma rozkład równomierny na zbiorze A , a więc na mocy drugiej tezy tw. 3.3A X ma rozkład o gęstości $f(x)$.

Podstawowy algorytm metody eliminacji ma, więc następującą postać:

ALGORYTM 3.1

Repeat

Generuj X o rozkładzie z gęstością g

Generuj U o rozkładzie równomiernym $U(0,1)$

until

$$cUg(X) \leq f(X)$$

Return X

Zauważmy, że do tej pory nie powiedzieliśmy dokładnie, jak wyznacza się stałą c w opisanym algorytmie; sformułowaliśmy dla niej tylko jeden warunek: (3.4). Jest oczywiste, że jeżeli pewna liczba c_1 spełnia warunek (3.4), to również spełnia go każda liczba $c_1 > c_2$. Wyznaczenie „dobrej” stałej c opiera się na następującym rozumowaniu: pewnym elementem algorytmu jest powtarzanie losowania punktów (X, U) dopóty, dopóki nie zostanie spełniony warunek (3.5), powinniśmy więc postarać się o to, żeby został on spełniony jak najszybciej. Możemy to osiągnąć wybierając taką stałą c , aby prawdopodobieństwo spełnienia warunku (3.5) było możliwie jak największe. Ale

$$P\{Ucg(X) \leq f(X)\} = \int_{R^k} g(x) dx \int_0^{f(x)/cg(x)} du = \frac{1}{c}$$

więc optymalna stała c jest najmniejszą stałą spełniającą zależność (3.4). Jest nią oczywiście

$$c = \sup \frac{f(x)}{g(x)} \quad (3.6)$$

Przykład 1 (generowanie rozkładu normalnego za pomocą rozkładu wykładniczego). Generujemy zmienną losową o rozkładzie normalnym $N(0, 1)$. W tym celu generujemy najpierw X o gęstości $f(x) = \sqrt{2/\pi} \exp(-x^2/2)$ (dodatnia połówka rozkładu normalnego), a później wyposażamy X w znak $+$ lub $-$, każdy z prawdopodobieństwem $1/2$. Za gęstość dominującą przyjmijmy gęstość $g(x) = e^{-x}$, $x > 0$, rozkładu wykładniczego. Jest to rzeczywiście łatwy rozkład, bo jeżeli U ma rozkład równomierny $U(0,1)$, to $-\ln U$ ma rozkład o gęstości g (patrz p. 3.1.1). Otrzymujemy stałą $c = \sqrt{2e/\pi}$ i algorytm przyjmuje postać

ALGORYTM 3.2

Repeat

Generuj X o rozkładzie wykładniczym $E(0,1)$

Generuj U o rozkładzie równomiernym $U(0,1)$

until

$$\sqrt{\frac{2e}{\pi}} U e^{-x} \leq \sqrt{\frac{2}{\pi}} e^{-x^2/2}$$

Return X

Warunek zakończenia pętli powtórzeń jest oczywiście równoważny z warunkiem

$$(X - 1)^2 \leq -2 \ln U$$

W celu zakończenia generowania zmiennej losowej o rozkładzie $N(0,1)$ wystarczy teraz wyprodukować kolejne U o rozkładzie $U(0,1)$ i zamienić X na $-X$, gdy np. $U \leq 0.5$. Można ewentualnie skorzystać z generatora liczb losowych V o rozkładzie równomiernym $U(-1,1)$ i zakończyć obliczenia działaniem $X = X \text{sign} V$. Dalsze uproszczenie, polegające na tym, że generuje się tylko dwie zmienne losowe V i X zamiast trzech U , V i X , opiera się na następującym lemacie:

LEMAT 3.1. Jeżeli zmienna losowa W ma rozkład równomierny $U(-1,1)$, to zmienna losowa $|W|$ ma rozkład równomierny $U(0,1)$, a dwupunktowa zmienna losowa $\text{sign} W$ ma rozkład równomierny na zbiorze $\{-1,1\}$ i te dwie zmienne losowe są niezależne.

Łatwy dowód tego lematu, jak również odpowiednią modyfikację algorytmu, pozostawiamy Czytelnikowi. •

Przykład zastosowania: ogon rozkładu normalnego

Jako przykład praktycznego zastosowania metody eliminacji, a zarazem przykład różnych trików stosowanych w programowaniu generatorów liczb losowych o zadanych rozkładach prawdopodobieństwa, omówimy zagadnienie generowania zmiennej losowej z prawego ogona rozkładu normalnego $N(0, 1)$, tzn. zmiennej losowej X o rozkładzie z gęstością

$$f_t(x) = \sqrt{\frac{2}{\pi}} \frac{e^{-x^2/2}}{2(1 - \Phi(t))}, \quad x \geq t \quad (3.7)$$

Przedstawimy trzy algorytmy generowania tej zmiennej.

Przykład 2 (ogon rozkładu normalnego I). Generator liczb losowych X o rozkładzie z gęstością (3.7) można skonstruować metodą eliminacji, korzystając z oczywistej nierówności

$$e^{-x^2/2} \leq \frac{x}{t} e^{-x^2/2}, \quad x \geq t$$

Prawą stronę tej nierówności można przedstawić w postaci $c(t) * g_t(x)$, gdzie $c(t)$ jest stałą (zależną od parametru obcięcia t) oraz $g_t(x)$ jest gęstością pewnego rozkładu prawdopodobieństwa. Żeby z funkcji $x \exp(-x^2/2)$ zrobić gęstość prawdopodobieństwa na przedziale $(t, +\infty)$, obliczamy stałą normującą

$$\int_t^{+\infty} x e^{-x^2/2} dx = e^{-t^2/2}$$

i otrzymujemy

$$g_t(x) = x \exp\left(\frac{1}{2}(t^2 - x^2)\right) 1_{(t, +\infty)}(x)$$

gdzie $1_A(x)$, jak zwykle, oznacza funkcję wskaźnikową zbioru A , tzn. funkcję przyjmującą wartość 1 na zbiorze A oraz 0 poza nim.

Ograniczenie z góry gęstości (3.7) przyjmuje postać

$$f_t(x) = \sqrt{\frac{2}{\pi}} \frac{e^{-x^2/2}}{2(1 - \Phi(t))} \leq c(t) g_t(x)$$

gdzie:

$$c(t) = \frac{\varphi(t)}{t(1 - \Phi(t))}$$

W standardowym algorytmie, konstruowanym metodą eliminacji, generujemy X o rozkładzie z gęstością $g_t(x)$ oraz U o rozkładzie równomiernym $U(0, 1)$ i sprawdzamy warunek akceptacji

$$c(t) U g_t(x) \leq f_t(x)$$

który po prostych przekształceniach przyjmuje postać

$$UX \leq t$$

Jeżeli X ma rozkład prawdopodobieństwa o gęstości $g_t(x)$, to

$$P\{X \leq x\} = \int_t^x g_t(v)dv = 1 - \exp\left(\frac{1}{2}(t^2 - x^2)\right)$$

więc zmienną losową X możemy generować metodą odwracania dystrybuanty: $X = \sqrt{t^2 - 2 \ln V}$, gdzie V jest zmienną losową o rozkładzie równomiernym $U(0, 1)$. Ta operacja wymaga obliczania logarytmu naturalnego i pierwiastka kwadratowego w głównej pętli algorytmu. Obliczanie logarytmu można ewentualnie zastąpić generatorem zmiennej losowej o rozkładzie wykładniczym, natomiast obliczanie pierwiastka da się wyprowadzić poza główną pętlę algorytmu, gdzie będzie on obliczany tylko jeden raz dla każdej generowanej liczby losowej zamiast wielokrotnie, przy okazji każdego kandydata na tę liczbę. Można to wykonać w następujący sposób. Warunek $\{UX \leq t\}$ jest tutaj równoważny z warunkiem $\{U^2 X^2 \leq t^2\}$. Ale $X^2 = t^2 - 2 \ln V = 2(t^2/2 - \ln V)$. Wprowadzając stałą $r = t^2/2$ i zmienną losową Y o rozkładzie wykładniczym $E(r, 1)$, otrzymujemy algorytm

ALGORYTM 3.3

Repeat

 Generuj Y o rozkładzie wykładniczym $E(r, 1)$

 Generuj U o rozkładzie równomiernym $U(0, 1)$

until

$$2U^2Y \leq r$$

Return $X = \sqrt{2Y}$

Przykład 3 (ogon rozkładu normalnego II). Weźmy pod uwagę następującą nierówność:

$$\exp(-x^2/2) \leq \exp(t^2/2 - tx), \quad x \geq t$$

Gęstość dominująca wyraża się wzorem

$$g_t(x) = \frac{\exp(t^2/2 - tx)1_{(t,+\infty)}(x)}{\int_t^{+\infty} \exp(t^2/2 - ty)dy} = t \exp(t^2 - tx)1_{(t,+\infty)}(x)$$

Algorytm przyjmuje postać: generuj X o rozkładzie z gęstością $g_t(x)$ oraz U o rozkładzie równomiernym $U(0, 1)$ dopóty, dopóki nie zostanie spełniony warunek

$$U \exp(t^2/2 - tx) \leq \exp(-x^2/2)$$

czyli warunek

$$(X - t)^2 \leq -2 \ln U$$

Pozostaje wybór metody generowania zmiennej losowej X o rozkładzie z gęstością $g_t(x)$.

LEMAT 3.2. Jeżeli zmienna losowa Y ma rozkład wykładniczy $E(0,1)$, to zmienna losowa $X = t + Y/t$ ma rozkład o gęstości $g_t(x)$.

Po podstawieniu w warunku akceptacji Y/t zamiast $X - t$ oraz zmiennej losowej W o rozkładzie wykładniczym $E(0,1)$ zamiast $-\ln U$, otrzymujemy

ALGORYTM 3.4

Repeat

Generuj Y o rozkładzie wykładniczym $E(0,1)$

Generuj W o rozkładzie wykładniczym $E(0,1)$

until

$Y^2 \leq 2t^2W$

Return $X = t + Y/t$

W przykładach 2 i 3 mamy stałą $c(t) = \varphi(t) / (t(1 - \Phi(t)))$, czyli efektywność $\text{eff}(t) = 1/c(t)$ mierzona prawdopodobieństwem akceptacji, jest równa odpowiednio: $\text{eff}(1) = 0.6556$, $\text{eff}(2) = 0.8427$, $\text{eff}(3) = 0.9138$, $\text{eff}(4) = 0.9470$, itd.

Algorytm przedstawiony w następnym przykładzie jest bardziej efektywny, ale wymaga nieco dłuższego czasu przygotowawczego, potrzebnego do obliczenia stałych algorytmu.

Przykład 4 (ogon rozkładu normalnego III). Tak samo jak w przykładach 2 i 3, generujemy zmienną losową X o rozkładzie z gęstością (3.7), ale za gęstość dominującą przyjmujemy teraz gęstość przesuniętego rozkładu wykładniczego

$$g_{t,\lambda}(x) = \lambda e^{-\lambda(x-t)}, \quad x \geq t$$

Dystrybuantą tego rozkładu jest $G_{t,\lambda}(x) = 1 - \exp(-\lambda(x-t))$. Generowanie zmiennej losowej X o takim rozkładzie można wykonać metodą odwracania dystrybuanty za pomocą wzoru $X = t - (1/\lambda) \ln(1-U)$ gdzie U jest zmienną losową o rozkładzie równomiernym $U(0,1)$ lub wzoru $X = t - Y_1/\lambda$, gdzie Y_1 jest zmienną losową o rozkładzie wykładniczym $E(0,1)$. Zgodnie z ogólną zasadą metody eliminacji będziemy teraz generować niezależne zmienne losowe U i V o rozkładzie równomiernym $U(0,1)$ i obliczać $X = t - (1/\lambda) \ln U$ dopóty, dopóki nie zostanie spełniony warunek $cVg_{t,\lambda}(X) \leq f_t(X)$ przy odpowiednio wybranej stałej c . Dla ustalonych wartości parametrów t oraz λ optymalna stała $c = c(t, \lambda)$ wynosi

$$c(\lambda, t) = \sup_{x \geq t} \frac{f_t(x)}{g_{t,\lambda}(x)} = \sup_{x \geq t} \frac{\sqrt{2/\pi}}{2(1-\Phi(t))\lambda} \exp\left(\lambda(x-t)^2 - \frac{1}{2}x^2\right)$$

Maksimum po prawej stronie tego wzoru jest osiągnięte dla takiej wartości $x \geq t$, która maksymalizuje wielkość $\lambda(x-t)^2 - \frac{1}{2}x^2$, czyli dla $x = \lambda$, gdy $\lambda \geq t$ lub dla $x = t$ w przeciwnym przypadku. Otrzymujemy stąd optymalną wartość stałej $c(\lambda, t)$, wyrażającą się wzorem

$$c(\lambda, t) = \begin{cases} \frac{\sqrt{2/\pi}}{2\lambda(1-\Phi(t))} \exp\left(\frac{\lambda^2}{2} - \lambda t\right), & \text{gdy } \lambda \geq t \\ \frac{\sqrt{2/\pi}}{2\lambda(1-\Phi(t))} \exp(-t^2/2), & \text{gdy } \lambda < t \end{cases}$$

Parametr λ jest do naszej dyspozycji, możemy zatem przypisać mu taką wartość, żeby przy danym t zminimalizować $c(\lambda, t)$. Tą wartością jest

$$\lambda_{opt} = \lambda(t) = \frac{1}{2} \left(\sqrt{t^2 + 4} + t \right) \quad (3.8)$$

i stąd otrzymujemy optymalną wartość $c_{opt}=c(t)$:

$$c(t) = \frac{\sqrt{2/\pi}}{(\sqrt{t^2 + 4 + t})(1 - \Phi(t))} \exp\left(\frac{(\sqrt{t^2 + 4} - 3t)(\sqrt{t^2 + 4 + t})}{8}\right) \quad (3.9)$$

Po podstawieniu w warunku akceptacji optymalnych wartości parametrów λ i c , po prostych przekształceniach warunek ten przyjmuje postać

$$V \leq \exp\left(-\frac{1}{2}(X - \lambda)^2\right)$$

skąd ostatecznie otrzymujemy algorytm

ALGORYTM 3.5

Repeat

Generuj Y_1 o rozkładzie wykładniczym $E(0, 1)$

Generuj Y_2 o rozkładzie wykładniczym $E(0, 1)$

$$X = t + Y_1 / \lambda$$

until

$$\frac{1}{2}(X - \lambda)^2 \leq Y_2$$

Return X

Efektywność $\text{eff}(t) = 1/c(t)$, na mocy wzoru (3.9), wynosi $\text{eff}(1) = 0.8765$, $\text{eff}(2) = 0.9336$, $\text{eff}(3) = 0.9609$, $\text{eff}(4) = 0.9749$, itd. .

Metoda eliminacji dla gęstości $f(x) = cp(x)q(x)$

Podamy kilka wersji metody eliminacji, związanych ze szczególnymi postaciami gęstości.

1. Jeżeli gęstość f można przedstawić w postaci $f(x) = cp(x)q(x)$, gdzie $p(x)$ jest także gęstością pewnego rozkładu, a funkcja $q(x)$ przyjmuje wartości tylko w przedziale $[0,1]$, to algorytm przybiera postać

ALGORYTM 3.6

Repeat

Generuj X o rozkładzie z gęstością p

Generuj U o rozkładzie równomiernym $U(0,1)$

until

$$U \leq q(X)$$

Return X

2. Jeżeli gęstość f można przedstawić w postaci $f(x) = cp(x)q(x)$, gdzie $p(x)$ oraz $q(x)$ są, odpowiednio, gęstością i dystrybuantą pewnych rozkładów prawdopodobieństwa, to algorytm przyjmuje postać

ALGORYTM 3.7

Repeat

Generuj X o rozkładzie z gęstością p

Generuj Y o rozkładzie z dystrybuantą q

until

$Y \leq X$

Return X

3. Jeżeli gęstość f można przedstawić w postaci $f(x) = cp(x)q(t(x))$, gdzie $p(x)$ oraz $q(x)$ są, odpowiednio, gęstością i dystrybuantą pewnych rozkładów prawdopodobieństwa, to algorytm przyjmuje postać

ALGORYTM 3.8

Repeat

Generuj X o rozkładzie z gęstością p

Generuj Y o rozkładzie z dystrybuantą q

until

$Y \leq t(X)$

Return X

4. Jeżeli gęstość f można przedstawić w postaci $f(x) = cp(x)(1 - q(x))$, gdzie $p(x)$ oraz $q(x)$ są, odpowiednio, gęstością i dystrybuantą pewnych rozkładów prawdopodobieństwa, to algorytm przyjmuje postać

ALGORYTM 3.9

Repeat

Generuj X o rozkładzie z gęstością p

Generuj Y o rozkładzie z dystrybuantą q

until

$Y > X$

Return X

5. Jeżeli gęstość f można przedstawić w postaci $f(x) = cp(x)(1 - q(t(x)))$, gdzie $p(x)$ oraz $q(x)$ są, odpowiednio, gęstością i dystrybuantą pewnych rozkładów prawdopodobieństwa, to algorytm przyjmuje postać

ALGORYTM 3.10

Repeat Repeat

Generuj X o rozkładzie z gęstością p

Generuj Y o rozkładzie z dystrybuantą q

until

$Y > t(X)$

Return X

Przykład 5 (dodatnia połówka rozkładu normalnego $N(0,1)$). Dla dodatniej połówki rozkładu

normalnego $N(0,1)$ mamy $f(x) = \sqrt{2/\pi} \exp(-x^2/2)$, $x > 0$, co można zapisać w postaci

$$f(x) = \sqrt{\frac{2e}{\pi}} e^{-x} \left(1 - (1 - e^{-(x-1)^2/2})\right)$$

Algorytm przyjmuje zatem następującą postać:

ALGORYTM 3.11

Repeat

Generuj X o rozkładzie wykładniczym $E(0,1)$

Generuj Y o rozkładzie wykładniczym $E(0,1)$

until

$$Y = \frac{1}{2}(X - 1)^2$$

Return X

O pewnych dalszych rozwinięciach i zastosowaniach metody eliminacji powiemy w p. 3.1.4

Warunki szybkiej eliminacji lub szybkiej akceptacji

W przedstawionych wyżej algorytmach pojawiał się warunek eliminacji, który mógł przybierać różne postaci:

$$U \leq \frac{f(X)}{cg(X)}$$

lub

$$cUg(X) \leq f(X)$$

lub

$$X \leq h(U)$$

przy odpowiednio zdefiniowanej funkcji h , itp. Skoncentrujmy uwagę na warunku (3.10).

Może się okazać, że obliczenie prawej strony nierówności (3.10) jest czasochłonne i, wobec konieczności generowania dziesiątek lub setek tysięcy liczb losowych, może opłacać się wyznaczenie dwóch prostych funkcji $\alpha(x)$ i $\beta(x)$, bliskich funkcji $f(x)/cg(x)$, takich że

$$\alpha(x) \leq \frac{f(x)}{cg(x)} \leq \beta(x) \quad \text{dla wszystkich } x$$

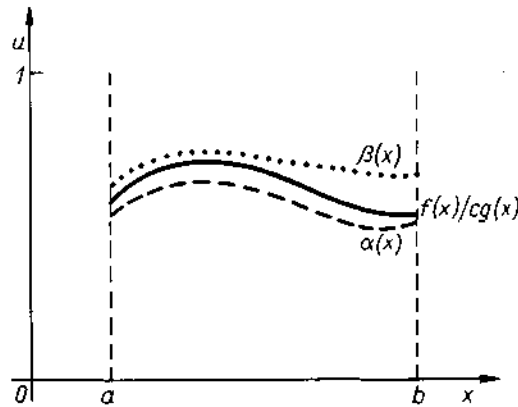
Zdarzenie

$$\{U \leq \alpha(X)\} \quad (3.11)$$

prowadzi oczywiście do zaakceptowania wygenerowanej liczby losowej X , a zdarzenie

$$\{U > \beta(X)\} \quad (3.12)$$

prowadzi do jej eliminacji i w konsekwencji do nowego generowania pary (X, U) (patrz rys. 3.3). Jeżeli żadne z tych zdarzeń nie zachodzi, o akceptacji lub eliminacji pary (X, U) decyduje spełnienie lub niespełnienie warunku (3.10), ale jeżeli różnice $\beta(x) - \alpha(x)$ są małe, to czasochłonna weryfikacja nierówności (3.10) będzie odbywała się rzadko.



Rys.3.3

Warunek (3.11) nazywa się *warunkiem szybkiej akceptacji*, a warunek (3.12) *warunkiem szybkiej eliminacji*.

Uogólnienie opisanego wyżej pomysłu polega na konstrukcji ciągu nierówności

$$\alpha_k(x) \leq \alpha_{k-1}(x) \leq \dots \alpha_1(x) \leq \frac{f(x)}{cg(x)} \leq \beta_1(x) \leq \beta_2(x) \dots \beta_l(x) \quad (3.13)$$

i kolejnym sprawdzaniu odpowiednich warunków, np. w następującej kolejności

$$U \leq \alpha_k(X), \quad U > \beta_l(X), \quad U \leq \alpha_{k-1}(X), \dots, U \leq \frac{f(x)}{cg(x)}$$

Oczywiście, pierwsze spełnienie jednej z kolejnych nierówności przerywa procedurę.

Przykład 6. W wielu algorytmach pojawiają się warunki akceptacji lub eliminacji postaci $V \leq e^{-x}$ lub $Y \leq \ln X$. Korzystając z nierówności

$$1 - x \leq e^{-x} \leq 1 - x + \frac{x^2}{2}$$

lub

$$-\frac{x}{1-x} \leq \ln(1-x) \leq -x, \quad 0 \leq x \leq 1$$

łatwo możemy skonstruować odpowiednie warunki szybkiej akceptacji i szybkiej eliminacji. •

Jeden z efektywnych sposobów konstruowania ciągu nierówności (3.13) polega na rozwinięciu w szereg naprzemienny funkcji $f(x)/cg(x)$ występującej w warunku 3.10. Szczegółowe sformułowanie tej idei pozostawiamy Czytelnikowi.

3.1.3. Metoda superpozycji rozkładów

Konstrukcja ogólna

Przedstawmy gęstość f rozkładu zmiennej losowej X w następującej postaci

$$f(x) = \int_{-\infty}^{\infty} g_t(x)h(t)dt \quad (3.14)$$

gdzie $g_t(x)$ jest dla każdej ustalonej wartości parametru t gęstością pewnego rozkładu prawdopodobieństwa oraz $h(t)$ jest również pewną gęstością.

Wzór (3.14) można również odczytać w następujący sposób: zmienna losowa X ma rozkład o gęstości $g_i(x)$ zależnej od pewnego parametru T , który z kolei jest zmienną losową o gęstości $h(t)$; wtedy $f(x)$ jest gęstością bezwarunkowego rozkładu zmiennej losowej X .

Rozkład prawdopodobieństwa o gęstości (3.14) nazywa się *rozkładem złożonym*, (patrz np. Fisz (1967), Krysicki i in. (1989)), a w literaturze angielskiej (np. Devroye (1986)), w takich przypadkach mówi się o *dekompozycji* rozkładu f .

Sposób generowania zmiennej losowej X o rozkładzie z gęstością (3.14) jest całkiem naturalny:

1. Wygenerować zmienną losową T według rozkładu o gęstości h .
2. Dla wygenerowanej wartości t zmiennej losowej T wygenerować X według rozkładu z gęstością $g_i(x)$.

Metoda superpozycji rozkładów została zaproponowana w pracy Butlera (1956), gdzie również podano pierwsze przykłady jej zastosowania.

Reprezentacja (3.14) gęstości $f(x)$ za pomocą całki jest bardzo ogólna i w szczególności obejmuje przypadek, w którym

$$f(x) = \sum_{i=1}^{\infty} p_i g_i(x) \quad (3.15)$$

gdzie $p_i \geq 0$, $\sum_{i=1}^{\infty} p_i = 1$ oraz g_i gęstościami pewnych rozkładów prawdopodobieństwa. W praktycznych zastosowaniach najczęściej używa się dekompozycji

$$f(x) = \sum_{i=1}^K p_i g_i(x) \quad (3.16)$$

gdzie K jest skończoną, zwykle niedużą liczbą. Jeden ze sposobów dokonania takiej dekompozycji polega na tym, że przedział, na którym gęstość $f(x)$ jest dodatnia, rozbija się na sumę rozłącznych przedziałów (lub ogólniej: zbiorów) A_1, A_2, \dots, A_K w taki sposób, żeby procedury generowania liczb losowych na każdym z tych małych zbiorów były łatwe i szybkie.

Po rozbiciu przedziału określoności gęstości na przedziały A_1, A_2, \dots, A_K i wyznaczeniu liczb

$$p_i = \int_{A_i} f(x) dx$$

otrzymujemy następującą reprezentację dla gęstości f :

$$f(x) = \sum_{i=1}^K p_i g_i(x), \quad g_i(x) = \frac{1_{A_i}(x) f(x)}{p_i}$$

Stąd otrzymujemy algorytm:

1. Wygenerować zmienną losową I o wartościach w zbiorze $\{1, 2, \dots, K\}$.
2. Dla wygenerowanej wartości $I = i$ wygenerować X według rozkładu z gęstością $g_i(x) = 1_{A_i}(x) f(x) / p_i$.

Na małym przedziale można łatwiej znaleźć „dobrą” gęstość h_i majoryzującą g_i w tym sensie, że $g_i \leq c_i h_i$ dla odpowiedniej stałej c_i (por. p.3.1.2). Powiemy o tym dokładniej poniżej. W

szczególności, gęstość h_i może być wielomianem, który dobrze przybliży z góry funkcję g_i . O generowaniu zmiennych losowych o rozkładach z gęstościami wielomianowymi powiemy dokładniej na s. 59.

Kombinacja metody superpozycji z metodą eliminacji

W przypadku gdy liczby losowe o rozkładach zdeterminowanych przez dekompozycję (3.15) są generowane metodą eliminacji za pomocą odpowiednio wybranych gęstości dominujących h_i , punkt 2 w podanym wyżej algorytmie przyjmuje postać:

2'. Dla wygenerowanej wartości $I = i$ wygenerować X o rozkładzie z gęstością h_i oraz U o rozkładzie równomiernym $U(0, 1)$. Zaakceptować ten wynik, gdy $c_i U h_i(X) \leq g_i(X)$ lub powtórzyć losowanie pary (X, U) . Przypadek dekompozycji (3.14) jest analogiczny, więc nie będziemy się nim zajmowali.

Dwa następujące przykłady, oba związane z rozkładem normalnym, ilustrują opisaną metodę.

Przykład 1. (dodatnia połówka rozkładu normalnego $N(0, 1)$). Generujemy zmienną losową X o rozkładzie z gęstością

$$f(x) = \sqrt{\frac{2}{\pi}} e^{-x^2/2}, \quad x \geq 0 \quad (3.17)$$

Przedstawiamy tę gęstość w postaci

$$f(x) = \alpha_1 f_1(x) g_1(x) + \alpha_2 f_2(x) g_2(x) \quad (3.18)$$

gdzie

$$\begin{aligned} \alpha_1 &= \sqrt{\frac{2}{\pi}} & \alpha_2 &= \frac{1}{\sqrt{2\pi}} \\ f_1(x) &= \begin{cases} 1 & \text{dla } 0 \leq x \leq 1 \\ 0 & \text{pozatym} \end{cases} & f_2(x) &= \begin{cases} 2e^{-2(x-1)} & \text{dla } x \geq 1 \\ 0 & \text{pozatym} \end{cases} \\ g_1(x) &= e^{-x^2/2} & g_2(x) &= e^{-(x-2)^2/2} \end{aligned}$$

Ponieważ $\alpha_1/(\alpha_1 + \alpha_2) = 2/3$ oraz $\alpha_2/(\alpha_1 + \alpha_2) = 1/3$, więc z prawdopodobieństwem $2/3$ będziemy generowali zmienną losową X o rozkładzie z gęstością f_1 i dokonywali ewentualnej eliminacji według funkcji g_1 lub z prawdopodobieństwem $1/3$ będziemy generowali zmienną losową X o rozkładzie z gęstością f_2 i dokonywali eliminacji według funkcji g_2 . Otrzymujemy następujący algorytm:

ALGORYTM 3.12

Generuj V o rozkładzie równomiernym $U(0,1)$

$$\text{If } V < \frac{2}{3}$$

then

Repeat

Generuj U o rozkładzie równomiernym $U(0,1)$

```

    Generuj  $X$  o rozkładzie równomiernym  $U(0,1)$ 
    until  $U \leq \exp(-X^2/2)$ 
else
    Repeat
    Generuj  $U$  o rozkładzie równomiernym  $U(0,1)$ 
    Generuj  $X$  z gęstością  $2e^{-2(x-1)}$ 
    until  $U \leq \exp(-(X-2)^2/2)$ 
Return  $X$ 

```

Ze względów dydaktycznych pozostawiamy ten algorytm w takiej właśnie postaci, nawiązującej bezpośrednio do wzoru (3.18), ale Czytelnik z łatwością zauważy, że jeżeli w pierwszym kroku algorytmu zamiast U będzie generowana zmienna losowa o rozkładzie wykładniczym $E(0,1)$, to z warunków akceptacji zniknie funkcja \exp . Przedstawiony tu algorytm pochodzi z pracy Butchera (1960).

