3.4. OSTATNÍ TYPY NÁHODNÝCH VELIČIN

VIDĚLI JSME, jak na počítači vygenerovat posloupnost čísel $U_0, U_1, U_2, \ldots$, která se chová stejně, jako bychom každé z nich vybrali náhodně a nezávisle z intervalu od 0 do 1 s rovnoměrným rozdělením. Aplikace náhodných čísel si ovšem často žádají jiné typy rozdělení; jestliže například provádíme náhodný výběr z k možností, potřebujeme náhodné *celé číslo* od 1 do k. Pokud v nějakém simulačním procesu potřebujeme náhodnou dobu čekání mezi výskyty nezávislých událostí, znamená to náhodné číslo s *exponenciálním rozdělením*. Někdy dokonce nepotřebujeme náhodná *čísla*, ale náhodnou *permutaci* (tedy náhodné uspořádání n objektů), nebo náhodnou *kombinaci* (náhodný výběr k objektů či prvků z množiny n).

Principiálně můžeme každou z těchto ostatních náhodných proměnných neboli veličin snadno získat z rovnoměrných veličin $U_0, U_1, U_2, \ldots$; lidé již navrhli celou řadu důležitých „náhodných triků“ pro efektivní transformaci těchto rovnoměrných veličin. Při zkoumání těchto technik nahlédneme také do správného užití náhodných čísel v jakékoli aplikaci typu Monte Carlo. (Jak jistě bystrý čtenář nahlédne, je výraz „rovnoměrná veličina“ stručnějším označením proměnné nebo veličiny s rovnoměrným rozdělením; podobnou logikou jsou vedeny i výrazy „normální veličina“, „exponenciální veličina“ apod. Pozn. překl.)

Lze si představit, že jednoho dne někdo vynalezne generátor náhodných čísel, jenž bude vytvářet tyto ostatní náhodné veličiny *přímo*, nikoli nepřímo z rovnoměrného rozdělení. Zatím ale žádná přímá metoda není příliš praktická, kromě generátoru „náhodných bitů“ popsaného v části 3.2.2. (Viz též cvičení 3.4.1–31, kde rovnoměrné rozdělení používáme zejména pro inicializaci, po níž běží metoda téměř celá přímo.)

Výklad v následující části předpokládá existenci náhodné posloupnosti rovnoměrně rozdělených reálných čísel od 0 do 1. Kdykoli potřebujeme nové číslo, vygenerujeme novou rovnoměrnou veličinu U. Tato čísla se obvykle reprezentují pomocí počítačového slova, přičemž úplně vlevo se předpokládá řádová čárka.

3.4.1. Číselná rozdělení

V této části textu shrneme nejlepší známé techniky pro vytváření čísel z různých důležitých rozdělení. Mnohé z metod původně navrhl začátkem 50. let John von Neumann a jiní lidé je postupně zdokonalovali, především George Marsaglia, J. H. Ahrens a U. Dieter.

A. Náhodné výběry z konečné množiny. Nejjednodušším a nejběžnějším typem v praxi používaného rozdělení je náhodné *celé číslo*. Číslo od 0 do 7 můžeme například na dvojkovém počítači převzít ze tří bitů proměnné U; konkrétně je v takovém případě vhodné brát bity z *nejvýznamnější* (nejlevější) části počítačového slova, protože nejméně významné bity v mnoha generátorech náhodných čísel nebývají dostatečně náhodné. (Viz výklad v části 3.2.1.1.)

Obecně pro získání náhodného celého čísla X od 0 do $k-1$ můžeme *vynásobit* k a položit $X = \lfloor kU \rfloor$. Na počítači MIXmůžeme napsat

```
LDA  U
MUL  K
```
(1)

a po provedení těchto dvou instrukcí se požadované celé číslo objeví v registru A. Potřebujeme-li namísto toho náhodné celé číslo od 1 do k, stačí k výsledku přičíst jedničku. (Znamená to doplnit za (1) instrukci „`INCA 1`“.)

U této metody mají všechna celá čísla zhruba stejnou pravděpodobnost. Dochází ovšem k malé chybě, protože velikost počítačového slova je konečná (viz cvičení 2); pro malé k je nicméně chyba zanedbatelná, například pro $k/m < < 1/10\,000$.

V obecnější situaci můžeme chtít každému celému číslu přiřadit jinou váhu. Dejme tomu, že hodnotu $X = x_1$ chceme získat s pravděpodobností p_1, $X = x_2$ s pravděpodobností p_2, ..., a $X = x_k$ s pravděpodobností p_k. Můžeme tedy vygenerovat rovnoměrně rozdělené číslo U a převést

$$X = \begin{cases} x_1, & \text{je-li } 0 \le U < p_1; \\ x_2, & \text{je-li } p_1 \le U < p_1 + p_2; \\ \quad\vdots & \\ x_k, & \text{je-li } p_1 + p_2 + \cdots + p_{k-1} \le U < 1. \end{cases} \tag{2}$$

(Všimněte si, že $p_1 + p_2 + \cdots + p_k = 1$.)

Existuje ale „nejlepší možný“ způsob porovnání U s různými hodnotami $p_1 + p_2 + \cdots + p_s$, naznačenými v (2); tuto situaci jsme rozebírali v části 2.3.4.5. Pro speciální případy jsou vhodné efektivnější metody; pro získání jedné z 11 hodnot 2, 3, ..., 12, jejichž pravděpodobnosti odpovídají hodu dvěma kostkami, $\frac{1}{36}, \frac{2}{36}, \ldots, \frac{6}{36}, \ldots, \frac{2}{36}, \frac{1}{36}$, stačí například vypočítat dvě nezávislá náhodná celá čísla od 1 do 6 a sečíst je.

Existuje ale rychlejší způsob výběru $x_1, \ldots, x_k$ s libovolně zadanými pravděpodobnostmi, který je postavený na důmyslném postupu od A. J. Walkera [*Electronics Letters* **10**, 8 (1974), 127–128; *ACM Trans. Math. Software* **3** (1977), 253–256]. Dejme tomu, že vytvoříme kU a uvážíme samostatně jeho celou část $K = \lfloor kU \rfloor$ a zlomkovou část $V = (kU) \bmod 1$; po provedení kódu (1) bude například K v registru A a V v registru X. Potom můžeme vždy získat požadované rozdělení pomocí operací

$$\text{je-li} \quad V < P_K \quad \text{pak} \quad X \leftarrow x_{K+1} \quad \text{jinak} \quad X \leftarrow Y_K, \tag{3}$$

pro nějaké vhodné tabulky $(P_0, \ldots, P_{k-1})$ a $(Y_0, \ldots, Y_{k-1})$. Ve cvičení 7 si ukážeme obecný postup výpočtu takovýchto tabulek. Walkerova metoda se někdy nazývá „metoda aliasů“.

Na dvojkovém počítači je obvykle užitečné předpokládat, že k je mocninou 2, takže násobení pak můžeme nahradit bitovým posuvem. To lze provést bez újmy na obecnosti; stačí zavést další hodnoty x s nulovou pravděpodobností výskytu. Uvažujme například opět hrací kostku a požadujme výskyt $X = j$ s následujícími 16 pravděpodobnostmi:

$j =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$p_j =$	0	0	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$	0	0	0

Toho snadno dosáhneme pomocí (3), a to pokud $k = 16$ a $x_{j+1} = j$ pro $0 \le j < 16$, a pokud jsou tabulky P a Y definovány následovně:

$$\begin{array}{rcccccccccccccccc} j = & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ P_j = & 0 & 0 & \frac{4}{9} & \frac{8}{9} & 1 & \frac{7}{9} & 1 & 1 & 1 & \frac{7}{9} & \frac{7}{9} & \frac{8}{9} & \frac{4}{9} & 0 & 0 & 0 \\ Y_j = & 5 & 9 & 7 & 4 & * & 6 & * & * & * & 8 & 4 & 7 & 10 & 6 & 7 & 8 \end{array}$$

(Pokud je $P_j = 1$, Y_j se nepoužívá.) Hodnota 7 se zde tedy vyskytuje s pravděpodobností $\frac{1}{16} \cdot \big((1 - P_2) + P_7 + (1 - P_{11}) + (1 - P_{14})\big) = \frac{6}{36}$, což jsme požadovali. Je to dosti zvláštní způsob házení kostkami, ale výsledky jsou k nerozeznání od skutečných.

Pravděpodobnosti p_j je možné reprezentovat implicitně pomocí nezáporných vah $w_1, w_2, \ldots, w_k$; označíme-li součet vah jako W, pak $p_j = w_j/W$. Jednotlivé váhy se v řadě aplikacích mohou měnit i dynamicky. Matias, Vitter a Ni [*SODA* **4** (1993), 361–370] ukázali, jak provést aktualizaci váhy a generování X v konstantním očekávaném čase.

B. Obecné metody pro spojitá rozdělení. Nejobecnější rozdělení reálných hodnot můžeme vyjádřit pomocí příslušné „distribuční funkce“ $F(x)$, která určuje pravděpodobnost, že náhodná veličina X nepřevýší x:

$$F(x) = \Pr(X \le x). \tag{4}$$

Tato funkce je vždy monotónní a roste od 0 do 1; to znamená, že

$$F(x_1) \le F(x_2), \quad \text{je-li } x_1 \le x_2; \qquad F(-\infty) = 0, \qquad F(+\infty) = 1. \tag{5}$$

Příklady distribučních funkcí jsme si uváděli v části 3.3.1, na Obr. 3. Pokud je $F(x)$ spojitá a ostře rostoucí (takže pro $x_1 < x_2$ je ostře $F(x_1) < F(x_2)$), nabývá všech hodnot od 0 do 1 a existuje k ní *inverzní funkce* $F^{[-1]}(y)$ taková, že pro $0 < y < 1$ je

$$y = F(x) \qquad \text{právě tehdy, když} \qquad x = F^{[-1]}(y). \tag{6}$$

Obecně pro spojitou a ostře rostoucí funkci $F(x)$ můžeme spočítat náhodnou veličinu X s rozdělením $F(x)$ podle by setting

$$X = F^{[-1]}(U), \tag{7}$$

kde U je rovnoměrné. To je správně, protože pravděpodobnost $X \le x$ je rovna pravděpodobnosti $F^{[-1]}(U) \le x$, tedy pravděpodobnosti $U \le F(x)$, což je $F(x)$.

Tím se problém redukuje na problém numerické analýzy, konkrétně na problém nalezení vhodných metod výpočtu $F^{[-1]}(U)$ s požadovanou přesností. Numerická analýza je mimo rámec této knihy zaměřené na seminumerické algoritmy; existuje nicméně řada důležitých obratů, které vedou k urychlení obecného postupu (7), a nyní si o nich něco řekneme.

Za prvé, je-li X_1 náhodná proměnná s rozdělením $F_1(x)$ a X_2 na ní nezávislá náhodná proměnná s rozdělením $F_2(x)$, pak

$$\begin{array}{ll} \max(X_1, X_2) & \text{má rozdělení} \quad F_1(x)F_2(x), \\ \min(X_1, X_2) & \text{má rozdělení} \quad F_1(x) + F_2(x) - F_1(x)F_2(x). \end{array} \tag{8}$$

(Viz cvičení 4.) Například rovnoměrná veličina (neboli veličina s rovnoměrným rozdělením) U má rozdělení $F(x) = x$, pro $0 \le x \le 1$; jestliže $U_1, U_2, \ldots, U_t$ jsou nezávislé rovnoměrně rozdělené veličiny, pak $\max(U_1, U_2, \ldots, U_t)$ má distribuční funkci $F(x) = x^t$, pro $0 \le x \le 1$. Tento vzorec je základem testu „maximum z t“, popsaného v části 3.3.2; inverzní funkce je $F^{[-1]}(y) = \sqrt[t]{y}$. Ve speciálním případě $t = 2$ tak vidíme, že oba vztahy

$$X = \sqrt{U} \qquad \text{a} \qquad X = \max(U_1, U_2) \tag{9}$$

dávají u náhodné proměnné X ekvivalentní rozdělení, i když to na první pohled není zřejmé. Nemusíme tedy počítat druhou odmocninu rovnoměrné veličiny.

Počet podobných triků je doslova nekonečný: *Libovolný* algoritmus, jenž přebírá na vstupu náhodná čísla, dá na výstupu náhodnou veličinu s *nějakým* rozdělením. Problémem ale je nalézt obecné metody pro konstrukci algoritmu ze zadané distribuční funkce výstupu. Tyto metody nebudeme probírat čistě abstraktně, nýbrž budeme zkoumat jejich aplikaci na nejdůležitější případy.

C. Normální rozdělení. Zřejmě nejdůležitějším nerovnoměrným, spojitým rozdělením je *normální rozdělení s nulovou střední hodnotou a standardní odchylkou rovnou jedné:*

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, \mathrm{d}t. \tag{10}$$

O významu tohoto rozdělení jsme se zmínili již v části 1.2.10. V tomto případě se bohužel inverzní funkce $F^{[-1]}$ nedá spočítat příliš snadno, uvidíme ale, že je k dispozici několik jiných technik.

1) *Polární metoda*, kterou představili G. E. P. Box, M. E. Muller a G. Marsaglia. (Viz *Annals Math. Stat.* **29** (1958), 610–611; a Boeing Scientific Res. Lab. zpráva D1-82-0203 (1962).)

Algoritmus P (*Polární metoda pro normální veličiny*). Tento algoritmus vypočte dvě nezávislé proměnné (veličiny) s normálním rozdělením X_1 a X_2.

P1. [Vytvoření rovnoměrných proměnných.] Vygenerujte dvě nezávislé náhodné proměnné U_1 a U_2 s rovnoměrným rozdělením od 0 do 1. Přiřaďte $V_1 \leftarrow 2U_1 - 1$, $V_2 \leftarrow 2U_2 - 1$. (Nyní V_1 a V_2 mají rovnoměrné rozdělení od -1 do $+1$. Na většině počítačů bude vhodnější reprezentovat V_1 a V_2 jako čísla s pohyblivou řádovou čárkou.)

P2. [Výpočet S.] Přiřaďte $S \leftarrow V_1^2 + V_2^2$.

P3. [Je $S \ge 1$?] Je-li $S \ge 1$, vraťte se na P1. (Kroky P1 až P3 se provádějí v průměru 1,27krát, se standardní odchylkou 0,59; viz cvičení 6.)

