Obliczenia inspirowane Naturą

Wykład 14 - Losowość

Jarosław Miszczak

IITiS PAN Gliwice

09/06/2016

Na poprzednim wykładzie

1. . . .
2. . . .
3. . . .

1. Liczby pseudolosowe
   - Oszacowanie π
   - Generatory liczb pseudolosowych
   - Blum Blum Shub
   - Mersenne Twister
   - Ekstraktory losowości
   - Dowolne rozkłady
2. Czynnik ludzki
3. Liczby losowe
   - Architektura Ivy Bridge
   - Serwis Random.org
   - Kwantowe generatory liczb losowych

Zastosowania

Gdzie stosujemy liczby (pseudo)losowe

- w algorytmach metaheurystycznych
- w metodzie Monte Carlo (np. do całkowania numerycznego)
- w symulacjach komputerowych
- w kryptografii
- w grach hazardowych

Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin. – John von Neumann

Liczby pseudolosowe

Monte Carlo

Metody Monte Carlo to wspólna nazwa grupy technik komputerowych bazujących na obserwacji, iż wyniki dotyczące układu nie wymagają znajomości wszystkich możliwych stanów, a jedynie zbadania zachowania w stanach (konfiguracjach) *typowych*.

Początki metod Monte Carlo sięgają prac nad projektem Manhatan, a ich pomysłodawcą byli Stanisław Ulam oraz John von Neuman.

Liczby pseudolosowe
Oszacowanie π

Najstarszy przykład zastosowania metody Monte Carlo to estymacja wartości π.

Igła Buffona (1777)

Dana jest (nieskończona) kartka papieru liniami odległymi od siebie o t. Mamy do dyspozycji igłę od długości l. Ile wynosi prawdopodobieństwo zdarzenia, że rzucona igła dotknie linii?

Liczby pseudolosowe
Oszacowanie π

Co zakładamy?

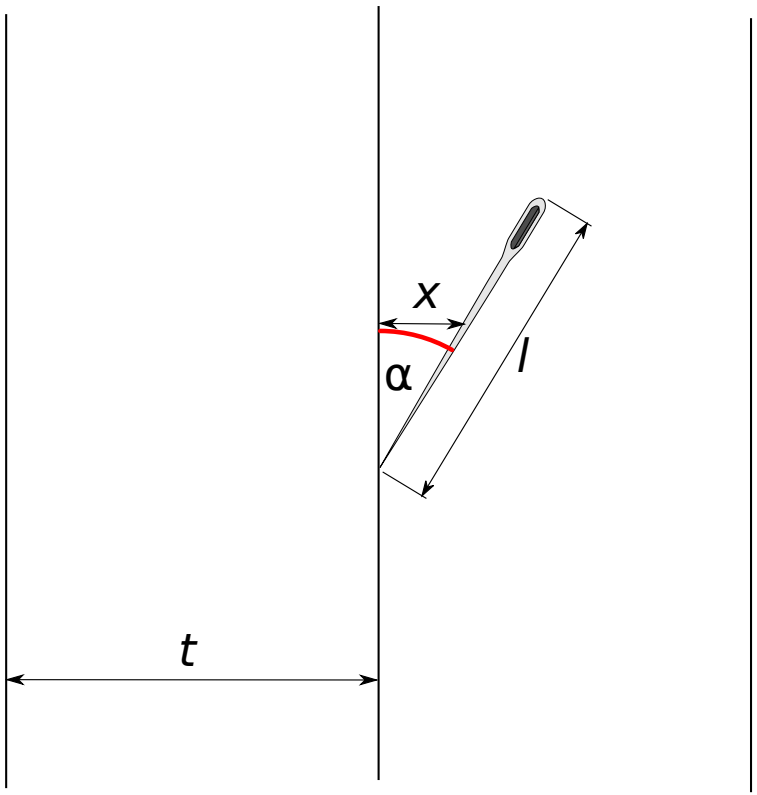
- Odległość środka igły od linii jest *losowana* z rozkładem równomiernym na $[0, \frac{t}{2}]$.
- Kąt między linią a igła jest *losowany* z rozkładem równomiernym na $[0, \frac{\pi}{2}]$.

Liczby pseudolosowe
Oszacowanie π

Co obserwujemy?

- Częstość zdarzenia „igła dotknęła linii” R w n eksperymentach, czyli $\frac{R}{n}$.