Przykład 2. Pewne uogólnienie metody przedstawionej w poprzednim przykładzie polega na wprowadzeniu parametrów liczbowych do reprezentacji (3.18), którą teraz traktujemy jako reprezentację gęstości rozkładu normalnego $N(0,1)$ na całej prostej, i optymalizacji algorytmu przez odpowiedni wybór wartości tych parametrów. W tym celu w dekompozycji (3.18) definiujemy

$$\alpha_1 = \mu \sqrt{\frac{2}{\pi}}$$

$$f_1(x) = \begin{cases} \frac{1}{2\mu} & \text{dla } -\mu \leq x \leq \mu \\ 0 & \text{poza tym} \end{cases}$$

$$g_1(x) = \exp(-x^2/2)$$

$$\alpha_2 = \frac{1}{\lambda} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{\lambda^2}{2} - \lambda\mu\right)$$

$$f_2(x) = \begin{cases} \frac{1}{2} \lambda e^{-\lambda(|x|-\mu)} & \text{dla } |x| > \mu \\ 0 & \text{poza tym} \end{cases}$$

$$g_2(x) = \exp(-(|x| - \lambda)^2/2)$$

Proponujemy Czytelnikowi sprawdzenie, że oczekiwana liczba obliczeń potrzebnych do uzyskania jednej wartości zmiennej losowej X jest najmniejsza wtedy, gdy $\lambda = \sqrt{2}$ oraz $\mu = 1/\sqrt{2}$. Przedstawione tu uogólnienie pochodzi z książki Hammersleya i Handscomba (1961). •

Rozkłady o gęstościach wielomianowych

Przypuśćmy, że zmienna losowa X ma rozkład o gęstości f , którą można przedstawić w postaci szeregu potęgowego

$$f(x) = \sum_{i=1}^M c_i x^i, \quad 0 \leq x \leq 1, \quad c_i \geq 0$$

przy czym $M \leq \infty$. Wtedy $\sum_{i=1}^M c_i / (i+1) = 1$ i otrzymujemy prosty algorytm:

1. Wygenerować indeks $I \in \{1, 2, \dots\}$ według rozkładu prawdopodobieństwa $P\{I=i\} = c_i / (i+1)$.
2. Dla danej wartości $I=i$ wygenerować zmienną losową o rozkładzie z gęstością $(i+1)x^i$, $0 \leq x \leq 1$.

Zmienną losową o rozkładzie z gęstością $(i + 1)x^i$, $0 \leq x \leq 1$, można łatwo wygenerować metodą odwracania dystrybuanty albo za pomocą wzoru

$$X = \max\{U_1, U_2, \dots, U_{i+1}\}$$

gdzie U_1, \dots, U_i są niezależnymi zmiennymi losowymi o jednakowym rozkładzie równomiernym $U(0,1)$.

Może się okazać, że w „dobrym” wielomianie aproksymującym gęstość nie wszystkie współczynniki c_i są dodatnie. Zapiszmy c_i w postaci

$$c_i = c_i^+ - c_i^-, \quad c_i^+, c_i^- > 0$$

Wtedy

$$f(x) = \sum_{i=1}^M c_i x^i \leq \sum_{i=1}^M c_i^+ x^i$$

Oznaczmy $g(x) = \sum_{i=1}^M c_i^+ x^i$. Funkcja

$$\bar{g}(x) = g(x) / \sum_{i=1}^M c_i^+$$

jest gęstością pewnego rozkładu prawdopodobieństwa i w generowaniu zmiennej losowej X o rozkładzie z gęstością f może odgrywać rolę gęstości dominującej. Otrzymujemy algorytm

ALGORYTM 3.13

Repeat

Generuj X o rozkładzie z gęstością g

Generuj U o rozkładzie równomiernym $U(0,1)$

until $Ug(X) \leq f(X)$

Return X

Przykład 3. Zilustrujemy przypadek, w którym nośnikiem rozkładu generowanej zmiennej losowej X jest przedział $[-1,1]$, a nie, jak w algorytmie 3.13, przedział $[0,1]$,

Niech zmienna losowa X ma rozkład prawdopodobieństwa o gęstości określonej wzorem

$$f(x) = \begin{cases} \frac{3}{4}(1 - x^2) & \text{dla } -1 \leq x \leq 1 \\ 0 & \text{poza tym} \end{cases}$$

Otrzymujemy algorytm

ALGORYTM 3.14

Repeat

Generuj X o rozkładzie równomiernym $U(-1,1)$

Generuj U o rozkładzie równomiernym $U(0,1)$

until $U \leq 1 - X^2$

Return X

Zauważmy, że ten sposób traktowania wielomianów z ujemnymi współczynnikami łatwo

uogólnia się na przypadek dekompozycji (3.14), (3.15) i (3.16).

W procesie projektowania algorytmów generowania liczb losowych o rozkładach z wielomianowymi gęstościami są przydatne dwa następujące lematy (Devroye (1986)).

Lemat 3.3. *Jeżeli U_1 i U_2 są niezależnymi zmiennymi losowymi o jednakowym rozkładzie równomiernym $U(0,1)$ oraz $a > 1$ jest pewną stałą, to zmienna losowa $X = U_1^{1/a} U_2$ ma rozkład o gęstości*

$$f(x) = \frac{a}{a-1}(1-x^{a-1}), \quad 0 \leq x \leq 1$$

Dowód jest elementarny.

Lemat 3.4. *Zakładamy, że U_1, \dots, U_n , $n \geq 2$, jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym $U(0,1)$ i że L jest numerem pierwszej zmiennej losowej U_1, \dots, U_n , która nie jest równa $\max\{U_1, \dots, U_n\}$. Niech $f(x)$ będzie gęstością rozkładu prawdopodobieństwa zmiennej losowej U_L . Wtedy*

$$f(x) = \frac{n}{n-1}(1-x^{n-1}), \quad 0 \leq x \leq 1 \quad (3.19)$$

Dowód. Każda zmienna losowa U_i ma rozkład równomierny $U(0,1)$, tzn. ma gęstość równą 1 na przedziale $(0,1)$. Każda losowo wybrana zmienna losowa U_i ma taki rozkład. Ale losowo wybrana zmienna losowa U_i jest równa $\max\{U_1, \dots, U_n\}$ z prawdopodobieństwem $1/n$ oraz nie równa się $\max\{U_1, \dots, U_n\}$ z prawdopodobieństwem $(n-1)/n$. W pierwszym przypadku ma rozkład o gęstości nx^{n-1} , a w drugim rozkład o gęstości $f(x)$.

Mamy zatem

$$\frac{n-1}{n}f(x) + \frac{1}{2}nx^{n-1} = 1, \quad 0 \leq x \leq 1$$

a stąd wzór (3.19).

Przypadek $K = 2$. Metoda AC

Przedstawmy gęstość f zmiennej losowej X w postaci $f=f_1+f_2$, gdzie f_1 oraz f_2 są pewnymi funkcjami nieujemnymi. Oznaczmy przez p całkę $\int f_1(x)dx$. Wtedy f_1/p oraz $f_2/(1-p)$ są gęstościami pewnych rozkładów prawdopodobieństwa. Jeżeli funkcje f_1 i f_2 są tak wybrane, że łatwo i szybko potrafimy generować liczby losowe o rozkładzie z gęstością f_1/p metodą eliminacji (powiedzmy, z dominującą gęstością g) oraz liczby losowe o rozkładzie z gęstością $f_2/(1-p)$, to generowanie zmiennej losowej X możemy wykonać za pomocą następującego algorytmu:

ALGORYTM 3.15

Generuj X o rozkładzie z gęstością g

Generuj U o rozkładzie równomiernym $U(0,1)$

If $Ug(X) > f_1(X)$

then generuj X według rozkładu z gęstością $f_2/(1-p)$

Return X

W celu przekonania się o poprawności tego algorytmu, prześledźmy następujące rachunki. Dla zbioru B w przestrzeni wartości zmiennej losowej X , zgodnie z tym algorytmem mamy

$$P\{X \in B\} = P\{X \in B, U_g(X) \leq f_1(X)\} + \\ + P\{U_g(X) > f_1(X)\} \cdot \int_B f_2(x) dx / (1 - p)$$

Ale

$$P\{x \in B, U_g(X) \leq f_1(X)\} = \int_B \left(g(x) \int_0^{\frac{f_1(x)}{g(x)}} du \right) dx = \int_B f_1(x) dx$$

oraz

$$P\{U_g(X) > f_1(X)\} = \int \left(g(x) \int_{\frac{f_1(x)}{g(x)}}^1 dx \right) = \int g(x) dx - \int f_1(x) dx = 1 - p$$

więc

$$P\{X \in B\} = \int_B f(x) dx$$

Zauważmy, że w tym algorytmie mamy w istocie rzeczy do czynienia z dekompozycją

$$f(x) = 1_A(x) \frac{f_1(x)}{\int f_1(x) dx} + 1_{A^c}(x) \frac{f_2(x)}{\int f_2(x) dx}$$

gdzie A^c jest dopełnieniem zbioru A , natomiast zbiór A jest zdefiniowany w taki sposób, że zdarzenie $\{X \in A\}$, gdzie X jest zmienną losową o rozkładzie z gęstością g , jest równoważne ze zdarzeniem $\{U_g(X) \leq f_1(X)\}$, gdzie X jest zmienną losową o rozkładzie z gęstością g oraz U jest zmienną losową o rozkładzie równomiernym $U(0,1)$ i te dwie zmienne losowe są niezależne.

Zajście zdarzenia losowego $\{U_g(X) \leq f_1(X)\}$ przesądza o tym, że wylosowana w pierwszym kroku algorytmu liczba losowa X zostanie zaakceptowana, a zajście zdarzenia przeciwnego („uzupełniającego”) prowadzi do ostatecznego losowania X zgodnie z rozkładem o gęstości $f_2(x)/(1-p)$. Stąd właśnie pochodzi angielska nazwa tej metody: *acceptance-complement method* lub krótko: *metoda AC*. Autorami tego pomysłu są Kronmal i Pe-terson (1981, 1984).

Algorytm 3.15 staje się szczególnie prosty i szybki, gdy $f_2 = \text{const}$; generowanie według gęstości $f_2/(1-p)$ jest wtedy generowaniem zmiennej losowej o rozkładzie równomiernym na odpowiednim przedziale. Taki algorytm udaje się skonstruować, gdy oryginalna gęstość f jest „odcięta od zera”: można wtedy przyjąć $f_2(x) = \inf_x f(x)$. Ten pomysł wykorzystujemy w konstrukcji generatora liczb losowych o rozkładzie Cauchy'ego (p. 3.2.6).

3.1.4. Metoda ROU

Nazwa metody zaproponowanej przez Kindermana i Monahana (1977) pochodzi od angielskiego *ratio-of-uniforms method*. Opiera się ona na podanym niżej twierdzeniu.

Dla danej nieujemnej i całkownej funkcji $f(x)$, $-\infty < x < \infty$, zdefiniujmy zbiór

$$A = \left\{ (u, v) : 0 \leq u \leq \sqrt{F\left(\frac{v}{u}\right)} \right\}$$

Twierdzenie 3.4. *Jeżeli punkt losowy (U, V) ma rozkład równomierny na zbiorze A , to zmienna losowa $X = V/U$ ma rozkład o gęstości f/c , gdzie $c = \int f = 2l_2(A)$.*

Dowód. Niech (U, V) będzie punktem losowym o rozkładzie równomiernym na zbiorze A , tzn. punktem losowym o rozkładzie z gęstością

$$f_{U,V}(u, v) = \frac{1}{l_2(A)} 1_A(u, v)$$

Weźmy pod uwagę wzajemnie jednoznaczne przekształcenie $X = V/U$, $Y = U$. Jakobian

przekształcenia $u = y$, $v = xy$, jest równy $\left\| \det \begin{bmatrix} 0 & 1 \\ y & x \end{bmatrix} \right\| = y$,

więc gęstość rozkładu dwuwymiarowej zmiennej losowej (X, Y) wyraża się wzorem

$$f_{X,Y}(x, y) = \frac{1}{l_2(A)} 1_A(y, xy) \cdot y = \frac{y}{l_2(A)} 1_{[0, \sqrt{f(x)}]}(y)$$

Otrzymujemy zatem gęstość rozkładu zmiennej losowej X

$$f(x) = \int f_{X,Y}(x, y) dy = \frac{1}{l_2(A)} \int_0^{\sqrt{f(x)}} y dy = \frac{1}{2l_2(A)} f(x)$$

co kończy dowód.

Z tego twierdzenia wynika następujący algorytm: wygenerować punkt losowy (U, V) o rozkładzie równomiernym na zbiorze A i obliczyć $X = V/U$. Od strony technicznej realizacja tej metody sprowadza się do sprawnego generowania punktów losowych (U, V) o rozkładzie równomiernym na zbiorze A . Jeżeli zbiór A można zamknąć w pewnym prostokącie $\{(u, v) : 0 < u \leq b, a^- \leq v \leq a^+\}$, to algorytm przyjmuje postać

ALGORYTM 3.16

Repeat

Generuj U o rozkładzie równomiernym $U(0, b)$
Generuj V o rozkładzie równomiernym $U(a^-, a^+)$
 $X = V/U$
until $U^2 \leq f(X)$

Return X

W celu optymalnego wyboru stałych a^- , a^+ , b zauważmy przede wszystkim, że w

przestrzeni R^2 punktów (u, v)

- zbiór A leży w półpłaszczyźnie $u \geq 0$,
- jeżeli funkcja f jest symetryczna względem zera, to zbiór A jest symetryczny względem osi O_u ,
- jeżeli funkcja f jest gęstością pewnego rozkładu prawdopodobieństwa, skoncentrowanego na dodatniej półosi $0 < x < \infty$, to zbiór A zawiera się w dodatniej ćwiartce układu współrzędnych O_{uv} (bo wtedy $f(v/u) > 0$ tylko dla $v/u > 0$, czyli dla $v > 0$).

Wprowadzając parametr $t = v/u$, możemy napisać równania parametryczne brzegu tego zbioru:

$$u = \sqrt{f(t)}, \quad v = t\sqrt{f(t)}$$

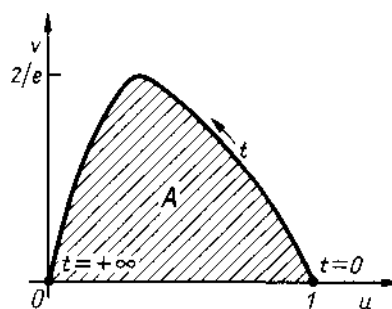
Jeżeli zbiór A ma być zamknięty w prostokącie $[a^-, a^+] \times (0, b)$, to stałe a^-, a^+, b muszą spełniać warunki

$$b \geq \sup_t \sqrt{f(t)}, \quad a^- \leq \inf_t t \cdot \sqrt{f(t)}, \quad a^+ \geq \sup_t t \cdot \sqrt{f(t)}$$

a optymalne stałe a^-, a^+, b spełniają te warunki ze znakami równości.

Przykład 1 (rozkład wykładniczy $E(0, 1)$). Zilustrujemy w możliwie najprostszy sposób metodę ROU; skonstruowany tu algorytm nie wytrzyma konkurencji z prostym generatorem $X = -\ln U$, gdzie U jest zmienną losową o rozkładzie równomiernym $U(0, 1)$.

Dla rozkładu wykładniczego $E(0, 1)$ mamy $f(x) = e^{-x}$, $x \geq 0$. W tym przypadku $a^- = 0$, $a^+ = 2/e$ oraz $b = 1$. Zbiór A przedstawiono na rys. 3.4.



Rys.3

Otrzymujemy algorytm

ALGORYTM 3.17

Repeat

Generuj U o rozkładzie równomiernym $U(0, 1)$

Generuj V o rozkładzie równomiernym $U(0, 2/e)$

$X = V/U$

until $U^2 \leq e^{-X}$

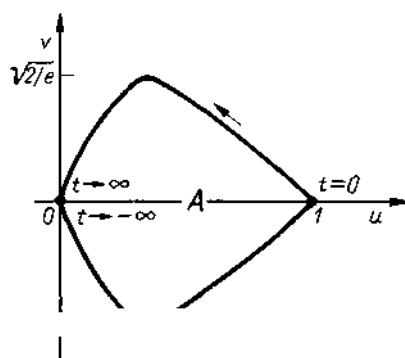
Return X

Jeden ze sposobów przyspieszenia tego algorytmu pokażemy w p. 3.2.2.

Przykład 2 (rozkład normalny $N(0,1)$). Obecnie zaprezentujemy algorytm, który jest bardzo efektywny i z sukcesem konkuruje z innymi generatorami liczb losowych o rozkładzie normalnym $N(0,1)$. W konstrukcji algorytmu użyjemy funkcji $f(x) = \exp(-x^2/2)$, bo odpowiednia stała normująca i tak pojawia się automatycznie.

Mamy $a^- = -\sqrt{2/e}$, $a^+ = \sqrt{2/e}$, $b = 1$. Zbiór A jest symetryczny względem osi Ou (patrz rys. 3.5), a równania parametryczne brzegu tego zbioru mają postać

$$u = e^{-t^2/4}, \quad v = t \cdot e^{-t^2/4}, \quad -\infty < t < \infty$$



Rys. 3.5

Otrzymujemy algorytm

ALGORYTM 3.18

Repeat

Generuj U o rozkładzie równomiernym $U(0,1)$

Generuj V o rozkładzie równomiernym $U(-\sqrt{2/e}, \sqrt{2/e})$

$X = V/U$

until $U^2 \leq e^{-X^2/2}$

Return X

Podobnie jak w poprzednim przykładzie, odpowiednie warunki szybkiej akceptacji i szybkiej eliminacji mogą znacznie przyspieszyć działanie algorytmu (patrz p. 3.2.3). •

Zwróćmy uwagę na następujące dwa lematy:

LEMAT 3.5. Jeżeli punkt losowy (U, V) ma rozkład równomierny na zbiorze $\{(u, v) : 0 \leq u \leq f(u + v)\}$, to zmienna losowa $U + V$ ma rozkład gęstości proporcjonalnej do f .

LEMAT 3.6. Jeżeli punkt losowy (U, V) ma rozkład równomierny na zbiorze $\{(u, v) : 0 \leq u \leq (f(v/\sqrt{u}))^{2/3}\}$ to zmienna losowa V/\sqrt{U} ma rozkład o gęstości proporcjonalnej do f .

Te dwa lematy sugerują inne warianty oraz uogólnienia metody ROU.

3.1.5. Rozkłady dyskretne

Metoda odwracania dystrybuanty

Metoda odwracania dystrybuanty, o której mówiliśmy w p. 3.1.1, w przypadku rozkładów dyskretnych prowadziła do wzoru (3.3). Dla rozkładu prawdopodobieństwa $P\{X = k\} = p_k$, $k = 0, 1, 2, \dots$ otrzymujemy prosty algorytm

ALGORYTM 3.19

$X = 0$, $S = p_0$

Generuj U o rozkładzie równomiernym $U(0, 1)$

While $U > S$

do $X = X + 1$, $S = S + p_x$

Return X

Łatwe uogólnienie tego algorytmu na przypadek rozkładu

$$P\{X = x_k\} = p_k, \quad k = 0, 1, 2, \dots \quad (3.20)$$

pozostawiamy Czytelnikowi.

Liczba N kroków w pętli *while* ma rozkład $P\{N = k\} = p_k$, może się więc okazać, że wartość oczekiwana tej zmiennej losowej jest bardzo duża lub nawet równa $+\infty$. Wynika stąd, że ten prosty algorytm może pracować bardzo wolno. Tak jest np. w niektórych dyskretnych wersjach rozkładu Pareto. Pewne przyspieszenie algorytmu można uzyskać przez taką permutację ciągu liczb p_0, p_1, p_2, \dots , żeby w nowym ciągu $p_{(0)}, p_{(1)}, p_{(2)}, \dots$ były spełnione nierówności

$$p_{(0)} \geq p_{(1)} \geq p_{(2)} \geq \dots$$

Jeżeli dokonamy takiej samej permutacji liczb x_0, x_1, x_2, \dots , to zmienna losowa Y o rozkładzie $P\{Y = X_{(k)}\} = p_{(k)}$, $k = 0, 1, 2, \dots$, będzie miała taki sam rozkład jak zmienna losowa X , ale średnia liczba kroków w pętli *while* będzie mniejsza.

Metoda równomiernego rozbicia przedziału (0, 1)

Metodę odwracania dystrybuanty można interpretować w następujący sposób. Rozbijamy przedział $(0, 1)$ na rozłączne podprzedziały o długościach równych p_0, p_1, p_2, \dots . Każdemu podprzedziałowi przyporządkowujemy odpowiednią wartość zmiennej losowej: podprzedziałowi o długości p_i wartość x_i . Generujemy liczbę losową U o rozkładzie równomiernym $U(0, 1)$ i za wynik generowania zmiennej losowej X przyjmujemy tę wartość x_i , która odpowiada podprzedziałowi, do którego wpadło to U .

Metoda równomiernego rozbicia przedziału $(0, 1)$ polega na podzieleniu przedziału $(0, 1)$ na jednakowo długie podprzedziały, sprawdzeniu, do którego z nich wpadło U , a następnie na identyfikowaniu wartości generowanej zmiennej losowej X już tylko w tym małym podprzedziale. Oto szczegóły.

Rozważamy zmienną losową X przyjmującą skończoną liczbę różnych wartości

$$P\{X = k\} = p_k, \quad k = 0, 1, \dots, K \quad (3.21)$$

Dzielimy przedział $(0,1)$ na $K+1$ podprzedziałów $\left(\frac{i-1}{K+1}, \frac{i}{K+1}\right)$ o jednakowej długości i umawiamy się, że odcinek $\left(0, \frac{1}{K+1}\right)$ ma numer 1. Zatem zmienna U wpada do podprzedziału o numerze $[(K+1)U + 1]$.

Dla danego rozkładu (3.21) tworzymy ciąg $q_i = \sum_{j=0}^i p_j, i = 0, 1, \dots, K,$ i przyjmujemy $q_{-1} = 0$. Następnie tworzymy pomocniczy ciąg liczb

$$g_i = \max \left\{ j : q_j < \frac{i}{K+1} \right\}, \quad i = 1, 2, \dots, K+1$$

umawiając się przy tym, że maksimum na zbiorze pustym jest równe zero. Po tych przygotowaniach algorytm generowania liczb losowych o rozkładzie (3.21) przyjmuje następującą postać:

ALGORYTM 3.20

Generuj U o rozkładzie równomiernym $U(0,1)$

$X = [(K+1)U + 1]$

$X = q_X + 1$

While $q_{X-1} > U$ do $X = X - 1$

Return X

Najważniejszą własność tego algorytmu jest opisana w następującym twierdzeniu:

Twierdzenie 3.5. *Oczekiwana liczba porównań $\{q_{X-1} > U\}$ jest nie większa od 2.*

Dowód. Jeżeli w pierwszym kroku algorytmu został wybrany podprzedział o numerze i , to liczba porównań $\{q_{X-1} > U\}$ nie będzie większa od liczby różnych wartości q_j w tym podprzedziale, plus jeden. Każdy z $K+1$ podprzedziałów jest wybierany z jednakowym prawdopodobieństwem, więc oczekiwana liczba porównań \leq

$$\leq 1 + \frac{1}{K+1} \sum_{i=0}^K (\text{liczba różnych } q_j \text{ w } i\text{-tym przedziale}) = 2$$

Pomysł metody równomiernego rozbicia przedziału $(0,1)$ pochodzi z pracy Chena i Asau (1974), a opis tej metody podaliśmy według monografii Devroya (1986). W literaturze anglojęzycznej ta metoda nosi nazwę *method of guide tables*.

Mieszanka rozkładów dwupunktowych

Mówimy, że rozkład o gęstości $f(x)$ jest *mieszaniną* k rozkładów o gęstościach $f_i(x)$, $i = 0, 1, \dots, k-1$, jeżeli

$$f(x) = \sum_{i=0}^{k-1} \lambda_i f_i(x)$$

gdzie

$$\sum_{i=0}^{k-1} \lambda_i = 1, \quad \lambda_i > 0, \quad i = 0, 1, \dots, k-1$$

Weźmy pod uwagę zmienną losową X o rozkładzie dyskretnym

$$P\{X = x_i\} = p_i, \quad i = 0, 1, \dots, k-1 \quad (3.23)$$

Mówimy, że rozkład (3.23) jest *równomierną mieszaniną rozkładów dwupunktowych*, jeżeli istnieją ciąg par liczb (u_j, v_j) , $j = 0, 1, \dots, k-1$, oraz ciąg liczb $q_j \in (0, 1)$, $j = 0, 1, \dots, k-1$, takie że

$$p_i = \frac{1}{k} \sum_{j=0}^{k-1} (q_j 1_{\{u_j\}}(x_i) + (1 - q_j) 1_{\{v_j\}}(x_i)), \quad i = 0, 1, \dots, k-1 \quad (3.24)$$

Generowanie liczb losowych X o rozkładzie dyskretnym (3.23), który jest równomierną mieszaniną rozkładów dwupunktowych, jest bardzo łatwe i szybkie: wystarczy wygenerować indeks J według rozkładu równomiernego na zbiorze $\{0, 1, \dots, k-1\}$, a następnie wygenerować X według J -tego rozkładu dwupunktowego

$$P\{X = u_j\} = q_j, \quad P\{X = v_j\} = 1 - q_j$$

Twierdzenie 3.6. *Każdy rozkład dyskretny postaci (3.23) można przedstawić w postaci równomiernej mieszaniny (3.24) rozkładów dwupunktowych.*

Dowód (indukcyjny). Dla $k = 1$ mamy oczywiście $u_0 = x_0$ i $q_0 = p_0 (= 1)$. Przypuśćmy, że teza jest prawdziwa dla rozkładów $(k-1)$ -punktowych i rozpatrzmy k -punktowy rozkład (3.23).

Niech i_0 będzie takim wskaźnikiem, że

$$p_{i_0} = \min_{0 \leq i \leq k} p_i$$

Podstawmy $u_0 = x_{i_0}$ oraz $q_0 = kp_{i_0}$ (mamy oczywiście $p_{i_0} \leq 1/k$, więc $kp_{i_0} \leq 1$). Niech j_0 będzie takim wskaźnikiem, że

$$p_{j_0} = \max_{0 \leq i \leq k} p_i$$

Mamy oczywiście $p_{j_0} \geq 1/k$, więc $(1 - q_0)/k \leq p_{j_0}$. Podstawmy $v_0 = x_{j_0}$. Wielkości (u_0, v_0) oraz q_0 potraktujmy jako pierwszy składnik sumy (3.24).

Liczby

$$p'_i = p_i - \frac{1}{k} (q_0 1_{\{u_0\}}(x_i) + (1 - q_0) 1_{\{v_0\}}(x_i)), \quad 0 \leq i \leq k$$

są nieujemne i sumują się do $1 - \frac{1}{k}$, przy czym $p'_{i_0} = 0$. Teraz

$$\begin{pmatrix} x_0 & x_1 & \dots & x_{i_0-1} & x_{i_0+1} & \dots & x_{k-1} \\ \tilde{p}_0 & \tilde{p}_1 & \dots & \tilde{p}_{i_0-1} & \tilde{p}_{i_0+1} & \dots & \tilde{p}_{k-1} \end{pmatrix}$$

gdzie $\tilde{p}_i = kp_i/(k-1)$, jest $(k-1)$ -punktowym rozkładem prawdopodobieństwa, który na mocy założenia indukcyjnego może być przedstawiony w postaci (3.24), co kończy dowód.

Algorytm korzysta z trzech następujących bloków liczb: $(q_0, q_1, \dots, q_{k-1})$, $(u_0, u_1, \dots, u_{k-1})$ oraz $(v_0, v_1, \dots, v_{k-1})$ i przebiega według następującego schematu:

ALGORYTM 3.21

Generuj U_1 o rozkładzie równomiernym $U(0, 1)$

Generuj U_2 o rozkładzie równomiernym $U(0, 1)$

$I = \lfloor kU_1 \rfloor$

If $U_2 \leq q_I$ then $X = u_I$ else $X = v_I$

Return X

Ten algorytm dwukrotnie odwołuje się do generowania liczb losowych o rozkładzie równomiernym $U(0, 1)$; korzystając z następującego lematu, można te dwa odwołania zredukować do jednego.

LEMAT 3.7. *Jeżeli zmienna losowa U ma rozkład równomierny $U(0, 1)$, to zmienna losowa $\lfloor kU \rfloor$ ma rozkład równomierny na zbiorze $\{0, 1, \dots, k-1\}$, zmienna losowa $\{kU\}$ ma rozkład równomierny $U(0, 1)$ i te zmienne losowe są niezależne.*

Dowód pozostawiamy Czytelnikowi.

3.2. Metody konstrukcji generatorów dla podstawowych rozkładów prawdopodobieństwa

3.2.1. Rozkłady dyskretne

Rozkład dwumianowy

Zmienna losowa X ma rozkład dwumianowy $b(n, p)$, jeżeli

$$p\{X = x\} = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, \dots, n \quad (3.25)$$

Najprostszy algorytm generowania zmiennej losowej o rozkładzie dwumianowym, oparty na jej definicji (liczba sukcesów w schemacie Bernoulliego), nie wymaga komentarzy:

ALGORYTM 3.22

$X = 0$

For $i = 1$ to n do

Generuj U o rozkładzie równomiernym $U(0, 1)$

If $U \leq p$ then $X = X + 1$

Return X

Ten algorytm wymaga wielokrotnego odwoływania się do generatora liczb losowych o rozkładzie równomiernym $U(0, 1)$, co czasami może okazać się jego wadą.

Jeżeli n nie jest bardzo duże, to może opłacać się stabilizowanie dystrybuanty tego rozkładu, tzn. obliczenie liczb

$$p_k = \sum_{i=1}^k P\{X = i\} \quad (3.26)$$

i skorzystanie z następującego algorytmu, który wymaga tylko jednej liczby losowej o rozkładzie $U(0,1)$:

ALGORYTM 3.23

Generuj U o rozkładzie równomiernym $U(0, 1)$

$X = 0$

While $U > p_x$ do $X = X + 1$

Return X

Jeżeli n jest duże, ale może być przedstawione w postaci $n = km$, gdzie z kolei m nie jest bardzo duże, to może opłacać się generowanie k liczb losowych o rozkładach dwumianowych $b(m,p)$ za pomocą algorytmu 3.23 i obliczenie X jako sumy tak wygenerowanych liczb.

Inny algorytm, który tylko raz odwołuje się do generatora liczb losowych o rozkładzie równomiernym $U(0,1)$, jest oparty na następującym lemacie (patrz Devroye (1986)):

LEMAT 3.8. *Jeżeli zmienna losowa U ma rozkład równomierny $U(0,1)$, to zmienne losowe*

$$1_{(0,p)}(U) \quad \text{oraz} \quad V = \min\left\{\frac{U}{p}, \frac{1-U}{1-p}\right\}$$

są niezależne i V ma rozkład równomierny $U(0,1)$.