P4. [Výpočet X_1, X_2.] Je-li$S = 0$, přiřaďte $X_1 \leftarrow X_2 \leftarrow 0$; jinak přiřaďte

$$X_1 \leftarrow V_1 \sqrt{\frac{-2 \ln S}{S}}, \qquad X_2 \leftarrow V_2 \sqrt{\frac{-2 \ln S}{S}}. \tag{11}$$

To jsou požadované proměnné s normálním rozdělením. ∎

Pro důkaz správnosti této metody využijeme elementární analytickou geometrii a diferenciální počet: Je-li v kroku P3 $S < 1$, pak bod v rovině s kartézskými souřadnicemi (V_1, V_2) *je náhodným bodem s rovnoměrným rozdělením uvnitř jednotkového kruhu.* Po transformaci na polární souřadnice $V_1 = R\cos\Theta$, $V_2 = R\sin\Theta$, dostáváme

$$S = R^2, \quad X_1 = \sqrt{-2\ln S}\cos\Theta, \quad X_2 = \sqrt{-2\ln S}\sin\Theta.$$

Vezmeme-li dále polární souřadnice $X_1 = R'\cos\Theta'$, $X_2 = R'\sin\Theta'$, odvodíme, že $\Theta' = \Theta$ a $R' = \sqrt{-2\ln S}$. Je zřejmé, že R' a Θ' jsou nezávislé, protože R a Θ jsou nezávislé uvnitř jednotkového kruhu. Navíc, Θ' má rovnoměrné rozdělení od 0 do 2π, a pravděpodobnost, že $R' \le r$, je rovna pravděpodobnosti, že $-2\ln S \le r^2$, tedy pravděpodobnosti $S \ge e^{-r^2/2}$. To se rovná $1 - e^{-r^2/2}$, protože $S = R^2$ je rovnoměrně rozdělené od 0 do 1. Pravděpodobnost, že R' leží mezi r a $r + \mathrm{d}r$, je proto diferenciál z $1 - e^{-r^2/2}$, tedy $re^{-r^2/2}\mathrm{d}r$. Podobně pravděpodobnost, že Θ' leží mezi θ a $\theta + \mathrm{d}\theta$, je $(1/2\pi)\,\mathrm{d}\theta$. Nyní můžeme vypočítat spojenou pravděpodobnost, že $X_1 \le x_1$ a $X_2 \le x_2$:

$$\begin{aligned}\int_{\{(r,\theta)\,|\,r\cos\theta\le x_1,\ r\sin\theta\le x_2\}} & \frac{1}{2\pi}e^{-r^2/2}\,r\,\mathrm{d}r\,\mathrm{d}\theta \\ &= \frac{1}{2\pi}\int_{\{(x,y)\,|\,x\le x_1,\ y\le x_2\}} e^{-(x^2+y^2)/2}\,\mathrm{d}x\,\mathrm{d}y \\ &= \left(\sqrt{\frac{1}{2\pi}}\int_{-\infty}^{x_1} e^{-x^2/2}\,\mathrm{d}x\right)\left(\sqrt{\frac{1}{2\pi}}\int_{-\infty}^{x_2} e^{-y^2/2}\,\mathrm{d}y\right).\end{aligned}$$

Tímto výpočtem jsme dokázali, že X_1 a X_2 jsou nezávislé a že mají normální rozdělení, jak jsme požadovali.

2) *Metoda obdélník-klín-okraj*, kterou zavedl G. Marsaglia. Využijeme zde funkci

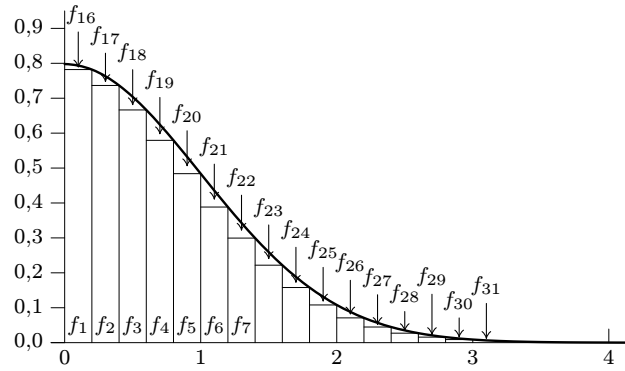
$$F(x) = \mathrm{erf}(x/\sqrt{2}\,) = \sqrt{\frac{2}{\pi}}\int_0^x e^{-t^2/2}\,\mathrm{d}t, \qquad x \ge 0, \tag{12}$$

která dává rozdělení *absolutní hodnoty* normální veličiny. Jakmile vypočteme X podle rozdělení (12), doplníme k jeho hodnotě náhodné znaménko a vytvoříme z něj tak skutečnou normální veličinu.

Metoda obdélník-klín-okraj je založena na několika důležitých obecných technikách, které budeme zkoumat během vývoje algoritmu. První klíčovou myšlenkou je považovat $F(x)$ za „směs“ neboli *kombinaci* několika dalších funkcí, konkrétně

$$F(x) = p_1F_1(x) + p_2F_2(x) + \cdots + p_nF_n(x), \tag{13}$$

kde F_1, F_2, …, F_n jsou vhodná rozdělení a p_1, p_2, …, p_n jsou nezáporné pravděpodobnosti, jejichž součet je 1. Jestliže vygenerujeme náhodnou proměnnou X s tím, že zvolíme rozdělení F_j s pravděpodobností p_j, snadno nahlédneme, že X bude mít celkové rozdělení F. S některými rozděleními $F_j(x)$ se třeba obtížně pracuje, ještě obtížněji než se samotným F, obvykle ale můžeme vše uspořádat takovým způsobem, že je pravděpodobnost p_j v tomto případě velmi



Obr. 9. Funkce hustoty rozdělená do 31 částí. Plocha každé části reprezentuje průměrný počet případů, kdy z výpočtu vyjde náhodné číslo s danou hodnotou.

malá. Většina různých rozdělení $F_j(x)$ se dá zvládnout poměrně snadno, protože budou triviálními modifikacemi rovnoměrného rozdělení. Výsledná metoda vede k mimořádně efektivnímu programu, protože jeho *průměrná* doba provádění je velmi krátká.

Činnost metody snáze pochopíme, jestliže namísto samotných distribučních funkcí budeme pracovat s jejich *derivacemi*. Nazvěme

$$f(x) = F'(x), \qquad f_j(x) = F_j{}'(x)$$

funkcemi hustoty rozdělení pravděpodobnosti. Výraz (13) přechází na

$$f(x) = p_1 f_1(x) + p_2 f_2(x) + \cdots + p_n f_n(x). \tag{14}$$

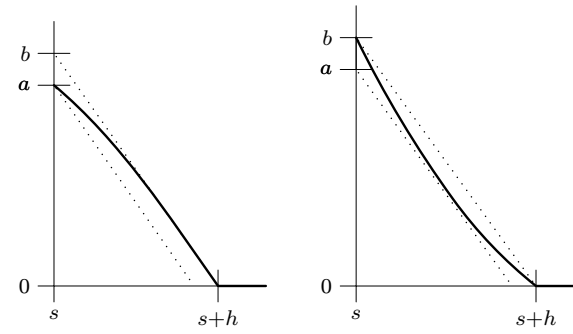
Zde každé $f_j(x)$ je ≥ 0 a celková plocha pod grafem $f_j(x)$ je 1. Celou relaci (14) můžeme tudíž pohodlně vyjádřit graficky: Plochu pod $f(x)$ rozdělíme do n částí, přičemž část odpovídající $f_j(x)$ má plochu p_j. Viz Obr. 9, který ilustruje pro nás zajímavou situaci, kdy $f(x) = F'(x) = \sqrt{2/\pi}\, e^{-x^2/2}$ a plocha pod křivkou je rozdělena do $n = 31$ částí. Je tu 15 obdélníků, jež reprezentují $p_1 f_1(x)$, …, $p_{15} f_{15}(x)$; dále je tu 15 částí tvaru klínku, jež vyjadřují $p_{16} f_{16}(x)$, …, $p_{30} f_{30}(x)$; zbývající část $p_{31} f_{31}(x)$ je „okraj“, tedy celý zbytek grafu $f(x)$ pro $x \geq 3$.

Obdélníkové části $f_1(x)$, …, $f_{15}(x)$ reprezentují *rovnoměrné rozdělení.* Například $f_3(x)$ tak představuje náhodnou proměnnou s rovnoměrným rozdělením od $\frac{2}{5}$ do $\frac{3}{5}$. Výška $p_j f_j(x)$ je $f(j/5)$, takže plocha j-tého obdélníku je

$$p_j = \frac{1}{5} f(j/5) = \sqrt{\frac{2}{25\pi}}\, e^{-j^2/50}, \qquad \text{pro } 1 \leq j \leq 15. \tag{15}$$

Pro generování takovýchto obdélníkových částí rozdělení stačí jednoduše vypočítat

$$X = \tfrac{1}{5} U + S, \tag{16}$$



Obr. 10. Funkce hustoty, pro něž může Algoritmus L generovat náhodná čísla

kde U má rovnoměrné rozdělení a S nabývá hodnoty $(j-1)/5$ s pravděpodobností p_j. Protože $p_1 + \cdots + p_{15} = 0,9183$, můžeme takovéto jednoduché rovnoměrné veličiny používat zhruba v 92 procentech případů.

Ve zbývajících 8 procentech případů musíme obvykle vygenerovat jedno z „klínových" rozdělení $F_{16}, \ldots, F_{30}$. Typické příklady funkcí, které potřebujeme, jsou na Obr. 10. Je-li $x < 1$, pak je část křivky konkávní, a je-li $x > 1$, pak je konvexní, ale v obou případech se rozumně blíží přímce a dá se podle obrázku uzavřít mezi dvě rovnoběžky.

Tato rozdělení klínového tvaru vyřešíme pomocí ještě jiné obecné techniky, kterou je von Neumannova *zamítací metoda*. Zde pomocí ní odvodíme komplikovanou hustotu z jiné, která ji „uzavírá". Jednoduchým příkladem tohoto postupu je výše popsaná polární metoda, kde v krocích P1–P3 získáme náhodný bod uvnitř jednotkového kruhu. K tomu nejprve vygenerujeme náhodný bod ve větším čtverci a pokud se nachází mimo kruh, zamítneme jej a začneme znovu.

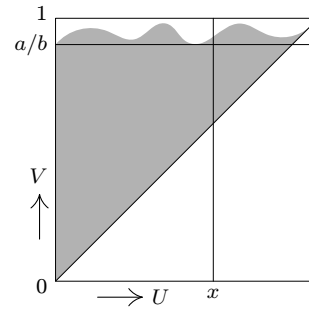
Obecná zamítací metoda je ale ještě silnější. Pro vygenerování náhodné proměnné X s hustotou f, nechť g je jiná funkce hustoty rozdělení pravděpodobnosti taková, že

$$f(t) \le cg(t) \tag{17}$$

pro všechna t, kde c je konstanta. Nyní podle funkce hustoty g vygenerujeme veličinu X a také vygenerujeme nezávislou rovnoměrnou veličinu U. Je-li $U \ge \ge f(X)/cg(X)$, zamítneme X a začneme znovu s jiným X a U. Jakmile je podmínka $U < f(X)/cg(X)$ konečně splněna, má výsledné X požadovanou hustotu f. $\big[$*Důkaz*: $X \le x$ nastává s pravděpodobností $p(x) = \int_{-\infty}^{x} \big(g(t)\,\mathrm{d}t \cdot \cdot f(t)/cg(t)\big) + qp(x)$, kde veličina $q = \int_{-\infty}^{\infty} \big(g(t)\,\mathrm{d}t \cdot (1 - f(t)/cg(t))\big) = 1 - 1/c$ je pravděpodobnost zamítnutí; odtud $p(x) = \int_{-\infty}^{x} f(t)\,\mathrm{d}t$.$\big]$

Zamítací technika je nejefektivnější pro malé c, protože v průměru potřebujeme pro přijetí hodnoty provést c iterací. (Viz cvičení 6.) V některých případech je $f(x)/cg(x)$ vždy rovno jedné; potom U nemusíme generovat. Jindy, pokud je výpočet $f(x)/cg(x)$ obtížný, ji můžeme ohraničit dvěma funkcemi

$$r(x) \le f(x)/cg(x) \le s(x) \tag{18}$$



Obr. 11. Oblast „přijetí“ v Algoritmu L

které jsou mnohem jednodušší; přesnou hodnotu $f(x)/cg(x)$ pak není nutné vypočítávat, jestliže není $r(x) \le U < s(x)$. Následující algoritmus rozvíjí zamítací metodu a tím řeší klínový problém.

Algoritmus L (*Téměř lineární hustoty*). Pomocí tohoto algoritmu je možné generovat náhodnou proměnnou X s libovolným rozdělením, jehož funkce hustoty $f(x)$ splňuje následující podmínky (viz Obr. 10):

$$\begin{aligned} f(x) &= 0, \qquad && \text{pro } x < s \text{ a pro } x > s+h; \\ a - b(x-s)/h \le f(x) &\le b - b(x-s)/h, && \text{pro } s \le x \le s+h. \end{aligned} \tag{19}$$

L1. [Načtení $U \le V$.] Vygenerujte dvě nezávislé náhodné proměnné U a V s rovnoměrným rozdělením od 0 do 1. Je-li $U > V$, vyměňte $U \leftrightarrow V$.

L2. [Snadný případ?] Je-li $V \le a/b$, jděte na krok L4.

L3. [Nový pokus ?] Je-li $V > U + (1/b)f(s+hU)$, vraťte se na krok L1. (Pokud je a/b blízko 1, nebude nutné tento krok provádět příliš často.)

L4. [Výpočet X.] Přiřaďte $X \leftarrow s + hU$. ▮

Po dosažení kroku L4 je bod (U, V) náhodným bodem v oblasti, která je na Obr. 11 stínovaná, tedy v oblasti $0 \le U \le V \le U + (1/b)f(s+hU)$. Podmínky (19) zaručují, že

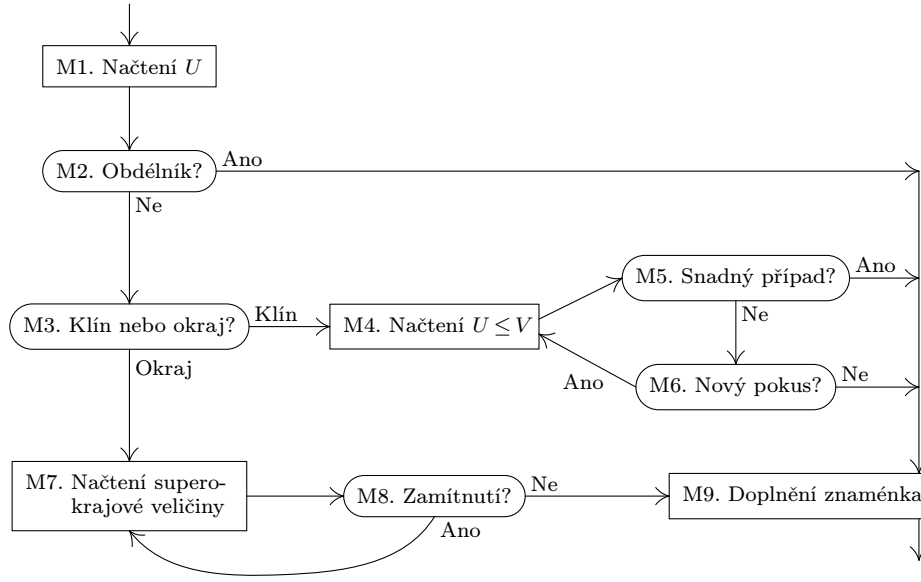
$$\frac{a}{b} \le U + \frac{1}{b} f(s+hU) \le 1.$$

Nyní pravděpodobnost, že $X \le s + hx$, pro $0 \le x \le 1$, je plocha ležící na Obr. 11 vlevo od svislé čáry $U = x$, děleno celkovou plochou, tedy

$$\int_0^x \frac{1}{b} f(s+hu)\,\mathrm{d}u \Big/ \int_0^1 \frac{1}{b} f(s+hu)\,\mathrm{d}u = \int_s^{s+hx} f(v)\,\mathrm{d}v;$$

a proto X má správné rozdělení.