Liczby pseudolosowe
Oszacowanie π



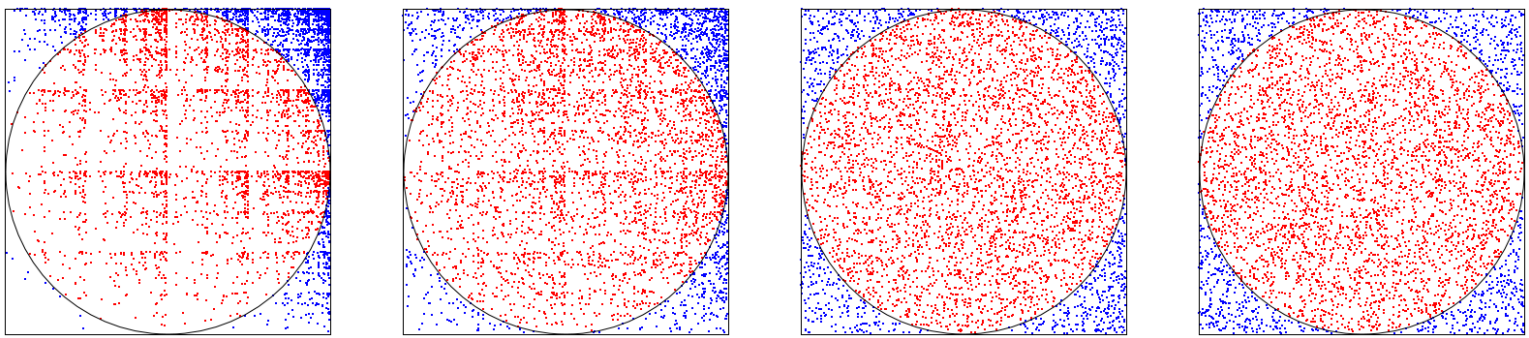
Liczby pseudolosowe
Oszacowanie π

Ile wynosi prawdopodobieństwo zdarzenia „igła dotknęła linii” $\frac{R}{n}$?

- Zdarzenie zachodzi gdy $x \leqslant \frac{l}{2} \sin \alpha$.
- Zatem

$$\frac{R}{n} \approx \int_0^{\pi/2} \int_0^{l/2 \sin \theta} \mathrm{d}x\mathrm{d}\theta.$$

Liczby pseudolosowe
Oszacowanie π



(a) $\pi \approx 2.4168$ (b) $\pi \approx 2.9792$ (c) $\pi \approx 3.1120$ (d) $\pi \approx 3.1416$

Rysunek: Reprezentacja próbek wykorzystanych do obliczenia wartości π dla próbek z prawdopodobieństwem wylosowania 1 zwiększonym o 0.2 (a), 0.1 (b), 0.025 (c) oraz dla równego prawdopodobieństwa wylosowani 0 i 1 (d). Każda próbka zawiera 5000 punktów.

Liczby pseudolosowe
Oszacowanie π

- Żeby wyniki numeryczne były poprawne konieczne jest dostarczenie do algorytmu liczb (pseudo)losowych które są równomiernie rozłożone na badanym zbiorze.
- Małe zaburzenie rozkładu liczb, może mieć duże znacznie dla dokładności uzyskiwanych wyników.

Liczby pseudolosowe
Oszacowanie π

Doby generator powinien

- generować liczby, które mają rozkład jak najbliższy do jednostajnego;
- produkować ciągi w których podciągi są wzajemnie niezależne;
- mieć długi *okres* – czyli najmniejszą liczbę p taką, że $x_n = x_{n+p}$;
- być przenośny, efektywny i powtarzalny.

Liczby pseudolosowe

Determinizm

Generatory liczb pseudolosowych są deterministyczne.

- Wada: ...
- Zaleta: możliwe jest powtórzenie wyników eksperymentów (lub przetestowania kodu) dla tego samego ciągu liczb losowych.

Liczby pseudolosowe

- Niemal każdy język programowania dostarcza mechanizmu generowania liczby „losowych”.
- Generatory te mają za zadanie dostarczanie liczb z rozkładem jednorodnym.
- Mając do dyspozycji ciąg liczb losowych z przedziału $[0, 1)$ można go stosunkowo łatwo przekształcić w ciąg o dowolnym rozkładzie.

Liczby pseudolosowe

Najprostszy generator liczb pseudolosowych to generator udostępniany w bibliotece standardowej języka C przez funkcję rand. Jest to tak zwany Liniowy Generator Kongruentny ang. *linear congruential generator (LCG)*, który działa zgodnie z regułą

$$x_n = ax_{n-1} + c \pmod{m},$$

gdzie a (mnożnik) c (przyrost) i m (moduł) to odpowiednio dobrane parametry.
Jeżeli $c = 0$, to generator nazywamy multiplikatywnym.