Dowód. Prawdopodobieństwo łącznego zajścia zdarzeń $\{1_{(0,p)}(U) = 1\}$ oraz $\{V \leq x\}$, dla $x \leq 1$, jest równe px . Prawdopodobieństwo zdarzenia $\{1_{(0,p)}(U) = 1\}$ jest równe p . Zdarzenie $\{V \leq x\}$ przedstawiamy w postaci sumy rozłącznych zdarzeń $\{V \leq x, U \leq p\}$ i $\{V \leq x, U > p\}$. Prawdopodobieństwo pierwszego z tych zdarzeń jest równe px , a drugiego $(1-p)x$, więc prawdopodobieństwo zdarzenia $\{V \leq x\}$ jest równe x .

Z lematu 3.8 wynika, że zmienną losową $1_{(0,p)}(U)$ możemy traktować jako wskaźnik pojawienia się sukcesu, a zmienną losową V możemy użyć jako zmienną losową o rozkładzie $U(0,1)$ w niezależnym powtórzeniu doświadczenia w schemacie Bernoulliego. Prowadzi to do następującego algorytmu:

ALGORYTM 3.24

Generuj U o rozkładzie równomiernym $U(0,1)$

$X = 0$

For $i = 1$ to n do

If $U \leq p$ then $X = X + 1$, $U = U/p$

else $U = (1-U)/(1-p)$

Return X

Podkreślamy jednak, że manipulacje liczbą losową U w tym algorytmie mogą być na niektórych komputerach bardziej czasochłonne niż wielokrotne odwoływanie się do generatora liczb losowych o rozkładzie równomiernym $U(0,1)$ (algorytm 3.22) lub korzystanie z tablic (algorytm 3.23). Jeszcze inny algorytm, dla przypadku rozkładów dwumianowych z dużymi wartościami oczekiwanymi, podali Kachitvichyanukul i Schmeiser (1988).

Rozkład Poissona

Zmienna losowa X ma rozkład Poissona $P(\lambda)$, jeżeli

$$P\{X = x\} = \frac{\lambda^x}{x!} e^{-\lambda}, \quad x = 0, 1, \dots \quad (3.27)$$

Najprostszy algorytm generowania zmiennej losowej o tym rozkładzie jest oparty na następującym lemacie:

LEMAT 3.9. *Jeżeli $\xi_0, \xi_1, \xi_2, \dots$ jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie wykładniczym $E(0, 1)$, to zmienna losowa*

$$\min \left\{ j: \sum_{i=0}^j \xi_i > \lambda \right\}$$

ma rozkład Poissona $P(\lambda)$.

Dowód. Zdarzenia losowe $\{X \leq k\}$ oraz $\{\sum_{i=0}^k \xi_i > \lambda\}$ są równoważne. Zmienna losowa $\{\sum_{i=0}^k \xi_i\}$ ma rozkład $\Gamma(k+1, 1)$ gdzie $\Gamma(\alpha, \theta)$ oznacza rozkład gamma o gęstości

$$f_{\alpha, \theta}(x) = \frac{1}{\Gamma(\alpha) \theta^\alpha} x^{\alpha-1} e^{-x/\theta}$$

Zatem $P\{X \geq k\} = \int_k^\infty f_{k+1,1}(x) dx$ i wystarczy pokazać, że $P\{X = k\} = P\{X \geq k\} - P\{X \geq k-1\}$ jest równe prawdopodobieństwu (3.27).

Odpowiedni algorytm przyjmuje postać:

ALGORYTM 3.25

$X = -1, S = 0$

While $S \leq \lambda$ *do*

 Generuj Y o rozkładzie $E(0, 1)$, $S = S+Y, X = X+1$

Return X

Oczywista jest następująca odmiana tego algorytmu:

ALGORYTM 3.26

$X = -1, S = 1, q = e^{-\lambda}$

While $S > q$ *do*

 Generuj U o rozkładzie $U(0, 1)$, $S = S * U, X = X + 1$

Return X

Oba algorytmy mają tę wadę, że wymagają wielokrotnego odwoływania się do generatora liczb losowych o rozkładzie wykładniczym (algorytm 3.25) lub równomiernym (algorytm 3.26), przy czym liczba tych odwołań rośnie wraz z λ . Następujący algorytm, oparty na metodzie odwracania dystrybucyj, wymaga tylko jednej liczby losowej U o rozkładzie równomiernym $U(0,1)$

ALGORYTM 3.27

$q = e^{-\lambda}$, $X = 0$, $S = q$, $P = q$

Generuj U o rozkładzie $U(0, 1)$

While $U > S$ do $X = X + 1$, $P = P * \lambda / X$, $S = S + P$

Return X

Przy dużych wartościach λ następuje kumulacja błędów zaokrągleń, związana z sumowaniem dużej liczby składników (algorytm 3.25), mnożeniem dużej liczby czynników (algorytm 3.26) lub z obliczaniem prawdopodobieństw P (algorytm 3.27), tak że każdy z tych algorytmów może być bez dodatkowych zabezpieczeń używany tylko dla małych lub niezbyt wielkich wartości λ .

Dla dużych wartości λ , powiedzmy $\lambda > 30$, rozważymy dwie sytuacje:

a) Jeżeli dla pewnego, ustalonego, dużego λ mamy wygenerować dużo liczb losowych X o rozkładzie $P(\lambda)$, to może się opłacać, kosztem pewnych obliczeń przygotowawczych, wyznaczenie odpowiednich parametrów dla metody eliminacji i generowanie X tą metodą;

b) Jeżeli mamy generować liczby losowe o rozkładach $P(\lambda)$, ale λ zmienia się przy kolejnych odwołaniach do generatora (co zdarza się np. w symulacjach związanych z procesem Poissona o zmiennej intensywności), to może się opłacić przygotowanie tablic rozkładów $P(1)$, $P(2)$, $P(4)$, $P(8)$... oraz programu generującego zmienną losową Z o rozkładzie $P(\mu)$ dla $\mu < 1$ i, np. zmienną losową X o rozkładzie $P(75.3)$ generować w postaci sumy zmiennych losowych o rozkładach $P(0.3)$, $P(1)$, $P(2)$, $P(8)$, $P(64)$.

W przypadku a) dobrze się spisuje następujący algorytm zaproponowany przez Atkinsona (1979a) (patrz też Atkinson (1979b)), skonstruowany metodą eliminacji z gęstością rozkładu logistycznego jako gęstością dominującą. Oto szczegóły.

Generujemy zmienną losową X o rozkładzie Poissona $P(\lambda)$. Weźmy pod uwagę rozkład logistyczny o dystrybuancie

$$G_{\lambda}(x) = \frac{1}{1 + \exp(\alpha - \beta x)}, \quad -\infty < x < +\infty$$

gdzie parametry

$$\beta = \frac{\pi}{\sqrt{3\lambda}}, \quad \alpha = \beta \lambda$$

są tak dobrane, żeby wartość oczekiwana α/β i wariancja $\pi/(3\beta^2)$ w tym rozkładzie były równe odpowiednio wartości oczekiwanej i wariancji w rozkładzie Poissona $P(\lambda)$. Zaproponowana przez Atkinsona „poprawka na ciągłość” polega na tym, żeby po wylosowaniu X według rozkładu logistycznego, za kandydata na liczbę losową o rozkładzie Poissona przyjąć $N = [X]$. Standardowa realizacja metody eliminacji przybiera teraz postać: generować X o rozkładzie logistycznym z dystrybuantą $G_{\lambda}(x)$ oraz U o rozkładzie równomiernym $U(0,1)$ dopóty, dopóki nie zostanie spełniony standardowy warunek akceptacji, który - w nawiązaniu do oryginalnych oznaczeń Atkinsona - zapiszemy w postaci

$$\frac{U\beta \exp(\alpha - \beta x)}{p(1 + \exp(\alpha - \beta x))^2} \leq \frac{\lambda^N}{N!} e^{-\lambda}$$

gdzie p jest taką stałą, że

$$\frac{\lambda^N}{N!} e^{-\lambda} \leq \frac{\beta \exp(\alpha - \beta x)}{p(1 + \exp(\alpha - \beta x))^2} \text{ dla wszystkich } x$$

Stała p jest równa prawdopodobieństwu spełnienia warunku akceptacji, a więc powinna być największą stałą spełniającą ten warunek. Taka stała p jest określona wzorem

$$p = \beta \exp(\alpha + \lambda) \min_x \frac{N! \exp(-\beta x)}{\lambda^N (1 + \exp(\alpha - \beta x))^2}$$

Każdorazowe obliczanie stałej p może być zbyt czasochłonne i dlatego Atkinson (1979a) podaje następującą tabelkę:

λ	10	20	30	50	70	100	150	200
p	0.5925	0.6506	0.6760	0.7049	0.7202	0.7348	0.7472	0.7552

i sugeruje interpolację liniową względem $1/\lambda$. Taka interpolacja między $\lambda = 70$ i $\lambda = 100$ prowadzi do wygodnego wzoru $p = 0.767 - 3.36 / \lambda$. Ostatecznie otrzymujemy następujący algorytm:

ALGORYTM 3.28

$$\beta = \pi / \sqrt{3\lambda}, \alpha = \beta \lambda, k = \ln p - \lambda - \ln \beta$$

Repeat

Repeat

Generuj U o rozkładzie równomiernym $U(0, 1)$

$$X = \beta^{-1} (\alpha - \ln \frac{1-U}{U})$$

until $X > -1/2$

$$N = [X]$$

Generuj U o rozkładzie równomiernym $U(0, 1)$

$$\text{until } \alpha - \beta X + \ln \frac{U}{(1 + \exp(\alpha - \beta X))^2} \leq k + N \ln \lambda - \ln(N!)$$

Return N

Istnieje wiele innych metod generowania zmiennej losowej o rozkładzie Poissona. Kilka algorytmów, o których tutaj nie mówiliśmy, można znaleźć w cytowanych wyżej dwóch pracach Atkinsona oraz w dużej monografii Devroye'a (1985).

Rozkład geometryczny

Zmienna losowa X ma rozkład geometryczny $G(p)$, jeżeli

$$P\{X = x\} = (1-p)p^x, \quad x = 0, 1, 2, \dots$$

Najprostszy algorytm generowania zmiennej losowej o tym rozkładzie jest oparty na następującym lemacie:

LEMAT 3.10. *Jeżeli zmienna losowa X ma rozkład wykładniczy o gęstości ae^{-ax} , to zmienna losowa $[X]$ ma rozkład geometryczny $G(e^{-a})$.*

Na mocy tego lematu, w celu wygenerowania zmiennej losowej X o rozkładzie (3.28) wystarczy wygenerować zmienną losową U o rozkładzie równomiernym $U(0, 1)$ i obliczyć $X = [-\ln U / \ln p]$.

3.2.2. Rozkład wykładniczy

Gęstość prawdopodobieństwa rozkładu wykładniczego $E(\theta, \lambda)$ wyraża się wzorem

$$f_{\theta, \lambda}(x) = \frac{1}{\lambda} \exp\left(-\frac{x - \theta}{\lambda}\right), \quad 0 < x < +\infty$$

Jeżeli zmienna losowa Y ma rozkład wykładniczy $E(\theta, \lambda)$, to zmienna losowa $X = (Y - \theta) / \lambda$ ma rozkład wykładniczy o gęstości

$$f(x) = e^{-x}, \quad 0 < x < +\infty \quad (3.29)$$

więc w dalszym ciągu będziemy zajmowali się tylko rozkładem (3.29).

Dystrybuanta rozkładu $E(0, 1)$ ma postać

$$F(x) = 1 - e^{-x}, \quad 0 < x < +\infty \quad (3.30)$$

skąd wynika prosty sposób generowania liczb losowych o rozkładzie wykładniczym $E(0, 1)$ metodą odwracania dystrybuanty: $X = -\ln U$, gdzie U jest zmienną losową o rozkładzie równomiernym $U(0, 1)$. Jest to sposób dokładny i szybki, ale gdy trzeba generować dużo liczb losowych o rozkładzie wykładniczym, bardziej przydatne mogą okazać się jeszcze szybsze algorytmy, nie odwołujące się do funkcji standardowej \ln . Omówimy dwa takie algorytmy. Jeden z nich (algorytm 3.17), skonstruowany metodą ROU, przedstawiliśmy w p. 3.1.4. Warunek akceptacji $\{U^2 \leq e^{-x}\}$ w algorytmie (3.17) można zapisać w postaci $\{X \leq -2 \ln U\}$. Korzystając z nierówności

$$-\frac{t}{1-t} \leq \ln(1-t) \leq -t, \quad 0 \leq t < 1$$

otrzymujemy warunek szybkiej akceptacji $\{X \leq 2(1 - U)\}$ oraz warunek szybkiej eliminacji $\{X > 2/U - 2\}$. Prowadzi to do następującego szybkiego algorytmu:

ALGORYTM 3.29

Repeat

Generuj U o rozkładzie równomiernym $U(0,1)$

Generuj V o rozkładzie równomiernym $U(0, 2/e)$

$X=V/U$

If $X \leq 2(1-U)$ then akceptuj=true

else If $X \leq 2/U - 2$ then

If $X \leq -2 \ln U$ then akceptuj=true

until akceptuj

Return X

Proponujemy Czytelnikowi obliczenie, z jakim prawdopodobieństwem dochodzi do sprawdzania oryginalnego warunku $\{X \leq -2 \ln U\}$.

Drugi algorytm, o którym chcemy powiedzieć, oparty na *metodzie serii monotonicznych* zaproponowanej przez von Neumanna (1951), pochodzi jeszcze z epoki wolnych komputerów o czasochłonnych podprogramach dla funkcji standardowych. Korzysta on tylko z porównania liczb i dwóch liczników. Oto jego konstrukcja.

Generujemy ciąg U_1, U_2, \dots niezależnych liczb losowych o jednakowym rozkładzie równomiernym $U(0,1)$. Obserwujemy kolejne serie postaci

$$U_1 \geq U_2 \geq \dots \geq U_n < U_{n+1}$$

i numerujemy je kolejnymi liczbami $0, 1, 2, \dots$. Numer pierwszej serii, w której n (długość serii) jest liczbą nieparzystą, przyjmujemy za część całkowitą generowanej liczby losowej X , natomiast wartość R_1 pierwszej liczby w tej serii - za jej część ułamkową.

Aby przekonać się, że generowana w ten sposób liczba losowa X ma rozkład wykładniczy $E(0,1)$, rozpatrzmy następujące zdarzenia losowe:

$$E_n(u) = \{u \geq U_1 \geq U_2 \geq \dots \geq U_n\}$$

$$A_n(u) = \{u \geq U_1 \geq U_2 \geq \dots \geq U_n < U_{n+1}\}$$

Zauważmy, że $A_n(u) = E_n(u) - E_{n+1}(u)$ oraz że zdarzenie $B(1)$ polega na zaobserwowaniu serii „nieparzystej”.

Niech $B_0(u), B_1(u), \dots$ będzie ciągiem niezależnych zdarzeń losowych, takich jak zdarzenie $B(u)$. Wtedy

$$P\{m < X \leq m + u\} = P\{B_0^c(1) \cap B_1^c(1) \cap \dots \cap B_{m-1}^c(1) \cap B_m(u)\}$$

gdzie B^c jest zdarzeniem przeciwnym do zdarzenia B .

W wyniku obliczeń otrzymujemy

$$P\{E_n(u)\} = \int_{u \geq u_1 \geq u_2 \geq \dots \geq u_n} du_1 du_2 \dots du_n = \frac{u^n}{n!}$$

$$P\{A_n(u)\} = P\{E_n\} - P\{E_{n+1}\} = \frac{u^n}{n!} - \frac{u^{n+1}}{(n+1)!}$$

$$P\{B(u)\} = \sum_{n=0}^{\infty} \left(\frac{u^{2n+1}}{(2n+1)!} - \frac{u^{2n+2}}{(2n+2)!} \right) = 1 - e^{-u}$$

$$\begin{aligned} P\{m < X \leq m + u\} &= P\{(B^c(1))^m\} P\{B(u)\} = \\ &= (e^{-1})^m (1 - e^{-u}) = F(m+u) - F(u) \end{aligned}$$

gdzie F jest dystrybuantą rozkładu wykładniczego (3.30).

Generator liczb losowych o rozkładzie wykładniczym jest jednym z najczęściej używanych generatorów - bądź bezpośrednio do produkcji liczb losowych o tym rozkładzie, bądź jako pewien element w generatorach liczb losowych o innych rozkładach. W bogatej literaturze można znaleźć kilkanaście innych generatorów liczb losowych o rozkładzie wykładniczym (patrz np. monografia Devroye'a (1986)).

Jako pewną ciekawostkę odnotujemy, że n niezależnych zmiennych losowych o jednakowym rozkładzie wykładniczym $E(0, 1)$ można wygenerować za pomocą $(n-1)$ liczb losowych o rozkładzie równomiernym $U(0, 1)$ i jednej liczby losowej Z o rozkładzie gamma $\Gamma(n, 1)$: jeżeli $U_{1:n-1}, \dots, U_{n-1:n-1}$ jest statystyką pozycyjną z rozkładu równomiernego $U(0,1)$, to $ZU_{1:n-1}, Z(2:n-1) - U_{1:n-1}, \dots, Z(1-U_{n-1:n-1})$ jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie wykładniczym $E(0, 1)$.

3.2.3. Rozkład normalny

Gęstość prawdopodobieństwa rozkładu normalnego $N(\mu, \sigma)$ wyraża się wzorem

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

Jeżeli zmienna losowa X ma rozkład normalny $N(\mu, \sigma)$, to $(X-\mu)/\sigma$ ma rozkład normalny $N(0,1)$; dalej będziemy zajmowali się więc tylko zmiennymi losowymi o rozkładzie normalnym $N(0, 1)$. Gęstość prawdopodobieństwa tego rozkładu oznaczamy przez φ , a dystrybuantę przez Φ .

Ze względu na liczne i bardzo różnorodne zastosowania rozkładu normalnego opracowano wiele algorytmów generowania liczb losowych o tym rozkładzie.

Metodę odwracania dystrybuanty dla rozkładu normalnego przedstawiliśmy w p. 3.1.1. Jest to metoda dokładna i przy dobrej implementacji komputerowej (schemat Homera!) na tyle szybka, że warto jej się przyjrzeć na swoim komputerze. Podstawowa jej zaleta polega na tym, że używa tylko jednej liczby losowej o rozkładzie równomiernym $U(0,1)$ (por. p. 3.1.1).

Metodę eliminacji z gęstością rozkładu wykładniczego jako gęstością dominującą podaliśmy w przykładzie 1 w p. 3.1.2. Inny algorytm opracowany według tej metody, oparty na pewnej szczególnej faktoryzacji gęstości, podaliśmy w tym samym punkcie w przykładzie 5.

Metodę superpozycji rozkładów ilustrowaliśmy w p. 3.1.3 generatorem liczb losowych o rozkładzie normalnym w przykładach 1 i 2.

Metodę ROU dla rozkładu normalnego przedstawiliśmy w p. 3.1.4 w przykładzie 2. Warunek akceptacji w podanym tam algorytmie 3.18 miał postać $\{U^2 \leq e^{-X^2/2}\}$. W punkcie 3.1.2 mówiliśmy o warunkach szybkiej akceptacji i szybkiej eliminacji, opartych na odpowiednio dobranych nierównościach. Rozważany teraz warunek akceptacji można zapisać w postaci $\{X^2 \leq -4 \ln U\}$. Korzystając np. z nierówności

$$\frac{3}{2} - 2u + \frac{1}{2}u^2 < -\ln u < \frac{1}{2}\left(\frac{1}{u} - u\right)$$

otrzymujemy szybki i elegancki algorytm

ALGORYTM 3.30

Repeat

Generuj U o rozkładzie równomiernym $U(0,1)$

Generuj V o rozkładzie równomiernym $U(-\sqrt{2/e}, \sqrt{2/e})$

$X = V/U$

If $X^2 \leq 2(3 - U(4 + U))$ then akceptuj=true *Else*

If $X^2 \leq 2/U - 2U$ then

If $X^2 \leq -4 \ln U$ then akceptuj=true

until akceptuj

Return X

Metoda Marsaglii i Braya z 1964 roku jest tak piękna i pouczająca, że mimo upływu lat ciągle przyciąga uwagę wszystkich zajmujących się symulacjami komputerowymi i w różnych wersjach jest cytowana prawie w każdej książce na temat generatorów liczb losowych (Zieliński (1972, 1979); Devroye (1986), Ripley (1987) i in.). Implementacja, którą podajemy, pochodzi od Kindermana i Ramage'a (1976).

Istota rzeczy tkwi w odpowiednio wykonanej dekompozycji gęstości rozkładu normalnego. Rozpocznijmy tę dekompozycję od obcięcia ogonów; eleganckie i efektywne algorytmy generowania zmiennych losowych z ogona rozkładu $N(0, 1)$ pokazaliśmy w przykładach w p. 3.1.2. Dokonajmy obcięcia na poziomie $a = 3$ i gęstość tej „ogonowej” zmiennej losowej skoncentrowanej na sumie przedziałów $(-\infty, -3]$ i $[3, +\infty)$, oznaczmy przez f_4 . Uwzględniając wzór (3.16) dla $K = 4$, otrzymamy $p_4 = P\{|N(0,1)| > 3\} = 0.002699796063\dots$

Po dokonaniu obcięcia pozostaje część rozkładu normalnego, skupiona na przedziale $(-3,3)$. Krzywa gęstości $\varphi(x)$ tego rozkładu, o punktach przegięcia $x = -1$ i $x = 1$, wklęsła na odcinku $(-1,1)$ i wypukła poza nim, wydaje się być całkiem dobrze przybliżana za

pomocą odpowiednich fragmentów paraboli drugiego stopnia:

$$f_1(x) = \begin{cases} (3 - x^2)/8 & \text{dla } |x| < 1 \\ (3 - |x|^2)/16 & \text{dla } 1 \leq |x| \leq 3 \\ 0 & \text{poza tym} \end{cases}$$

Aby wykorzystać najbardziej efektywnie tę prostą aproksymację, wybierzemy maksymalne p_1 , dla którego

$$\varphi(x) - p_1 f_1(x) \geq 0 \quad \text{dla wszystkich } |x| < 3$$

Otrzymamy $p_1 = 16/\sqrt{2\pi e} = 0,86385546$ Oznacza to - tu powołujemy się znowu na dekompozycję (3.16) - że z pokaznym prawdopodobieństwem p_1 będziemy dokonywali losowania zmiennej losowej X z gęstością $f_1(x)$. Zauważmy, że jest to gęstość rozkładu sumy $V_1 + V_2 + V_3$, gdzie V_1, V_2, V_3 są niezależnymi zmiennymi losowymi o jednakowym rozkładzie równomiernym $U(-1,1)$, a więc z dużym prawdopodobieństwem p_1 generowanie zmiennej losowej X o rozkładzie normalnym $N(0,1)$ redukuje się do wygenerowania trzech niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym $U(-1,1)$ i dodania ich do siebie.

Do dalszej dekompozycji pozostaje funkcja $\varphi - p_1 f_1(x)$, którą - jak się okazuje - można dobrze przybliżyć za pomocą gęstości

$$f_2(x) = \begin{cases} \frac{4}{9} \left(\frac{3}{2} - |x| \right) & \text{dla } |x| \leq \frac{3}{2} \\ 0 & \text{poza tym} \end{cases}$$

Jest to gęstość rozkładu zmiennej losowej $3(U_1 + U_2 - 1)/2$, gdzie U_1 i U_2 są niezależnymi zmiennymi losowymi o jednakowym rozkładzie $U(0, 1)$. Wyznaczając maksymalne p_2 , dla którego

$$\varphi(x) - p_1 f_1(x) \geq 0 \quad \text{dla wszystkich } |x| < 3$$

otrzymujemy $p_2 = 0.1108179673\dots$ Z pozostałym prawdopodobieństwem $p_3 = 1 - p_1 - p_2 - p_4 = 0.02262677245\dots$ trzeba będzie losować X według rozkładu o gęstości

$$f_3 = \frac{1}{p_3} (\varphi - p_1 f_1 - p_2 f_2 - p_4 f_4)$$

Gęstość ta wyraża się wzorem

$$f_3(x) = \begin{cases} c_1 e^{-x^2/2} - c_2(3-x^2) - c_3\left(\frac{3}{2} - |x|\right) & \text{dla } |x| \leq 1 \\ c_1 e^{-x^2/2} - c_4(3-|x|)^2 - c_3\left(\frac{3}{2} - |x|\right) & \text{dla } 1 < |x| \leq \frac{3}{2} \\ c_1 e^{-x^2/2} - c_4(3-|x|)^2 & \text{dla } \frac{3}{2} < |x| \leq 3 \\ 0 & \text{poza tym} \end{cases}$$

gdzie

$$c_1 = 17.49731196, \quad c_2 = 4.73570326, \quad c_3 = 2.15787544, \quad c_4 = 2.36785163$$

Gęstość $f_3(x)$ skupiona na przedziale $(-3,3)$ i jest ograniczona przez

$$M = \max f_3(x) = 0.357070192\dots$$

a losowanie zmiennej losowej X o tej gęstości można wykonać prostą metodą eliminacji z gęstością rozkładu równomiernego na tym przedziale jako gęstością dominującą. Wymaga to pewnej liczby rachunków, ale odbywa się dość rzadko. (Odnotujmy, że Devroye (1986) podaje inną wartość dla M , ale cytując na s. 390 implementację Kindermana-Ramage'a podaje stałe, które pochodzą raczej od liczby M podanej tutaj przez nas, a nie przez niego.)

Oto zapowiadana implementacja Kindermana i Ramage'a tego algorytmu. Zauważmy, że w tej implementacji liczba losowa U , wygenerowana w pierwszym kroku algorytmu i służąca do wyboru odpowiedniego przypadku, jest później wykorzystywana w trakcie obliczeń na dalszych etapach tego algorytmu.

ALGORYTM 3.31 (Marsaglia-Bray)

Generuj U o rozkładzie równomiernym $U(0,1)$

PRZYPADEK $0 \leq U \leq p_1$

Generuj V i W o rozkładzie równomiernym $U(1,1)$

$$\text{Return } X = \frac{2U}{p_1} - 1 + V + W$$

PRZYPADEK $p_1 < U \leq p_1 + p_2$

Generuj V o rozkładzie równomiernym $U(0,1)$

$$\text{Return } X = \frac{3}{2} \left(\frac{U - p_1}{p_2} - 1 + V \right)$$

PRZYPADEK $1 - p_4 < U \leq 1$

Repeat

Generuj V i W o rozkładzie równomiernym $U(0,1)$

$$X = \frac{9}{2} - \ln W$$

$$\text{until } XV^2 \leq \frac{9}{2}$$

$$\text{Return } X = \sqrt{2X} \cdot \text{sign}\left(U - \left(1 - \frac{p_4}{2}\right)\right)$$

PRZYPADEK $p_1 + p_2 < U \leq 1 - p_4$

Repeat

Generuj X o rozkładzie równomiernym $U(-3, 3)$

Generuj U o rozkładzie równomiernym $U(0, 1)$

$V = |X|$

$$W = \frac{c_4}{M}(3 - V)^2$$

$S = 0$

$$\text{If } V < \frac{3}{2} \text{ then } S = \frac{c_3}{M}\left(\frac{3}{2} - V\right)$$

$$\text{If } V < 1 \quad \text{then } S = S + \frac{c_2}{M}(3 - V^2) - W$$

$$\text{until } U \leq \frac{c_1}{M}e^{-V^2/2} - S - W$$

Return X

3.2.4. Rozkład gamma

Gęstość rozkładu gamma $\Gamma(\alpha, \lambda)$ wyraża się wzorem

$$f_{\alpha\lambda}(x) = \frac{1}{\Gamma(\alpha)\lambda^\alpha} x^{\alpha-1} e^{-x/\lambda}, \quad 0 < x < \infty \quad (3.32)$$

gdzie $\alpha > 0$ oraz $\lambda > 0$ są parametrami rozkładu.

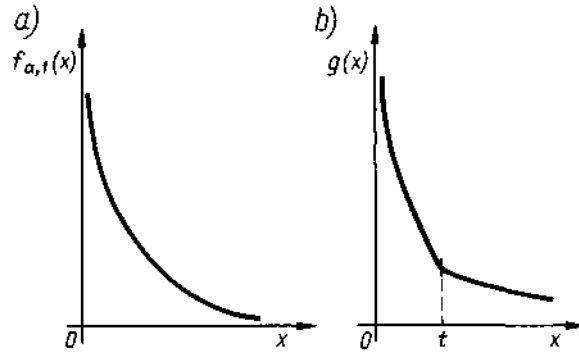
Jeżeli zmienna X ma rozkład $\Gamma(\alpha, \lambda)$ to zmienna losowa X/λ ma rozkład $\Gamma(\alpha, 1)$, więc dalej będziemy zajmowali się tylko zmiennymi losowymi o rozkładzie gamma $\Gamma(\alpha, 1)$.

Ze względu na liczne i bardzo różnorodne zastosowania rozkładu gamma opracowano wiele różnych algorytmów generowania liczb losowych o tym rozkładzie. Współczesny przegląd tych algorytmów oraz wyniki komputerowych studiów porównawczych można znaleźć w pracy Szczuki i Zielińskiego (1993). Tutaj ograniczamy się do przedstawienia metod i algorytmów, które - według naszego doświadczenia - okazały się stosunkowo proste (więc łatwe do implementacji) i zarazem zadowolająco szybkie.

Rozkład gamma z $\alpha = 1$ jest przypadkiem rozkładu wykładniczego, którym zajmowaliśmy się obszernie w p. 3.2.2.

Rozkład $\Gamma(\alpha)$ w przypadku $\alpha < 1$

Gdy $\alpha < 1$, wówczas gęstość rozkładu gamma ma kształt przedstawiony na rys. 3.6a. Zastosujemy w tym przypadku metodę eliminacji z gęstościami dominującymi, takimi jak gęstość na rys. 3.6b. Odpowiedni algorytm, opracowany przez Ahrensa i Dietera (1974) oraz Besta (1983), jest znany jako algorytm RGS. Oto szczegóły z nim związane.