Při odpovídajících konstantách a_j, b_j, s_j, se Algoritmus L postará o hustoty klínového tvaru z Obr. 9, pro $1 \le j \le 15$. Poslední rozdělení F_{31} stačí vzít v úvahu jen asi v jednom případě z 370; použijeme je při výpočtu výsledku $X \ge 3$. Ve cvičení 11 si ukážeme, že i pro tento „okraj“ grafu lze použít standardní zamítací schéma. Nyní tedy konečně můžeme napsat celou proceduru:



Obr. 12. Algoritmus „obdélník-klín-okraj“ pro generování normálních veličin

Algoritmus M (*Metoda obdélník-klín-okraj pro normální veličiny*). Při výpočtu v algoritmu budeme používat pomocné tabulky $(P_0, \ldots, P_{31})$, $(Q_1, \ldots, Q_{15})$, $(Y_0, \ldots, Y_{31})$, $(Z_0, \ldots, Z_{31})$, $(S_1, \ldots, S_{16})$, $(D_{16}, \ldots, D_{30})$, $(E_{16}, \ldots, E_{30})$, sestavené podle cvičení 10; příklady jsou uvedeny v Tabulce 1. Dále předpokládáme dvojkový počítač; podobnou proceduru bychom ale mohli vytvořit i pro desítkový stroj.

M1. [Načtení U.] Vygenerujte náhodné číslo s rovnoměrným rozdělením $U = (.b_0 b_1 b_2 \ldots b_t)_2$. (Zde b jsou jednotlivé bity ve dvojkové reprezentaci U. Pro přijatelnou úroveň přesnosti by t mělo být nejméně 24.) Přiřaďte $\psi \leftarrow b_0$. (Později podle ψ stanovíme znaménko výsledku.)

M2. [Obdélník?] Přiřaďte $j \leftarrow (b_1 b_2 b_3 b_4 b_5)_2$, což je dvojkové číslo stanovené podle vedoucích bitů U, a přiřaďte $f \leftarrow (.b_6 b_7 \ldots b_t)_2$, což je zlomek ze zbývajících bitů. Je-li $f \ge P_j$, přiřaďte $X \leftarrow Y_j + fZ_j$ a jděte na M9. Jinak je-li $j \le 15$ (tedy je-li $b_1 = 0$), přiřaďte $X \leftarrow S_j + fQ_j$ a jděte na M9. (Jedná se o upravenou verzi Walkerovy metody aliasů (3).)

M3. [Klín nebo okraj?] (Nyní je $16 \le j \le 31$ a každá jednotlivá hodnota j se vyskytuje s pravděpodobností p_j.) Je-li $j = 31$, jděte na M7.

M4. [Načtení $U \le V$.] Vygenerujte dvě nové rovnoměrné veličiny U a V; je-li $U > V$, vyměňte $U \leftrightarrow V$. (Zde zpracováváme speciální případ Algoritmu L.) Přiřaďte $X \leftarrow S_{j-15} + \frac{1}{5}U$.

M5. [Snadný případ?] Je-li $V \le D_j$, jděte na M9.

Tabulka 1

PŘÍKLADY TABULEK POUŽÍVANÝCH V ALGORITMU M*

j	P_j	P_{j+16}	Q_j	Y_j	Y_{j+16}	Z_j	Z_{j+16}	S_{j+1}	D_{j+15}	E_{j+15}
0	0,000	,067		0,00	0,59	0,20	0,21	0,0		
1	0,849	,161	0,236	− 0,92	0,96	1,32	0,24	0,2	,505	25,00
2	0,970	,236	0,206	− 5,86	−0,06	6,66	0,26	0,4	,773	12,50
3	0,855	,285	0,234	− 0,58	0,12	1,38	0,28	0,6	,876	8,33
4	0,994	,308	0,201	−33,16	1,31	34,96	0,29	0,8	,939	6,25
5	0,995	,304	0,201	−39,51	0,31	41,31	0,29	1,0	,986	5,00
6	0,933	,280	0,214	− 2,57	1,12	2,97	0,28	1,2	,995	4,06
7	0,923	,241	0,217	− 1,61	0,54	2,61	0,26	1,4	,987	3,37
8	0,727	,197	0,275	0,67	0,75	0,73	0,25	1,6	,979	2,86
9	1,000	,152	0,200		0,56		0,24	1,8	,972	2,47
10	0,691	,112	0,289	0,35	0,17	0,65	0,23	2,0	,966	2,16
11	0,454	,079	0,440	− 0,17	0,38	0,37	0,22	2,2	,960	1,92
12	0,287	,052	0,698	0,92	−0,01	0,28	0,21	2,4	,954	1,71
13	0,174	,033	1,150	0,36	0,39	0,24	0,21	2,6	,948	1,54
14	0,101	,020	1,974	− 0,02	0,20	0,22	0,20	2,8	,942	1,40
15	0,057	,086	3,526	0,19	0,78	0,21	0,22	3,0	,936	1,27

*V praxi budeme tato data počítat s mnohem větší přesností; tabulka obsahuje jen ukázku hodnot, podle nichž si zvídavý čtenář může vyzkoušet činnost algoritmu pro přesnější výpočet. Hodnoty Q_0, Y_9, Z_9, D_{15} a E_{15} jsou nevyužité.

M6. [Nový pokus?] Je-li $V > U + E_j(e^{(S_{j-14}^2 - X^2)/2} - 1)$, vraťte se na krok M4; jinak jděte na M9. (Tento krok se provádí s nízkou pravděpodobností.)

M7. [Načtení superokrajové veličiny.] Vygenerujte dvě nezávislé rovnoměrné veličiny U a V, a přiřaďte $X \leftarrow \sqrt{9 - 2\ln V}$.

M8. [Zamítnutí?] Je-li $UX \geq 3$, vraťte se na krok M7. (Tento přechod proběhne v kroku M8 jen asi v jedné dvanáctině případů.)

M9. [Doplnění znaménka.] Je-li $\psi = 1$, přiřaďte $X \leftarrow -X$. ∎

Tento algoritmus je velice pěkným příkladem matematické teorie, těsně propletené s programátorskou genialitou – krásná ilustrace umění programování! Většinou stačí provést jen kroky M1, M2 a M9 a ani ostatní kroky nejsou příliš pomalé. Poprvé metodu obdélník-klín-okraj publikoval G. Marsaglia, *Annals Math. Stat.* **32** (1961), 894–899; G. Marsaglia, M. D. MacLaren a T. A. Bray, *CACM* **7** (1964), 4–10. Další zdokonalení Algoritmu M pak navrhli G. Marsaglia, K. Ananthanarayanan a N. J. Paul, *Inf. Proc. Letters* **5** (1976), 27–30.

3) *Metoda sudá-lichá*, objevil G. E. Forsythe. Překvapivě jednoduchou techniku generování náhodných veličin s hustotou obecného exponenciálního tvaru

$$f(x) = Ce^{-h(x)}\,[a \leq x < b], \tag{20}$$

kde

$$0 \leq h(x) \leq 1 \qquad \text{pro } a \leq x < b, \tag{21}$$

představili John von Neumann a G. E. Forsythe kolem roku 1950. Myšlenka vychází z výše popsané zamítací metody, kdy vezmeme $g(x)$ jako rovnoměrné rozdělení na intervalu $[a\,..\,b)$: Přiřadíme $X \leftarrow a + (b-a)U$, kde U je rovnoměrná veličina, a poté chceme přijmout X s pravděpodobností $e^{-h(X)}$. Pro provedení druhé operace stačí porovnat $e^{-h(X)}$ s V, nebo $h(X)$ s $-\ln V$, kde V je jiná rovno-

měrná veličina, avšak můžeme ji provést také následujícím zajímavým způsobem, bez využití jakékoli transcendentní funkce. Přiřadíme $V_0 \leftarrow h(X)$, a potom generujeme rovnoměrné veličiny V_1, V_2, ... tak dlouho, až najdeme nějaké $K \geq 1$, pro něž $V_{K-1} < V_K$. Pro pevné X a k je pravděpodobnost, že $h(X) \geq V_1 \geq \cdots \geq V_k$, rovna $1/k!$ krát pravděpodobnost, že $\max(V_1, \ldots, V_k) \leq h(X)$, tedy $h(X)^k/k!$; odtud pravděpodobnost $K = k$ je $h(X)^{k-1}/(k-1)! - h(X)^k/k!$, a pravděpodobnost, že K je liché, je

$$\sum_{k \text{ liché}, k \geq 1} \left(\frac{h(X)^{k-1}}{(k-1)!} - \frac{h(X)^k}{k!} \right) = e^{-h(X)}. \tag{22}$$

Je-li tedy K sudé, zamítneme X a provedeme nový pokus; je-li K liché, přijímáme X jako náhodnou proměnnou s hustotou (20). Pro stanovení K nepotřebujeme obvykle generovat příliš mnoho V, protože průměrná hodnota K (při daném X) je rovna $\sum_{k \geq 0} \Pr(K > k) = \sum_{k \geq 0} h(X)^k/k! = e^{h(X)} \leq e$.

O několik roků později si Forsythe uvědomil, že tento postup vede k efektivní metodě výpočtu normálních veličin, bez nutnosti pomocných rutin pro výpočet odmocnin či logaritmů jako v Algoritmech P a M. Jeho proceduru, kterou J. H. Ahrens a U. Dieter zdokonalili o výběr intervalů $[a\,..\,b)$, můžeme shrnout následovně.

Algoritmus F (*Metoda sudá-lichá pro normální veličiny*). Tento algoritmus generuje normální veličiny na dvojkovém počítači, při přibližně $t + 1$ bitech přesnosti. Vyžaduje tabulku hodnotu $d_j = a_j - a_{j-1}$, pro $1 \leq j \leq t + 1$, kde a_j je definováno vztahem

$$\sqrt{\frac{2}{\pi}} \int_{a_j}^{\infty} e^{-x^2/2}\, \mathrm{d}x = \frac{1}{2^j}. \tag{23}$$

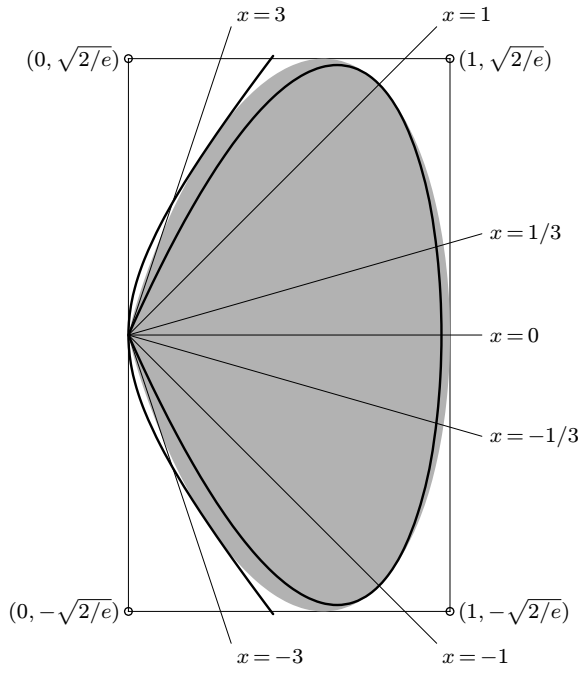
F1. [Načtení U.] Vygenerujte rovnoměrné náhodné číslo $U = (.b_0 b_1 \ldots b_t)_2$, kde b_0, b_1, ..., b_t označují bity ve dvojkovém zápise. Přiřaďte $\psi \leftarrow b_0$, $j \leftarrow 1$ a $a \leftarrow 0$.

F2. [Nalezení prvního nulového b_j.] Je-li $b_j = 1$, přiřaďte $a \leftarrow a + d_j$, $j \leftarrow j + 1$ a opakujte tento krok. (Je-li $j = t + 1$, považujte b_j za nulové.)

F3. [Generování kandidáta.] (Nyní je $a = a_{j-1}$ a aktuální hodnota j se vyskytuje s pravděpodobností $\approx 2^{-j}$. Vygenerujeme X v intervalu $[a_{j-1}\,..\,a_j)$, pomocí výše popsané zamítací metody, při $h(x) = x^2/2 - a^2/2 = y^2/2 + ay$, kde $y = x - a$. Ve cvičení 12 dokážeme, že $h(x) \leq 1$, jak požaduje (21).) Přiřaďte $Y \leftarrow d_j$ krát $(.b_{j+1} \ldots b_t)_2$ a $V \leftarrow (\frac{1}{2}Y + a)Y$. (Protože průměrná hodnota j je 2, bude obvykle $(.b_{j+1} \ldots b_t)_2$ obsahovat dostatečný počet bitů pro odpovídající přesnost. Výpočty je možné provést v aritmetice s pevnou řádovou čárkou.)

F4. [Zamítnutí?] Vygenerujte rovnoměrnou veličinu U. Je-li $V < U$, jděte na krok F5. Jinak do V přiřaďte novou rovnoměrnou veličinu a opakujte krok F4, pokud nové V je $\leq U$. Jinak (tedy pokud je K sudé jako v předchozím výkladu), nahraďte U novou rovnoměrnou veličinou $(.b_0 b_1 \ldots b_t)_2$ a vraťte se na F3.

F5. [Vrácení X.] Přiřaďte $X \leftarrow a + Y$. Je-li $\psi = 1$, přiřaďte $X \leftarrow -X$. ∎



Obr. 13. Oblast „přijetí“ v metodě poměru rovnoměrných veličin pro normální veličiny. Délky čar s poměrem souřadnic x mají normální rozdělení.