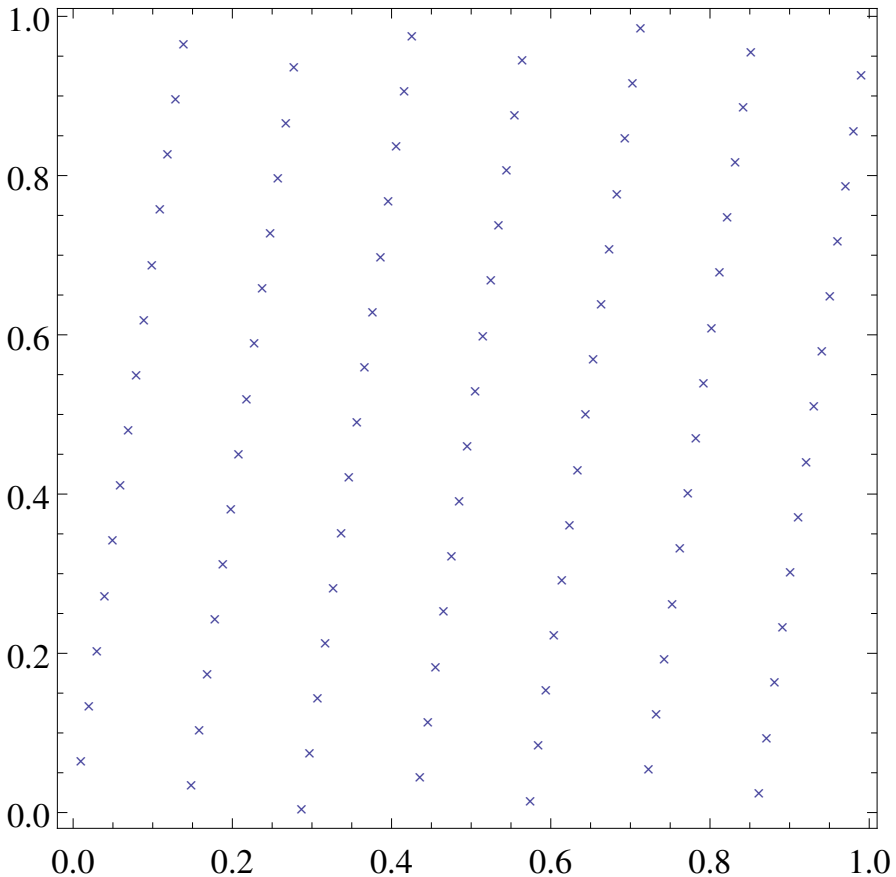
Liczby pseudolosowe

- Wartość początkowa ciągu x_0 to *ziarno* (ang. seed)
- Zły dobór ziarna może zepsuć generator.
- Generator można zainicjować
  - aktualnym czasem
  - innym generatorem

Liczby pseudolosowe

- Generatory kongruentne są bardzo wrażliwe na dobór parametrów.
- Ich zaletą jest wydajna implementacja.
- Generator powinien osiągać maksymalny okres.
- Okres postaci $2^{31} - 1$ jest popularny ze względu na łatwość implementacji, ale jest zbyt mały na współczesne potrzeby.

Liczby pseudolosowe



Znormalizowane losowe punkty $\frac{x_n}{m}$ uzyskane z generatora multiplikatywnego $x_{n+1} = 7x_n \pmod{101}$ dla wartości początkowych $x_0 = 10$ i $x_0 = 13$.

Liczby pseudolosowe

RANDU

Jednym z najgorszych, powszechnie stosowanych, generatorów liczb pseudolosowych jest RANDU opisany formułą

$$x_{n+1} = 65539 \cdot x_n \pmod{2^{31}}$$

Generator ten był powszechnie stosowany w latach '60 i '70.

Liczby pseudolosowe
Blum Blum Shub

- Lenore Blum, Manuel Blum i Michael Shub, 1986[1]
- Generator postaci

$$x_{n+1} = (x_n)^2 \pmod{M},$$

  gdzie $M = pq$ jest iloczynem dwóch dużych liczb pierwszych.
- Ziarnem x_0 powinna być liczba względnie pierwsza z M.
- Wyjściem jest tylko część bitów x_n.

[1] L. Blum, M. Blum, and M. Shub. A Simple Unpredictable Pseudo-Random Number Generator. SIAM Journal on Computing, vol. 15, p. 364-383, May 1986

Liczby pseudolosowe
Blum Blum Shub

- Dość wolny.
- Bardzo bezpieczny.
- Odróżnienie jego wyników od szumu jest równie trudne jak faktoryzacja M.