Gęstość rozkładu gamma $\Gamma(\alpha - 1)$ spełnia następujące nierówności:

$$f_{\alpha, t}(x) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} \leq \begin{cases} \frac{1}{\Gamma(\alpha)} x^{\alpha-1}, & \text{gdy } x > 0 \\ \frac{1}{\Gamma(\alpha)} t^{\alpha-1} e^{-x}, & \text{gdy } x \geq t \end{cases}$$

co zapisujemy w postaci

$$f_{\alpha, t}(x) \leq g_0(x)$$

gdzie

$$g_0(x) = \frac{1}{\Gamma(\alpha)} (x^{\alpha-1} 1_{(0,t)}(x) + t^{\alpha-1} e^{-x} 1_{(t,+\infty)}(x))$$

Pole $s(t)$ powierzchni pod krzywą gęstości dominującej wyraża się wzorem

$$s(t) = \frac{1}{\Gamma(\alpha)} \left(\frac{t^\alpha}{\alpha} + t^{\alpha-1} e^{-t} \right)$$

Będzie ono najmniejsze, gdy wybierzemy t takie, żeby $\frac{d}{dt} s(t) = 0$, czyli tak, żeby $t + (\alpha - 1 - t)e^{-t} = 0$. W pracy na temat tego algorytmu Best (1983) wynalazł przybliżenie $t \approx 0.07 + 0.75\sqrt{1 - \alpha}$. Gęstość dominująca przyjmuje postać

$$g(x) = \frac{1}{s(t)} g_0(x) = \frac{\alpha}{t + \alpha e^{-t}} \left(\left(\frac{x}{t} \right)^{\alpha-1} 1_{(0,t)}(x) + e^{-x} 1_{(t,+\infty)}(x) \right) \quad (3.34)$$

Otrzymujemy algorytm ALGORYTM 3.32

Repeat

Generuj X o rozkładzie z gęstością $g(x)$

Generuj U o rozkładzie równomiernym $U(0,1)$

until

$$Ug_0(X) \leq f_{a,1}(X)$$

Return X

Generowanie zmiennej losowej X o rozkładzie z gęstością dominującą (3.34) można wykonać metodą superpozycji rozkładów, generując z prawdopodobieństwem

$$p = \frac{\alpha}{t + \alpha e^{-t}} \int_0^t g(x) dx = \frac{t}{t + \alpha e^{-t}}$$

punkt w przedziale $(0, t)$ o rozkładzie z gęstością proporcjonalną do $x^{\alpha-1}$ oraz z prawdopodobieństwem $1-p$ punkt w przedziale $(t, +\infty)$ o rozkładzie z gęstością proporcjonalną do e^{-x} .

Dystrybuanta rozkładu skoncentrowanego na przedziale $(0, t)$ i o gęstości proporcjonalnej do $x^{\alpha-1}$ (oznaczymy tę dystrybuantę przez G_1) wyraża się wzorem $G_1(x) = (x/t)^\alpha$. Ponieważ $G_1^{-1}(u) = tu^{1/\alpha}$, więc otrzymujemy algorytm: generuj U o rozkładzie równomiernym $U(0,1)$ i oblicz $X = tU^{1/\alpha}$

Dystrybuanta rozkładu skoncentrowanego na przedziale (t, ∞) i o gęstości proporcjonalnej do e^{-x} (oznaczymy tę dystrybuantę przez G_2) wyraża się wzorem $G_2(x) = 1 - e^{-(x-t)}$. Ponieważ $G_2^{-1}(u) = t - \ln(1-u)$, więc otrzymujemy algorytm: generuj U o rozkładzie równomiernym $U(0,1)$ i oblicz $X = t - \ln U$. Fragment

Generuj X o rozkładzie z gęstością $g(x)$

algorytmu 3.32 rozwija się teraz do postaci następującej:

Generuj U o rozkładzie równomiernym $U(0,1)$

If $U \leq p$ then Generuj X o rozkładzie z dystrybuantą $G_1(x)$

else Generuj X o rozkładzie z dystrybuantą $G_2(x)$

Liczbę losową U , za pomocą której rozstrzygaliśmy, według którego rozkładu generować X , możemy powtórnie wykorzystać, tym razem do generowania zmiennej losowej X . Jeżeli $U \leq p$, to U/p ma rozkład równomierny $U(0,1)$ i dla zmiennej losowej X otrzymujemy wzór: $X = t(U/p)^{1/\alpha}$. Jeżeli $U > p$, to $(U - p)/(1 - p)$ ma rozkład równomierny $U(0,1)$ i dla zmiennej

losowej X otrzymujemy wzór: $X = t - \ln\left(1 - \frac{U - p}{1 - p}\right)$ Po wprowadzeniu oznaczeń

$$b = \frac{1}{p} = 1 + \frac{\alpha e_t}{t}, \quad c = \frac{1}{\alpha}$$

mamy $X = -\ln(ct(b-bU))$. Prowadzi to do algorytmu

ALGORYTM 3.33

Repeat

Generuj U o rozkładzie równomiernym $U(0,1)$

Generuj W o rozkładzie równomiernym $U(0,1)$

$V = bU$

If $V \leq 1$ then

$X = tV^c$

```

    Akceptuj = (W ≤ e-X)
  Ele
    X = -ln(ct(b-V))
    akceptuj =  $\left( W \leq \left( \frac{X}{t} \right)^{\alpha-1} \right)$ 
  until akceptuj
  Return X

```

Możemy jeszcze dokonać drobnych ulepszeń, powodujących przyspieszenie warunków akceptacji (por. s. 54). Możemy to zrobić, np. korzystając z nierówności

$$b = \frac{1}{p} = 1 + \frac{\alpha e_t}{t}, \quad c = \frac{1}{\alpha}$$

w celu przyspieszenia warunku $W \leq e^{-X}$ oraz z nierówności

$$\frac{1}{1+cx} \leq (1+x)^{-c}, \quad x \geq 0 \quad c \in [0,1]$$

w celu przyspieszenia warunku $W \leq (X/t)^{\alpha-1}$. Ostatecznie dla przypadku $\alpha < 1$ możemy zaproponować następujący algorytm:

ALGORYTM 3.34

```

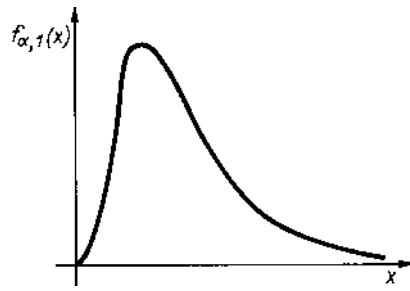
Repeat
  Generuj U o rozkładzie równomiernym U(0,1)
  Generuj W o rozkładzie równomiernym U(0,1)
  V = bU
  If V ≤ 1 then
    X = tVc
    akceptuj =  $\left( W \leq \frac{2-X}{2+X} \right)$ 
    if akceptuj = false then akceptuj = (W ≤ e-X)
  else
    X = -ln(ct(b-V))
    Y = X/t
    akceptuj = (W(α+Y-αY) ≤ 1)
    If akceptuj = false then akceptuj =  $\left( W \leq \left( \frac{X}{t} \right)^{\alpha-1} \right)$ 
until akceptuj
Return X

```

Rozkład $\Gamma(\alpha)$ w przypadku $\alpha > 1$

Gdy $\alpha > 1$, wówczas gęstość rozkładu gamma ma kształt przedstawiony na rys.3.7. W tym przypadku proponujemy metodę eliminacji z gęstością dominującą

$$g_{\lambda, \mu}(x) = \lambda \mu \frac{x^{\lambda-1}}{(\mu + x^{\lambda})^2}, \quad \lambda, \mu > 0, \quad x \geq 0 \quad (3.35)$$



Rys. 3.7

Jest to rozkład XII z rodziny rozkładów Burra (Burr (1942), patrz też Devroye (1986))-Dystrybuanta i funkcja odwrotna do dystrybuanty tego rozkładu wyrażają się wzorami

$$G_{\lambda, \mu}(x) = \frac{x^\lambda}{\mu + x^\lambda}, \quad G_{\lambda, \mu}^{-1}(u) = \left(\mu \frac{u}{1-u} \right)^{1/\lambda} \quad (3.36)$$

co pozwala na łatwe generowanie zmiennej losowej o tym rozkładzie metodą odwracania dystrybuanty. Pewną trudność może stanowić odpowiedni wybór parametrów rozkładu Burra: wybór ten powinien być na tyle prosty i uniwersalny, żeby algorytm generowania zmiennej losowej $\Gamma(\alpha)$, który przy każdej nowej wartości α musi dokonywać tego wyboru, robił to szybko i dokładnie. Pewne rozumowania heurystyczne (Cheng (1977), Devroye (1986)), uwzględniające punkty położenia maksimumów funkcji gęstości obu rozkładów oraz maksimum ilorazu tych funkcji, prowadzą do propozycji

$$\lambda = \sqrt{2\alpha - 1}, \quad \mu = \alpha^\mu \quad (3.37)$$

Optymalna stała c w metodzie eliminacji przyjmuje wtedy postać

$$c = \frac{4\alpha^\alpha e^{-\alpha}}{\lambda \Gamma(\alpha)} \quad (3.38)$$

Generowanie zmiennej losowej X o rozkładzie z dystrybuantą (3-36) zrealizujemy w następujący sposób:

Generuj U_1 o rozkładzie równomiernym $U(0, 1)$

$$V = \frac{1}{\lambda} \ln \frac{U_1}{1 - U_1}$$

$$X = \alpha e^V$$

Z kolei warunek akceptacji, przy użyciu zmiennej losowej U_2 o rozkładzie równomiernym $U(0,1)$, niezależnej od zmiennej losowej U_1 , przyjmuje postać

$$cU_2 g_{\lambda, \mu}(X) \leq \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x}$$

Korzystając z (3.37) dla μ oraz ze wzoru (3.36) dla dystrybuanty, otrzymujemy

$$4U_2 \left(\frac{\alpha}{e} \right)^\alpha \left(\frac{\alpha^\lambda}{X^{\lambda+1}} \right) G_{\lambda, \mu}^2(X) < x^{\alpha-1} e^{-X}$$

Ze sposobu, w jaki generowaliśmy zmienną losową X wynika, że $G_{\lambda, \mu}(X) = U_1$ oraz $\ln(X/\alpha) = V$, więc po zlogarytmowaniu tej nierówności stronami otrzymujemy warunek

$$\ln Z \leq R$$

gdzie

$$Z = U_1^2 U_2, \quad R = b + dV - X$$

przy czym stałe

$$b = \alpha - \ln 4, \quad d = \alpha + \lambda = \alpha + \sqrt{2\alpha - 1}$$

mogą być obliczone przy inicjowaniu algorytmu.

Warunek akceptacji (3.39) można przyspieszyć, poprzedzając go warunkiem

$$\gamma t - \ln \gamma - 1 \leq R$$

co wynika z nierówności

$$\ln t \leq \gamma t - \ln \gamma - 1, \quad \gamma > 0$$

która jest prawdziwa dla każdego dodatniego γ . W oryginalnej propozycji Chenga (1977) jest $\gamma = 9/2$. Ostatecznie otrzymujemy następujący algorytm:

ALGORYTM (3.35)

$$d = \frac{1}{\sqrt{2\alpha - 1}}, \quad c = \alpha + \frac{1}{d}, \quad b = \alpha - \ln 4$$

Repeat

Generuj U_1 o rozkładzie równomiernym $U(0, 1)$

Generuj U_2 o rozkładzie równomiernym $U(0, 1)$

$$V = d \cdot \ln \frac{U_1}{1 - U_1}$$

$$X = \alpha e^V, Z = U_2^2 U_1, R = b + cV - X$$

if akceptuj = false then akceptuj = (R ≥ ln Z)

until akceptuj

Return X

Jeżeli zmienne losowe X_1, \dots, X_n są niezależne i zmienna losowa X_i ma rozkład $\Gamma(\alpha_i)$, to zmienna losowa $X = \sum_{i=1}^n X_i$ ma rozkład gamma $\Gamma(\alpha)$, gdzie $\alpha = \sum_{i=1}^n \alpha_i$. Ten fakt może być wykorzystany w następujący sposób: jeżeli $\alpha > 1$, to zmienną losową o tym rozkładzie można wygenerować jako sumę $[\alpha]$ niezależnych zmiennych losowych o rozkładzie wykładniczym $E(0, 1)$ i jednej zmiennej losowej o rozkładzie $\Gamma(\{\alpha\})$.

3.2.5. Rozkład beta

Gęstość prawdopodobieństwa rozkładu beta $B(a, b)$ wyraża się wzorem

$$f_{a,b}(x) = \frac{1}{B(a,b)} x^{a-1} (1-x)^{b-1}, \quad 0 \leq x \leq 1$$

gdzie $B(a,b) = \Gamma(a) \Gamma(b) / \Gamma(a+b)$ oraz $a > 0$ i $b > 0$ są parametrami rozkładu. Zwracamy uwagę na to, że zarówno rozkład $B(a, b)$, stałą $B(a, b)$, jak i czasami samą zmienną losową o tym rozkładzie, oznaczamy takim samym symbolem; z kontekstu zawsze wiadomo o co chodzi, a unikamy w ten sposób nadmiaru oznaczeń.

Jeżeli zmienna losowa X ma rozkład beta $B(a,b)$, to zmienna losowa $1-X$ ma rozkład beta $B(b, a)$, więc będziemy rozważali tylko przypadek, gdy $a \leq b$.

Trzy najprostsze algorytmy generowania zmiennych losowych o rozkładach beta są oparte na trzech następujących lematach:

LEMAT 3.11 (Przypadek całkowitych a oraz b). Statystyka pozycyjna $U_{k:n}$ z n -elementowej próby z rozkładu równomiernego $U(0,1)$ ma rozkład beta $B(k, n-k+1)$

LEMAT 3.12. Jeżeli $\Gamma(a)$ oraz $\Gamma(b)$ są dwiema niezależnymi zmiennymi losowymi o rozkładach gamma z parametrami odpowiednio równymi a i b , to zmienna losowa $\Gamma(a) / (\Gamma(a) + \Gamma(b))$ ma rozkład beta $B(a, b)$.

LEMAT 3.13 (Algorytm Jdhka (1964)). Jeżeli U, V są niezależnymi zmiennymi losowymi o jednakowym rozkładzie równomiernym $U(0,1)$ oraz a, b są dodatnimi stałymi, to rozkład warunkowy zmiennej losowej

$$\frac{U^{1/a}}{U^{1/a} + V^{1/b}}$$

przy warunku $\{U^{1/a} + V^{1/b} \leq 1\}$, jest rozkładem beta $B(a,b)$.

Ze względu na czas generowania liczb losowych, pierwszy z tych algorytmów może konkurować z dwoma pozostałymi tylko w przypadku małych całkowitych wartości a oraz b , a drugi tylko wtedy, gdy dysponujemy szybkimi generatorami liczb losowych o rozkładach gamma. Algorytm Jdhka jest efektywny w przypadku małych wartości parametrów a i b :

prawdopodobieństwo spełnienia podanego w nim warunku jest równe $\frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+1)}$

W literaturze można znaleźć kilkanaście szybkich i łatwych do implementacji algorytmów rozkładu beta. Ze względu na metodykę konstrukcji, warto zwrócić uwagę na algorytm zaproponowany przez Chenga (1978), oparty na następującym lemacie:

LEMAT 3.14 Jeżeli zmienna losowa Y ma rozkład beta H rodzaju o gęstości

$$\varphi_{a,b}(y) = \frac{y^{a-1}}{B(a,b)(1+y)^{a+b}}, \quad y > 0 \quad (3.41)$$

to zmienna losowa $X = Y/(1+Y)$ ma rozkład beta $B(a, b)$.

Zmienną losową Y o rozkładzie (3.41) można generować metodą eliminacji z gęstością

dominującą

$$g_{\lambda, \mu}(y) = \frac{\lambda \mu y^{\lambda-1}}{(\mu + y^\lambda)^2}, \quad y > 0$$

Odpowiednia dystrybuanta ma postać

$$G_{\lambda, \mu}(y) = \frac{y^\lambda}{\mu + y^\lambda}, \quad y > 0$$

Z kolei zmienną losową Y o takiej dystrybuancie łatwo otrzymuje się metodą odwracania dystrybuanty. Standardowy algorytm przyjmuje postać: generować Y o rozkładzie z dystrybuanta $G_{\lambda, \mu}(y)$ i zmienną losową U o rozkładzie równomiernym $U(0, 1)$ dopóki, dopóki nie zostanie spełniony warunek $\{cUg_{\lambda, \mu}(Y) \leq \varphi_{a,b}(Y)\}$ i obliczyć $X = Y/(1+Y)$. Stała c powinna oczywiście zależeć od λ oraz μ , a te parametry chcielibyśmy wybrać tak, żeby ta stała była jak najmniejsza. Wybór takich optymalnych λ i μ zależy od parametrów a i b rozkładu generowanej zmiennej losowej, ale - jak zwykle - wolimy tu zdecydować się na jakieś proste, uniwersalne (dla dużego zakresu wartości parametrów a i b) i łatwe do zaprogramowania procedury wyznaczania λ oraz μ , niż na rozwiązywanie odpowiedniego zadania optymalizacji przy każdym odwołaniu się do generatora. Oryginalną propozycją Chenga jest

$$\lambda = \begin{cases} a, & \text{gdzie } a \leq 1 \\ \sqrt{\frac{2ab - (a+b)}{a+b-2}}, & \text{gdzie } a > 1 \end{cases}$$

oraz $\mu = \left(\frac{a}{b}\right)^\lambda$

(Przypominamy że zgodnie z przyjętymi na początku p.3.2.5 oznaczeniami, $a \leq b$.) Wtedy

$$c = \frac{4a^a b^b}{\lambda B(a, b)(a+b)^{a+b}} \quad (3.42)$$

Po wykonaniu odpowiednich przekształceń otrzymujemy algorytm

ALGORYTM 3.36

```

a = a + b
If a ≤ 1 then β = a else β = √(2ab - a) / (a - 2)
γ = a + β
Repeat
  Generuj U1 o rozkładzie równomiernym U(0, 1)
  V = 1/β ln(U1 / (1 - U1))   W = aeV
  Generuj U2 o rozkładzie równomiernym U(0, 1)
until   α ln(α / (b + W)) + γV - ln 4 ≥ ln(U12U2)
Return X = Y / (b + Y)

```

Stalą (3.42.) nigdy nie przekracza liczby 4, a jeżeli $a \geq 1$, to ta stała jest nie większa od $4/e \approx 1.47$.

3.2.6. Rozkład Cauchy'ego

Gęstość prawdopodobieństwa rozkładu Cauchy'ego $C(\theta, \lambda)$ wyraża się wzorem

$$f_{\theta, \lambda}(x) = \frac{\lambda}{\pi} \frac{1}{\lambda^2 + (x - \theta)^2}, \quad -\infty < x < +\infty$$

Jeżeli zmienna losowa Y ma rozkład Cauchy'ego $C(\theta, \lambda)$ to zmienna losowa $X = (Y - \theta)/\lambda$ ma rozkład Cauchy'ego $C(0, 1)$ o gęstości

$$f(x) = \frac{1}{\pi} \frac{1}{1 + x^2}, \quad -\infty < x < +\infty \quad (3.43)$$

więc w dalszym ciągu będziemy zajmowali się tylko rozkładem $C(0,1)$ o gęstości (3.43).

Dystrybuanta rozkładu $C(0,1)$ ma postać

$$F(x) = \frac{1}{\pi} \operatorname{arctg}(x) + \frac{1}{2} \quad (3.44)$$

skąd wynika prosty sposób generowania liczb losowych o tym rozkładzie metodą odwracania dystrybuanty; $X = \operatorname{tg}(U\pi)$, gdzie U jest zmienną losową o rozkładzie równomiernym $U\left(-\frac{1}{2}, \frac{1}{2}\right)$.

Przedstawimy szybki i elegancki algorytm generowania liczb losowych o rozkładzie Cauchy'ego $C(0,1)$ za pomocą pewnej kombinacji metody superpozycji rozkładów i metody eliminacji.

LEMAT 3.15. *Jeżeli zmienna losowa X ma ucięty rozkład Cauchy'ego $C(0,1)$ o gęstości*

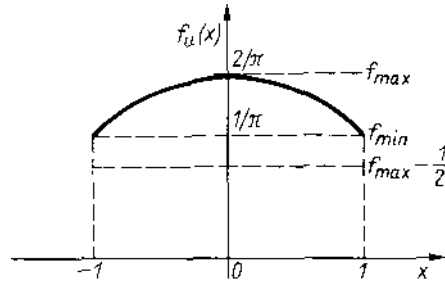
$$f_u(x) = \begin{cases} \frac{2}{\pi} \frac{1}{1+x^2}, & \text{gdy } -1 \leq x \leq 1 \\ 0, & \text{poza tym} \end{cases} \quad (3.45)$$

to zmienna losowa Y , która z prawdopodobieństwem $1/2$ jest identyczna ze zmienną losową X oraz z prawdopodobieństwem $1/2$ jest identyczna ze zmienną losową $1/X$, ma rozkład Cauchy'ego $C(0,1)$.

Dowód: Pamiętając o tym, że $\int_a^b dt/(1+t^2) = \operatorname{arctg}(b) - \operatorname{arctg}(a)$, dla zmiennej losowej Y i dla wartości argumentu $y \leq -1$ łatwo otrzymujemy:

$$\begin{aligned} P\{Y \leq y\} &= \frac{1}{2} P\{X \leq y\} + \frac{1}{2} P\left\{\frac{1}{X} \leq y\right\} = \\ &= 0 + \frac{1}{2} P\left\{\frac{1}{y} \leq X < 0\right\} = \frac{1}{2} \frac{2}{\pi} \int_{1/y}^0 \frac{dt}{1+t^2} = \\ &= \frac{1}{\pi} \operatorname{arctg} y + \frac{1}{2} \end{aligned}$$

czyli dystrybuantę (3.44). Podobnie dowodzimy lematu, gdy $y \in (-1, 1)$ oraz gdy $y \geq 1$.



Rys. 3.8

Zmienną losową X o rozkładzie z gęstością $f_u(x)$ wygenerujemy metodą AC (patrz algorytm 3.15), przyjmując za gęstość dominującą g gęstość rozkładu równomiernego $U(-1, 1)$ oraz za f_2 gęstość stałą (używamy tu oznaczeń jak w algorytmie 3.15). Ponieważ $g(x) = 1/2$, więc za f_1 przyjmujemy $f_u(x) - (f_{\max} - 1/2)$, gdzie $f_{\max} = \max f_u(x)$. Wtedy $f_2(x) = f_{\max} - 1/2$ (patrz rys. 3.8)

Otrzymujemy następujący algorytm:

ALGORYTM 3.37

Generuj X o rozkładzie równomiernym $U(-1, 1)$

Generuj U o rozkładzie równomiernym $U(-1, 1)$

If $\frac{1}{2}U > f_u(X) - \left(f_{\max} - \frac{1}{2}\right)$

then Generuj X o rozkładzie równomiernym $U(-1, 1)$

Return X

Ponieważ $f_{\max} = 2/\pi$, więc dodatkowego generowania X według rozkładu równomiernego $U(-1, 1)$ przyjmuje postać $(U + 0.27324)(1 + X^2) > 1.27324$. ten warunek jest spełniony z prawdopodobieństwem $\int_{-1}^1 f_2(x) dx = 2(f_{\max} - 1/2) = 0.27324$

3.2.7. Rozkłady α -stabilne

Rozkłady α -stabilne tworzą obszerną klasę, która zawiera m.in. rozkład normalny ($\alpha = 2$) i rozkład Cauchy'ego ($\alpha = 1$). Kłopoty z generowaniem zmiennych losowych o takich rozkładach polegają przede wszystkim na tym, że poza wyjątkowymi sytuacjami nie jest znany wzór dla gęstości lub dystrybuanty tego rozkładu, a więc takie metody, jak metoda odwracania dystrybuanty lub metoda eliminacji, nie znajdują tutaj zastosowania.

Zmienna losowa X ma rozkład α -stabilny, $0 < \alpha < 2$. z parametrem skali $\sigma > 0$, z parametrem asymetrii $\beta \in [-1, 1]$ i z parametrem położenia $\mu \in \mathbb{R}$, jeżeli jej funkcja charakterystyczna $\varphi(t)$ jest określona wzorem

$$\ln \varphi(t) = \begin{cases} -\sigma^\alpha |t|^\alpha \left(1 - i\beta \operatorname{sign} t \operatorname{tg} \frac{\pi \alpha}{2}\right) + i\mu t, & \text{gdy } \alpha \neq 1 \\ -\sigma |t| \left(1 + i\beta \operatorname{sign} t \frac{2}{\pi} \ln |t|\right) + i\mu t & \text{gdy } \alpha = 1 \end{cases}$$

Ten rozkład oznaczamy przez $S_\alpha(\sigma, \beta, \mu)$. Jeżeli zmienna losowa X ma rozkład $S_\alpha(1, \beta, 0)$, to zmienna losowa Y określona wzorem

$$Y = \begin{cases} \sigma X + \mu, & \text{gdy } \alpha \neq 1 \\ \sigma X + \frac{2}{\pi} \beta \sigma \ln \sigma + \mu, & \text{gdy } \alpha = 1 \end{cases}$$

ma rozkład $S_\alpha(\sigma, \beta, \mu)$, wystarczy więc skonstruowanie algorytmu generowania zmiennej losowej X o rozkładzie α -stabilnym $X_\alpha(1, \beta, 0)$. Można to zrobić w następujący sposób, podany po raz pierwszy w pracy Chambersa, Mallowsa i Stucka (1976): wygenerować zmienną losową U o rozkładzie równomiernym $U(-\pi/2, \pi/2)$ oraz niezależnie zmienną losową W o rozkładzie wykładniczym $E(0, 1)$, a następnie:

1) jeżeli $\alpha = 1$, obliczyć

$$X = \frac{2}{\pi} \left(\left(\frac{\pi}{2} + \beta U \right) \operatorname{tg} U - \beta \ln \frac{\frac{\pi}{2} W \cos U}{\frac{\pi}{2} + \beta U} \right),$$

1) jeżeli $\alpha \neq 1$, obliczyć

$$X = S_{\alpha, \beta} \cdot \frac{\sin \alpha (U + B_{\alpha, \beta})}{(\cos U)^{1/\alpha}} \cdot \left(\frac{\cos(U - \alpha (U + B_{\alpha, \beta}))}{W} \right)^{(1-\alpha)/\alpha}$$

gdzie

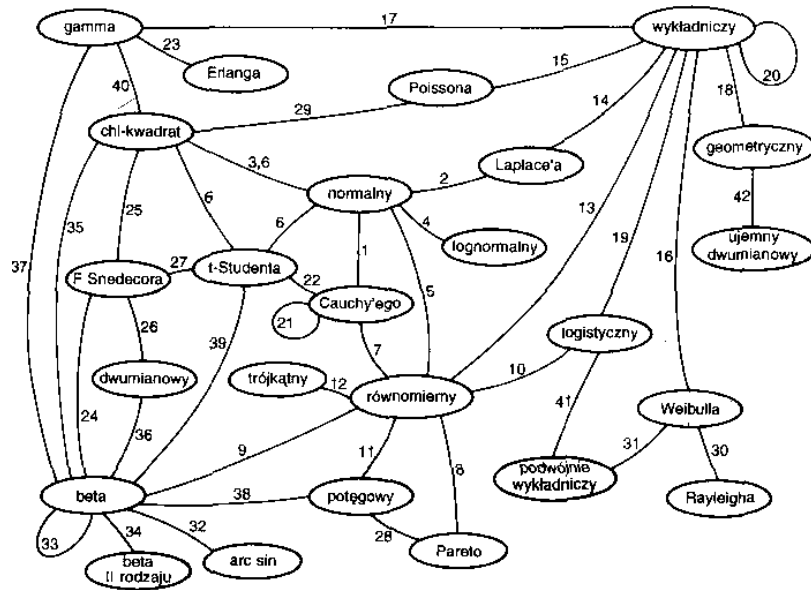
$$B_{\alpha, \beta} = \alpha^{-1} \operatorname{arctg} \left(\beta \operatorname{tg} \frac{\pi \alpha}{2} \right), \quad S_{\alpha, \beta} = \left(1 + \beta^2 \operatorname{tg}^2 \frac{\pi \alpha}{2} \right)^{1/(2\alpha)}$$

Nie podajemy tu żadnych dowodów; zwięzły wykład na ten temat oraz bardziej specjalistyczną bibliografię można znaleźć w książce Janickiego i Werona (1994) oraz w pracach Werona (1996a, 1996b).

3.3. Związki między rozkładami

Podstawą konstrukcji generatorów liczb losowych o zadanych rozkładach prawdopodobieństwa są związki teoretyczne między różnymi zmiennymi losowymi. W bieżącym podrozdziale zamieszczamy listę takich związków. Odnalezienie informacji na temat konkretnych, interesujących Czytelnika rozkładów, ułatwi schemat graficzny przedstawiony na rys. 3.9

1. **Rozkład normalny i rozkład Cauchy'ego.** Jeżeli zmienne X i Y są niezależnymi zmiennymi losowymi o rozkładzie normalnym $N(0, 1)$, to zmienna losowa $Z = X/Y$ ma rozkład Cauchy'ego $C(0, 1)$.
2. **Rozkład normalny i rozkład Laplace'a.** Jeżeli X_1, X_2, X_3, X_4 są niezależnymi zmiennymi losowymi o rozkładzie normalnym $N(0, 1)$, to zmienna $X_1 X_4 - X_2 X_3$ ma rozkład Laplace'a z gęstością $f(x) = \frac{1}{4} \exp\left(-\frac{1}{2}|x - \theta|\right)$.
3. **Rozkład normalny i rozkład chi-kwadrat.** Jeżeli X_1, \dots, X_n są niezależnymi zmiennymi losowymi o jednakowym rozkładzie normalnym $N(0, 1)$, to zmienna losowa $\sum_{i=1}^n X_i^2$ ma rozkład chi-kwadrat X_n^2 o n stopniach swobody.
4. **Rozkład normalny i rozkład lognormalny.** Jeżeli zmienna losowa X ma rozkład normalny $N(\mu, \sigma)$, to zmienna losowa $Y = e^X$ ma rozkład lognormalny z parametrami (μ, σ) .