Hodnoty d_j pro $1 \le j \le 47$ uvádějí ve svém článku Ahrens a Dieter, *Math. Comp.* **27** (1973), 927–937; rozebírají zde zdokonalení algoritmu, která vedou k urychlení, i když za cenu více tabulek. Algoritmus F je velmi zajímavý, protože je téměř stejně rychlý jako Algoritmus S, ale snáze se implementuje. Průměrný počet rovnoměrných veličin na normální veličinu je 2,539 47; R. P. Brent [*CACM* **17** (1974), 704–705] ukázal, jak tento počet snížit na 1,374 46 za cenu dvou odčítání a jednoho dělení na uloženou rovnoměrnou veličinu.

4) *Poměry rovnoměrných veličin.* Existuje ještě jeden způsob generování normálních veličin, s nímž přišli A. J. Kinderman a J. F. Monahan v roce 1976. Jejich myšlenkou je vygenerovat náhodný bod (U, V) v oblasti definované nerovnostmi

$$0 < u \le 1, \qquad -2u\sqrt{\ln(1/u)} \le v \le 2u\sqrt{\ln(1/u)}, \tag{24}$$

a poté vypsat poměr $X \leftarrow V/U$. Stínovaná plocha na Obr. 13 je „magickou“ oblastí (24), díky níž může metoda fungovat. Než se pustíme do příslušné teorie, popíšeme samotný algoritmus, abychom jasně viděli, jak je efektivní a jednoduchý:

Algoritmus R (*Poměrová metoda pro normální veličiny*). Tento algoritmus generuje normální veličiny X.

R1. [Načtení U, V.] Vygenerujte dvě nezávislé rovnoměrné veličiny U a V, kde U je nenulové, a přiřaďte $X \leftarrow \sqrt{8/e}\,\left(V - \frac{1}{2}\right)/U$. (Nyní je X poměrem souřadnic $\left(U, \sqrt{8/e}\,\left(V - \frac{1}{2}\right)\right)$ náhodného bodu v obdélníku, který uzavírá stínovanou plochu na Obr. 13. Číslo X přijímáme, pokud odpovídající bod skutečně leží „ve stínu“, jinak provedeme nový pokus.)

R2. [Nepovinný test horní hranice.] Je-li $X^2 \le 5 - 4e^{1/4}U$, vypište X a ukončete algoritmus. (Tento krok je podle potřeby možné vynechat; testujeme v něm, jestli se vybraný bod nachází ve vnitřní ploše Obr. 13 nebo ne, takže pak není nutné počítat logaritmus.)

R3. [Nepovinný test dolní hranice.] Je-li $X^2 \ge 4e^{-1.35}/U + 1{,}4$, vraťte se na R1. (I tento krok je možné vynechat; testujeme v něm, jestli se vybraný bod nachází vně vnější plochy Obr. 13, takže pak není nutné počítat logaritmus.)

R4. [Závěrečný test.] Je-li $X^2 \le -4 \ln U$, vypište X a ukončete algoritmus. Jinak se vraťte na R1. ▮

Analýza časování algoritmu nás čeká ve cvičeních 20 a 21; konkrétně analyzujeme čtyři různé verze, protože kroky R2 a R3 je možné jednotlivě ponechat nebo vynechat. Následující tabulka uvádí, kolikrát se budou jednotlivé kroky v průměru provádět, a to podle zahrnutí nepovinných testů:

Krok	Žádný	Jen R2	Jen R3	Oba
R1	1,369	1,369	1,369	1,369
R2	0	1,369	0	1,369
R3	0	0	1,369	0,467
R4	1,369	0,467	1,134	0,232

(25)

Nepovinné testy se tedy vyplatí vynechat jen v případě, že máme velmi rychlou operaci logaritmování, ale pokud je pomalejší, bude lépe testy ponechat.

Proč ale celý algoritmus funguje? Jedním důvodem je, že můžeme vypočítat pravděpodobnost $X \le x$, která, jak se ukazuje, je správnou hodnotou (10). Takovýto výpočet není ale právě snadný, pokud nepřijdeme na ten správný trik, a vždy je lepší nejprve pochopit, jak byl zřejmě algoritmus objeven. Tento odvodili Kinderman a Monahan, a to z rozpracování následující teorie, kterou je možné využít ve spojení s libovolnou rozumně se chovající funkcí hustoty $f(x)$ [viz *ACM Trans. Math. Software* **3** (1977), 257–260].

Obecně předpokládejme, že byl vygenerován bod (U, V) s rovnoměrným rozdělením přes oblast (u, v)-roviny definovanou nerovnostmi

$$u > 0, \qquad u^2 \le g(v/u) \tag{26}$$

pro nějakou nezápornou integrovatelnou funkci g. Jestliže přiřadíme $X \leftarrow V/U$, pak pravděpodobnost $X \le x$ můžeme vypočítat integrací $\mathrm{d}u\,\mathrm{d}v$ přes plochu definovanou dvěma relacemi (26) plus dodatečnou podmínkou $v/u \le x$, a potom vydělit stejným integrálem bez této podmínky. Položíme-li $v = tu$, takže $\mathrm{d}v = u\,\mathrm{d}t$, přejde integrál ve tvar

$$\int_{-\infty}^{x} \mathrm{d}t \int_{0}^{\sqrt{g(t)}} u\,\mathrm{d}u = \frac{1}{2}\int_{-\infty}^{x} g(t)\,\mathrm{d}t.$$

Pravděpodobnost $X \le x$ je tudíž rovna

$$\int_{-\infty}^{x} g(t)\,\mathrm{d}t \Bigg/ \int_{-\infty}^{+\infty} g(t)\,\mathrm{d}t. \tag{27}$$

Pro $g(t) = e^{-t^2/2}$ vychází odtud normální rozdělení a podmínka $u^2 \le g(v/u)$ se v tomto případě zjednoduší na $(v/u)^2 \le -4 \ln u$. Snadno nahlédneme, že množina všech takovýchto dvojic (u, v) je celá obsažena v obdélníku na Obr. 13.

Hranice v krocích R2 a R3 definují vnitřní a vnější oblasti pomocí jednodušších hraničních nerovnic. Pomocí známé nerovnosti

$$e^x \ge 1 + x,$$

která platí pro všechna reálná x, můžeme ukázat, že

$$1 + \ln c - cu \;\le\; -\ln u \;\le\; 1/(cu) - 1 + \ln c \tag{28}$$

pro libovolnou konstantu $c > 0$. Ve cvičení 21 dokážeme, že nejlepší možnou konstantou pro krok R2 je $c = e^{1/4}$. V kroku R3 je situace komplikovanější a zdá se, že v tomto případě nelze optimální c vyjádřit nějakým jednoduchým výrazem, ale podle experimentálních výpočtů je nejlepší hodnota pro R3 $\approx e^{1,35}$. Pro $u = 1/c$ jsou aproximační křivky (28) tečnami ke skutečné hranici.

Při zdokonalené aproximaci oblasti přijetí [viz J. L. Leva, *ACM Trans. Math. Software* **18** (1992), 449–455] můžeme fakticky snížit očekávaný počet výpočtů logaritmu jen na 0,012.

Jestliže celou oblast rozdělíme do podoblastí, z nichž většinu je možné zpracovat rychleji, dostaneme celkově rychlejší metodu. Samozřejmě to znamená jisté pomocné tabulky, podobně jako v Algoritmech M a F. Zajímavou alternativu, která vyžaduje méně položek v pomocných tabulkách, navrhli Ahrens a Dieter v *CACM* **31** (1988), 1330–1337.

5) *Normální veličiny z normálních veličin.* Ve cvičení 31 rozebíráme zajímavý přístup, který nestaví vše jen na rovnoměrných veličinách, nýbrž pracuje přímo s normálními veličinami a tím šetří jistý čas. Metodu představil v roce 1996 C. S. Wallace, zatím má relativně slabou teoretickou podporu, ale úspěšně splnila řadu empirických testů.

6) *Varianty normálního rozdělení.* Zatím jsme uvažovali normované normální rozdělení, se střední hodnotou nula a standardní odchylkou rovnou jedné. Má-li veličina X toto rozdělení, pak

$$Y = \mu + \sigma X \tag{29}$$

má normální rozdělení se střední hodnotou μ a standardní odchylkou σ. Navíc, jsou-li X_1 a X_2 nezávislé normální veličiny se střední hodnotou nula a standardní odchylkou jednak a je-li

$$Y_1 = \mu_1 + \sigma_1 X_1, \qquad Y_2 = \mu_2 + \sigma_2\big(\rho X_1 + \sqrt{1-\rho^2}\, X_2\big), \tag{30}$$

pak Y_1 a Y_2 jsou *závislé* náhodné proměnné s normálním rozdělením, se středními hodnotami μ_1, μ_2 a standardními odchylkami σ_1, σ_2, a s korelačním koeficientem ρ. (Zobecnění na n proměnných viz cvičení 13.)

D. Exponenciální rozdělení. Další nejdůležitější náhodnou veličinou kromě rovnoměrných a normálních veličin je *exponenciální veličina.* Takováto čísla se

vyskytují v situacích, které charakterizuje „doba příchodu“; jestliže například radioaktivní látka emituje alfa částice takovou rychlostí, že jedna je emitována v průměru jednou za μ sekund, pak doba mezi dvěma následujícími částicemi má exponenciální rozdělení se střední hodnotou μ. Toto rozdělení je definováno vztahem

$$F(x) = 1 - e^{-x/\mu}, \qquad x \geq 0. \tag{31}$$

1) *Logaritmická metoda.* Zřejmě jestliže $y = F(x) = 1 - e^{-x/\mu}$, pak $x = F^{[-1]}(y) = -\mu \ln(1-y)$. Podle (7) má tudíž $-\mu \ln(1-U)$ exponenciální rozdělení. A protože $1-U$ má rovnoměrné rozdělení, pokud je má i U, dospějeme k závěru, že

$$X = -\mu \ln U \tag{32}$$

má exponenciální rozdělení se střední hodnotou μ. (Případ $U = 0$ musíme ošetřit zvlášť; namísto 0 můžeme dosadit libovolnou vhodnou hodnotu ϵ, protože pravděpodobnost tohoto případu je extrémně malá.)

2) *Náhodná minimalizační metoda.* V Algoritmu F jsme viděli, že existují jednoduché a rychlé alternativy k výpočtu logaritmu rovnoměrné veličiny. Následující mimořádně efektivní postup vyvinuli G. Marsaglia, M. Sibuya a J. H. Ahrens [viz *CACM* **15** (1972), 876–877]:

Algoritmus S (*Exponenciální rozdělení se střední hodnotou* μ). Tento algoritmus vytváří na dvojkovém počítači exponenciální veličinu, a to s využitím rovnoměrné veličiny s $(t+1)$-bitovou přesností. Konstanty

$$Q[k] = \frac{\ln 2}{1!} + \frac{(\ln 2)^2}{2!} + \cdots + \frac{(\ln 2)^k}{k!}, \qquad k \geq 1, \tag{33}$$

je vhodné vypočítat předem a postupovat až do $Q[k] > 1 - 2^{-t}$.

S1. [Načtení U a posuv.] Vygenerujte $(t+1)$-bitový rovnoměrný náhodný dvojkový zlomek $U = (.b_0 b_1 b_2 \ldots b_t)_2$; najděte v něm první nulový bit b_j a posuňte pryč vedoucích $j+1$ bitů, takže přiřadíte $U \leftarrow (.b_{j+1} \ldots b_t)_2$. (Stejně jako v Algoritmu F je průměrný počet zahozených bitů 2.)

S2. [Okamžité přijetí?] Je-li $U < \ln 2$, přiřaďte $X \leftarrow \mu(j \ln 2 + U)$ a ukončete algoritmus. (Všimněte si, že $Q[1] = \ln 2$.)

S3. [Minimalizace.] Najděte nejmenší $k \geq 2$ takové, že $U < Q[k]$. Vygenerujte k nových rovnoměrných veličin $U_1, \ldots, U_k$ a přiřaďte $V \leftarrow \min(U_1, \ldots, U_k)$.

S4. [Předání odpovědi.] Přiřaďte $X \leftarrow \mu(j+V)\ln 2$. ∎

Exponenciální veličiny můžeme generovat také jinými, alternativními postupy (jako je například poměrová metoda rovnoměrných veličin z Algoritmu R.

E. Ostatní spojitá rozdělení. Podívejme se nyní stručně na výpočty některých jiných rozdělení, které se v praxi poměrně často vyskytují.

1) *Gama rozdělení* řádu $a > 0$ je definováno vztahem

$$F(x) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t}\, \mathrm{d}t, \qquad x \geq 0. \tag{34}$$

Pro $a = 1$ přechází vztah v exponenciální rozdělení se střední hodnotou 1; pro $a = \frac{1}{2}$ se jedná o rozdělení $\frac{1}{2}Z^2$, kde Z má normální normované rozdělení se střední hodnotou 0 a rozptylem 1. Jsou-li X a Y nezávislé náhodné proměnné s gama rozdělením řádu a, resp. b, pak také veličina $X + Y$ má gama rozdělení, a to řádu $a + b$. Proto například součet k nezávislých exponenciálních veličin se střední hodnotou 1 má gama rozdělení řádu k. Jestliže tyto exponenciální veličiny generujeme pomocí logaritmické metody (32), stačí nám vypočítat jediný logaritmus, a sice $X \leftarrow -\ln(U_1 \ldots U_k)$, kde $U_1, \ldots, U_k$ jsou nenulové rovnoměrné veličiny. Tato technika pracuje s libovolným celočíselným řádem a; pro úplnost se ještě ve cvičení 16 objeví vhodná metoda pro $0 < a < 1$.

Prostá logaritmická metoda je ale pro velké a příliš pomalá, protože potřebuje $\lfloor a \rfloor$ rovnoměrných veličin. Navíc existuje velké riziko, že při součinu $U_1 \ldots U_{\lfloor a \rfloor}$ dojde k podtečení reálné aritmetiky. Pro velká a navrhl tedy J. H. Ahrens následující rozumně efektivní algoritmus, který se navíc snadno dá zapsat pomocí standardních podprogramů. [Viz *Ann. Inst. Stat. Math.* **13** (1962), 231–237.]

Algoritmus A (*Gama rozdělení řádu* $a > 1$).

A1. [Generování kandidáta.] Přiřaďte $Y \leftarrow \tan(\pi U)$, kde U je rovnoměrná veličina, a přiřaďte $X \leftarrow \sqrt{2a-1}\,Y + a - 1$. (Namísto $\tan(\pi U)$ můžeme využít polární metodu a vypočítat poměr V_2/V_1 jako v kroku P4 Algoritmu P.)