Liczby pseudolosowe
Mersenne Twister

- Opracowany W 1997 roku przez Makoto Matsumoto i Takuji Nishimura[2].
- Zaprojektowany z myślą o metodach Monte Carlo i innych symulacjach statystycznych.

[2] M. Matsumoto, T. Nishimura, *Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator*, ACM Trans. Model. Comput. Simul. 8, 3 (1998).

Liczby pseudolosowe
Mersenne Twister

- Na okres wybierana jest liczba pierwsza Mersenne'a – np. $2^{19937} - 1$.
- Generator osiąga pełny okres i daje bardzo dobre źródło losowości.
- $2^{19937} - 1 \approx 4.3 \times 10^{6001}$ czyli więcej niż liczba cząstek w obserwowanym Wszechświecie (czyi 10^{87}).

Liczby pseudolosowe
Mersenne Twister

Liczby Mersenne'a – liczby pierwsze postaci $2^p - 1$, gdzie p jest liczba pierwszą.

- Nie wiadomo czy jest ich nieskończenie wiele.
- Dotychczas znamy 49 takich liczb.
- Największa z nich to $2^{74207281} - 1$ odkryta w styczniu 2016.

GIMPS

Projekt który ma na celu znajdowanie nowych liczb Mersenne'a: http://www.mersenne.org/

Liczby pseudolosowe
Mersenne Twister

Mersenne Twister to obecnie domyślny generator stosowany w wielu popularnych językach programowania, m.in. PHP, Python, R oraz Ruby, a także wielu systemach algebry liniowej, m.in. Maple, Matlab, GNU Octave oraz Scilab.

Generator też został wprowadzony do standardu C++11 i dostępny jest w pliku nagłówkowym $<\mathrm{random}>$.

Liczby pseudolosowe
Mersenne Twister

W systemie *Mathematica* domyślnie zastosowana metoda generowania liczb pseudolosowych oparta jest na automatach komórkowych. Wykorzystuje on chaotyczne zachowanie ciągów bitów generowanych przez automaty komórkowe.
Prostym przykładem zastosowania automatów komórkowych do generowania ciągów pseudolosowych jest automat Rule30.

Liczby pseudolosowe

Ekstraktor losowości (ang. randomness extractor), to funkcja, która zastosowana do wyjścia ze źródła losowości o małej entropii generuje losowe wyjście niezależne od wejścia.

Liczby pseudolosowe

Ekstraktor von Neumanna

Dla ciągu bitów $x_1, x_2, \ldots, x_{n-1}, x_n$ ekstraktor von Neumanna dla nienakładających się par bitów daje

$$\begin{array}{ll} \epsilon & \text{jeżeli } x_i = x_{i+1} \\ x_i & \text{jeżeli } x_i \neq x_{i+1} \end{array}$$

gdzie ϵ to napis pusty.

Liczby pseudolosowe

- Ekstraktor losowości redukuje ilość bitów na wyjściu.

Liczby pseudolosowe
Dowolne rozkłady

Metoda odwracania dystrybuanty:

- Zakładamy, że mamy dostęp do źródła $U(0,1)$.
- Niech F będzie dystrybuantą interesującego nas rozkładu.
- Definiujemy $X = F^{-1}(U)$.
- Zmienna losowa ma rozkład o dystrybuancie F.

Liczby pseudolosowe
Dowolne rozkłady

Metoda odwracania dystrybuanty

RandomRealNormal[d] :=
Mean[d] + Sqrt[2]Sqrt[Variance[d]] InverseErf[-1+2
RandomReal[0,1]];

gdzie

- RandomReal[0,1] daje liczbę z $U(0,1)$.
- Erf[z] $= \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} \mathrm{d}t$

Liczby pseudolosowe
Dowolne rozkłady

- Działa zawsze.
- Jest mało wydajne.
- Wymaga znajomości analitycznej postaci funkcji odwrotnej do dystrybuanty.

Liczby pseudolosowe
Dowolne rozkłady

Metoda eliminacji (John von Neumann):
Załóżmy, że gęstość prawdopodobieństwa f rozkładu jest nieujemna na (a, b) i ograniczona z góry przez d.

- Wygeneruj U_1 z $U(a, b)$ oraz U_2 z $U(0, d)$.
- Jeżeli $U_2 \leqslant f(U_1)$, to przyjmujemy $X = U_1$
- W przeciwnym wypadku powtarzamy.