Rys. 3.9

5. Rozkład normalny i rozkład równomierny.

(a) Jeżeli X i Y są niezależnymi zmiennymi losowymi o rozkładzie równomiernym na przedziale $(0,1)$, to $U = \sqrt{-2 \ln X} \cdot \cos(2\pi Y)$, $V = \sqrt{-2 \ln X} \cdot \sin(2\pi Y)$ są niezależnymi zmiennymi losowymi o rozkładach normalnych $N(0,1)$.

(b) Jeżeli X i Y są niezależnymi zmiennymi losowymi o rozkładzie normalnym $N(0,1)$, to zmienne losowe $U = X^2 + Y^2$, $V = \arcsin\left(X / \sqrt{X^2 + Y^2}\right)$ są niezależne, U ma rozkład wykładniczy $E(0,2)$, V ma rozkład równomierny na przedziale $(-\pi/2, \pi/2)$.

6. Rozkład normalny, rozkład chi-kwadrat i rozkład t-Studenta.

Jeżeli zmienna losowa X ma rozkład normalny $N(0,1)$, zmienna losowa Y ma rozkład chi-kwadrat o n stopniach swobody i te zmienne są niezależne, to zmienna losowa $t = X / \sqrt{Y/n}$ ma rozkład Studenta (rozkład t-Studenta) o n stopniach swobody.

7. **Rozkład równomierny i rozkład Cauchy'ego.** Jeżeli zmienna losowa X ma rozkład równomierny na przedziale $(-\pi/2, \pi/2)$, to zmienna losowa $Y = tgX$ ma rozkład Cauchy'ego $C(0,1)$.

8. **Rozkład równomierny i rozkład Pareto.** Jeżeli zmienna losowa U ma rozkład równomierny $U(0,1)$, to zmienna losowa $X = bU^{-1/\alpha}$ ma rozkład Pareto o dystrybuancie $F_{a,b}(x) = 1 - (b/x)^a, x \geq b$.

9. **Rozkład równomierny i rozkład beta.** (a) Jeżeli X_1, \dots, X_n są niezależnymi zmiennymi losowymi o rozkładzie równomiernym na przedziale $(0, 1)$, to k -ta statystyka pozycyjna $X_{k:n}$ ma rozkład beta $B(k, n - k + 1)$, a zmienna losowa $X_{l:n} - X_{k:n}$ (dla $l > k$) ma rozkład beta $B(l - k, n - l + k + 1)$.

(b) Jeżeli C, V są niezależnymi zmiennymi losowymi o jednakowym rozkładzie równomiernym $U(0, 1)$ oraz a, b są dodatnimi stałymi, to rozkład warunkowy zmiennej losowej

$$\frac{U^{1/a}}{U^{1/a} + V^{1/b}}$$

przy warunku $\{U^{1/a} + V^{1/b} \leq 1\}$, jest rozkładem beta $B(a,b)$.

10. Rozkład równomierny i rozkład logistyczny.

(a) Jeżeli zmienna losowa X ma rozkład równomierny na przedziale $(0,1)$, to zmienna losowa $Y = a - b \ln(1/X - 1)$ ma rozkład logistyczny $L(a, b)$ o gęstości

$$\frac{\exp(-(x-a)/b)}{b(1 + \exp(-(x-a)/b))^2}, \quad b > 0$$

(b) Jeżeli zmienna losowa U ma rozkład równomierny $U(0,1)$, to zmienna losowa $X = \ln(U/(1-U))$ ma rozkład logistyczny o dystrybuancie $F(x) = 1/(1+e^x)$.

11. Rozkład równomierny i rozkład potęgowy.

(a) Jeżeli X_1, \dots, X_k są niezależnymi zmiennymi losowymi o rozkładzie równomiernym na przedziale $(0,1)$, to $\max(X_1, \dots, X_k)$ jest zmienną losową o gęstości $f(x) = kx^{k-1}$, $0 \leq x \leq 1$.

(b) Jeżeli zmienna losowa X ma rozkład równomierny na przedziale $(0,1)$, to zmienna losowa $Y = X^{1/\alpha}$ ma rozkład potęgowy o gęstości $f(y) = \alpha y^{-\alpha-1}$, $0 \leq y \leq 1$, $\alpha > 0$.

12. Rozkład równomierny i rozkład trójkątny. Jeżeli X i Y są niezależnymi zmiennymi losowymi o rozkładzie równomiernym na przedziale $(-a/2, b/2)$, to zmienna losowa $Z = X+Y$ ma rozkład trójkątny na przedziale (a, b) o gęstości

$$f(x) = \frac{2}{b-a} - \frac{2}{(b-a)^2} |a+b-2x|, \quad x \in (a, b)$$

13. Rozkład równomierny i rozkład wykładniczy.

(a) Jeżeli zmienna losowa X ma rozkład równomierny na przedziale $(0,1)$, to zmienna losowa $Y = -\lambda \ln X$ ma rozkład wykładniczy $E(0, \lambda)$.

(b) Jeżeli X_1, \dots, X_n jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie wykładniczym $E(0,1)$, to

$$U_{i:n} = \exp\left(-\frac{X_1}{n} - \frac{X_2}{n-1} - \dots - \frac{X_{n-1+1}}{i}\right)$$

ma taki sam rozkład jak i -ta statystyka pozycyjna z próby U_1, \dots, U_n z rozkładu równomiernego $U(0, 1)$.

14. Rozkład wykładniczy i rozkład Laplace'a.

(a) Jeżeli X i Y są niezależnymi zmiennymi losowymi o rozkładzie wykładniczym $E(0, \lambda)$, to zmienna losowa $Z = X - Y + \alpha$ ma rozkład Laplace'a (podwójnie wykładniczy lub dwustronny rozkład wykładniczy) o gęstości

$$f(x) = (2\lambda)^{-1} \exp(-|x - \alpha|/\lambda), \quad x \in \mathbb{R}^1$$

(b) Jeżeli zmienna losowa X ma rozkład Laplace'a z parametrami (λ, α) , to zmienna losowa $Y = |X - \alpha|$ ma rozkład wykładniczy $E(0, \lambda)$.

15. **Rozkład wykładniczy i rozkład Poissona.** Jeżeli T_1, T_2, \dots są niezależnymi zmiennymi losowymi o rozkładzie wykładniczym $E(0, 1/\lambda)$ oznaczającymi kolejne chwile skoku procesu stochastycznego X_t , takiego że $X_0 = 0$, to proces ten jest jednorodnym procesem Poissona z intensywnością λ .

16. **Rozkład wykładniczy i rozkład Weibulla.** Jeżeli zmienna losowa X ma rozkład wykładniczy $E(0, \lambda)$, to zmienna losowa $Y = X^{1/\alpha}$, $\alpha, \lambda > 0$, ma rozkład Weibulla $W(\lambda, \alpha)$ o dystrybuancie $P\{Y \leq y\} = 1 - \exp(-y^\alpha/\lambda)$, $y > 0$.

17. **Rozkład wykładniczy i rozkład gamma.** Jeżeli X_1, \dots, X_n są niezależnymi zmiennymi losowymi o jednakowym rozkładzie wykładniczym $E(0, \lambda)$, to zmienna losowa $\sum_{i=1}^n X_i$ ma rozkład gamma $\Gamma(n, \lambda)$.

18. **Rozkład wykładniczy i rozkład geometryczny.** Jeżeli zmienna losowa X ma rozkład wykładniczy $E(0, \lambda)$, to zmienna losowa $Y = [X]$ ma rozkład geometryczny

$$P\{Y = y\} = (1 - p)p^y, \quad y = 0, 1, 2, \dots$$

z parametrem $p = \exp(-1/\lambda)$.

19. **Rozkład wykładniczy i rozkład logistyczny.** Jeżeli niezależne zmienne losowe X i Y mają rozkład wykładniczy $E(0, 1)$, to zmienna losowa $Z = \frac{1}{2} \ln(X/Y)$ ma rozkład logistyczny $L(0, 1)$.

20. **Rozkład wykładniczy i statystyki pozycyjne.** Jeżeli X_1, \dots, X_n są niezależnymi zmiennymi losowymi o rozkładzie wykładniczym $E(0, \lambda)$, to również $nX_{1:n}, (n-1)(X_{2:n} - X_{1:n}), (n-2)(X_{3:n} - X_{2:n}), \dots, X_{n:n} - X_{n-1:n}$ są niezależnymi zmiennymi losowymi o rozkładzie wykładniczym $E(0, \lambda)$.

21. **Rozkład Cauchy'ego i rozkład Cauchy'ego.**

(a) Jeżeli X jest zmienną losową o rozkładzie Cauchy'ego $C(0, 1)$, to zmienna losowa $1/X$ też ma rozkład Cauchy'ego $C(0, 1)$.

(b) Jeżeli X_1, \dots, X_n są niezależnymi zmiennymi losowymi o rozkładzie Cauchy'ego $C(0, 1)$, to zmienna losowa $(X_1 + X_2 + \dots + X_n)/n$ też ma rozkład Cauchy'ego $C(0, 1)$.

22. **Rozkład Cauchy'ego i rozkład t-Studenta.** Rozkład Cauchy'ego $C(0, 1)$ jest identyczny z rozkładem t-Studenta z jednym stopniem swobody.

23. **Rozkład Erlanga i rozkład gamma.** Suma n niezależnych zmiennych losowych o rozkładzie $\Gamma(1, \lambda)$ ma rozkład gamma $\Gamma(n, \lambda)$ (jest to tzw. *rozkład Erlanga*).

24. **Rozkład F Snedecora i rozkład beta.**

(a) Jeżeli X jest zmienną losową o rozkładzie F z (m, n) stopniami swobody, to zmienna losowa $Y = n/(mX + n)$ ma rozkład beta $B(n/2, m/2)$.

(b) Jeżeli $B(\alpha, \beta)$ jest zmienną losową o rozkładzie beta z parametrami α i β oraz $F_{m,n}$ jest zmienną losową o rozkładzie F z (m, n) stopniami swobody, to

$$P\left\{B(\alpha, \beta) \leq \frac{\alpha}{\alpha + \beta}\right\} = P\{F_{2\beta, 2\alpha} > x\}$$

25. **Rozkład F Snedecora i rozkład chi-kwadrat.** Jeżeli zmienna losowa X ma rozkład chi-kwadrat o m stopniach swobody, zmienna losowa Y ma rozkład chi-kwadrat o n

stopniach swobody i te zmienne są niezależne, to zmienna losowa $F = \frac{X/m}{Y/n}$ ma rozkład F Snedecora (krótko *rozkład F* , czasami używa się też nazwy *rozkład F Fishera*) o (m, n) stopniach swobody.

26. **Rozkład F Snedecora i rozkład dwumianowy.** Między dystrybuantami rozkładu dwumianowego z parametrami (n, p) i rozkładu F Snedecora zachodzi następujący związek

$$\sum_{j=0}^k \binom{n}{k} p^k (1-p)^{n-k} = P \left\{ F_{2(n-k), 2(k+1)} \leq \frac{k+1}{n-k} \frac{1-p}{p} \right\}$$

gdzie $F_{a,b}$ oznacza zmienną losową o rozkładzie F z (a, b) stopniami swobody.

27. **Rozkład F Snedecora i rozkład t -Studenta.** Jeżeli zmienna losowa X ma rozkład Studenta z n stopniami swobody, to zmienna losowa $Y = X^2$ ma rozkład F Snedecora o $(1, n)$ stopniach swobody.

28. **Rozkład Pareto i rozkład potęgowy.** Rozkład Pareto o gęstości $f(x) = (\alpha/x_0)(x_0/x)^{\alpha+1}$ dla $x \geq x_0$, $\alpha > 0$, jest rozkładem potęgowym o gęstości $f(x) = \alpha x^{-(\alpha+1)}$, $x > 0$, uciętym do przedziału (x_0, ∞) .

29. **Rozkład Poissona i rozkład chi-kwadrat.** Jeżeli X jest zmienną losową o rozkładzie Poissona z parametrem λ , to

$$P\{X \leq k\} = 1 - G_{2(k+1)}(2\lambda), \quad k = 0, 1, \dots$$

przy czym $G_m(x)$ jest wartością dystrybuanty rozkładu chi-kwadrat o m stopniach swobody w punkcie x .

30. **Rozkład Weibulla i rozkład Rayleigha.** Rozkład Weibulla $W(\lambda, 2)$ jest nazywany *rozkładem Rayleigha* (czyli \sqrt{X} ma rozkład Rayleigha, gdy X ma rozkład wykładniczy $E(0, \lambda)$).

31. **Rozkład Weibulla i rozkład podwójnie wykładniczy.** Jeżeli zmienna losowa X ma rozkład Weibulla $W(1, 1)$, to zmienna losowa $Y = -\ln X$ ma rozkład podwójnie wykładniczy (rozkład wartości ekstremalnych) o gęstości $f(y) = \exp(-\exp(-x))$, $x > 0$.

32. **Rozkład arcusa sinusa i rozkład beta.** Rozkład beta $B(1/2, 1/2)$ nazywa się *rozkładem arcusa sinusa*.

33. **Rozkład beta i rozkład beta.** Jeżeli zmienna losowa X ma rozkład beta $B(a, b)$, to zmienna losowa $1 - X$ ma rozkład beta $B(b, a)$.

34. **Rozkład beta i rozkład beta II rodzaju.** Jeżeli X jest zmienną losową o rozkładzie beta $B(a, b)$, to $Y = X/(1 - X)$ jest zmienną losową o rozkładzie beta II rodzaju, którego gęstość wyraża się wzorem

$$f(x) = \frac{x^{a-1}}{B(a, b)(1+x)^{a+b}}, \quad x > 0$$

35. **Rozkład beta i rozkład chi-kwadrat.** Jeżeli niezależne zmienne losowe X i Y mają odpowiednio rozkłady X_{2b}^2 i X_{2a}^2 zmienna losowa $Z = Y/(X + Y)$ ma rozkład beta $B(a, b)$.

36. **Rozkład beta i rozkład dwumianowy.** Dla zmiennej losowej X o rozkładzie dwumianowym z parametrami (n, p) zachodzi związek

$$P\{X \leq x\} = I_{1-p}(n-k, k+1) = 1 - I_p(k+1, n-k)$$

gdzie $I_x(a, b)$ oznacza wartość dystrybuanty rozkładu beta $B(a, b)$ w punkcie x (jest to tzw. niekompletna funkcja beta).

37. **Rozkład beta i rozkład gamma.** Jeżeli X i Y są niezależnymi zmiennymi losowymi o rozkładach gamma $\Gamma(a)$ i $\Gamma(b)$, to $Y = X/(X + Y)$ jest zmienną losową o rozkładzie beta $B(a, b)$.

38. **Rozkład beta i rozkład potęgowy.** Zmienna losowa o rozkładzie beta $B(\alpha, 1)$ jest zmienną o rozkładzie potęgowym z gęstością $f(x) = \alpha x^{\alpha-1}$, $x \in (0, 1)$

39. **Rozkład beta i rozkład t-Studenta.** Między dystrybuantą $S_n(x)$ rozkładu t-Studenta z n stopniami swobody i dystrybuantą $I_x(\alpha, \beta)$ rozkładu beta $B(\alpha, \beta)$ zachodzi związek

$$2S_n(|t|) - 1 = I_{t^2/(n+t^2)}(1/2, n/2)$$

40. Rozkład chi-kwadrat i rozkład gamma,

(a) Rozkład chi-kwadrat o n stopniach swobody jest rozkładem gamma $\Gamma(n/2, 2)$.

(b) Jeżeli X_1, \dots, X_n są niezależnymi zmiennymi losowymi o rozkładzie gamma $\Gamma(1, \lambda)$,

to zmienna losowa $T = (2\lambda)^{-1} \sum_{i=1}^n X_i$ ma rozkład chi-kwadrat o $2n$ stopniach swobody.

41. Rozkład logistyczny i rozkład podwójnie wykładniczy. Jeżeli niezależne zmienne losowe X i Y mają rozkład podwójnie wykładniczy, to zmienna losowa $X - Y$ ma rozkład logistyczny $L(0, 1)$.

42. Rozkład ujemny dwumianowy i rozkład geometryczny. Ujemny rozkład dwumianowy z parametrami (r, p) , określony wzorem

$$P\{X = k\} = \binom{r+k-1}{k} (1-p)^r p^k, \quad k = 0, 1, 2, \dots$$

jest rozkładem geometrycznym, jeśli przyjmiemy $r = 1$.

4. Generatory liczb losowych o rozkładach wielowymiarowych

4.1. Przypadek ogólny

Przyjmijmy, że m -wymiarowa zmienna losowa \mathbf{X} ma rozkład prawdopodobieństwa o

gęstości $f(x_1, \dots, x_m)$. Wszystko to co mówiliśmy ogólnie o metodzie eliminacji (p.3.1.2) oraz o metodzie superpozycji rozkładów (p.3.1.3) odnosi się również do generowania wielowymiarowej zmiennej losowej. Pojawiają się tu jednak większe trudności techniczne. W każdym konkretnym przypadku można starać się znaleźć odpowiednią gęstość dominującą lub poszukać reprezentacji

$$f(x_1, \dots, x_m) = \sum_{j=1}^k p_j f_j(x_1, \dots, x_m), p_j > 0, \sum_{j=1}^k 1$$

za pomocą prostszych gęstości $f_j(x_1, \dots, x_m)$, $j = 1, 2, \dots, k$ ale te prostsze gęstości wielowymiarowe wcale nie muszą być proste i nie muszą pojawiać się w tak naturalny sposób, jak to obserwowaliśmy w przypadku jednowymiarowym. Ponadto przy implementacji metody eliminacji (w postaci czystej lub w stosunku do gęstości $f_j(x_1, \dots, x_m)$), pojawia się pewien efekt, nazywany czasami „demonem wielowymiarowości”. Zilustrujemy to prostym przykładem zastosowania metody eliminacji do generowania zmiennej losowej \mathbf{X} o rozkładzie równomiernym na kuli jednostkowej $K_m(0,1)$ w przestrzeni m -wymiarowej. Przykład ten polega na generowaniu punktu $U = (U_1, \dots, U_m)$ o rozkładzie równomiernym na m -wymiarowej kostce $[-1,1]^m$ co sprowadza się do generowania m niezależnych zmiennych losowych U_1, U_2, \dots, U_m o rozkładach równomiernych na przedziale $[-1,1]$, i zaakceptowaniu U jako \mathbf{X} , gdy to U wpadnie do kuli jednostkowej $K_m(0,1)$. Prawdopodobieństwo p_m tego zdarzenia (tzn. prawdopodobieństwo akceptacji) jest oczywiście równe ilorazowi objętości kuli $K_m(0,1)$ i objętości opisanej na niej kostki $[-1, 1]^m$, a średnia liczba N_m punktów U , potrzebnych do uzyskania jednej realizacji zmiennej losowej \mathbf{X} , jest równa odwrotności tego prawdopodobieństwa. Mamy oczywiście $p_m = \pi^{m/2}/(2^m \Gamma(m/2+1))$, a odpowiednie liczby dla kilku wybranych wartości m podaje następująca tabelka:

m	p_m	N_m
2	$7.854 \cdot 10^{-1}$	1.27
5	$1.645 \cdot 10^{-1}$	6.08
10	$2.490 \cdot 10^{-3}$	$4.015 \cdot 10^2$
20	$2.461 \cdot 10^{-8}$	$4.063 \cdot 10^7$
50	$1.537 \cdot 10^{-28}$	$6.507 \cdot 10^{27}$

Inna metoda ogólna generowania m -wymiarowej zmiennej losowej, gdy $m > 1$, polega na przedstawieniu gęstości $f(x_1, \dots, x_m)$ w postaci iloczynu gęstości rozkładów brzegowych i warunkowych

$$f(x_1, \dots, x_m) = f(x_1) f(x_2 | x_1) f(x_3 | x_1, x_2) \dots f(x_n | x_1, x_2, \dots, x_{n-1}) \quad (4.1)$$

i generowaniu „współrzędna po współrzędnej”: współrzędnej X_i według rozkładu warunkowego $f(x_i | x_1, x_2, \dots, x_{i-1})$ po uprzednim wygenerowaniu współrzędnych X_1, \dots, X_{i-1}

Dla niektórych rozkładów wielowymiarowych znane są efektywne metody wykorzystujące specyficzne własności danych rozkładów; np. algorytm generowania m -wymiarowej zmiennej losowej o niezależnych składowych z jednakowymi rozkładami wykładniczymi przedstawiliśmy w p.3.2.2. W niniejszym rozdziale przedstawimy kilka zagadnień związanych z generowaniem wielowymiarowych rozkładów równomiernych (podrozdz. 4.2) oraz algorytm generowania

wielowymiarowego rozkładu normalnego o zadanej macierzy kowariancji (podrozdz. 4.3).

4.2. Rozkłady równomierne w R^m

4.2.1. Uwagi ogólne

Przyjmijmy, że Ω jest pewnym zbiorem w przestrzeni R^m i niech \mathbf{X} będzie zmienną losową o rozkładzie równomiernym na tym zbiorze. Przy prezentacji metody eliminacji w p.3.1.2 dokładniej precyzowaliśmy, co rozumiemy przez rozkład równomierny na danym podzbiorze przestrzeni R^m .

Uogólniając pojęcia *pole powierzchni* i *objętość*, za zbiór Ω , na którym określamy nasz rozkład, będziemy zawsze przyjmowali zbiór, którego miara Lebesgue'a $l_m(\Omega)$ w przestrzeni R^m jest dodatnia i skończona. Mówimy, że zmienna losowa \mathbf{X} ma *rozkład równomierny na zbiorze Ω* (krótko: *zmienna losowa \mathbf{X} ma rozkład $U(\Omega)$*), jeżeli dla *dowolnego* zbioru $A \subset \Omega$ zachodzi $P\{X \in A\} = l_m(A)/l_m(\Omega)$. Przymiotnik *dowolny* wyróżniliśmy w celu zaznaczenia, że chodzi nam tylko o takie zbiory A , dla których $l_m(A)$ ma sens. Po tych wstępnych uwagach przedstawimy dwie ogólne metody generowania zmiennej losowej \mathbf{X} o rozkładzie $U(\Omega)$.

1. Przy sformułowanych wyżej założeniach zbiór Ω możemy zamknąć w m -wymiarowym przedziale $\prod_{j=1}^m [a_j, b_j]$, $-\infty < a_j < b_j < +\infty, j = 1, 2, \dots$. Najprostszy algorytm generowania zmiennej losowej \mathbf{X} polega na wygenerowaniu m niezależnych, jednowymiarowych zmiennych losowych U_j o rozkładach równomiernych $U(a_j, b_j)$ i zaakceptowaniu $\mathbf{U} = (U_1, \dots, U_m)$ jako \mathbf{X} , gdy $\mathbf{U} \in \Omega$, lub na powtórzeniu losowania \mathbf{U} w przeciwnym przypadku. Pokazaliśmy wyżej na przykładzie losowania \mathbf{X} o rozkładzie równomiernym na kuli $K_m(0,1)$, jak niedoskonały (żeby nie powiedzieć - bezużyteczny) jest to algorytm, ale niestety jest to jedyny znany dotąd, ogólny, uniwersalny sposób postępowania. Istotne udoskonalenie może polegać na pokryciu zbioru Ω wieloma rozłącznymi przedziałami, losowaniu najpierw przedziału, a następnie \mathbf{U} o rozkładzie równomiernym na tym wybranym przedziale, wymaga to jednak dokładniejszego uwzględnienia kształtu zbioru Ω i ogólnie niewiele możemy tu sugerować.

2. Niektóre zbiory Ω można otrzymać przez nieosobliwe liniowe przekształcenie pewnych prostszych zbiorów: np. elipsoidy w R^m otrzymujemy przez nieosobliwe liniowe przekształcenie kuli $K_m(0, 1)$, powierzchnię elipsoidy - przez nieosobliwe liniowe przekształcenie sfery $S_m(0,1)$, niektóre wielościany wypukłe - przez nieosobliwe liniowe przekształcenie sympleksu

$W_m = \{(x_1, \dots, x_m) : \sum_{j=1}^m x_j \leq 1, x_j \geq 0, j = 1, 2, \dots, m\}$ a powierzchnie takich wielościanów - przez przekształcenia zbioru $V_m = \{(x_1, \dots, x_m) : \sum_{j=1}^m x_j = 1, x_j \geq 0, j = 1, 2, \dots, m\}$. Jest oczywiste, że jeżeli \mathbf{X} jest zmienną losową o rozkładzie równomiernym na zbiorze Ω oraz F jest nieosobliwym liniowym przekształceniem przestrzeni R^m , to $F(\mathbf{X})$ jest zmienną losową o rozkładzie równomiernym na obrazie $F(\Omega)$ zbioru Ω w tym przekształceniu.

4.2.2. Rozkład równomierny na sferze i na kuli w R^m

Przypuśćmy, że zmienna losowa $\mathbf{X} = (X_1, \dots, X_m)$ ma rozkład równomierny na sferze

$$S_m = \left\{ (x_1, \dots, x_m) : \sum_{j=1}^m x_j^2 = 1 \right\} \quad (4.2)$$

Zauważmy, że jeżeli \mathbf{A} jest macierzą ortonormalną (tzn. macierzą pewnego obrotu w R^m) i jeżeli \mathbf{X} ma rozkład równomierny na sferze S_m , to zmienna losowa \mathbf{AX} ma również rozkład równomierny na tej sferze.

Mówimy, że m -wymiarowa zmienna losowa $\mathbf{Z} = (Z_1, \dots, Z_m)$ ma *rozkład sferycznie konturowany*, jeżeli gęstość $g_z(z_1, \dots, z_m)$ jej rozkładu zależy tylko od $\|\mathbf{z}\|^2 = \sum_{j=1}^m z_j^2$. Najprostszym przykładem takiego rozkładu jest łączny rozkład m niezależnych zmiennych losowych o rozkładach normalnych $N(0,1)$:

$$g_z(z_1, \dots, z_m) = (2\pi)^{-m/2} \exp\left(-\frac{1}{2}\|\mathbf{z}\|^2\right)$$

Ponieważ dla każdej ortonormalnej macierzy \mathbf{A} mamy $\|\mathbf{Az}\|^2 = \|\mathbf{z}\|^2$, więc jeżeli \mathbf{Z} ma rozkład sferycznie konturowany, to \mathbf{AZ} ma taki sam rozkład jak \mathbf{Z} . Prowadzi to do następującego algorytmu generowania zmiennej losowej \mathbf{X} o rozkładzie równomiernym na sferze S_m :

ALGORYTM 4.1

Wygeneruj m niezależnych zmiennych losowych Z_1, \dots, Z_m o rozkładzie normalnym $N(0,1)$

Oblicz $X_j = Z_j/\|\mathbf{z}\|$, $j = 1, 2, \dots, m$

Return $\mathbf{X} = (X_1, \dots, X_m)$

Inna metoda generowania zmiennej losowej \mathbf{X} oparta jest na następującym lemacie:

Lemat 4.1. Jeżeli Z_1, \dots, Z_k ma rozkład równomierny na k -wymiarowej sferze S_k , R jest zmienną losową o rozkładzie z gęstością

$$h(r) = \begin{cases} \frac{cr^{k-1}}{\sqrt{1-r^2}} & \text{jeżeli } 0 \leq r \leq 1 \\ 0, & \text{poza tym} \end{cases} \quad (4.3)$$

oraz s jest „losowym znakiem” tzn.

$$P\{s = 1\} = P\{s = -1\} = \frac{1}{2} \quad (4.4)$$

to $(k+1)$ -wymiarowa zmienna losowa $(RZ_1, RZ_2, \dots, RZ_k, s\sqrt{1-R^2})$ ma rozkład równomierny na sferze S_{k+1} .

Otrzymujemy następujący algorytm:

ALGORYTM 4.2

Wygeneruj X_1 jako punkt losowy na sferze jednowymiarowej, tzn., wylosuj z jednakowym prawdopodobieństwem jedną z liczb -1 lub 1

For $k = 2$ to $m - 1$ do

Wygeneruj R według rozkładu o gęstości (4.3) oraz s według rozkładu (4.4)

Oblicz $X_j = RX_j, j = 1, 2, \dots, k, X_{k+1} = s\sqrt{1 - R^2}$

Return $\mathbf{X} = (X_1, \dots, X_m)$

W celu zaprojektowania generatora liczb losowych o rozkładzie z gęstością (4.3) wystarczy zauważyć, że zmienna losowa o tym rozkładzie może być potraktowana jako pierwiastek kwadratowy zmiennej losowej o rozkładzie beta $B(k/2, 1/2)$, czyli jako zmienna losowa $\sqrt{Y_1/(Y_1 + Y_2)}$, gdzie Y_1 oraz Y_2 są niezależnymi zmiennymi losowymi: Y_1 o rozkładzie gamma $\Gamma(k/2, 1)$ i Y_2 o rozkładzie gamma $\Gamma(1/2, 1)$.

Niech zmienna losowa $\mathbf{Y} = (Y_1, \dots, Y_m)$ ma rozkład równomierny na kuli

$$K_m = \{(y_1, \dots, y_m) : \sum_{j=1}^m y_j^2 \leq 1\} \quad (4.5)$$

Jest to oczywiście rozkład sferycznie konturowany. O prostym algorytmie generowania zmiennej losowej \mathbf{Y} metodą eliminacji z m -wymiarowej kostki opisanej na tej kuli mówiliśmy na początku rozdziału. Zwróciliśmy tam jednak uwagę na to, że jest to algorytm wysoce nieefektywny, a w przypadku dużego wymiaru m wręcz bezużyteczny.