A2. [Přijetí?] Je-li $X \le 0$, vraťte se na A1. Jinak generujte rovnoměrnou veličinu V a vraťte se na A1, je-li $V > (1+Y^2)\exp\big((a-1)\ln\big(X/(a-1)\big) - \sqrt{2a-1}\,Y\big)$. Jinak přijměte X. ∎

Pro $a \ge 3$ je průměrný počet opakování kroku A1 $< 1{,}902$.

Existuje také další zajímavý způsob výpočtu pro velké a, který vychází z pozoruhodného faktu, že gama veličiny jsou přibližně rovné aX^3, kde X je proměnná s normálním rozdělením se střední hodnotou $1 - 1/(9a)$ a standardní odchylkou $1/\sqrt{9a}$; viz E. B. Wilson a M. M. Hilferty, *Proc. Nat. Acad. Sci.* **17** (1931), 684–688; G. Marsaglia, *Computers and Math.* **3** (1977), 321–325.*

Poněkud komplikovaný, ale výrazně rychlejší algoritmus, jenž generuje gama veličinu ve zhruba dvojnásobném čase než normální veličinu, představili J. H. Ahrens a U. Dieter, *CACM* **25** (1982), 47–54. Článek obsahuje podnětný výklad principů návrhu tohoto algoritmu.

2) *Beta rozdělení* s kladnými parametry a a b je definováno vztahem

$$F(x) = \frac{\Gamma(a+b)}{\Gamma(a)\,\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1}\,\mathrm{d}t, \qquad 0 \le x \le 1. \tag{35}$$

Jsou-li X_1 a X_2 nezávislé gama veličiny řádu a a b, přiřadíme $X \leftarrow X_1/(X_1+X_2)$. Při jiném postupu, užitečném při malém a a b, přiřazujeme opakovaně

$$Y_1 \leftarrow U_1^{1/a} \qquad \text{a} \qquad Y_2 \leftarrow U_2^{1/b}$$

tak dlouho, až je $Y_1 + Y_2 \le 1$; potom $X \leftarrow Y_1/(Y_1 + Y_2)$. [Viz M. D. Jöhnk, *Metrika* **8** (1964), 5–15.] Ještě jiný postup pro nepříliš velká celá a a b znamená

* V kroku 3 algoritmu na straně 323 citované publikace změňte „$+(3a-1)$“ na „$-(3a-1)$“.

do X přiřadit b-tou největší z $a+b-1$ nezávislých rovnoměrných veličin (viz cvičení 9 na začátku kapitoly 5). Viz též přímější metoda, kterou popsal R. C. H. Cheng, *CACM* **21** (1978), 317–322.

3) *chí-kvadrát rozdělení* s ν stupni volnosti (definice viz 3.3.1–(22)) získáme přiřazením $X \leftarrow 2Y$, kde Y je náhodná proměnná s gama rozdělením řádu $\nu/2$.

4) *F rozdělení* (Fisherovo rozdělení neboli rozdělení poměrů rozptylů) s ν_1 a ν_2 stupni volnosti je definováno vztahem

$$F(x) = \frac{\nu_1^{\nu_1/2}\,\nu_2^{\nu_2/2}\,\Gamma\big((\nu_1+\nu_2)/2\big)}{\Gamma(\nu_1/2)\,\Gamma(\nu_2/2)} \int_0^x t^{\nu_1/2-1}(\nu_2+\nu_1 t)^{-\nu_1/2-\nu_2/2}\,\mathrm{d}t, \quad (36)$$

kde $x \ge 0$. Nechť Y_1 a Y_2 jsou nezávislé proměnné s chí-kvadrát rozdělením s ν_1 a ν_2 stupni volnosti; přiřaďme $X \leftarrow Y_1\nu_2/Y_2\nu_1$. Nebo přiřadíme $X \leftarrow$ $\leftarrow \nu_2 Y/\nu_1(1-Y)$, kde Y je beta veličina s parametry $\nu_1/2$ a $\nu_2/2$.

5) *Studentovo t-rozdělení* s ν stupni volnosti je definováno vztahem

$$F(x) = \frac{\Gamma\big((\nu+1)/2\big)}{\sqrt{\pi\nu}\,\Gamma(\nu/2)} \int_{-\infty}^x (1+t^2/\nu)^{-(\nu+1)/2}\,\mathrm{d}t. \quad (37)$$

Nechť Y_1 je normovaná normální veličina (střední hodnota 0, rozptyl 1) a nechť Y_2 je nezávislé na Y_1 a má chí-kvadrát rozdělení s ν stupni volnosti; přiřadíme $X \leftarrow Y_1/\sqrt{Y_2/\nu}$. Alternativně pro $\nu > 2$ nechť Y_1 je normální veličina a nechť Y_2 je nezávislá proměnná s exponenciálním rozdělením se střední hodnotou $2/(\nu-2)$; přiřadíme $Z \leftarrow Y_1^2/(\nu-2)$ a zamítneme (Y_1, Y_2), pokud $e^{-Y_2-Z} \ge 1-Z$, jinak přiřadíme

$$X \leftarrow Y_1/\sqrt{(1-2/\nu)(1-Z)}.$$

Druhou metodu popsal George Marsaglia, *Math. Comp.* **34** (1980), 235–236. [Viz též A. J. Kinderman, J. F. Monahan a J. G. Ramage, *Math. Comp.* **31** (1977), 1009–1018.]

6) *Náhodný bod na n-rozměrné kulové ploše s jednotkovým poloměrem.* Nechť $X_1, X_2, \ldots, X_n$ jsou nezávislé normální veličiny (střední hodnota 0, rozptyl 1); požadovaný bod na jednotkové kulové ploše je

$$(X_1/r, X_2/r, \ldots, X_n/r), \qquad \text{kde } r = \sqrt{X_1^2+X_2^2+\cdots+X_n^2}. \quad (38)$$

Jestliže tyto veličiny X vypočítáme polární metodou Algoritmu P, vypočteme tím pokaždé dvě nezávislé proměnné X a v notaci zmíněného algoritmu máme $X_1^2+X_2^2 = -2\ln S$; tím si ušetříme určitý čas výpočtu r. Platnost vztahu (38) vyplývá z toho, že funkce hustoty rozdělení bodu $(X_1, \ldots, X_n)$ je závislá jen na jeho vzdálenosti od počátku, takže po projekci na jednotkovou kulovou plochu dostáváme rovnoměrné rozdělení. Tuto metodu jako první navrhl G. W. Brown, v knize *Modern Mathematics for the Engineer*, první série, redigoval E. F. Beckenbach (New York: McGraw–Hill, 1956), str. 302. Pro odvození náhodného bodu *uvnitř* n-rozměrné koule doporučuje R. P. Brent vzít bod na kulové ploše a vynásobit jej $U^{1/n}$.

V trojrozměrném prostoru můžeme využít významně jednodušší metodu, protože každá ze souřadnic má rovnoměrné rozdělení od -1 do 1: Pomocí kroků P1–P3 Algoritmu P najdeme V_1, V_2 a S; požadovaný náhodný bod na kulové ploše je pak $(\alpha V_1, \alpha V_2, 2S-1)$, kde $\alpha = 2\sqrt{1-S}$. [Robert E. Knop, *CACM* **13** (1970), 326.]

F. Významná rozdělení s celočíselnými hodnotami. Rozdělení pravděpodobnosti, tvořené jen celočíselnými hodnotami, můžeme v podstatě vyřešit technikami popsanými na začátku této části; některá z těchto rozdělení jsou ale v praxi natolik důležitá, že si zde zasluhují zvláštní pozornost.

1) *Geometrické rozdělení.* Jestliže nějaká událost nastává s pravděpodobností p, pak počet N nezávislých pokusů nutných pro další opakování události (nebo počet pokusů před jejím prvním nastoupením) má geometrické rozdělení. Je tedy $N = 1$ s pravděpodobností p, $N = 2$ s pravděpodobností $(1-p)p$, ..., $N = n$ s pravděpodobností $(1-p)^{n-1}p$. To je v podstatě stejná situace, jakou jsme uvažovali v mezerovém testu z části 3.3.2; zároveň přímo souvisí s počtem provádění určitých smyček v algoritmech této části, například kroků P1–P3 polární metody.

Proměnnou tohoto rozdělení můžeme šikovně vygenerovat přiřazením

$$N \leftarrow \lceil \ln U / \ln(1-p) \rceil. \tag{39}$$

Nyní ověříme správnost vzorce a všimneme si, že $\lceil \ln U / \ln(1-p) \rceil = n$ tehdy a jen tehdy, je-li $n - 1 < \ln U / \ln(1-p) \le n$, to znamená, $(1-p)^{n-1} > U \ge (1-p)^n$, což nastává právě s požadovanou pravděpodobností $(1-p)^{n-1}p$. Veličinu $\ln U$ je možné nahradit za $-Y$, kde Y má exponenciální rozdělení se střední hodnotou 1.

Speciální případ $p = \frac{1}{2}$ je na dvojkovém počítači docela jednoduchý, protože předpis (39) se redukuje na přiřazení $N \leftarrow \lceil -\lg U \rceil$; to znamená, že N je o jedničku vyšší než počet vedoucích nulových bitů ve dvojkové reprezentaci U.

2) *Binomické rozdělení* (t, p). Jestliže nějaká událost nastává s pravděpodobností p a jestliže provedeme t nezávislých pokusů, pak se celkový počet N výskytů události rovná n s pravděpodobností $\binom{t}{n} p^n (1-p)^{t-n}$. (Viz část 1.2.10.) Jinými slovy, při generování veličin U_1, ..., U_t, potřebujeme zjistit, kolik z nich je $< p$. Pro malé t můžeme N získat právě takto.

Pro velká t můžeme vygenerovat beta veličinu X s celočíselnými parametry a a b, kde $a + b - 1 = t$; tím fakticky dostaneme b-tý největší z t prvků, aniž bychom museli generovat i ostatní. Nyní je-li $X \ge p$, přiřadíme $N \leftarrow N_1$, kde N_1 má binomické rozdělení $(a-1, p/X)$, protože vyjadřuje, kolik z $a-1$ náhodných čísel v intervalu $[0\,..\,X)$ je $< p$; a je-li $X < p$, přiřadíme $N \leftarrow a + N_1$, kde N_1 má binomické rozdělení $\big(b-1, (p-X)/(1-X)\big)$, protože N_1 ukazuje, kolik z $b-1$ náhodných čísel v intervalu $[X\,..\,1)$ je $< p$. Zvolíme-li $a = 1 + \lfloor t/2 \rfloor$, sníží se parametr t na rozumnou mez po zhruba $\lg t$ redukcích tohoto typu. (Postup objevil J. H. Ahrens, který také navrhl alternativní metodu pro středně velká t; viz cvičení 27.)

3) *Poissonovo rozdělení* se střední hodnotou μ. Poissonovo rozdělení má podobný vztah k exponenciálnímu rozdělení, jaký má binomické rozdělení ke geome-

trickému: Vyjadřuje za jednotku času počet výskytů události, která může nastat v kterémkoli okamžiku. Příkladem je počet alfa částic, emitovaných radioaktivní látkou za jednu sekundu; ten má právě Poissonovo rozdělení.

Na základě tohoto principu již můžeme vytvořit poissonovskou veličinu N; k tomu vygenerujeme nezávislé exponenciální veličiny X_1, X_2, ... se střední hodnotou $1/\mu$ a zastavíme se, jakmile $X_1 + \cdots + X_m \geq 1$; potom je $N \leftarrow m-1$. Pravděpodobnost $X_1+\cdots+X_m \geq 1$ je rovna pravděpodobnosti, že gama veličina řádu m je $\geq \mu$, což vede k $\int_\mu^\infty t^{m-1}e^{-t}\,\mathrm{d}t/(m-1)!$; pravděpodobnost $N = n$ je tudíž rovna

$$\frac{1}{n!}\int_\mu^\infty t^n e^{-t}\,\mathrm{d}t - \frac{1}{(n-1)!}\int_\mu^\infty t^{n-1}e^{-t}\,\mathrm{d}t = e^{-\mu}\frac{\mu^n}{n!}, \qquad n \geq 0. \tag{40}$$

Při generování exponenciálních veličin pomocí logaritmické metody se výše uvedený předpis zastaví, jakmile je $-(\ln U_1 + \cdots + \ln U_m)/\mu \geq 1$. Po zjednodušení výrazu uvidíme, že požadovanou poissonovskou veličinu můžeme získat výpočtem hodnoty $e^{-\mu}$, kterou pak převedeme do pevné řádové čárky a potom vygenerujeme jednu nebo více rovnoměrných veličin U_1, U_2, ..., až součin splní nerovnost $U_1 \ldots U_m \leq e^{-\mu}$; nakonec přiřadíme $N \leftarrow m-1$. V průměru k tomu potřebujeme vygenerovat $\mu + 1$ rovnoměrných veličin, a proto je postup velmi vhodný pro nepříliš velké μ.

Je-li μ velké, můžeme metodu řádu $\log\mu$ odvodit z toho, že již umíme pracovat s gama a binomickým rozdělením velkých řádů: Nejprve vygenerujeme X s gama rozdělením řádu $m = \lfloor \alpha\mu \rfloor$, kde α je vhodná konstanta. (Protože X je ekvivalentní $-\ln(U_1 \ldots U_m)$, v podstatě z předchozí metody vynecháváme m kroků.) Je-li $X < \mu$, přiřadíme $N \leftarrow m + N_1$, kde N_1 je poissonovská veličina se střední hodnotou $\mu - X$; a je-li $X \geq \mu$, přiřadíme $N \leftarrow N_1$, kde N_1 má binomické rozdělení $(m-1, \mu/X)$. Tuto metodu odvodili J. H. Ahrens a U. Dieter, z jejichž experimentů vyplývá, že vhodnou hodnotou α je $\frac{7}{8}$.

Platnost uvedené redukce pro $X \geq \mu$ je důsledkem následujícího důležitého principu: „Nechť X_1, ..., X_m jsou nezávislé exponenciální veličiny se stejnou střední hodnotou; položme $S_j = X_1 + \cdots + X_j$ a dále $V_j = S_j/S_m$ pro $1 \leq \leq j \leq m$. Potom rozdělení V_1, V_2, ..., V_{m-1} je stejné jako rozdělení $m-1$ nezávislých rovnoměrných veličin seřazených do rostoucího pořadí.“ Formálně tento princip zavedeme tak, že vypočteme pravděpodobnost, s níž je $V_1 \leq v_1$, ..., $V_{m-1} \leq v_{m-1}$, je-li dána hodnota $S_m = s$, pro libovolné hodnoty $0 \leq v_1 \leq \leq \cdots \leq v_{m-1} \leq 1$: Nechť $f(v_1, v_2, \ldots, v_{m-1})$ je $(m-1)$-násobný integrál

$$\int_0^{v_1 s} \mu e^{-t_1/\mu}\,\mathrm{d}t_1 \int_0^{v_2 s - t_1} \mu e^{-t_2/\mu}\,\mathrm{d}t_2 \ldots \times \int_0^{v_{m-1}s - t_1 - \cdots - t_{m-2}} \mu e^{-t_{m-1}/\mu}\,\mathrm{d}t_{m-1} \cdot \mu e^{-(s - t_1 - \cdots - t_{m-1})/\mu};$$

potom

$$\frac{f(v_1, v_2, \ldots, v_{m-1})}{f(1, 1, \ldots, 1)} = \frac{\int_0^{v_1}\,\mathrm{d}u_1 \int_{u_1}^{v_2}\,\mathrm{d}u_2 \ldots \int_{u_{m-2}}^{v_{m-1}}\,\mathrm{d}u_{m-1}}{\int_0^1\,\mathrm{d}u_1 \int_{u_1}^1\,\mathrm{d}u_2 \ldots \int_{u_{m-2}}^1\,\mathrm{d}u_{m-1}},$$

provedeme-li substituci $t_1 = su_1$, $t_1 + t_2 = su_2$, ..., $t_1 + \cdots + t_{m-1} = su_{m-1}$. Druhý uvedený poměr je roven příslušné pravděpodobnosti, že rovnoměrné veličiny $U_1, \ldots, U_{m-1}$ splňují nerovnosti $U_1 \le v_1, \ldots, U_{m-1} \le v_{m-1}$, pokud zároveň splňují $U_1 \le \cdots \le U_{m-1}$.

Efektivnější, i když poněkud komplikovanější techniku pro binomické a poissonovské veličiny si načrtneme ve cvičení 22.

G. Zdroje dalších informací. V článku *Stanislaw Ulam 1909–1984*, zvláštní vydání *Los Alamos Science* (Los Alamos National Lab., 1987), 135–136, je otištěna faksimile dopisu od von Neumanna s datem 21. května 1947, v němž poprvé spatřila světlo světa zamítací metoda. Kniha *Non-Uniform Random Variate Generation*, kterou napsal L. Devroye (Springer, 1986), popisuje mnohem více algoritmů pro generování náhodných proměnných s nerovnoměrnými rozděleními, a také pečlivě rozebírá efektivitu každého z nich na typickém počítači.

W. Hörmann a G. Derflinger poukázali [*ACM Trans. Math. Software* **19** (1993), 489–495], že může být nebezpečné používat zamítací metodu ve spojení s lineárními kongruentními generátory s malými násobiteli, $a \approx \sqrt{m}$.

Z teoretického pohledu je zajímavé uvažovat *optimální* metody generování náhodných proměnných daného rozdělení, a to v tom smyslu, že metoda má požadovaných výsledků dosáhnout z minimálního možného počtu náhodných bitů. Počátky teorie k těmto otázkám popisují D. E. Knuth a A. C. Yao, *Algorithms and Complexity*, redigoval J. F. Traub (New York: Academic Press, 1976), 357–428.

Přehled různých technik z této části nás čeká ve cvičení 16.

CVIČENÍ

1. [*10*] Pokud jsou α a β reálná čísla, pro něž $\alpha < \beta$, jak vygenerujete náhodné reálné číslo s rovnoměrným rozdělením mezi α a β?

2. [*M16*] Předpokládejme, že mU je náhodné celé číslo od 0 do $m-1$; jaká je *přesná* pravděpodobnost, že $\lfloor kU \rfloor = r$, pokud je $0 \le r < k$? Výsledek porovnejte s požadovanou pravděpodobností $1/k$.

▸ **3.** [*14*] Rozeberte postup, kdy budeme U považovat za celé číslo a náhodné celé číslo od 0 do $k-1$ budeme generovat pomocí výpočtu jeho *zbytku* mod k, nikoli násobením jako je uvedeno v textu. Instrukce (1) tudíž změníme na

```
ENTA 0;  LDX U;  DIV K,
```

přičemž výsledek se zapíše do registru X. Je tato metoda dobrá?

4. [*M20*] Dokažte obě relace v (8).

▸ **5.** [*21*] Navrhněte efektivní způsob výpočtu náhodné proměnné s rozdělením $F(x) = px + qx^2 + rx^3$, kde $p \ge 0$, $q \ge 0$, $r \ge 0$ a $p + q + r = 1$.

6. [*VM21*] Veličinu X vypočítáme následujícím postupem:

Krok 1. Vygenerujte dvě nezávislé rovnoměrné veličiny U a V.

Krok 2. Je-li $U^2 + V^2 \ge 1$, vraťte se na krok 1; jinak přiřaďte $X \leftarrow U$.

Jaká je distribuční funkce X? Kolikrát se bude provádět krok 1? (Uveďte střední hodnotu a standardní odchylku.)

▸ **7.** [*20*] (A. J. Walker.) Předpokládejme, že máme hromadu kostek k různých barev, dejme tomu n_j kostek barvy C_j pro $1 \le j \le k$, a dále máme k krabic, $\{B_1, \ldots, B_k\}$ z nichž do každé se vejde právě n kostek. Navíc je $n_1 + \cdots + n_k = kn$, takže kostky se přesně vejdou do připravených krabic. Podejte konstruktivní důkaz, že vždy existuje takový způsob rozmístění kostek do krabic, že každá krabice obsahuje kostky nejvýše dvou různých barev; ve skutečnosti to lze provést dokonce tak, že kdykoli krabice B_j obsahuje dvě barvy, je jednou z těchto barev C_j. Ukažte, jako pomocí tohoto principu vypočítat tabulky P a Y potřebné v (3), je-li dáno rozdělení pravděpodobnosti $(p_1, \ldots, p_k)$.

8. [*M15*] Ukažte, že je možné operaci (3) změnit na

$$\text{je-li} \quad U < P_K \quad \text{pak} \quad X \leftarrow x_{K+1} \quad \text{jinak} \quad X \leftarrow Y_K$$

(to znamená, že namísto V použijeme původní hodnotu U), pokud je to vhodnější, a to pomocí příslušné modifikace P_0, P_1, …, P_{k-1}.

9. [*VM10*] Proč je křivka $f(x)$ na Obr. 9 konkávní pro $x < 1$ a konvexní pro $x > 1$?

▸ **10.** [*VM24*] Vysvětlete, jak vypočítat pomocné konstanty P_j, Q_j, Y_j, Z_j, S_j, D_j, E_j tak, aby Algoritmus M vydal odpovědi ve správném rozdělení.

▸ **11.** [*VM27*] Dokažte, že kroky M7–M8 Algoritmu M vygenerují náhodnou proměnnou s příslušným okrajem normálního rozdělení; jinými slovy pravděpodobnost $X \le x$ bude přesně

$$\int_3^x e^{-t^2/2}\,\mathrm{d}t \Big/ \int_3^\infty e^{-t^2/2}\,\mathrm{d}t, \qquad x \ge 3.$$

[*Nápověda*: Ukažte, že se jedná o speciální případ zamítací metody, kde $g(t) = Cte^{-t^2/2}$ pro nějaké C.]

12. [*VM23*] (R. P. Brent.) Dokažte, že čísla a_j definovaná ve vztahu (23) splňují relaci

$$a_j^2 - a_{j-1}^2 < 2\ln 2 \qquad \text{pro všechna } j \ge 1.$$

[*Nápověda*: Je-li $f(x) = e^{x^2/2}\int_x^\infty e^{-t^2/2}\,\mathrm{d}t$, pak ukažte, že $f(x) > f(y)$ pro $0 \le x < y$.]

13. [*VM25*] Je-li dána množina n nezávislých normálních veličin X_1, X_2, …, X_n se střední hodnotou 0 a rozptylem 1, ukažte, jak nalézt konstanty b_j a a_{ij}, $1 \le j \le i \le n$, takové, že je-li

$$Y_1 = b_1 + a_{11}X_1, \quad Y_2 = b_2 + a_{21}X_1 + a_{22}X_2, \quad \ldots, \quad Y_n = b_n + a_{n1}X_1 + \cdots + a_{nn}X_n,$$

pak Y_1, Y_2, …, Y_n jsou závislé proměnné s normálním rozdělením, Y_j má střední hodnotu μ_j a proměnné Y mají danou kovarianční matici. (c_{ij}). (Kovariance c_{ij} z Y_i a Y_j je definována jako průměrná hodnota z $(Y_i - \mu_i)(Y_j - \mu_j)$. Zejména c_{jj} je rozptyl Y_j, tedy čtverec její standardní odchylky. Ne všechny matice (c_{ij}) mohou být kovarianční maticí a v úloze samozřejmě stačí, pokud za podmínek, kdy existuje řešení, bude vaše konstrukce fungovat.)

14. [*M21*] Je-li X náhodná proměnná se spojitým rozdělením $F(x)$ a je-li c konstanta (může být i záporná), jaké je rozdělení cX?

15. [*VM21*] Pokud jsou X_1 a X_2 dvě nezávislé náhodné proměnné s příslušnými rozděleními $F_1(x)$ a $F_2(x)$, a s funkcemi hustoty $f_1(x) = F_1'(x)$, $f_2(x) = F_2'(x)$, jaká je distribuční funkce a hustota pravděpodobnosti veličiny $X_1 + X_2$?

▸ **16.** [*VM22*] (J. H. Ahrens.) Navrhněte algoritmus pro generování gama veličin řádu a pro $0 < a \le 1$, postavený na zamítací metodě s $cg(t) = t^{a-1}/\Gamma(a)$ pro $0 < t < 1$, a s $cg(t) = e^{-t}/\Gamma(a)$ pro $t \ge 1$.

▸ **17.** [*M24*] Jaká je *distribuční funkce* $F(x)$ pro geometrické rozdělení s pravděpodobností p? Jaká je příslušná *generující funkce* $G(z)$? A jakou má toto rozdělení střední hodnotu a standardní odchylku?

18. [*M24*] Navrhněte metodu pro výpočet náhodného celého čísla N, kde N nabývá hodnoty n s pravděpodobností $np^2(1-p)^{n-1}$, $n \ge 0$. (Zvláště zajímavý případ je pro poměrně malé p.)

19. [*22*] *Záporné binomické rozdělení* (t, p) nabývá celočíselných hodnot $N = n$ s pravděpodobností $\binom{t-1+n}{n} p^t (1-p)^n$. (Na rozdíl od běžného binomického rozdělení zde t nemusí být celé, protože tato veličina je nezáporná pro všechna n, kde $t > 0$.) Zobecněte cvičení 18 a vysvětlete, jak vygenerovat celá čísla N s tímto rozdělením, je-li t malé kladné celé číslo. Jakou metodu navrhnete, pokud je $t = p = \frac{1}{2}$?

20. [*M20*] Nechť A je velikost stínované plochy na Obr. 13 a nechť R je plocha opsaného obdélníku. Dále nechť I je velikost vnitřní plochy přijaté v kroku R2 a E velikost plochy mezi vnější plochou zamítnutou v kroku E3 a vnějším obdélníkem. Určete, kolikrát se bude provádět každý z kroků Algoritmu R, pro každou z jeho čtyř variant popsaných v (2_5), a vyjádřete pomocí A, R, I a E.

21. [*VM29*] Odvoďte vztahy pro veličiny A, R, I a E definované ve cvičení 20. (Pro I a zejména E je vhodné využít interaktivní počítačový algebraický systém.) Ukažte, že $c = e^{1/4}$ je nejlepší možná konstanta v kroku R2 pro testy tvaru „$X^2 \le 4(1+\ln c) - 4cU$“.

22. [*VM40*] Můžeme přesné Poissonovo rozdělení pro velké μ získat vygenerováním odpovídající normální veličiny, kterou následně vhodným způsobem převedeme na celé číslo a na malé procento případů aplikujeme jistou korekci (i když může být komplikovaná)?

23. [*VM23*] (J. von Neumann.) Jsou následující dvě metody generování náhodné veličiny X ekvivalentní (tedy má veličina X stejné rozdělení)?

Metoda 1: Přiřaďte $X \leftarrow \sin\big((\pi/2)U\big)$, kde U má rovnoměrné rozdělení.

Metoda 2: Vygenerujte dvě rovnoměrné veličiny U a V; je-li $U^2 + V^2 \ge 1$, opakujte krok až je $U^2 + V^2 < 1$. Potom přiřaďte $X \leftarrow |U^2 - V^2|/(U^2 + V^2)$.

24. [*VM40*] (S. Ulam, J. von Neumann.) Nechť V_0 je náhodně zvolené reálné číslo od 0 do 1 a nechť je definována posloupnost $\langle V_n \rangle$ podle pravidla $V_{n+1} = 4V_n(1 - V_n)$. Provedeme-li výpočet s dokonalou přesností, bude výsledkem posloupnost s rozdělením $\sin^2 \pi U$, kde U je rovnoměrné, tedy s distribuční funkcí $F(x) = \int_0^x \mathrm{d}x/\sqrt{2\pi x(1-x)}$. Jestliže totiž napíšeme $V_n = \sin^2 \pi U_n$, zjistíme, že $U_{n+1} = (2U_n) \bmod 1$; a protože téměř všechna reálná čísla mají náhodný dvojkový rozvoj (viz část 3.5), je posloupnost U_n ekvidistribuovaná. Pokud ale výpočet V_n provádíme jen s konečnou přesností, zdůvodnění již neplatí, protože nás brzy ovlivní šum ze zaokrouhlovací chyby. [Viz von Neumann, *Collected Works* **5**, 768–770.]

Analyzujte takto definovanou posloupnost $\langle V_n \rangle$ za podmínky jen konečné přesnosti, a to jak empiricky (pro různě zvolená V_0), tak i teoreticky. Podobá se skutečné rozdělení posloupnosti očekávanému?

25. [*M25*] Nechť $X_1, X_2, \ldots, X_5$ jsou dvojková slova, v nichž každý bit má nezávislou hodnotu 0 nebo 1 s pravděpodobností $\frac{1}{2}$. S jakou pravděpodobností bude daná bitová pozice čísla $X_1 \mid (X_2 \,\&\, (X_3 \mid (X_4 \,\&\, X_5)))$ obsahovat 1? Zobecněte.

26. [*M18*] Nechť N_1 a N_2 jsou nezávislé poissonovské veličiny se středními hodnotami μ_1 a μ_2, kde $\mu_1 > \mu_2 \geq 0$. Dokažte nebo vyvraťte: (a) $N_1 + N_2$ má Poissonovo rozdělení se střední hodnotou $\mu_1 + \mu_2$. (b) $N_1 - N_2$ má Poissonovo rozdělení se střední hodnotou $\mu_1 - \mu_2$.

27. [*22*] (J. H. Ahrens.) Na většině dvojkových počítačů existuje efektivní způsob spočtení jedniček ve dvojkovém slově (viz část 7.1). Existuje tedy i šikovný způsob získání binomického rozdělení (t, p) pro $p = \frac{1}{2}$; stačí vygenerovat t náhodných bitů a spočítat jedničky.