Prosta metoda generowania zmiennej losowej \mathbf{Y} jest oparta na spostrzeżeniu, że długość promienia wodzącego R punktu \mathbf{Y} o rozkładzie równomiernym na m -wymiarowej kuli $K_m(0, 1)$ jest zmienną losową o rozkładzie potęgowym z gęstością $h(r) = mr^{m-1}, 0 \leq r \leq 1$. Wystarczy zatem wygenerować punkt $\mathbf{X} = (X_1, \dots, X_m)$ o rozkładzie równomiernym na sferze $S_m(0, 1)$ oraz zmienną losową R o rozkładzie z gęstością $h(r)$ i obliczyć $\mathbf{Y} = (RX_1, \dots, RX_m)$:

ALGORYTM 4.3

Wygeneruj \mathbf{X} jako punkt losowy na sferze $S_m(0, 1)$

Wygeneruj R według rozkładu o dystrybucji $H(r) = r^m, 0 \leq r \leq 1$

Return $\mathbf{X} = (RX_1, \dots, RX_m)$

Inna metoda generowania \mathbf{Y} jest oparta na rozkładach warunkowych, co opisaliśmy wyżej. Na przykład, w przypadku $m = 2$, tzn. w przypadku rozkładu równomiernego na kole jednostkowym $K_2(0, 1)$, możemy skorzystać z następującego algorytmu:

ALGORYTM 4.4

Wygeneruj Y_1 o rozkładzie z gęstością $g_1(y_1) = \frac{2}{\pi} \sqrt{1 - y_1^2}, |y_1| \leq 1$

Wygeneruj Y_2 o rozkładzie równomiernym na przedziale $(-\sqrt{1 - Y_1^2}, \sqrt{1 - Y_1^2})$

Return $\mathbf{Y} = (Y_1, Y_2)$

Rozwinięcie tej metody na ogólny przypadek m -wymiarowego rozkładu równomiernego na kuli $K_m(0, 1)$ pozostawiamy Czytelnikowi. Metodę generowania punktu losowego na kuli m -wymiarowej przez odpowiednie składanie dwóch niezależnych punktów losowych o jednakowych rozkładach równomiernych na kuli $(m/2)$ -wymiarowej podali Banerija i Dwyer (1993).

4.2.3. Rozkład równomierny na sympleksie i na powierzchni sympleksu

Lemat 4.2. Jeżeli U_1, \dots, U_m jest ciągiem niezależnych zmiennych losowych o rozkładzie równomiernym $U(0,1)$ oraz $U_{1:m}, \dots, U_{m:m}$ jest ciągiem odpowiednich statystyk pozycyjnych, to m -wymiarowa zmienna losowa $\mathbf{X} = (X_1, \dots, X_m)$ określona wzorami

$$\begin{aligned} X_1 &= U_{1:m} \\ X_2 &= U_{2:m} - U_{1:m} \\ &\dots, \dots \\ X_m &= U_{m:m} - U_{m-1:m} \end{aligned}$$

ma rozkład równomierny na sympleksie

$$W_m = \{(x_1, \dots, x_m) : \sum_{j=1}^m x_j \leq 1, x_j \geq 0, j = 1, 2, \dots, m\}$$

Lemat 4.3. Jeżeli U_1, \dots, U_{m-1} jest ciągiem niezależnych zmiennych losowych o rozkładzie równomiernym $U(0, 1)$ oraz $U_{1:m-1}, \dots, U_{m-1:m-1}$ jest ciągiem odpowiednich statystyk pozycyjnych, to m -wymiarowa zmienna losowa $\mathbf{X} = (X_1, \dots, X_m)$ określona wzorami

$$\begin{aligned} X_1 &= U_{1:m-1} \\ X_2 &= U_{2:m-1} - U_{1:m-1} \\ &\dots \\ X_{m-1} &= U_{m-1,m-1} - U_{m-2,m-1} \\ X_m &= 1 - U_{m-1:m-1} \end{aligned}$$

ma rozkład równomierny na powierzchni sympleksu

$$V_m = \{(x_1, \dots, x_m) : \sum_{j=1}^m x_j = 1, x_j \geq 0, j = 1, 2, \dots, m\}$$

Z lematów wynika, że generowanie punktów losowych na m -wymiarowym sympleksie lub na jego powierzchni sprowadza się do sprawnego wyznaczenia statystyk pozycyjnych w m - lub $(m-1)$ -elementowym ciągu niezależnych zmiennych losowych o rozkładzie równomiernym $U(0,1)$. Najprostszy algorytm generowania ciągu $U_{1:m}, \dots, U_{m:m}$ polega oczywiście na uporządkowaniu ciągu U_1, \dots, U_m w ciąg niemalejący, jednak znane są bardziej efektywne algorytmy. Oto dwa z nich:

ALGORYTM 4.5

Wygeneruj ciąg E_1, \dots, E_{m+1} i niezależnych zmiennych losowych o rozkładzie wykładniczym $E(0,1)$

$$S = \sum_{j=1}^{m+1} E_j; \quad U_{0:m} = 0$$

For $j = 1$ to m do $U_{j:m} = U_{j-1:m} + E_j/S$

Return $U_{1:m}, \dots, U_{m:m}$

ALGORYTM 4.6

$U_{m+1:m} = 1$
 For $j = m$ downto 1 do
 Generuj U o rozkładzie równomiernym $U(0,1)$
 $U_{j:m} = U^{1/j} U_{j+1:m}$
 Return $U_{1:m}, \dots, U_{m:m}$

Algorytm 4.5 jest oparty na następującym lemacie:

LEMAT 4.4. Niech $U_{1:m}, \dots, U_{m:m}$ będzie ciągiem statystyk pozycyjnych ciągu U_1, \dots, U_m niezależnych zmiennych losowych o rozkładzie równomiernym i niech $U_{0:m} = 0$ oraz $U_{m+1:m} = 1$. Zmienne losowe $U_{j:m} - U_{j-1:m}$, $j = 1, 2, \dots, m+1$, mają taki sam rozkład jak zmienne losowe $E_1/S, E_2/S, \dots, E_{m+1}/S$, gdzie E_1, \dots, E_{m+1} jest ciągiem niezależnych zmiennych losowych o rozkładzie wykładniczym $E(0,1)$ oraz $S = \sum_{j=1}^{m+1} E_j$

Wynika stąd prosty algorytm generowania punktów o rozkładzie równomiernym na powierzchni sympleksu: wygeneruj ciąg E_1, \dots, E_m niezależnych zmiennych losowych o rozkładzie wykładniczym $E(0,1)$ i oblicz $E_1/S, E_2/S, \dots, E_m/S$, gdzie $S = E_1 + E_2 + \dots + E_m$.

Algorytm 4.6 jest oparty na następującym lemacie:

Lemat 4.5. Niech $U_{1:m}, \dots, U_{m:m}$ będzie ciągiem statystyk pozycyjnych ciągu U_1, \dots, U_m niezależnych zmiennych losowych o rozkładzie równomiernym. Dla każdego $k = 0, 1, \dots, m-1$, zmienne losowe $U_{m:m}, U_{m-1:m}, \dots, U_{k:m}$ mają taki sam rozkład jak zmienne losowe $U_n^{1/n}, U_n^{1/n} U_{n-1}^{1/(m-1)}, \dots, U_n^{1/n} U_{n-1}^{1/(m-1)} \dots U_{m-k}^{1/(m-k)}$

4.3. Wielowymiarowy rozkład normalny

Prosta metoda Boxa-Mullera (1958) generowania dwuwymiarowej zmiennej losowej o rozkładzie normalnym jest oparta na następującym spostrzeżeniu: jeżeli U_1 i U_2 są dwiema niezależnymi zmiennymi losowymi o jednakowym rozkładzie równomiernym $U(0,1)$, to zmienne losowe X_1 i X_2 określone wzorami

$$X_1 = \sqrt{-2 \ln(U_1)} \cos(2\pi U_2) \quad X_2 = \sqrt{-2 \ln(U_1)} \sin(2\pi U_2)$$

są niezależne i mają jednakowy rozkład normalny $N(0,1)$. Za pomocą tej metody otrzymujemy jednocześnie dwie niezależne zmienne losowe o jednakowym rozkładzie normalnym $N(0, 1)$.

W zastosowaniach często trzeba korzystać z wielowymiarowych rozkładów normalnych, w których poszczególne zmienne losowe są ze sobą skorelowane. Niech

$$A = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \dots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_{2,2} & \dots & \sigma_{2,n} \\ \dots & \dots & \dots & \dots \\ \sigma_{n,1} & \sigma_{n,2} & \dots & \sigma_{n,n} \end{bmatrix}$$

będzie macierzą kowariancji n -wymiarowej zmiennej losowej $\mathbf{X} = (X_1, \dots, X_n)$ o rozkładzie normalnym, takim że wartość oczekiwana $EX_i = 0$ dla każdego $i = 1, 2, \dots, n$. Jeżeli $\mathbf{Z} = (Z_1, \dots, Z_n)$ oraz wszystkie Z_i , $i = 1, 2, \dots, n$, są niezależne i mają taki sam rozkład normalny $N(0,1)$, to zmienna losowa \mathbf{CZ} , gdzie \mathbf{C} jest pewną macierzą nieosobliwą, ma n -wymiarowy rozkład normalny z macierzą kowariancji \mathbf{CC}^T (\mathbf{C}^T jest macierzą transponowaną macierzy \mathbf{C}). W celu wygenerowania n -wymiarowej zmiennej losowej \mathbf{X} z daną macierzą kowariancji \mathbf{A} wystarczy więc skonstruować

odpowiednią macierz \mathbf{C} , taką żeby $\mathbf{C}\mathbf{C}^T = \mathbf{A}$, wygenerować n niezależnych zmiennych losowych Z_1, \dots, Z_n o jednakowym rozkładzie normalnym $N(0, 1)$ i obliczyć $\mathbf{X} = \mathbf{C}\mathbf{Z}$.

Macierz \mathbf{C} można skonstruować w następujący sposób:

$$c_{i,1} = \frac{\sigma_{i,1}}{\sqrt{\sigma_{1,1}}}$$

$$c_{i,i} = \left(\sigma_{i,i} - \sum_{r=1}^{i-1} c_{i,r}^2 \right)^{1/2}$$

$$c_{i,j} = c_{j,j}^{-1} \left(\sigma_{i,j} - \sum_{r=1}^{j-1} c_{i,r} c_{j,r} \right), \quad \text{gdy } i > j$$

$$c_{i,j} = 0, \quad \text{gdy } i < j$$

Inna metoda generowania zmiennej losowej $\mathbf{X} = (X_1, \dots, X_n)$ o n -wymiarowym rozkładzie normalnym z dowolną macierzą kowariancji (niekoniecznie nieosobliwą) polega na generowaniu składowych X_1, \dots, X_n , kolejno: X_1 z odpowiedniego (brzegowego) rozkładu normalnego, a gdy wygenerowano już X_1, \dots, X_{k-1} , generuje się X_k znowu z odpowiedniego (warunkowego) rozkładu normalnego. Ta metoda jest szczególnie przydatna do generowania różnych procesów gaussowskich.

5. Testowanie generatorów liczb losowych

5.1. Metodyka testowania generatorów

Rozważając własności statystyczne generatora (w tym również generatora programowego}, traktujemy go jako pewne urządzenie, takie jak np. moneta, urna z odpowiednim zapasem różnych kulek, ruletka, itp. Kolejną liczbę X produkowaną przez generator uznajemy za zmienną losową i testowanie generatora sprowadzamy do testowania odpowiednich hipotez o rozkładzie tej zmiennej losowej. Co więcej, na ogół interesuje nas nie pojedyncza liczba wyprodukowana przez generator, ale odpowiednio długie ciągi takich liczb: wtedy testowanie dotyczy hipotezy, że kolejno pojawiające się liczby X_1, X_2, \dots, X_n na wyjściu generatora są niezależnymi zmiennymi losowymi o jednakowym rozkładzie.

W zasadzie interesuje nas testowanie generatora liczb losowych U_n , $n = 1, 2, \dots$ o rozkładzie równomiernym $U(0,1)$; liczby losowe o innych rozkładach otrzymujemy z takich właśnie liczb (o czym mówiliśmy szczegółowo w rozdz. 3 i 4). Ich testowanie sprowadza się do sprawdzenia poprawności realizacji odpowiedniej procedury numerycznej, a ewentualne testy zgodności rozkładów dla tych nowych zmiennych losowych możemy traktować jako w istocie rzeczy pewne dodatkowe testy podstawowego generatora liczb losowych o rozkładzie równomiernym $U(0,1)$. W bieżącym rozdziale będziemy więc zajmowali się tylko testowaniem generatora liczb losowych o

rozkładzie równomiernym $U(0,1)$.

Sposób testowania generatora jest determinowany przez sposób, w jaki jest on używany. Wyobraźmy sobie generator jako skończony ciąg liczb (tu i dalej w tym rozdziale mamy w takim kontekście na myśli liczby z przedziału $(0,1)$) - długość tego ciągu jest równa okresowi generatora. Korzystając z generatora, wybieramy losowo pewną liczbę z tego ciągu i poczynając od niej generujemy potrzebną liczbę, powiedzmy n , kolejnych liczb. Losowy wybór początkowej liczby jest realizowany komendą *srand* w języku C, komendą *randomize* w Turbo Pascalu lub przez losowy wybór tej początkowej liczby jako *seed* przez nas samych w jakiś inny sposób. Oczekujemy, że wygenerowany w ten sposób ciąg u_1, \dots, u_n będziemy mogli traktować jako *próbę losową*, tzn. jako realizację ciągu U_1, \dots, U_n liczb losowych o rozkładzie równomiernym $U(0,1)$. Testowanie hipotezy, że aktualnie używany generator spełnia nasze oczekiwania, powinno zatem polegać na obliczeniu wartości odpowiedniej statystyki testowej i porównaniu jej z wartością krytyczną, właściwą dla zastosowanego testu (zwykle używamy w tym celu kilku lub kilkunastu różnych testów, wybranych odpowiednio do wykonywanego za pomocą tego generatora zadania). Pozytywny wynik testu (nie odrzucenie weryfikowanej hipotezy) jest pewnym argumentem za tym, że obliczenia wykonane przez nas za pomocą liczb losowych z tego generatora są poprawne.

Testowanie generatora nie powinno się jednak na tym zakończyć, nawet gdy n jest bardzo duże. Przemawiają za tym dwie następujące okoliczności. Po pierwsze, przystępując do rozwiązywania za pomocą tego generatora następnego zadania, wystartujemy zapewne z innej liczby początkowej, a przecież nie umiemy nic powiedzieć na temat wyników testów przy tej nowej liczbie startowej. Po drugie, jeżeli testujemy naszą hipotezę na zadanym poziomie istotności, np. α , to przeciętnie jeden raz na $1/\alpha$ przypadków natrafimy na wynik dyskwalifikujący hipotezę, ale jeżeli dzieje się tak właśnie przeciętnie jeden raz na $1/\alpha$ przypadków, to świadczy to na korzyść generatora, a nie na rzecz jego dyskwalifikacji. W związku z tym przyjmujemy następującą metodykę testowania generatora:

- 1) ustalamy liczbę n i startując z losowo wybranej liczby początkowej, generujemy n kolejnych liczb;
- 2) obliczamy wartość statystyki testowej - oznaczmy ją przez T ;
- 3) obliczamy wartość $F(T)$, gdzie F jest dystrybuantą rozkładu statystyki T , gdy weryfikowana hipoteza jest prawdziwa;
- 4) powtarzamy powyższą operację - oznaczmy przez N liczbę tych powtórzeń - obliczając w kolejnych krokach wartości statystyki: T_1, T_2, \dots, T_N . Jeżeli weryfikowana hipoteza jest prawdziwa, to $F(T_1), F(T_2), \dots, F(T_N)$ jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym $U(0,1)$. Testowanie generatora kończymy testowaniem właśnie tej hipotezy.

W pracy Zielińskiego (1966) testowano np. dwa generatory liniowe postaci $X_{n+1} = (aX_n + c) \bmod m$, $U_n = X_n/m$, dla $m = 2^{35}$. W generatorze nazwanym RNB wybrano $a = 2^2 * 23^{37} + 1$ oraz $c = 0$ (generator multiplikatywny) tak, że był to generator o maksymalnym okresie. W generatorze nazwanym RNC wybrano $a = 5$ oraz $c = 2^{-35} = \frac{1}{2} - \frac{1}{6}\sqrt{3}$, co zapewniało, że współczynnik autokorelacji tego generatora był równy zero (patrz Marsaglia (1962)). Dla $n = 1000$ oraz $N = 200$ w przypadku RNB i dla $N = 100$ w przypadku RNC uzyskano rozkłady empiryczne statystyki testu serii monofonicznych (por. podrozdz. 5.5), przedstawione w tab. 5.1.

Tabela 5.1

Przedział	RNB	RNC
0.0- 0.1	30	97
0.1 - 0.2	23	2

0.2 - 0.3	16	-
0.3-0.4	22	1
0.4-0.5	15	-
0.5 - 0.6	21	-
0.6 - 0.7	19	-
0.7- 0,8	15	-
0.8 - 0.9	20	-
0.9- 1.0	19	-

Zauważmy, że ostatecznie zdyskwalifikowany generator RNC w każdym ze stu wykonanych na nim testów wykazywał dobrą zgodność rozkładu empirycznego statystyki testowej z jej rozkładem teoretycznym!

W dalszym ciągu nie będziemy już wracać do tej dyskusji metodologicznej i ograniczymy się do prezentacji ważniejszych testów statystycznych stosowanych do badania generatorów liczb losowych.

W Internecie pod adresem <http://stat.fsu.edu/~geo/diehard.html> została udostępniona przez Marsaglię cała bateria testów statystycznych (procedury w języku C). Pakiet ten jest dostępny również na specjalnym CD-ROMie poświęconym generatorom liczb losowych (Marsaglia (1995)). Wiele klasycznych testów generatorów opisano w książce Knutha (1981). Dokładne opisy algorytmów i przykładowe implementacje zawiera również pozycja Dudewicza i Ralleya (1981).

W bieżącym rozdziale przyjmujemy następujące oznaczenia: X_1, \dots, X_n jest ciągiem liczb losowych wyprodukowanych przez testowany generator oraz U_1, \dots, U_n jest ciągiem liczb losowych o rozkładzie równomiernym $U(0,1)$, wyprodukowanym przez generator. Ogólnie, przez X lub odpowiednio przez U , oznaczamy zmienną losową produkowaną przez rozważany generator.

5.2. Testy zgodności z rozkładem $U(0,1)$

5.2.1. Test chi-kwadrat

Test chi-kwadrat jest jednym z najczęściej stosowanych testów zgodności, a jego opis i tablice wartości krytycznych można znaleźć w licznych podręcznikach, zbiorach tablic i pakietach komputerowych. W przypadku testowania generatorów znajduje on zastosowanie zarówno na pierwszym, jak i na drugim poziomie (tzn. także do testowania zgodności rozkładu zmiennej losowej $F(T)$ z rozkładem równomiernym $U(0, 1)$ na podstawie takich danych, jak dane w tab. 5.1).

Hipoteza statystyczna w rozważanych testach zgodności ma ogólną postać: zmienna losowa X ma rozkład prawdopodobieństwa o dystrybuancie F .

Niech a oraz b będą liczbami takimi, że $F(a) = 0$ oraz $F(b) = 1$ (nie wykluczamy przypadku $a = -\infty$ oraz/lub $b = +\infty$). Niech $a = a_0 < a_1 < a_2 < \dots < a_k = b$ będzie rozbiciem zbioru wartości zmiennej losowej X i niech $p_i = P\{a_{i-1} < X \leq a_i, i = 1, 2, \dots\}$. Oznaczmy przez n_i liczbę takich elementów X ciągu X_1, \dots, X_n , które spełniają warunek $a_{i-1} < X \leq a_i$. Statystyką testu jest

$$X_{k-1}^2 = \sum_{i=1}^k n_i^2 - n \quad (5.1)$$

Gdy n jest duże a rozbiecie takie, że liczby np_i „nie są zbyt małe”, statystyka (5.1) ma w przybliżeniu rozkład chi-kwadrat o $(k - 1)$ stopniach swobody. Inaczej niż w wielu zastosowaniach praktycznych, testując generatory liczb losowych mamy dużą swobodę w konstrukcji testu. Wygodnie jest dokonać takiego rozbiecia (a_1, \dots, a_k) , żeby $p_i = 1/k$ dla wszystkich $i = 1, 2, \dots$. Wtedy

statystyka (5.1) przyjmuje prostą postać

$$X_{k-1}^2 = \frac{k}{n} \sum_{i=1}^k n_i^2 - n \quad (5.2)$$

Jeżeli ponadto przyjmiemy pewne standardowe k , np. $k = 10$, będziemy mogli łatwo zautomatyzować obliczenia poziomów krytycznych $F(X_{k-1}^2)$ testu, gdzie F jest dystrybuantą odpowiedniego rozkładu chi-kwadrat.

5.2.2. Test zgodności z rozkładem wielowymiarowym

Test chi-kwadrat stosuje się bez żadnych ideowych zmian także w przypadku testowania zgodności rozkładu zmiennej losowej (X_1, \dots, X_m) z odpowiednim rozkładem wielowymiarowym, a ponieważ w przypadku testowania generatorów liczb losowych problem polega na testowaniu niezależności kolejno produkowanych liczb losowych, interesuje nas hipoteza, że ten rozkład wielowymiarowy jest odpowiednim rozkładem produktowym, czyli że zmienna losowa (X_1, \dots, X_m) ma rozkład o dystrybuancie

$$H(x_1, \dots, x_m) = F(x_1)F(x_2) \dots F(x_m) \quad (5.3)$$

W celu uniknięcia zbyt rozbudowanego systemu oznaczeń, przedstawimy konstrukcję statystyki testu dla przypadku testowania hipotezy, że zmienna losowa (X_1, \dots, X_m) ma m -wymiarowy rozkład równomierny na kostce jednostkowej $(0, 1)^m$. Rozbijmy każdy przedział $(0, 1)$ na k jednakowo długich podprzedziałów postaci $((j-1)/k, j/k)$, $j = 1, 2, \dots, k$. Otrzymamy w ten sposób rozbięcie kostki $(0, 1)^m$ na k^m małych kostek o jednakowej objętości (pedantycznemu Czytelnikowi powiedzmy dokładniej: o jednakowej mierze Lebesgue'a w R^m), równej oczywiście k^{-m} . Jeżeli teraz nm -elementowy ciąg liczb losowych z badanego generatora rozbijemy na n rozłącznych ciągów n -elementowych

$$\begin{aligned} & (X_1, X_2, \dots, X_m), (X_{m+1}, X_{m+2}, \dots, X_{2m}), \dots \\ & \dots, (X_{(j-1)m+1}, X_{(j-1)m+2}, \dots, X_{jm}), \dots \\ & \dots, (X_{(n-1)m+1}, X_{(n-1)m+2}, \dots, X_{nm}) \end{aligned} \quad (5.4)$$

i przez n_i oznaczymy liczbę takich elementów $(X_{(j-1)m+1}, X_{(j-1)m+2}, \dots, X_{jm})$ $j = 1, 2, \dots, n$, które wpadły do i -tej kostki, to zwykła statystyka testu chi-kwadrat, tzn. taka jaką przedstawiliśmy w podrozdz. 5.2.1, przyjmie prostą postać (5.2) po podstawieniu tam k^m w miejsce k . Przy konstruowaniu takiego testu w przypadku dużego m trzeba jednak zadbać o to, żeby liczba nk^{-m} nie była zbyt mała. W pracy Leeba (1991) pokazano, jak za pomocą tej prostej konstrukcji można testować równomierność poszczególnych zespołów bitów uzyskiwanych z generatora, a także zawarto wiele wyników takich badań dla popularnych generatorów.

Interesująco przedstawia się test zgodności z rozkładem wielowymiarowym, zbudowany na nakładających się na siebie elementach $(X_{(j-1)m+1}, X_{(j-1)m+2}, \dots, X_{jm})$:

$$(X_1, X_2, \dots, X_m), (X_2, X_3, \dots, X_{m+1}), (X_3, X_4, \dots, X_{m+2}), \dots \quad (5.5)$$

Jeżeli długość ciągu liczb losowych, na którego podstawie testujemy generator, jest równa N , to takich nakładających się elementów mamy oczywiście $N - m + 1$. Niech jak poprzednio n_i oznacza liczbę takich elementów (5.5), które należą do i -tej kostki. Zdefiniujmy statystyki

$$\psi_0^2 = 0$$

$$\psi_m^2 = \sum_i \frac{\left(n_i - \frac{N - m + 1}{k^m} \right)^2}{\frac{N - m + 1}{k^m}}, \quad m = 1, 2, \dots \quad (5.6)$$

Okazuje się, że dla dostatecznie dużych N zmienna losowa $\psi_m^2 - \psi_{m-1}^2$ w

przybliżeniu rozkład chi-kwadrat o $k^m - k^{m-1}$ stopniach swobody (por. Good (1953), Zieliński (1972)).

5.2.3. Test OPSO

Zaproponowany przez Marsaglię (1984) test OPSO (ang. *overlapping-pairs sparse-occupancy*) dotyczy analizy częstości nakładających się par liczb uzyskiwanych z generatora. Załóżmy, że dla badanego generatora liczb pseudolosowych analizujemy ciąg skończony X_1, X_2, \dots, X_n . Biorąc ustaloną liczbę b bitów z każdej liczby, otrzymujemy nowy ciąg I_1, I_2, \dots, I_n złożony z liczb całkowitych ze zbioru $\{0, 1, \dots, 2^b - 1\}$. Utwórzmy ciąg kolejnych, nakładających się par, tzn. ciąg

$$(I_1, I_2), (I_2, I_3), \dots, (I_{n-1}, I_n)$$

Niech Y oznacza liczbę takich par ze zbioru $\{(i, j): i, j = 0, \dots, 2^b - 1\}$, które nie pojawiły się w tym ciągu. Zmienna losowa Y ma asymptotycznie (dla dużych n) rozkład normalny $N(\mu, \sigma)$

Poniższa tabelka zawiera przykładowe parametry dla testu OPSO:

b	n	μ	σ
10	2^{21}	141909	290.26
11	2^{22}	1542998	638.75
11	2^{23}	567639	580.80

Przykładem generatora zakwestionowanego za pomocą testu OPSO może być generator multiplikatywny $X_n = 69069X_{n-1} \text{ mod } 2^{32}$, dla którego test z parametrami $b = 10$, $n = 2^{21}$, zrealizowany kilkakrotnie, dał wartości statystyki $(Y - \mu) / \sigma$ m.in. równe: 4.611, 4.682, 4.114, 5.591.

Dalsze rozwinięcie idei testu OPSO na przypadek trójek, czwórek, itd. liczb z generatora, można znaleźć w pracy Marsaglii (1993).

2.4. Test Kołmogorowa

Test zgodności Kołmogorowa służy do weryfikacji hipotezy, że rozważana zmienna losowa X ma rozkład o danej ciągłej dystrybuancie F , przy czym statystyka testu jest oparta na różnicy między hipotetyczną dystrybuantą F a dystrybuantą empiryczną F_n z próby X_1, X_2, \dots, X_n . Podobnie jak test chi-kwadrat, test Kołmogorowa znajduje zastosowanie w testowaniu generatorów liczb losowych zarówno na pierwszym jak i na drugim poziomie. Hipoteza przyjmuje w tym przypadku

postać: zmienna losowa X ma rozkład prawdopodobieństwa o ciągłej dystrybuancie F .

Następująca wielkość jest statystyką testową:

$$D_n = \sup_{-\infty < x < +\infty} |F_n(x) - F(x)|$$

gdzie dystrybuantę empiryczną definiujemy wzorem

$$F_n(x) = n^{-1} \sum_{j=1}^n 1_{(-\infty, x)}(X_j)$$

Jeżeli próba X_1, X_2, \dots, X_n pochodzi z rozkładu o dystrybuancie F , to $D_n \rightarrow 0$ z prawdopodobieństwem 1 (jest to tzw. podstawowe twierdzenie statystyki matematycznej, por. np. Zieliński (1990)). Duże wartości statystyki D_n przemawiają przeciwko wyjściowej hipotezie. Ponadto rozkład statystyki D_n nie zależy od postaci funkcji $F(x)$. Pozwala to na wyznaczenie wartości krytycznych testu, czyli takich liczb $D_n(\alpha)$, że dla zadanego poziomu istotności α zachodzi

$$P_F \{D_n > D_n(\alpha)\} = \alpha$$

Obszar krytyczny testu ma postać $[D_n(\alpha), 1]$. Wartości krytyczne zostały stabilicowane (np. Zieliński (1987), Zieliński (1990), Domański (1990), Krysicki (1994)). Praktyczne obliczenia statystyki D_n z próby X_1, X_2, \dots, X_n opierają się na spostrzeżeniu, że $\sup_{-\infty < x < +\infty} |F_n(x) - F(x)|$ jest osiągnięte w jednym z punktów skoku dystrybuanty empirycznej F_n . Ponieważ D_n nie zmienia się przy monotonicznych przekształceniach argumentu x , możemy wykonać obliczenia według następujących wzorów:

$$D_n^+ = \max_{1 \leq i \leq n} \left(\frac{i}{n} - F(X_{i:n}) \right)$$

$$D_n^- = \max_{1 \leq i \leq n} \left(F(X_{i:n}) - \frac{i-1}{n} \right)$$

$$D_n = \max \{D_n^+, D_n^-\}$$

gdzie $X_{i:n}$ oznacza i -tą statystykę pozycyjną z próby, czyli $X_{1:n} \leq X_{2:n} \leq \dots \leq X_{n:n}$.