Navrhněte algoritmus, který generuje binomické rozdělení (t, p) pro *libovolné* p, přičemž využije jen podprogram pro speciální případ $p = \frac{1}{2}$, který bude zdrojem náhodných dat. [*Nápověda*: Simulujte proces, který se nejprve podívá vždy na nejvýznamnější bit z t rovnoměrných veličin, potom u těch veličin, kde podle prvního bitu nepoznáme, jestli je hodnota $< p$, se podívá na druhý bit atd.]

28. [*VM35*] (R. P. Brent.) Navrhněte metodu pro generování náhodného bodu na povrchu elipsoidu, definovaného rovnicí $\sum a_k x_k^2 = 1$, kde $a_1 \geq \cdots \geq a_n > 0$.

29. [*M20*] (J. L. Bentley a J. B. Saxe.) Najděte jednoduchý způsob vygenerování n čísel $X_1, \ldots, X_n$, která mají rovnoměrné rozdělení od 0 do 1, ale jsou seřazená: $X_1 \leq \cdots \leq X_n$. Algoritmus by měl provádět jen $O(n)$ kroků.

30. [*M30*] Vysvětlete, jak lze vygenerovat množinu náhodných bodů (X_j, Y_j) takových, že pokud R je libovolný obdélník o ploše α, obsažený v jednotkovém čtverci, pak počet bodů (X_j, Y_j) ležících v R má Poissonovo rozdělení se střední hodnotou $\alpha\mu$.

31. [*VM39*] (*Přímé generování normálních veličin.*)

a) Dokažte, že pokud je $a_1^2 + \cdots + a_k^2 = 1$ a pokud jsou $X_1, \ldots, X_k$ nezávislé normální veličiny se střední hodnotou 0 a rozptylem 1, pak i $a_1 X_1 + \cdots + a_k X_k$ je normální veličina se střední hodnotou 0 a rozptylem 1.

b) V důsledku (a) můžeme generovat nové normální veličiny ze starších, stejně jako generujeme nové rovnoměrné veličiny ze starých hodnot. Můžeme například využít myšlenku 3.2.2–(7), avšak s rekurencí jako

$$X_n = (X_{n-24} + X_{n-55})/\sqrt{2} \qquad \text{nebo} \qquad X_n = \tfrac{3}{5}X_{n-24} + \tfrac{4}{5}X_{n-55},$$

po prvotním vypočtení množiny normálních veličin $X_0, \ldots, X_{54}$. Vysvětlete, proč tento postup *není* vhodný.

c) Ukažte ale, že *existuje* vhodný způsob rychlého generování normálních veličin z jiných normálních veličin, pokud vylepšíme myšlenky z (a) a (b). [*Nápověda*: Jsou-li X a Y nezávislé normální veličiny, jsou nezávislé normální veličiny i $X' = X\cos\theta + Y\sin\theta$ a $Y' = -X\sin\theta + Y\cos\theta$, pro libovolný úhel θ.]

32. [*VM30*] (C. S. Wallace.) Nechť X a Y jsou nezávislé exponenciální veličiny se střední hodnotou 1. Ukažte, že také X' a Y', které získáme z X a Y některým z následujících způsobů, jsou nezávislými exponenciálními veličinami se střední hodnotou 1:

a) Je-li dáno $0 < \lambda < 1$, pak

$$X' = (1-\lambda)X - \lambda Y + (X+Y)[(1-\lambda)X < \lambda Y], \qquad Y' = X + Y - X'.$$

b) $(X', Y') = \begin{cases} (2X, Y-X), & \text{je-li } X \leq Y; \\ (2Y, X-Y), & \text{je-li } X > Y. \end{cases}$

c) Je-li $X = (\ldots x_2 x_1 x_0 . x_{-1} x_{-2} x_{-3} \ldots)_2$ a $Y = (\ldots y_2 y_1 y_0 . y_{-1} y_{-2} y_{-3} \ldots)_2$ ve dvojkovém vyjádření, pak X' a Y' mají „pomíchané“ hodnoty

$$X' = (\ldots x_2 y_1 x_0 . y_{-1} x_{-2} y_{-3} \ldots)_2, \qquad Y' = (\ldots y_2 x_1 y_0 . x_{-1} y_{-2} x_{-3} \ldots)_2.$$

33. [*20*] Algoritmy P, M, F a R při generování normální veličiny spotřebovávají neznámý počet rovnoměrných náhodných proměnných $U_1, U_2, \ldots$. Jak je můžeme upravit, aby byl výstup funkcí jediné proměnné U?

3.4.2. Náhodné vzorkování a míchání

Mnoho aplikací pro zpracování dat vyžaduje nestranný výběr n náhodných záznamů ze souboru obsahujícího N záznamů. Problém vzniká například při řízení jakosti (kvality) či v jiných statistických výpočtech, kde potřebujeme vzorkování. Obvykle je N velmi velké, takže nelze načíst všechna data do paměti najednou; často i jednotlivé záznamy jsou velké, takže nemůžeme do paměti dát ani n vybraných záznamů. Hledáme proto efektivní proceduru pro výběr n záznamu, v níž se budeme průběžně rozhodovat, jestli každý záznam přijmout nebo zamítnout, přičemž přijaté záznamy zapíšeme do výstupního souboru.

Jako řešení tohoto problému bylo navrženo několik metod. Zřejmá možnost je vybrat každý záznam s pravděpodobností n/N; to může být někdy rozumné, ale ve vzorku tím dostaneme jen *průměrně* n záznamů. Standardní odchylka je $\sqrt{n(1-n/N)}$ a vzorek může být nakonec pro potřeby dané aplikace příliš velký, nebo naopak příliš malý, než aby z něj mohly vyplynout požadované výsledky.

Kýženou metodu naštěstí dostaneme po jednoduché úpravě této „zřejmé“ procedury: $(t+1)$-ní záznam stačí vybrat s pravděpodobností $(n-m)/(N-t)$, pokud jsme již vybrali m položek. To je správná pravděpodobnost, protože mezi všemi možnými způsoby výběru n prvků z N takových, že mezi prvními t bude m hodnot, vybírá právě

$$\binom{N-t-1}{n-m-1} \Big/ \binom{N-t}{n-m} = \frac{n-m}{N-t} \tag{1}$$

z nich $(t+1)$-ní prvek.

Myšlenka představená v předchozím odstavci vede okamžitě k následujícímu algoritmu:

Algoritmus S (*Technika vzorkování výběru*). Vybereme náhodně n záznamů z množiny N, kde $0 < n \leq N$.

S1. [Inicializace.] Přiřaďte $t \leftarrow 0$, $m \leftarrow 0$. (Během tohoto algoritmu reprezentuje m počet dosud vybraných záznamů a t je celkový počet vstupních záznamů, s nimiž jsme dosud pracovali.)

S2. [Generování U.] Vygenerujte náhodné číslo U s rovnoměrným rozdělením od 0 do 1.

S3. [Testování.] Je-li $(N-t)U \geq n-m$, jděte na krok S5.

S4. [Výběr.] Vyberte další záznam do vzorku a zvětšete m a t o 1. Je-li $m < n$, jděte na S2; jinak je vzorek hotový a algoritmus končí.

S5. [Přeskočení.] Přeskočte další záznam (nezahrnujte jej do vzorku), zvětšete t o 1, a vraťte se na krok S2. ∎

Tento algoritmus se může zdát na první pohled nespolehlivý a dokonce i nesprávný, avšak pečlivá analýza (viz níže uvedená cvičení) nám jasně ukáže, že je plně důvěryhodný. Není těžké si ověřit, že:

a) Vstupem je nejvýše N záznamů (to znamená, že před vybráním n položek se nikdy nedostaneme za konec souboru).

b) Vzorek je zcela nestranný. Zejména pravděpodobnost výběru libovolného konkrétního prvku souboru, třeba posledního, je vždy n/N.

Tvrzení (b) platí i přesto, že $(t+1)$-ní položku *nevybíráme* s pravděpodobností n/N, nýbrž s pravděpodobností podle (1)! I v literatuře to bylo zdrojem jistých nedorozumění. Dokáže bystrý čtenář tento zdánlivý rozpor vysvětlit?

(*Poznámka*: Při každém spuštění Algoritmu S je potřeba vzít jiný zdroj náhodných čísel U, protože jinak by vzorky získané v různých dnech byly na sobě závislé. K tomu stačí pokaždé zvolit jinou počáteční hodnotu X_0 pro lineární kongruenční metodu; můžeme do ní zapsat třeba aktuální datum, nebo poslední náhodné číslo X vygenerované při posledním běhu programu.)

Obvykle takto nemusíme procházet všech N záznamů. Protože, jak říká tvrzení (b), poslední záznam vybereme s pravděpodobností n/N, přesně v $(1-n/N)$ případů ukončíme algoritmus ještě *před* posledním záznamem. Průměrně pro $n=2$ projdeme zhruba $\frac{2}{3}N$ záznamů, přičemž obecné vzorce odvodíme ve cvičeních 5 a 6.

Nejen Algoritmus S, ale i řadu dalších technik vzorkování rozebírají ve svém článku C. T. Fan, Mervin E. Muller a Ivan Rezucha, *J. Amer. Stat. Assoc.* **57** (1962), 387–402. Metodu nezávisle na nich objevil také T. G. Jones, *CACM* **5** (1962), 343.

Problém ovšem nastává, pokud číslo N neznáme předem, protože jeho přesná hodnota je pro činnost Algoritmu S podstatná. Dejme tomu, že chceme vybrat ze souboru náhodně n položek, ale nevíme přesně, kolik záznamů v něm skutečně je. Mohli bychom v jednom průchodu záznamy spočítat a ve druhém je vybírat, ale obvykle je vhodnější udělat na první průchod vzorek $m \geq n$ z původních položek, kde m je podstatně menší než N, takže ve druhém průchodu již zpracováváme pouze m záznamů. Podstatné samozřejmě je, udělat tento průchod takovým způsobem, aby konečný výsledek byl opravdu náhodným vzorkem původního souboru.

Protože dopředu nevíme, kdy vstupní soubor skončí, musíme sledovat náhodný vzorek dosud načtených vstupních záznamů a být neustále připraveni na konec. Během načítání vstupu si vytvoříme „rezervoár“, jenž obsahuje jen ty záznamy, které se již mezi předchozími vzorky vyskytly. Prvních n záznamů jde tedy do rezervoáru vždy. Jakmile se na vstupu nachází $(t+1)$-ní záznam, pro $t \geq n$, máme již v paměti tabulku n indexů ukazujících na záznamy, které jsme vybrali z prvních t. Problém je, udržovat tuto situaci při postupném zvyšování t o jedničku; to znamená, že musíme vybrat nový náhodný vzorek vždy z $t+1$ aktuálně přítomných záznamů. Není těžké nahlédnout, že nový záznam musíme

do nového vzorku dát s pravděpodobností $n/(t+1)$, který v takovém případě nahradí náhodný prvek z předchozího vzorku.

Celou operaci provádí tudíž následující procedura:

Algoritmus R (*Vzorkování rezervoáru*). Vybírá náhodně n záznamů ze souboru o neznámé velikosti $\geq n$, kde $n > 0$. Pomocný soubor označovaný jako „rezervoár" obsahuje všechny záznamy, které jsou kandidátem do výsledného vzorku. Algoritmus používá tabulku různých indexů $I[j]$ pro $1 \leq j \leq n$, z nichž každý ukazuje na jeden ze záznamů v rezervoáru.

R1. [Inicializace.] Načtěte prvních n záznamů a zkopírujte je do rezervoáru. Přiřaďte $I[j] \leftarrow j$ pro $1 \leq j \leq n$, a dále přiřaďte $t \leftarrow m \leftarrow n$. (Pokud má vzorkovaný soubor méně než n záznamů, je samozřejmě nutné algoritmus zastavit a oznámit chybu. Během algoritmu ukazují indexy $I[1], \ldots, I[n]$ na záznamy v aktuálním vzorku; m je velikost rezervoáru a t je počet dosud přečtených vstupních záznamů.)

R2. [Konec souboru?] Pokud již nelze načíst další záznam, jděte na krok R6.

R3. [Generování a testování.] Zvětšete t o 1 a potom vygenerujte náhodné celé číslo M od 1 do t včetně. Je-li $M > n$, jděte na R5.

R4. [Přidání do rezervoáru.] Zkopírujte další záznam ze vstupního souboru do rezervoáru, zvětšete m o 1 a přiřaďte $I[M] \leftarrow m$. (Záznam, na který dříve ukazoval $I[M]$, je ve vzorku nahrazen novým záznamem.) Vraťte se na R2.

R5. [Přeskočení.] Přeskočte další záznam vstupního souboru (neukládejte jej do rezervoáru) a vraťte se na krok R2.

R6. [Druhý průchod.] Seřaďte tabulku položek I tak, že bude $I[1] < \cdots < I[n]$; potom projděte rezervoár a do výstupního souboru zkopírujte záznamy s těmito indexy, jež tvoří výsledný vzorek. ▮

Algoritmus R navrhl Alan G. Waterman. Čtenář si může vypracovat příklad jeho činnosti, uvedený ve cvičení 9.

Pokud jsou záznamy dostatečně krátké, není samozřejmě nutné vůbec pracovat s rezervoárem; všech n záznamů aktuálního vzorku můžeme mít stále v paměti a algoritmus se výrazně zjednoduší (viz cvičení 10).

K Algoritmu R se nyní naskýtá přirozená otázka: „Jaká je očekávaná velikost rezervoáru?" Ve cvičení 11 si ukážeme, že průměrná hodnota m je přesně $n(1 + H_N - H_n)$, což je přibližně rovno $n\big(1 + \ln(N/n)\big)$. Pro $N/n = 1\,000$ bude tedy rezervoár obsahovat jen asi 1/125 položek z původního souboru.

Všimněte si, že pomocí Algoritmů S a R je možné získat vzorky pro několik nezávislých kategorií současně. Máme-li například rozsáhlý soubor se jmény a adresami občanů USA, můžeme snadno vytvořit náhodné vzorky po 10 osobách v každém z 50 států, aniž bychom museli provádět 50 průchodů a aniž bychom museli nejprve data seřadit podle státu.

Algoritmy S i R je možné pro malé n/N významně zlepšit, pokud se nebudeme zvlášť rozhodovat o přeskočení každého záznamu, ale namísto toho nejprve vygenerujeme jednu náhodnou proměnnou, která pak bude udávat počet současně přeskočených záznamů. (Viz cvičení 8.)

Na problém vzorkování můžeme pohlížet jako na výpočet náhodné *kombinace* n prvků z N podle obvyklé definice (viz část 1.2.6). Nyní uvažujme problém výpočtu náhodné *permutace* t předmětů; hovoříme o problému *míchání*, protože míchání karet v balíčku není nic jiného, než provedení náhodné permutace.

Na základě těchto úvah nás jistě ihned napadne, že bychom mohli každého hráče karet přesvědčit, že tradiční postupy jsou naprosto neadekvátní. Těmito metodami naprosto nelze zaručit, že každou z $t!$ permutací dostaneme se zhruba stejnou pravděpodobností. Zkušení hráči bridge se prokazatelně takto rozhodují, jestli se mají pokusit přebít zdvih. Pro dosažení rozdělení, které se liší nejvýše o 10 % od rovnoměrného, je potřeba u balíčku 52 karet nejméně sedm kol „míchání zasouváním“ (riffle shuffles, někdy také styl farao, pozn. překl.), přičemž 14 náhodných zasunutí již stačí zaručeně [viz Aldous a Diaconis, *AMM* **93** (1986), 333–348].

Je-li t malé, můžeme náhodnou permutaci získat velmi rychle vygenerováním náhodného celého čísla od 1 do $t!$. Pro $t = 4$ stačí například k výběru náhodné permutace ze všech možností zvolit náhodné číslo od 1 do 24. Pro velké t již ale s výrokem o stejné pravděpodobnosti každé permutace musíme být opatrnější, protože $t!$ je mnohem větší než přesnost jednotlivého náhodného čísla.

Vhodnou proceduru pro míchání získáme na základě Algoritmu 3.3.2P, jenž dává jednoduchou korespondenci mezi každou z $t!$ možných permutací a posloupností čísel $(c_1, c_2, \ldots, c_{t-1})$, kde $0 \le c_j \le j$. Takovouto množinu čísel spočteme náhodně docela snadno a podle zmíněné korespondence již odvodíme náhodnou permutaci.

Algoritmus P (*Míchání*). Nechť X_1, X_2, …, X_t je množina t čísel k míchání.

P1. [Inicializace.] Přiřaďte $j \leftarrow t$.

P2. [Generování U.] Vygenerujte náhodné číslo U s rovnoměrným rozdělením od 0 do 1.

P3. [Výměna.] Přiřaďte $k \leftarrow \lfloor jU \rfloor + 1$. (Nyní je k náhodné číslo od 1 do j. Ve cvičení 3.4.1–3 jsme si vysvětlili, že k *nesmíme* počítat ze zbytku modulo j.) Vyměňte $X_k \leftrightarrow X_j$.

P4. [Zmenšení j.] Zmenšete j o 1. Je-li $j > 1$, vraťte se na krok P2. ▮

Tento algoritmus publikovali jako první R. A. Fisher a F. Yates [*Statistical Tables* (London, 1938), příklad 12] v běžném jazyce, a poté R. Durstenfeld [*CACM* **7** (1964), 420] v počítačovém jazyce. Pokud bychom namísto míchání zadané posloupnosti $(X_1, \ldots, X_t)$ chtěli vygenerovat jen náhodnou permutaci čísel $\{1, \ldots, t\}$, můžeme vynechat operaci výměny $X_k \leftrightarrow X_j$, nechat j probíhat od 1 do t a přiřadit $X_j \leftarrow X_k$, $X_k \leftarrow j$; viz D. E. Knuth, *The Stanford GraphBase* (New York: ACM Press, 1994), 104.

R. Salfi [*COMPSTAT 1974* (Vídeň: 1974), 28–35] poukázal, že Algoritmus P nemůže vygenerovat více než m různých permutací, jestliže rovnoměrně rozdělené proměnné U získáme z lineární kongruentní posloupnosti s modulem m, nebo jestliže skutečně použijeme rekurenci $U_{n+1} = f(U_n)$, ve které může U_n nabývat

jen m možných hodnot, protože výslednou permutaci v takových případech plně určuje hodnota prvního vygenerovaného U. Jestliže tedy například $m = 2^{32}$, některé permutace 13 prvků se nikdy nevyskytnou, protože $13! \approx 1{,}45 \times 2^{32}$. Ve většině aplikací sice fakticky *nepožadujeme* vygenerovat všech 13! permutací, přesto je ale znepokojivé vědět, že vyřazené permutace jsou určeny poměrně jednoduchým matematickým pravidlem, jako je struktura svazu (viz část 3.3.4).

U posunutého Fibonacciho generátoru s dostatečně dlouhou periodou jako 3.2.2–(7) tento problém nenastává. Ani s takovýmito metodami nebudou mít ale všechny permutace rovnoměrné rozdělení, pokud nedokážeme zadat nejméně $t!$ různých počátečních hodnot pro inicializaci generátoru. Jinými slovy, nemůžeme získat $\lg t!$ skutečně náhodných bitů, pokud nezadáme $\lg t!$ skutečně náhodných bitů na vstupu. V části 3.5 si ukážeme, že ovšem nemusíme propadat beznaději.

Algoritmus P můžeme snadno upravit tak, že dává náhodnou permutaci z náhodné kombinace (viz cvičení 15). Výklad náhodných kombinatorických objektů jiného typu (například rozkladů) viz část 7.2 a viz publikace *Combinatorial Algorithms*, Nijenhuis a Wilf (New York: Academic Press, 1975).

CVIČENÍ

1. [*M12*] Vysvětlete vztah (1).

2. [*20*] Dokažte, že se Algoritmus S nikdy nepokouší ze vstupního souboru načíst více než N záznamů.

▸ **3.** [*22*] Algoritmus S vybírá $(t+1)$-ní položku s pravděpodobností $(n-m)/(N-t)$, nikoli n/N, přesto se v textu tvrdí, že je vzorek nestranný, takže každá položka by měla být vybrána se *stejnou* pravděpodobností. Jak mohou být obě tvrzení pravdivá zároveň?

4. [*M23*] Nechť $p(m,t)$ je pravděpodobnost, že v technice vzorkování vybereme mezi prvními t položkami právě m. Ukažte přímo z Algoritmu S, že

$$p(m,t) = \binom{t}{m}\binom{N-t}{n-m}\Big/\binom{N}{n}, \qquad \text{pro } 0 \le t \le N.$$

5. [*M24*] Jaká je průměrná hodnota t při ukončení Algoritmu S? (Jinými slovy, kolik z n záznamů v průměru projdeme, než vytvoříme celý vzorek?)

6. [*M24*] Jaká je standardní odchylka hodnoty vypočtené ve cvičení 5?

7. [*M25*] Dokažte, že libovolný *daný* výběr n záznamů z množiny N dostaneme v Algoritmu S s pravděpodobností $1/\binom{N}{n}$. Vzorek je tudíž zcela nestranný.

▸ **8.** [*M39*] (J. S. Vitter.) Algoritmus S počítá na každý zpracovávaný vstupní záznam jednu rovnoměrnou veličinu. Úkolem tohoto cvičení je zvážit efektivně postup, kdy potřebný počet X vstupních záznamů, které je potřeba přeskočit před prvním výběrem, vypočteme rychleji.

a) S jakou pravděpodobností při daném k bude $X \ge k$?

b) Ukažte, že díky výsledku části (a) můžeme vypočítat X jen při vygenerování jediné rovnoměrné veličiny U a poté pomocí $O(X)$ dalších výpočtů.

c) Ukažte, že můžeme také přiřadit $X \leftarrow \min(Y_N, Y_{N-1}, \ldots, Y_{N-n+1})$, kde Y jsou nezávislé a každé Y_t je náhodné celé číslo v intervalu $0 \le Y_t < t$.

d) Ukažte, že pro maximální urychlení algoritmu je možné X vypočítat také průměrně v $O(1)$ krocích, a to pomocí metody ohraničení jako ve vztahu 3.4.1–(18).

9. [*12*] Nechť $n = 3$. Jestliže Algoritmus R aplikujeme na soubor s 20 záznamy očíslovanými od 1 do 20 a jestliže jako náhodná čísla postupně v kroku R3 vygenerujeme

$$4, 1, 6, 7, 5, 3, 5, 11, 11, 3, 7, 9, 3, 11, 4, 5, 4,$$

které záznamy se zapíší do rezervoáru? A které budou v konečném vzorku?

10. [*15*] Upravte Algoritmus R tak, že se obejde bez rezervoáru, za předpokladu, že se n záznamů aktuálního vzorku vejde do paměti.

▸ **11.** [*M25*] Nechť p_m je pravděpodobnost, že se do rezervoáru při prvním průchodu Algoritmu R dostane právě m prvků. Určete generující funkci $G(z) = \sum_m p_m z^m$ a najděte střední hodnotu a standardní odchylku. (Využijte myšlenky části 1.2.10.)

12. [*M26*] Podstatou Algoritmu P je, že libovolnou permutaci π je možné jednoznačně zapsat jako součin transpozicí ve tvaru $\pi = (a_t t)\dots(a_3 3)(a_2 2)$, kde $1 \le a_j \le$ $\le j$ pro $t \ge j > 1$. Dokažte, že existuje také jednoznačná reprezentace tvaru $\pi =$ $= (b_2 2)(b_3 3)\dots(b_t t)$, kde $1 \le b_j \le j$ pro $1 < j \le t$, a navrhněte algoritmus, který vypočte čísla b z čísel a v $O(t)$ krocích.

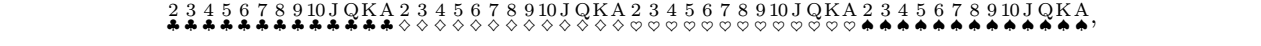
13. [*M23*] (S. W. Golomb.) Jedním z nejběžnějších způsobů míchání balíčku karet je rozdělit balíček do dvou zhruba stejných částí, „projet“ je palci a tím je jakoby „zasunout“ do sebe. (Podle výkladu hráčské etikety v Hoyleových pravidlech se „toto zasunutí má provést asi třikrát, aby byly karty důkladně promíchány“.) Uvažujme nyní balíček $2n-1$ karet $X_1, X_2, \dots, X_{2n-1}$; při „dokonalém míchání“ označeném s se rozdělí tento balíček na $X_1, X_2, \dots, X_n$ a $X_{n+1}, \dots, X_{2n-1}$, a poté se dokonale střídavě proloží, takže dostaneme $X_1, X_{n+1}, X_2, X_{n+2}, \dots, X_{2n-1}, X_n$. Při sejmutí neboli „říznutí“ c^j se pak změní $X_1, X_2, \dots, X_{2n-1}$ na $X_{j+1}, \dots, X_{2n-1}, X_1, \dots, X_j$. Ukažte, že při spojení dokonalého míchání zasunutím a dokonalého sejmutí je možných nejvýše $(2n-1)(2n-2)$ různých uspořádání balíčku, je-li $n > 1$.

14. [*22*] Permutace sejmutí a zasunutí změní balíček karet $a_0 a_1 \dots a_{n-1}$ na posloupnost, která obsahuje vybrané posloupnosti

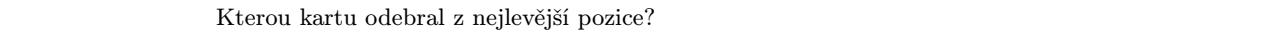
$$a_x\, a_{(x+1) \bmod n} \dots a_{(y-1) \bmod n} \qquad \text{a} \qquad a_y\, a_{(y+1) \bmod n} \dots a_{(x-1) \bmod n}$$

které jsou nějakým způsobem proložené, pro nějaké x a y. Permutace 3890145267 vznikla tudíž sejmutím a zasunutím z permutace 0123456789, pro $x = 3$ a $y = 8$.

a) Začínáme se sadou 52 hracích karet uspořádaných ve standardním pořadí



Mr. J. H. Quick (student) provedl náhodné sejmutí a zasunutí; potom odebral nejlevější kartu a vložil ji na náhodné místo, čímž dostal posloupnost



Kterou kartu odebral z nejlevější pozice?

b) Quick opět začal s balíčkem v původním pořadí; nejprve ale provedl *tři* operace sejmutí a zasunutí, a teprve potom přesunul nejlevější kartu na nové místo:

Kterou kartu přesouval tentokrát?

▸ **15.** [*30*] (Ole-Johan Dahl.) Je-li na začátku Algoritmu P $X_k = k$ pro $1 \le k \le t$ a jestliže algoritmus ukončíme, jakmile j dosáhne hodnoty $t-n$, je posloupnost X_{t-n+1}, $\ldots$, X_t náhodná permutace náhodné kombinace n prvků. Ukažte, jak simulovat efekt této procedury jen s pomocí $O(n)$ paměťových buněk.

▸ **16.** [*M25*] Navrhněte způsob, jak vypočítat náhodný vzorek n záznamů z N, pro dané N a n, a to na základě principu hašování (viz část 6.4). Metoda by měla využívat $O(n)$ paměťových pozic a průměrně $O(n)$ jednotek času, a měla by vracet vzorek v podobě seřazené (uspořádané) množiny celých čísel $1 \le X_1 < X_2 < \cdots < X_n \le N$.

17. [*M22*] (R. W. Floyd.) Dokažte, že následující algoritmus generuje náhodný vzorek S o n celých číslech z množiny $\{1, \ldots, N\}$: Přiřaďte $S \leftarrow \emptyset$; potom pro $j \leftarrow N-n+1$, $N-n+2$, $\ldots$, N (v tomto pořadí) přiřaďte $k \leftarrow \lfloor jU \rfloor + 1$ a

$$S \leftarrow \begin{cases} S \cup \{k\}, & \text{je-li } k \notin S; \\ S \cup \{j\}, & \text{je-li } k \in S. \end{cases}$$

▸ **18.** [*M32*] Lidé se někdy pokoušejí zamíchat n položek $(X_1, X_2, \ldots, X_n)$ postupnými výměnami

$$X_1 \leftrightarrow X_{k_1},\ X_2 \leftrightarrow X_{k_2},\ \ldots,\ X_n \leftrightarrow X_{k_n},$$

kde indexy k_j jsou nezávislé a s rovnoměrným rozdělením od 1 do n.

Uvažujte orientovaný graf s vrcholy $\{1, 2, \ldots, n\}$ a s hranami, které vedou z j do k_j pro $1 \le j \le n$. Popište orientované grafy tohoto typu, pro které, začneme-li s prvky $(X_1, X_2, \ldots, X_n) = (1, 2, \ldots, n)$, dostaneme popsanými výměnami příslušné permutace (a) $(n, 1, \ldots, 2)$; (b) $(1, 2, \ldots, n)$; (c) $(2, \ldots, n, 1)$. Na závěr zhodnoťte, jestli jsme tyto tři permutace získali s příliš rozdílnými pravděpodobnostmi.

Tak jako z kousku příze rozumíme celému klubku,
spokojíme se a ujistíme i s tímto vzorkem.

— Miguel de Cervantes, Důmyslný rytíř Don Quijote de la Mancha (1605)