Statystyki D_n^+ oraz D_n^- mają identyczne rozkłady (por. Koroluk i in. (1985) oraz Kendall i Stuart (1967)):

$$P\{D_n^+ \geq x\} = P\{D_n^- \geq x\} = \sum_{k=0}^{\lfloor n(1-x) \rfloor} \binom{n}{k} x \left(x + \frac{k}{n} \right)^{k-1} \left(1 - x - \frac{k}{n} \right)^{n-k}, \quad 0 < x < 1$$

Rozkład graniczny statystyk D_n^+ oraz D_n^- ma postać

$$\lim_{n \rightarrow \infty} P\left\{ \sqrt{n} D_n^+ \leq t \right\} = 1 - e^{-2t^2}, \quad t > 0$$

Dla odpowiednio dużych n (w praktyce już dla $n \geq 20$) wykorzystuje się również rozkład graniczny statystyki $\sqrt{n}D_n$, wyrażony wzorem

$$\lim P\{\sqrt{n}D_n \leq t\} = K(t) = \sum_{j=-\infty}^{\infty} (-1)^j \exp(-2j^2 t^2), \quad t > 0$$

Jest to tzw. *rozkład λ -Kolmogorowa*. Odpowiednie tablice wartości krytycznych λ_α podane są w zbiorze tablic Zielińskich (1990), a najczęściej używane wartości to: $\lambda_{0,1} = 1.224$, $\lambda_{0,05} = 1.358$, $\lambda_{0,01} = 1.628$.

5.3. Testy zgodności rozkładów statystyk

5.3.1. Wprowadzenie

Jeżeli X_1, X_2, \dots jest ciągiem niezależnych zmiennych losowych o rozkładzie równomiernym na przedziale $(0,1)$, to ciąg $(X_1, X_2, \dots, X_m), (X_{m+1}, X_{m+2}, \dots, X_{2m}), \dots$ jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym na m -wymiarowej kostce jednostkowej $(0,1)^m$.

Niech

$$y = h(x_1, x_2, \dots, x_m) \quad (5.7)$$

będzie funkcją m zmiennych, określoną na kostce jednostkowej $(0,1)^m$. O funkcji tej zakładamy tylko to, że jest ona zmienną losową, gdy jej argumenty są niezależnymi zmiennymi losowymi o rozkładzie równomiernym na przedziale $(0,1)$.

Przekształcając ciąg $(X_1, X_2, \dots, X_m), (X_{m+1}, X_{m+2}, \dots, X_{2m}), \dots$ według funkcji (5.7) otrzymujemy nowy ciąg

$$Y_j = h(X_{(j-1)m+1}, X_{(j-1)m+2}, \dots, X_{jm}), \quad j = 1, 2, \dots$$

Jest to ciąg niezależnych zmiennych losowych o jednakowym rozkładzie. Niech G będzie dystrybuantą rozkładu tych zmiennych losowych

$$G(y) = P\{Y_j \leq y\}$$

gdy zmienne losowe X_1, X_2, \dots są niezależne i mają jednakowy rozkład równomierny na przedziale $(0,1)$. Testowanie generatora na podstawie funkcji (5.7) przeprowadza się teraz w następujący sposób: obserwuje się ciąg zmiennych losowych X_1, X_2, \dots i przekształca go na ciąg zmiennych losowych Y_1, Y_2, \dots . Dla ciągu Y_1, Y_2, \dots weryfikuje się hipotezę, że jest on próbką prostą z populacji o dystrybuancie G . Odrzucenie tej hipotezy prowadzi do dyskwalifikacji generatora liczb losowych X_1, X_2, \dots

Funkcje (5.7) można wybierać w różny sposób; przedstawimy kilka takich funkcji, najczęściej używanych do testowania generatorów liczb losowych.

5.3.2. Testy oparte na statystykach pozycyjnych

Rozważmy funkcje postaci

$$u = \max\{x_1, x_2, \dots, x_m\}$$

$$v = \min\{x_1, x_2, \dots, x_m\}$$

$$r = u - v$$

Niech U_j , V_j oraz R_j będą zmiennymi losowymi otrzymanymi w wyniku przekształcenia danego ciągu X_1, X_2, \dots według funkcji, odpowiednio, u , v oraz r . Rozkłady tych zmiennych losowych wyrażają się za pomocą następujących wzorów:

$$P\{U_j \leq u\} = u^m, \quad 0 \leq u \leq 1$$

$$P\{V_j \leq v\} = 1 - (1 - v)^m, \quad 0 \leq v \leq 1$$

$$P\{R_j \leq r\} = mr^{m-1} - (m-1)r^m, \quad 0 \leq r \leq 1$$

Zadanie sprowadza się do weryfikacji hipotezy o zgodności rozkładu zmiennych U_j, V_j oraz R_j . Takie testy przeprowadza się najczęściej dla kilku wartości m , wybierając $m = 2, 3, \dots, 10$.

5.3.3. Test sum

Założmy, że funkcja (5.7) ma postać

$$y = x_1 + x_2 + \dots + x_m \quad (5.8)$$

Skonstruowane według tego przekształcenia zmienne losowe Y_j , mają rozkład o gęstości g_m , wyrażającej się wzorem

$$g_m = \begin{cases} \frac{1}{(m-1)} \left(y^{m-1} - \binom{m}{1} (y-1)^{m-1} + \binom{m}{2} (y-2)^{m-1} - \dots \right) & \text{dla } 0 \leq y \leq m \\ 0 & \text{poza tym} \end{cases}$$

gdzie sumowanie wykonuje się dopóty, dopóki $y, y-1, y-2, \dots$ są dodatnie.

Gdy $m = 2$, otrzymujemy rozkład trójkątny o gęstości

$$g_2(y) = \begin{cases} y & \text{dla } 0 \leq y \leq 1 \\ 2 - y & \text{dla } 1 \leq y \leq 2 \end{cases}$$

gdy zaś $m = 3$, otrzymujemy rozkład o gęstości

$$g_3(y) = \begin{cases} \frac{1}{2} y^2 & \text{dla } 0 \leq y \leq 1 \\ \frac{1}{2} (y^2 - 3(y-1)^2) & \text{dla } 1 < y \leq 2 \\ \frac{1}{2} (y^2 - 3((y-1)^2 + 3(y-2)^2)) & \text{dla } 2 < y \leq 3 \end{cases}$$

Opisane testy zgodności stosuje się zwykle dla małych wartości m , nie przekraczających 5.

Dla dużych m rozkład sumy (5.8) aproksymuje się za pomocą rozkładu normalnego.

5.3.4. Test d^2

We wzorze (5.7) przyjmijmy $m = 4$ i zdefiniujmy funkcję

$$y = (x_1 - x_3)^2 + (x_3 - x_4)^2 \quad (5.9)$$

Funkcję tą możemy interpretować w następujący sposób: potraktujmy (x_1, x_2) oraz (x_3, x_4) jako punkty w kwadracie jednostkowym $(0,1)^2$; wtedy y jest równe kwadratowi odległości między tymi punktami. Jeżeli X_1, X_2, X_3, X_4 są niezależnymi zmiennymi losowymi o jednakowym rozkładzie równomiernym na przedziale $(0,1)$, to zmienna losowa otrzymana w wyniku przekształcenia (5.9), oznaczana zwykle symbolem d^2 , ma rozkład

$$P\{d^2 \leq y\} = \begin{cases} \pi y - \frac{8}{3}y^{\frac{3}{2}} + \frac{1}{2}y^2 & \text{dla } 0 \leq y < 1 \\ \frac{1}{3} + (\pi - 2)y + 4(y-1)^{\frac{1}{2}} + \frac{8}{3}(y-1)^{\frac{3}{2}} - \frac{1}{2}y^2 - 4y \operatorname{arc} \sec(y^{\frac{1}{2}}) & \text{dla } 1 \leq y \leq 2 \end{cases}$$

(Grueberger (1951)). Weryfikacja generatora polega na sprawdzeniu zgodności rozkładu statystyki d^2 obliczonego dla danego generatora z podanym wyżej rozkładem teoretycznym.

5.3.5. Test urodzin dla spacji

Załóżmy że z generatora liczb pseudolosowych uzyskujemy ciąg liczb całkowitych I_1, I_2, \dots, I_m o wartościach ze zbioru $\{1, 2, \dots, n\}$. Porządkując I_1, \dots, I_m w ciąg niemalejący otrzymujemy ciąg $I_{1:m}, I_{2:m}, \dots, I_{m:m}$ a następnie tworzymy *ciąg spacji*, czyli ciąg

$$I_{1:m}, I_{2:m} - I_{1:m}, I_{3:m} - I_{2:m}, \dots, I_{m:m} - I_{m-1:m}$$

Niech Y będzie liczbą spacji, które występują więcej niż raz w powyższym ciągu. Wielkość Y jest zatem równa m minus liczba różnych spacji. Dowodzi się, że jeżeli ciąg I_1, \dots, I_m jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym na zbiorze $\{1, 2, \dots, n\}$, to zmienna losowa Y ma asymptotycznie rozkład Poissona z parametrem $\lambda = m^3/4n$. Nawiązując do sławnego problemu Von-Misesa i Fcllera dotyczącego rozkładu urodzin, Marsaglia (1984) nazwał opisywany test *testem urodzin dla spacji*.

Zauważmy, że na wartość statystyki Y nie ma wpływu kolejność liczb produkowanych przez generator. Spośród rozważanych przez nas generatorów test urodzin dla spacji z reguły spełniają generatory liniowe, generatory typu $F(r,s,*)$ oraz kombinacje generatorów. Zwykle nie spełniają jednak tego testu uogólnione generatory Fibonacciego typu $F(r,s,+)$, $F(r,s,-)$ oraz $F(r,s, \text{xor})$.

Stosując rozważany test w praktyce, wybiera się odpowiednio dużą wartość n , np.

$n \geq 10000$, i za wartości I_k przyjmuje się liczby utworzone z najbardziej znaczących bitów liczb z badanego generatora.

5.3.6. Test najmniejszej odległości w parach

Generujemy n punktów z kostki m wymiarowej $(0, 1)^m$. Weźmy pod uwagę wszystkie $\binom{n}{2}$ pary punktów i obliczmy odległość euklidesową między punktami każdej pary. Niech D oznacza najmniejszą z tych odległości. Przy założeniu równomierności generatora zmienna losowa $T = n^2 D^m / 2$ ma dla dużych n rozkład w przybliżeniu wykładniczy ze średnią $1/V_m$, gdzie V_m oznacza objętość m -wymiarowej kuli jednostkowej. Własność tę wykorzystuje się do tzw. *testu najmniejszej odległości w parach*. Stosując ten test w praktyce, zwykle generuje się Nn punktów w kostce $(0,1)^m$ uzyskując N realizacji statystyki T , a następnie dokonuje się porównania rozkładu empirycznego T z rozkładem wykładniczym (np. testem Kołmogorowa lub testem chi-kwadrat). Zauważmy, że duże wartości N wymagają również zwiększania n , gdyż rozkład wykładniczy jest rozkładem asymptotycznym statystyki T . Jakość tej aproksymacji pogarsza się ze wzrostem wymiaru m , co wymusza przyjęcie większego n lub zmniejszanie parametru N . Wartości n z kolei nie mogą być zbyt duże, gdyż powoduje to istotny wzrost złożoności obliczeń minimalnej odległości. L'Ecuyer (1995) proponuje np. następujące układy parametrów w omawianym teście:

- 1) $N=100, n = 10^5, m = 4,$
- 2) $N=20, n = 10^5, m = 6,$
- 3) $N = 20, n = 50000, m = 9.$

Generatory liniowe zwykle nie spełniają opisanego testu, ponieważ tworzą one regularne siatki w przestrzeniach \mathbf{R}^m .

Test momentów

Podstawę teoretyczną testów, o których teraz będziemy mówili, stanowi następujące twierdzenie:

TWIERDZENIE 5.1 *Niech $\zeta_1, \zeta_2, \dots, \zeta_N$ będą niezależnymi zmiennymi losowymi rozkładzie jednakowym rozkładzie z dystrybuantą F . Niech F_N oznacza dystrybuantę unormowanej zmiennej losowej $\bar{\zeta}_N = (\zeta_1 + \zeta_2 + \dots + \zeta_N) / N$. Jeżeli istnieją wszystkie momenty zmiennej losowej ζ oraz*

$$\lim_{|t| \rightarrow \infty} \sup |E(\exp(it\zeta))| < 1, \quad \text{to}$$

$$F_N(x) = \Phi(x) - \frac{1}{3!} \frac{\mu_3}{\sigma^3} \varphi^{(2)}(x) + \frac{1}{4!} \left(\frac{\mu_4}{\sigma^4} - 3 \right) \varphi^{(3)}(x) - \frac{1}{5!} \left(\frac{\mu_5}{\sigma^5} - 10 \frac{\mu_3}{\sigma^3} \right) \varphi^{(3)}(x) + \dots \quad (5.10)$$

gdzie $\Phi(x)$ i $\varphi(x)$ są, odpowiednio, dystrybuantą i gęstością rozkładu normalnego $N(0, 1)$, $\varphi^{(i)}(x)$ jest i -tą pochodną tej gęstości oraz μ_k jest k -tym momentem centralnym zmiennej losowej $\bar{\zeta}_N, \sigma^2 = \mu_2$. Błąd aproksymacji rozkładu zmiennej losowej $\bar{\zeta}_N$ za pomocą powyższego wzoru jest takiego samego rzędu jak pierwszy odrzucony wyraz rozwinięcia. D

Podane twierdzenie umożliwia obliczanie prawdopodobieństw różnych zdarzeń związanych ze zmiennymi losowymi $\bar{\zeta}_N$, za pomocą tablic rozkładu normalnego $N(0, 1)$. Dowód i dyskusje tego twierdzenia można znaleźć np. w pracach Cramera (1958) oraz Gniedenki i Kołmogorowa (1957).

W naszym przypadku zmienne losowe X_1, X_2, \dots, X_N mają rozkład równomierny na przedziale $(0,1)$. Założenia twierdzenia są spełnione, bo wszystkie całki $\int_0^1 x^m dx, m = 1, 2, \dots$, a więc wszystkie momenty zmiennych losowych X_i istnieją oraz

$$|E(\exp(itX_j))| = \left| \int_0^1 e^{itx} dx \right| = \left| \frac{e^{it} - 1}{it} \right| = \frac{\sqrt{2(1 - \cos t)}}{t}$$

więc

$$\lim_{|t| \rightarrow \infty} \sup |E(\exp(itX_j))| < 1$$

W celu zbudowania rozwinięcia (5.10) dla rozpatrywanej zmiennej losowej \bar{X}_N obliczymy kolejne momenty tej zmiennej. Rozważmy najpierw przypadek ogólny.

Niech $m = E(\xi)$ oraz niech $m_k = E((\xi - m)^k)$ będzie k -tym momentem centralnym zmiennej losowej ξ . Wtedy dla zmiennej losowej $\bar{\zeta}_N$ mamy

$$E(\bar{\zeta}_N) = m, \mu_2 = D^2(\bar{\zeta}_N) = m_2 / N, \mu_3 = m_3 / N^2, \mu_4 = 3m_2^2 / N^2 + (m_4 - 3m_2^2) / N^3, \\ \mu_5 = 10m_2m_3 / N^3 + (m_5 - 10m_2m_3) / N^4$$

i ogólnie

$$\mu_{2k-1} = 0(N^{-k}), \quad \mu_{2k} = 0(N^{-k})$$

W naszym przypadku dla zmiennej losowej X o rozkładzie równomiernym na przedziale $(0,1)$ jest: $E(X) = 1/2, m_2 = 1/12, m_3 = 0, m_4 = 1/80, m_5 = 0, \dots$ (wszystkie nieparzyste momenty centralne są równe zero). Dla zmiennej losowej \bar{X}_N otrzymujemy zatem: $\mu = E(\bar{X}_N) = 1/2, \sigma^2 = \mu_2 = 1/12N, \mu_3 = 0, \mu_4 = 1/80N^3 + (N-1)/48N^3, \mu_5 = 0, \dots$ Zgodnie z rozwinięciem (5.10) dla dystrybuanty zmiennej losowej \bar{X}_N otrzymujemy:

$$P\{\bar{X}_N \leq x\} = P\left\{ \frac{\bar{X}_N - \mu}{\sigma} \leq \frac{x - \mu}{\sigma} \right\} = \Phi\left(\left(x - \frac{1}{2} \right) \sqrt{12N} \right) - \frac{1}{20N} \phi^{(3)}\left(\left(x - \frac{1}{2} \right) \sqrt{12N} \right) + \dots \quad (5.11)$$

Wszystkie pozostałe wyrazy rozwinięcia aż do wyrazu zawierającego $\phi^{(8)}(x)$, są równe zero.

Postępowanie przy weryfikowaniu hipotezy $H: E(\bar{X}_N) = 1/2$ w stosunku do hipotezy alternatywnej $K: E(\bar{X}_N) \neq 1/2$ jest następujące (por. np. Fisz (1967)): obliczamy według wzoru (5.11) prawdopodobieństwo $P\{\bar{X}_N \leq \bar{x}_N\}$ (gdzie \bar{x}_N jest zaobserwowaną wartością zmiennej losowej \bar{X}_N) i porównujemy je z założonym poziomem istotności testu. Jeżeli to prawdopodobieństwo jest mniejsze od $\alpha/2$ albo większe od $1 - \alpha/2$, hipotezę H odrzucamy.

W analogiczny sposób konstruuje się test dla weryfikacji zgodności wariancji generatora z

wariancją teoretyczną. Ponieważ $D^2(X) = E(X^2) - (E(X))^2$, zgodność wariancji w ciągu liczb losowych z wariancją teoretyczną będziemy weryfikowali jako hipotezę o drugim momencie $E(X^2)$. Będziemy mianowicie rozważali statystykę

$$\overline{X}_N^2 = \sum_{j=1}^N X_j^2$$

Skorzystamy tu ze wszystkich podanych wyżej wyników dotyczących zmiennych losowych $\xi, \xi_1, \xi_2, \dots, \xi_N$, przy czym teraz $\xi_i = X_i^2$. Jeżeli X_i są zmiennymi losowymi o rozkładzie równomiernym na przedziale $(0,1)$, to zmienne losowe ξ_i mają gęstość prawdopodobieństwa równą $1/(2\sqrt{x})$ na tym przedziale i równą zero poza nim. Mamy więc $E(\xi) = 1/3$ i zadanie sprowadza się do zweryfikowania hipotezy $H: E(\overline{X}_N^2) = 1/3$ przy hipotezie alternatywnej $K: E(\overline{X}_N^2) \neq 1/3$. Powtarzając opisane wyżej obliczenia, otrzymujemy ostatecznie rozwinięcie

$$P\{\overline{X}_N^2 \leq x\} = \Phi\left(\left(x - \frac{1}{3}\right)\sqrt{\frac{45N}{4}}\right) - \frac{0,1065}{\sqrt{N}}\phi^{(2)}\left(\left(x - \frac{1}{3}\right)\sqrt{\frac{45N}{4}}\right) + \dots$$

Dalsze postępowanie jest takie samo, jak w przypadku weryfikowania hipotezy o wartości oczekiwanej.

Opisane procedury testowania można oczywiście przedłużyć również na wyższe momenty. Zauważmy, że testowanie momentów $E(X^k)$ jest równoważne ocenie generatora z punktu widzenia jego przydatności do obliczania metodą Monte Carlo całek $\int_0^1 x^k dx$.

5.5. Testy serii

Niech X_1, X_2, \dots, X_N będzie ciągiem liczb losowych o rozkładzie z dystrybuantą F . Oznaczmy przez X zmienną losową o rozkładzie z dystrybuantą F . Podzielmy zbiór wartości tej zmiennej losowej na dwa rozłączne podzbiory A i B i zdefiniujmy nową zmienną losową Y o wartościach a i b w następujący sposób:

$$Y = \begin{cases} a, & \text{gdy } X \in A \\ b, & \text{gdy } X \in B \end{cases} \quad (5.12)$$

Niech $p = P\{Y = a\}$; wtedy $P\{Y = b\} = 1 - p$.

Po przekształceniu każdego wyrazu ciągu według wzoru (5.12) otrzymujemy nowy ciąg zmiennych losowych Y_1, Y_2, \dots, Y_N . Oto przykładowa realizacja tego ciągu:

$$aa \ ba \ bbba \ bbb \quad (5.13)$$

Odcinkiem elementarnym ciągu postaci (5.13) nazwiemy każdy odcinek składający się z jednakowych elementów. *Serią* nazwiemy każdy taki odcinek elementarny, który przez przedłużenie w lewo lub w prawo przestaje być odcinkiem elementarnym. W przykładowym ciągu

(5.13) mamy więc następujące serie: aa, b, a, bbb, a, bbb .

Jest intuicyjnie oczywiste, że jeżeli zmienne losowe X_1, X_2, \dots, X_N są niezależne, to w ciągu (5.13) symbole a oraz b powinny być „dobrze przetasowane”; zbyt duża lub zbyt mała liczba serii w takim ciągu będzie świadczyła przeciwko hipotezie o niezależności. Te spostrzeżenia są punktem wyjścia do budowy dwóch testów opartych na statystykach związanych z seriami. Sformułujemy to dokładniej.

Niech y_1, y_2, \dots, y_N będzie pewną realizacją ciągu Y_1, Y_2, \dots, Y_N . Rozważmy najpierw takie ciągi y_1, y_2, \dots, y_N , w których jest n_a symboli a oraz $N - n_a = n_b$ symboli b . Wszystkich takich ciągów jest oczywiście $\binom{N}{n_a} = \binom{N}{n_b}$. Jeżeli hipoteza o niezależności jest prawdziwa, to prawdopodobieństwo każdej takiej realizacji jest jednakowe i oczywiście równe $1/\binom{N}{n_a}$. Rozkład liczby R serii przy tym warunku wyraża się wzorem

$$P\{R = r | n_a, n_b\} = \begin{cases} \frac{2 \binom{n_a - 1}{k - 1} \binom{n_b - 1}{k - 1}}{\binom{N}{n_a}} & \text{dla } r = 2k \\ \frac{\binom{n_a - 1}{k} \binom{n_b - 1}{k - 1} + \binom{n_a - 1}{k - 1} \binom{n_b - 1}{k}}{\binom{N}{n_a}} & \text{dla } r = 2k + 1 \end{cases}$$

Wzór ten i jego wyprowadzenie można znaleźć w różnych podręcznikach statystyki matematycznej (np. Fisz (1967), Domański (1990)).

Rozkład bezwarunkowy liczby R serii wyraża się wzorem

$$P\{R = r\} = \sum_{n_a=0}^{n_a+n_b} P\{R = r | n_a, n_b\} \binom{n_a + n_b}{n - a} p^{n_a} (1 - p)^{n_b}$$

Otrzymujemy

$$E(R) = 2Np(1 - p) + p^2 + (1 - p)^2 \quad (5.14)$$

$$D^2(R) = 4Np(1 - p)(1 - 3p(1 - p)) - 2p(1 - p)(3 - 10p(1 - p)) \quad (5.15)$$

Teoria weryfikowania hipotezy o niezależności zmiennych X_1, X_2, \dots, X_N oparta na liczbie R serii jest bardzo prosta: należy - dla ustalonego poziomu istotności α - znaleźć takie dwie wartości krytyczne R_1 oraz R_2 , żeby

$$P\{R < R_1\} = P\{R > R_2\} = \frac{\alpha}{2}$$

Jeżeli zaobserwowana liczba serii jest mniejsza od R_1 lub większa od R_2 , to weryfikowaną

hipotezę odrzucamy. Liczby R_1 i R_2 oblicza się rozwiązując równania

$$\sum_{j=0}^{R_1-1} P\{R = j\} = \frac{\alpha}{2}, \quad \sum_{j=R_2+1}^N P\{R = j\} = \frac{\alpha}{2}$$

Rozwiązywanie tych równań jest kłopotliwe, warto więc skorzystać z odpowiednich tablic (np. Domański (1990)). Dla dużych N dostatecznie dokładna do zastosowań praktycznych jest aproksymacja rozkładem normalnym z parametrami określonymi wzorami (5.14) i (5.15). Gdy $p = 1/2$, podane wzory znacznie się upraszczają, a dla dużych N rozkład statystyki R jest aproksymowany przez rozkład normalny ze średnią $N/2$ i z odchyleniem standardowym $\sqrt{N}/2$. W przypadku $p = 1/2$ zbiór wartości zmiennej losowej dzieli się zwykle na zbiór wartości większych niż mediana i zbiór wartości nie większych niż mediana (np. w rozkładzie równomiernym na przedziale (0,1) mediana jest równa 0.5). Takie serie nazywa się zwykle *seriami powyżej i poniżej mediany*.

Inny test jest oparty na tzw. *seriach monofonicznych*. Niech X_1, X_2, \dots, X_N będzie ciągiem liczb losowych o jednakowym, ciągłym rozkładzie prawdopodobieństwa. Rozważmy ciąg znaków $\text{zn}(X_2 - X_1), \text{zn}(X_3 - X_2), \dots, \text{zn}(X_N - X_{N-1})$, gdzie $\text{zn}(X) = +$, gdy $X > 0$ oraz $\text{zn}(X) = -$, gdy $X < 0$. Funkcja $\text{zn}(X)$ nie jest zdefiniowana dla $X = 0$, ale prawdopodobieństwo zdarzenia $X = 0$ przy założeniu ciągłości rozkładu zmiennych losowych X_1, X_2, \dots, X_N , jest równe zero. Ciąg znaków $+$ oraz $-$ możemy traktować jako ciąg dwóch symboli i rozważać różne statystyki związane z seriami jednakowych symboli. Sposób postępowania w takim przypadku dyskutowaliśmy już szczegółowo; omówienie teorii serii monofonicznych znaleźć można w pracy Edgingtona (1961), a tablice wartości krytycznych dla testu opartego na liczbie serii monotonicznych dostępne są w książce Domańskiego (1990). Zagadnienie zastosowania tego testu do badania generatorów liczb losowych opisano również w pracy Downhama (1969).

Niech L będzie liczbą serii monotonicznych w danym ciągu. Dla dużych N rozkład tej statystyki jest w przybliżeniu normalny ze średnią $(2N - 1)/3$ i wariancją $(16N - 29)/90$.

Rozważmy następujące zmienne losowe oparte na seriach symboli a i b : K_{1j} - liczba serii a długości j , K_{2j} - liczba serii b długości j , $S_{1r} = \sum_{j=r}^{n_a} K_{1j}$ - liczba serii a długości nie mniejszej niż r , $S_{2r} = \sum_{j=r}^{n_b} K_{2j}$ liczba serii b długości nie mniejszej niż r . Jeżeli hipoteza o niezależności jest prawdziwa, to

$$\begin{aligned} E(K_{1j}) &= (n - j)p^j(1 - p)^2 + p^j(1 - p)^2 \\ E(K_{2j}) &= (n - j)(1 - p)^j p^2 + (1 - p)^j(1 - p)^2 \\ E(S_{1r}) &= (n - r)p^r(1 - p) + p^r \\ E(S_{2r}) &= (n - r)(1 - p)^r p + (1 - p)^r \end{aligned}$$

Powyższe wzory są używane do weryfikowania hipotezy o niezależności za pomocą testu chi-kwadrat, który porównuje zaobserwowany rozkład liczby serii różnych długości z rozkładem teoretycznym. Przykłady użycia tego typu testów do badania generatorów liczb losowych znaleźć można w pracy Góralskiego (1978).

5.6. Testy kombinatoryczne

5.6.1. Test pokerowy

Opisana w tym podrozdziale grupa testów należy do tzw. *testów niezależności (losowości próby)*. Sprawdzamy, czy kolejno produkowane przez generator liczby są niezależnymi zmiennymi losowymi, czyli weryfikujemy hipotezę, że zmienna losowa (X_1, X_2, \dots, X_N) ma rozkład o dystrybuancie postaci

$$H(x_1, x_2, \dots, x_n) = F(x_1)F(x_2) \dots F(x_n)$$

Niech X_1, X_2, \dots, X_N będzie ciągiem liczb z pewnego generatora liczb losowych o rozkładzie z dystrybuantą F . Podzielmy zbiór wartości zmiennych losowych X na k rozłącznych „jednakowo prawdopodobnych” przedziałów za pomocą punktów $a_0 < a_1 < \dots < a_{k-1} < a_k$. Jeżeli zmienne losowe X_j mają rzeczywiście rozkład z dystrybuantą F , to dla każdego przedziału (a_{i-1}, a_i)

$$P\{a_{i-1} < X_j \leq a_i\} = \frac{1}{k}$$

Utwórzmy nowy ciąg zmiennych losowych Y_j zdefiniowanych wzorem

$$Y_j = i, \quad \text{jeżeli} \quad X_j \in (a_i, a_{i+1}), \quad i = 0, 1, \dots, k-1$$

Zmienne losowe Y_j przyjmują więc tylko wartości $0, 1, 2, \dots, k-1$, każdą z jednakowym prawdopodobieństwem.

Podzielmy ciąg $Y_1 Y_2, \dots$ na piątki (Y_1, Y_2, \dots, Y_5) , $(Y_6, Y_7, \dots, Y_{10}), \dots$. Ten nowy ciąg zbudowany jest z k^5 różnych piątek. Będziemy wyróżniali następujące typy piątek:

abcde (bust) - każda liczba w piątce jest inna
aabcd (para) - w piątce są dwie liczby jednakowe, wszystkie pozostałe są różne
aabbc (dwie pary)
aaabc (trójka)
aaabb (ful)
aaaab (czwórka)
aaaaa (piątka)

Przy założeniu, że każda z liczb $0, 1, \dots, k-1$ pojawia się w ciągu Y_1, Y_2, \dots z jednakowym prawdopodobieństwem oraz, że poszczególne wyrazy tego ciągu są niezależne, rozkład prawdopodobieństwa na wymienionym wyżej zbiorze wyróżnionych piątek opisuje się następującymi wzorami:

$$\begin{aligned}
P\{(abcde)\} &= \begin{cases} (k-1)(k-2)(k-3)(k-4)/k^4 & \text{dla } k \geq 5 \\ 0 & \text{dla } k < 5 \end{cases} \\
P\{(aabcd)\} &= \begin{cases} 10(k-1)(k-2)(k-3)/k^4 & \text{dla } k \geq 4 \\ 0 & \text{dla } k < 4 \end{cases} \\
P\{(aabbc)\} &= \begin{cases} 15(k-1)(k-2)/k^4 & \text{dla } k \geq 3 \\ 0 & \text{dla } k < 3 \end{cases} \\
P\{(aaabc)\} &= \begin{cases} 10(k-1)(k-2)/k^4 & \text{dla } k \geq 3 \\ 0 & \text{dla } k < 3 \end{cases} \\
P\{(aaabb)\} &= \begin{cases} 10(k-1)/k^4 & \text{dla } k \geq 2 \\ 0 & \text{dla } k < 2 \end{cases} \\
P\{(aaaaab)\} &= \begin{cases} 5(k-1)/k^4 & \text{dla } k \geq 2 \\ 0 & \text{dla } k < 2 \end{cases} \\
P\{(aaaaaa)\} &= \begin{cases} 1/k^4 & \text{dla } k \geq 2 \\ 0 & \text{dla } k < 2 \end{cases}
\end{aligned}$$

Wartościami najczęściej stosowanymi w praktyce są wartości $k = 2, 8, 10$. Zgodność rozkładu piętek różnych typów sprawdza się za pomocą zwykłego testu chi-kwadrat.

5.6.2. Test kolekcjonera

Utwórzmy ciąg Y_1, Y_2, \dots , tak jak dla testu pokerowego. Będziemy obserwowali ten ciąg dopóty, dopóki nie pojawia się w nim wszystkie k liczby $0, 1, 2, \dots, k-1$. Niech R będzie długością zaobserwowanego odcinka ciągu. Można udowodnić (por. np. Greenwood (1955)), że zmienna losowa R ma rozkład określony wzorem

$$P\{R = r\} = \frac{1}{k^{r-1}} \sum_{j=0}^{k-2} (-1)^j \binom{k-1}{j} (k-1-j)^{r-1}, \quad r = k, k+1, \dots$$

Zgodność tego rozkładu z rozkładem zaobserwowanym bada się za pomocą standardowego testu chi-kwadrat.

5.6.3. Test kolizji i test liczby pustych cel

Rozpatrzmy ciąg utworzony z nk kolejnych liczb z generatora

$$(X_1, X_2, \dots, X_k), (X_{k+1}, \dots, X_{2k}), \dots, (X_{k(n-1)+1}, \dots, X_{nk}) \quad (5.16)$$

Rozważmy podział k -wymiarowej kostki $[0, 1]^k$ na $m = s^k$ mniejszych kostek k -wymiarowych o jednakowych objętościach równych $1/m$. Za statystykę testową przyjmuje się tzw. *liczbę kolizji* C , tzn. liczbę przypadków, w których kolejny punkt z ciągu (5.16) wpada do kostki zajętej już przez jeden lub więcej punktów tego ciągu. Przy założeniu, że ciąg zmiennych losowych X_1, X_2, \dots jest

ciągami niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym $U(0,1)$, statystyka C ma rozkład określony wzorem

$$P\{C = c\} = \frac{m(m-1)\dots(m-n-c+1)}{m^n} \binom{n}{n-c}, \quad c = 0,1,2,\dots,n$$

Zgodność tego rozkładu z rozkładem zaobserwowanym bada się za pomocą testu chi-kwadrat.

W omawianej sytuacji można rozważać również statystykę C' , zdefiniowaną jako liczba kostek w których nie ma ani jednego elementu ciągu (5. 16). Test skonstruowany na podstawie statystyki C' znany jest w literaturze pod nazwą *testu liczby pustych cel* (ang. *the empty celi test* - patrz Wilks (1963)), a w polskiej literaturze ekonometrycznej występuje jako *test Hellwiga* lub *test Davida-Hellwiga*. Rozkład prawdopodobieństwa statystyki C' przy założeniu, że X_1, X_2, \dots jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym $U(0, 1)$, wyraża się wzorem

$$P\{C' = c\} = \binom{m}{c} \sum_{i=0}^{m-c} (-1)^i \binom{m-c}{i} \left(1 - \frac{c+i}{m}\right)^n, \quad c = 0,1,2,\dots,m$$

Tablice dystrybucyj i wartości krytycznych dla statystyki C' można znaleźć w pracach Hellwiga (1987) oraz Domańskiego (1990).

5.6.4. Test permutacji

Niech X_1, X_2, \dots - będzie ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie równomiernym na przedziale $(0,1)$ i rozpatrzmy ciąg punktów

$$(X_1, X_2, \dots, X_m), (X_{m+1}, X_{m+2}, \dots, X_{2m}), \dots$$

Przekształćmy każdy z punktów, zastępując współrzędne ich rangami wśród wszystkich współrzędnych danego punktu. Na przykład punkt $(0.72, 0.23, 0.47, 0.91, 0.50)$ przekształca się w opisany wyżej sposób na punkt $(4,1,2,5,3)$. Jeżeli ciąg X_1, X_2, \dots jest ciągiem niezależnych zmiennych losowych, to każda permutacja (n_1, n_2, \dots, n_m) liczb $(1, 2, \dots, m)$ jest jednakowo prawdopodobna. Testem zgodności chi-kwadrat weryfikuje się hipotezę, że tak jest rzeczywiście.

5.6.5. Test oparty na rzędzie losowych macierzy binarnych

W takich dziedzinach jak kombinatoryka i teoria grafów analizuje się macierze losowe o elementach 0 lub 1. W niniejszym teście korzysta się z następującego faktu dotyczącego takich macierzy: rząd losowej macierzy binarnej o wymiarach $m \times n$ przyjmuje wartości $r = 1, 2, \dots, \min(m, n)$ z prawdopodobieństwami

$$2^{r(n+m-r)-mn} \prod_{i=0}^{r-1} (1 - 2^{i-n}) \left(1 - \frac{2^{i-m}}{1 - 2^{i-r}}\right)$$

Na przykład dla $m = n$ oraz $n \geq 10$ rząd losowej macierzy binarnej jest równy m z

prawdopodobieństwem w przybliżeniu 0.30.

Marsaglia (1984) przytacza wyniki badań, z których wynika, że generatory oparte na rejestrach przesuwanych oraz generatory typu $F(r, s, \text{xor})$ zwykle nie spełniają tego testu, natomiast w większości przypadków spełniają go generatory $F(r, s, +)$ i $F(r, s, -)$.

5.7. Testowanie generatorów za pomocą zadań kontrolnych

Testowanie generatorów *za pomocą zadań kontrolnych* polega na rozwiązywaniu metodami Monte Carlo określonych zadań za pomocą liczb losowych z badanego generatora i porównywaniu otrzymanych wyników z wynikami otrzymanymi w inny sposób. Typowymi są tu zadania szacowania liczby π lub pewnych innych stałych matematycznych oraz zadania obliczania niektórych całek, np. zadanie szacowania objętości kuli jednostkowej w przestrzeni m -wymiarowej.

Estymator liczby π otrzymuje się za pomocą zadania Buffona (por. np. Ripley (1987)). Niech X_1, X_2, \dots będzie ciągiem liczb z badanego generatora liczb losowych o rozkładzie równomiernym na przedziale $(0, 1)$ i rozpatrzmy ciąg $(X_1, X_2), (X_3, X_4), \dots$. Każdej parze (X_{2j-1}, X_{2j}) , $j = 1, 2, \dots$, przyporządkujemy zdarzenie *sukces*, gdy $X_{2j-1} \leq 1/2 \cos(\pi/2 X_{2j})$ lub zdarzenie *porażka* w przypadku przeciwnym. Niech M będzie liczbą sukcesów w ciągu N par. Wtedy stosunek M/N jest nieobciążonym estymatorem liczby $\pi^{-1} = 0.3183098862\dots$. Wariancja estymatora M/N jest oczywiście równa $\pi^{-1}(1 - \pi^{-1})/N$, a dla dostatecznie dużych N rozkład tego estymatora można aproksymować rozkładem normalnym. Na tej podstawie można weryfikować generator, sprawdzając, czy obliczona z użyciem liczb losowych z tego generatora wartość M/N różni się istotnie od znanej wartości π^{-1} . Bardziej efektywne estymatory liczby π oraz listę prac źródłowych poświęconych tym problemom można znaleźć u Zielińskiego (1970).

Objętość kuli jednostkowej w przestrzeni m -wymiarowej wyraża się znanym wzorem $V_m = 2\pi^{m/2}/(m\Gamma(m/2))$. Rozważmy ciąg $(X_1, X_2, \dots, X_m), (X_{m+1}, X_{m+2}, \dots, X_{2m}), \dots$, gdzie X_1, X_2, \dots jest ciągiem liczb losowych o jednakowym rozkładzie równomiernym na przedziale $(-1, 1)$, i każdemu wyrazowi $(X_{(j-1)m+1}, X_{(j-1)m+2}, \dots, X_{jm})$, $j = 1, 2, \dots$, tego ciągu przyporządkujemy zdarzenie *sukces*, gdy $X_{(j-1)m+1}^2 + X_{(j-1)m+2}^2 + \dots + X_{jm}^2 \leq 1$, lub zdarzenie *porażka* w przypadku przeciwnym. Niech M będzie liczbą sukcesów w ciągu N -elementowym. Wówczas $2^m M/N$ jest nieobciążonym estymatorem liczby V_m . Na podstawie zaobserwowanej wartości statystyki $2^m M/N$ wnioskujemy o generatorze analogicznie jak w opisanym poprzednio przypadku szacowania stałej π^{-1} .

Opisane postępowanie można oczywiście zastosować do wielu innych zadań numerycznych, a przedstawione wyżej dwa przypadki są tylko przykładami.

Do testowania liczb pseudolosowych przydatne okazały się również pewne zjawiska fizyczne, dla których znane są charakteryzujące je wielkości. Parametry fizyczne badanego zjawiska można oszacować za pomocą odpowiedniego modelu symulacyjnego, a następnie porównać uzyskane wyniki ze znanymi wielkościami. Przykłady takich testów fizycznych i ich zastosowanie do badania kilku popularnych generatorów podał Yattulainen (1994). W tej grupie testów efektywny okazał się np. test oparty na zjawisku błędzenia przypadkowego, w którym modeluje się ruch Browna na płaszczyźnie.

6.Prace cytowane

- AFFERBACH L., GROTHE H.(1985): Calculation of Minkowski-Reduced Lattice Bases. *Computing* 35, s. 269-276.
- AHRENS J.H., DIETER U.(1974): Computer methods for sampling from gamma, beta, Poisson and binomial distributions. *Computing* 12, s. 223-246.
- ANDERSON S.L.(1990): Random number generators on vector supercomputers and other advanced architectures. *S/AM Review* 32,2, s. 221-251.
- ATKINSON A.C.(1979a): The computer generation of Poisson random variables. *Appl. Statist.* 28, s. 29-35.
- ATKINSON A.C.(1979b): Recent developments in the computer generation of Poisson random variables. *Appl. Statist.* 28, s. 260-263.
- BANERUA S., DWYER R.A.(1993): Generating Random Points in a Bali. *Commun. Statist. Simula.* 22(4), s. 1205-1209.
- BARBU G.(1982): On computer generation of a random variable by transformation of uniform variables. *Buli. Mati. Soc. Sci. Math. Romaniae* 26, s. 129-139. BERDNIKOV A.S., COMPAGNER A., TURCIA S.B.(1996): A MathLink Program for High-Quality Random Numbers. *Tie Mathematica Journal* 6(3), s. 65-69. BEST D.J.(1983): A note on gamma variate generators with shape parameter less than unity. *Computing* 3, s. 185-188.
- BIALYNICKI-BIRULA A.(1980): *Algebra*. Wyd.3, PWN Warszawa.
- BLUM L., BLUM M., SHUB M.(1986): A simple unpredictable pseudo-random number generator. *S/AM Journal on Computing* 15, s. 364-383.
- BOROSH I., NIEDERREITER H.(1983): Optima! multipliers for pseudorandom number generation by the linear congruential method. *BIT* 23, s. 65-74.
- BOSWELL M.T., GORE S.D., PATIL G.P., TAILLIE C.(1993): *The art of computer generation of random variables*. In: Handbook of Statistics, ed. Rao, C.R., Vol.9. Elsevier Science Publishers.
- BOX G.E.P., MULLER M.E.(1958): A note on the generation of random normal deviates. *Ann. Math. Statist* 29, s. 610-611.
- BRACHA Cz.(1996): *Teoretyczne podstawy metody reprezentacyjnej*. PWN Warszawa.
- BRATLEY P., FOX B.L., SCHRAGE L.E.(1987): *A guide to simulation*. Springer-Yerlag, New York, 2nd ed.
- BRIGHT H.S., ENISON R.L.(1979): Quasi-random number sequences from a long-period TLP generator with remarks on application to cryptography. *Comp. Surv.* 11, s. 357-370. BROWN M., SOLOMON H.(1979): On combining pseudorandom number generators. *Ann. Statist.* 1, s. 691-695.
- BURR I.W. (1942): Cumulative frequency distributions. *Ann. Math. Statist.* 13, s. 215-232.
- BUTCHER J.C. (1960): Random sampling from the normal distribution. *Computer J.* 3, s. 251-253.
- BUTLER J.W.(1956): *Machine sampling from given probability distribution*. In: Sympo-sium on Monte Carlo Methods, H.A.Meyer (ed.), Wiley, Chapman & Hali. CARTA D.C.(1990): Two fast implementations of the „minimal standard" random number generators. *Comm. ACM* 33, s. 87-88.
- CHAMBERS J.M., MALLOWES C.L., STUCK B.W.(1976): A method for simulating stable random variables. *J. Amer. Statist. Assoc.* 71, s. 340-344. (Patrz również poprawka w *J. Amer. Statist. Assoc.* (1987), 82, 704) CHEN H.C., ASAU Y.(1974): On generating random variates from an empirical distribution. *AHE Transactions* 6, s. 163-166.
- CHENG R.C.H.(1977): The Generation of Gamma Variables with Non-Integral Shape Parameter. *Appl. Statist.* 26, s. 71-75.
- CHENG R.C.H.(1978): Generating beta variates with nonintegral shape parameters. *Comm. ACM* 21,4, s. 317-322.
- COLLINGS B.J., HEMBREE G.B.(1986): Initializing generalized feedback shift register pseudo-random generators. *J. ACM* 33, s. 706-711.
- COMPAGNER A.(1995): Operational conditions for reliable random-number generation. *Phys. Rev. E* 52, s. 5634-5645.
- COYEYOU, R.R., MAC PHERSON, R.D.(1967): Fourier analysis of uniform random numbers generators. *J. ACM* 14, s. 100-119.
- CRAMER, H.(1958): *Metody matematyczne w statystyce*. PWN Warszawa.
- DE MATTEIS A., PAGNUTTI S.(1988): Parallelization of random number generators and long-range correlations. *Num. Math.* 53, s. 595-608.
- DENG G.(1990): Generation of uniform variates from several nearly uniform distributed variables. *Commun. Statist.-Simula.* 19(1), s. 145-154.
- DEYROE L.(1986): *Non-uniform random variate generation*. Springer-Yerlag. DOMAŃSKI C.(1990): *Testy statystyczne*. PWE Warszawa.
- DOWNHAM D.Y.(1969): The Runs Up and Down Test. *Computer J.* 12, s. 373-376. DUDEWICZ E.J., RALLEY T.G.(1981): *The Handbook of Random Number Generation and Testing with TESTRAND Computer Code*. American Science Press, Columbus, Ohio. DWYER J.(1995): Quick and Portable Random Number Generators. *C/C ++ Users J.* 13, 6, s. 33-44.
- EDIGINGTON, E.S.(1961): Probability Table for Number of Runs of Signs of First Differences in Ordered Series. *J. Amer. Statist. Assoc.* 56, s. 156-159.
- EICHENAUER-HERMANN J.(1991): Inversive congruential pseudorandom numbers avoid the planes. *Math. Comput.* 56, s. 297-301.
- EICHENAUER-HERMANN J.(1993a): Statistical independence of a new class of inversive congruential pseudorandom numbers. *Math. Comput.* 60, s. 375-384. EICHENAUER-HERMANN J.(1993b): Explicit inversive congruential pseudorandom numbers: The compound approach. *Computing* 51, s. 175-182.
- EICHENAUER-HERMANN J.(1994): On generalized inversive congruential pseudorandom numbers. *Math. Comput.* 63, s. 293-299.
- EICHENAUER-HERMANN J.(1995): Pseudorandom number generation by nonlinear methods. *intern. Statist. Rev.* 63, s. 247-255.
- EICHENAUER-HERMANN J., GROTHE H., LEHN J.(1989): On the period length of pseudo-random vector sequences generated by matrix generators. *Math. Comput.* 52, s. 145-148. EICHENAUER J., LEHN J.(1986): A non-linear congruential pseudo random number generator. *Statist. Pap.* 27, s. 315-326.
- FINCKE U., POHST M.(1985): Improved methods for calculating vectors of short length in a lattice, including a complexity

- analysis. *Math. Comput.* 44, s. 463-471. FISHMAN G.S., MOORE L.S.(1982): A statistical evaluation of multiplicative congruential random number generators with modulus 2^{31} —1. *J. Amer. Statist. Assoc.* 77, s. 129-136. FISHMAN G.S., MOORE L.S. (1986): An exhaustive analysis of multiplicative congruential random number generators with modulus 2^{31} —1. *SIAM J.Sci.Statist.Comput.* 7, s. 24-45. FISHMAN G.S.(1990): Multiplicative congruential random number generators with modulus 2^a : an exhaustive analysis for $a = 32$ and a partial analysis for $a = 48$. *Math. Comput.* 54, s. 331-344.
- Fisz M. (1967): *Rachunek prawdopodobieństwa i statystyka matematyczna*. Wyd. 2. PWN Warszawa.
- GNIEDENKO B., KOLMOGOROW A.N.(1957): *Rozkłady graniczne sum zmiennych losowych niezależnych*. PWN Warszawa.
- GOLOMB S.W.(1967): *Shift Register Sequences*. Holden-Day, San Francisco.
- GOOD I.J.(1953): The Serial Test for Sampling Numbers and Other Tests for Randomness. *Proc. Camb. Phil. Soc.* 49, s. 276-284.
- GÓRALSKI A.(1974): *Metody opisu i wnioskowania statystycznego w psychologii*. PWN Warszawa.
- GRAHAM R.L., KNUTH D.E., PATASHNIK O. (1996): *Matematyka konkretna*. PWN Warszawa.
- GREENBERGER M.(1961): An a priori determination of serial correlation in computer generated random numbers. *Math. Comput.* 15, s. 383-389 (Corrigenda 16, 1962, 126).
- GREENWOOD R.E.(1955): Coupon Collector's Test For Random Digit. *Mati. Tab. OtherAids Comp.* 9,1-5, s. 224-229.
- GRUENBERGER F., MARK A.M.(1951): The d^2 Test of Random Digits. *Math. Tab. OtherAids Comp.* 5, s. 109-110.
- HAMMER P.C.(1951): The mid-square method of generating digits. *JVat. Bur. Stand. Appl.Math. Ser.* 12, s. 33-39.
- HAMMERSLEY J.M., HANDSCOMB D.C.(1961): *Monte Carlo Methods*. Methuen - Wiley, London - New York.
- HELLWIG Z.(1987): *Elementy rachunku prawdopodobieństwa i statystyki matematycznej*. Wyd. 10, PWN Warszawa.
- JANICKI A., WERON A.(1994): *Simulation and Chaotic Behavior of Stable Stochastic Processes*. Marcel Dekker, New York.
- JANSSON B.(1964): Autocorrelation between pseudo-random numbers. *BIT* 4, s. 6-27.
- JANSSON B.(1966): *Random number generators*. Stockhohn, Yictor Pettersons Bokindustri Aktiebolag.
- JERMAKOW S.M.(1976): *Metoda Monte Carlo i zagadnienia pokrewne*. PWN Warszawa. JÖHNK M.D.(1964): Erzeugung von betaverteilten und gammaverteilten Zufallszahlen. *Metriia* 8, s. 5-15.
- KACHITYICHYANUKUL V., SCHMEISER B.W.(1988): Binomial Random Yariate Generation. *Comm. ACM* 31,2, s. 216-222.
- KENDALL M.G., STUART A.(1961): *The Advanced Theory of Statistics*. Vol. 2, Griffin, London (wyd. roś. 1973).
- KINDERMAN J.M., MONAHAN J.F.(1977): Computer generation of random variables using the ratio of uniform deviates. *ACM Trans. Math. Soft.* 3, s. 257-260.
- KINDERMAN A.J., RAMAGE, J.G.(1976): The computer generation of normal random variables. *J. Amer. Statist. Assoc.* 71, s. 893-896.
- KIRKPATRICK S., STOLL E.(1981): A very fast shift-register sequence random number generator. *J. Comput. Phys.* 40, s. 517-526.
- KNUTH D.E. (1981): *The Art of Computer Programming. Vol 2: Seminumerical Algorithms*. Addison-Wesley, Reading, MA. Koc C.(1995): Recurring-with-carry sequences. *J. Appl. Probl.* 32, s. 966-971.
- KOROLUK W.S., PORTENKO N.I., SKOROCHOD A.W., TURBIN A.F.(1985): *Sprawocznik po teorii wiorojatnostiej i matematycznej statistikie*. Wyd. 2. Nauka Moskwa.
- KRONMAL R.A., PETERSON A.V.(1981): A variant of the acceptance-rejection method for computer generation of random variables. *J. Amer. Statist. Assoc.* 76, s. 446-451.
- KRONMAL R.A., PETERSON A.V.(1984): An acceptance-complement analogue of the mixture-plus-acceptance-rejection method for generating random variables. *ACM Trans. Math. Soft.* 10, s. 271-281.
- KRYSICKI W., BARTOS J., DYCZKA W., KRÓLIKOWSKA K., WASILEWSKI M.(1989): *Rachunek prawdopodobieństwa i statystyka matematyczna w zadaniach*. Cz. I, PWN Warszawa.
- KRYSICKI W., BARTOS J., DYCZKA W., KRÓLIKOWSKA K., WASILEWSKI M.(1994): *Rachunek prawdopodobieństwa i statystyka matematyczna w zadaniach*. Cz. II. Wyd.2. PWN Warszawa.
- L'ECUYER P.(1988): Efficient and portable combined random number generators. *Comm. ACM* 31, s. 742-749,774.
- L'ECUYER P.(1990): Random numbers for simulation. *Comm. ACM* 33(10), s. 85-97. L'EcuYER P.(1995): *Testing Random Number Generators*. Artykuł dostępny w Internecie pod adresem <http://random.mat.sbg.ac.at>.
- L'ECUYER P.(1996a): *Bad lattice structures for vectors of non-succesive values produced by some linear recurrences*. Artykuł dostępny w Internecie pod adresem <http://random.mat.sbg.ac.at> - ukaże się w ORSA J.Computing.
- L'ECUYER P.(1996b): *Combined multiple recursive generators*. Artykuł dostępny w Internecie pod adresem <http://random.mat.sbg.ac.at> - ukaże się w Operations Research.
- L'ECUYER P., BLOUIN F., COUTURE R.(1993): A search for good multiple recursive random number generators. *ACM Trans. Mód. Comp. Sim.* 3, s. 87-98. L'ECUYER P., COTE S.(1991): Implementing a random number package with splitting facilities. *ACM Trans. Math. Soft.* 17,1, s. 98-111.
- L'EcuYER P., TEZUKA S.(1991): Structural properties for two classes of combined random number generators. *Math. Comput.* 57, s. 735-746.
- LEEB H.(1991): On the Digit Test. Artykuł dostępny w Internecie pod adresem: <http://random.mat.sbg.ac.at>.
- LEWIS T.G., PAYNE W.H.(1973): Generalized feedback shift register pseudorandom number algorithms. *J. ACM* 20, s. 456-468.
- MACLAREN N.M.(1992): A limit on the usable length of a pseudorandom sequence. *J. Statist. Comput. Sim.* 42, s. 47-54.
- MACLAREN M.D., MARSAGLIA G.(1965): Uniform random number generators. *J. ACM* 12, s. 83-89.
- MARSAGLIA G.(1962): Random VariaWes and Computers. **In:** Transactions of the Third Prague Conference 1962. Information Theory, Statistical Decision Functions, Random Processes.
- MARSAGLIA G.(1968): Handom numbers fail mainly in the planes. *Proc. Nat. Acad. Sci. USA* 61, s. 25-28.
- MARSAGLIA G.(1972): *The structure of lincair cougrucntial sequuces*. **In:** Applications of Number Theory to Nimierical Analysis. Ed. S.K. Zaremba, Academic Press, londou, s. 249-285
- MARSAGLIA G.(1984): A current vicw of raudoin number generators. **In:** Computer Science and Statistics: 16th Symposium iu the Interface, Atlanta.
- MARSAGLIA G.(1993): Monkey tests for raudom number generatora. *Comput. Matb- Appl.* 9, s. 1-10.
- MARSAGLIA G.(1995): *The Managlia ran dom number CD-ROM, including the DIEHARD battery of tssts of randomness*. Department of Statistics and Supercomputer Computa-tions Research. Institute, Florida State

University.

- MARSAGLIA G., BRAY T.A.(1964): A convenient method for generating normal variates. *SIAM Review* 6, s. 260-264.
- MARSAGLIA G., BRAY T.A.(1968): One-line random number generators and their use in combination. *Commun. ACM* 11, s. 757-759.
- MARSAGLIA G., TSAY L.H.(1985): Matrices and the structure of random number sequences, *Linear Alg. Appl.* 67, s. 147-156.
- MARSAGLIA G., ZAMAN A., TSANG W.W.(1990): Toward a universal random number generator. *Statist. Probl. Lett. S*, s. 35-39.
- MARSAGLIA G., ZAMAN A.(1991): A new class of random number generators. *Ann. Appl. Probab.* 1,3, s. 462-480.
- NIOMANN, VOM J.(1951): Various Techniques Used in Connection With Random Digits. *Monte Carlo Method - Proceedings of a symposium held June, 29,30, and July 1, 1949, in Los Angeles, California., National Bureau of Standards, Appl. Math. Series* 12, s. 36-38.
- NIEDERREITER H.(1994): On a new class of pseudorandom numbers for simulation methods. *J. Comp. Appl. Math.* 56, s. 159-167.
- NIEDERREITER H., SHU P.J.S., EDS.(1995): *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. Springer.
- ODEH R.E., EYANS J.O.(1974): The Percentage Points of the Normal Distribution. *Appl. Statist.* 23, s. 96-97.
- PARK S.K., MILLER K.W.(1988): Random numbers generators: Good ones are hard to find. *Comm. ACM* 31, s. 1192-1200.
- Random numbers (1991-1995): Opracowanie dostępne w formie elektronicznej w Internecie.
- RIPLEY B.D.(1983): The lattice structure of pseudo-random number generators. *Proc. Roy. Soc. A* 389, s. 197-204.
- RIPLEY B.D.(1987): *Stochastic simulation*. Wiley.
- RIPLEY B.D.(1990): Thoughts on pseudorandom number generators. *J. Comput. Appl. Math.* 31, s. 153-163.
- SCHNEIDER B.(1995): *Kryptografia dla praktyków. Protokoły, algorytmy i programy źródłowe w języku C*. WNT Warszawa.
- SZCZURA A., ZIELIŃSKI R.(1993): Generatory liczb losowych o rozkładzie gamma. *Roczniki PTM, Seria Ul: Matematyka Stosowana XXXVI*, s. 99-114.
- TAUSSKY O., TODD J.(1956): *Generation and Testing of Pseudo-Random Numbers*. In: Symposium on Monte Carlo Methods. H.A.Meyer (Ed.), Wiley, Chapman & Hall.
- TAUSWORTHE H.C.(1965): Random numbers generated by linear recurrence modulo two. *Math. Comp.* 19, s. 201-209.
- TEZUKA S.(1994): A unified view of large-period random number generators. *J. Oper. Res. Soc. Japan* 37, s. 211-227.
- TEZUKA S.(1995): *Uniform random numbers. Theory and practice*. Kluwer Academic Publishers, Boston, Dordrecht, London.
- TOOTILL J.P.R., ROBINSON W.D., EAGLE D.J.(1973): An asymptotically random Tausworthe sequence. *J. ACM* 20, s. 469-481.
- YATTULAINEN I., ALA-NISSILA T., KANKAALA K.(1994): Physical tests for random numbers in simulations. *Physical Review Letters* 70(19), s. 2513-2520.
- WERON R.(1996a): On the Chambers-Mallows-Stuck Method for Simulating Skewed Stable Random Variables. *Statist. Probl. Lett.* 28, s. 165-171.
- WERON R.(1996b): A note on the Chambers-Mallows-Stuck Method for Simulating Skewed Random Variables. Res. Rep., TU Wrocław.
- WICHMANN B.A., HILL I.D.(1982): An efficient and portable pseudorandom number generator. *Appl. Statist.* 31, s. 188-190.
- (Patrz również poprawki i uwagi w tym samym czasopiśmie: WICHMANN, HILL (1984) 33, s. 123; McLEOD (1985) 34, s. 198-200; ZEISEL (1986) 35, s. 89.)
- WIECZORKOWSKI, R.(1995): Algorytmy stochastyczne w optymalizacji dyskretnej przy zaburzonych wartościach funkcji. *Matematyka Stosowana XXXVIII*, s. 119-153.
- WILKS S.(1963): *Mathematical Statistics*. New York.
- ZASĘPA R.(1962): *Badania statystyczne metodą reprezentacyjną. Zarys teorii i praktyki*. PWN Warszawa.
- ZASĘPA R.(1972): *Metoda reprezentacyjna*. PWE Warszawa.
- ZIEMŃSKI R.(1966): Generator liczb losowych o rozkładzie równomiernym dla maszyny ZAM-2. *Algorytmy Q*, s. 103-125.
- ZIELIŃSKI R.(1970): *Metody Monte Carlo*. WNT Warszawa.
- ZIELIŃSKI R.(1972): Generatory liczb losowych. WNT Warszawa, (1979 wyd. 2). ZIELIŃSKI R., NEUMANN P.(1986): *Stochastyczne metody poszukiwania minimum funkcji*. WNT Warszawa.
- ZIELIŃSKI R., ZIELIŃSKI W.(1987): *Podręczne tablice statystyczne*. WNT Warszawa.
- ZIELIŃSKI R., ZIELIŃSKI W.(1990): *Tablice statystyczne*. PWN Warszawa.
- ZIELIŃSKI R.(1990): *Siedem wykładów wprowadzających do statystyki matematycznej*. PWN Warszawa.