Metoda to odpowiada całkowaniu gęstości rozkładu poprzez próbkowanie.

Czynnik ludzki

Ciekawym przykładem złego mechanizmu generowania liczb losowych jest wybieranie ich przez ludzi. Okazuje się, iż ludzie poproszeni o wypisanie losowego ciągu liczb (np. zer i jedynek) mają tendencję do unikania powtórzeń liczb.

Zjawisko to jako pierwszy opisał w 1953 roku amerykański psycholog Alphonse Chapanis.

Czynnik ludzki

Prostym sposobem na wykrycie nie-losowego ciągu cyfr jest zastosowanie prawa Benforda.

Prawo Benforda zaobserwował po raz pierwszy w 1881 roku amerykański astronom Simon Newcomb. Zauważył on, iż tablice logarytmów są najbardziej zabrudzone na początku.

Fizyk Frank Benford dokonał podobnej obserwacji w 1938 roku i poparł ją wieloma wynikami pomiarów (m.in. ciężarów atomowych czy powierzchni dorzecza rzek).

Czynnik ludzki

. . . [generowanie liczb nie-losowych]

Czynnik ludzki

Prawo Benforda mówi, iż w losowym zbiorze danych prawdopodobieństwo wystąpienia na pierwszym miejscu cyfry $d = 1, 2, \ldots, 9$ wynosi

$$P(d) = \log\left(1 + \frac{1}{d}\right). \tag{1}$$

Wynika stąd, iż prawdopodobieństwo tego, że na pierwszym miejscu jest jedynka wynosi ok. 30%.

Liczby losowe

Liczby losowe możemy podzielić na takie które:

- wyglądają na losowe,
- są losowe.

Liczby losowe
Architektura Ivy Bridge

Intel® Math Kernel Library (Intel® MKL) – zestaw bibliotek do obliczeń numerycznych udostępniana przez firmę Intel i zoptymalizowana dla procesorów tej firmy.

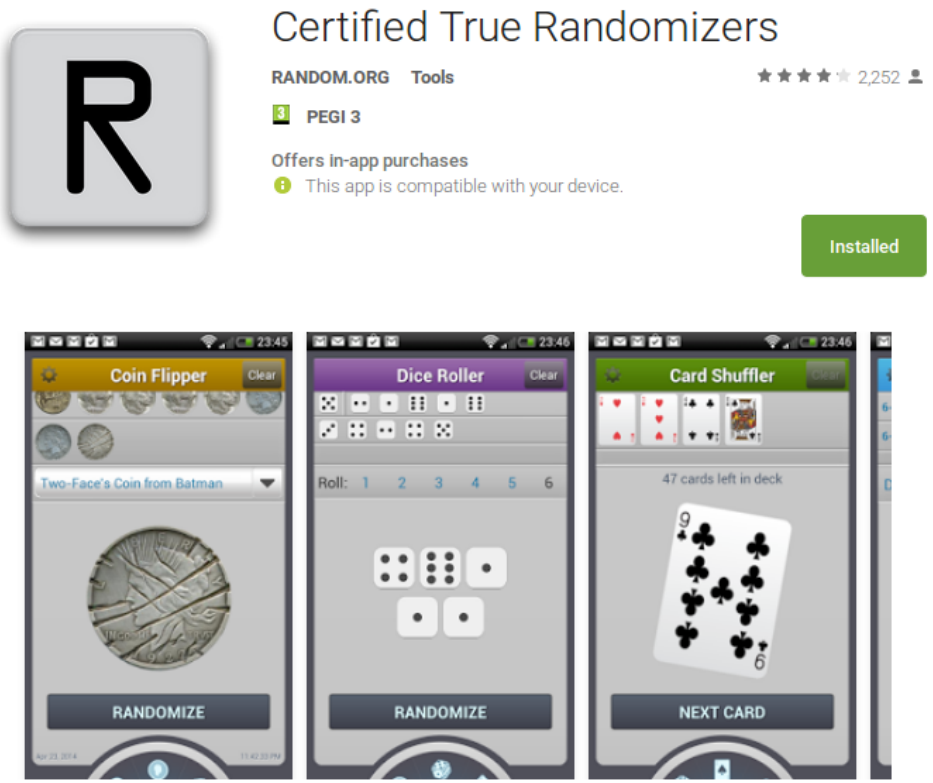
Ivy Bridge

- W roku 2011 firma Intel wprowadziła na rynek mikroarchitekturę procesorów o nazwie kodowej Ivy Bridge w technologii 22 nm.
- Jedną z nowości tej architektury jest pojawienie się instrukcji rdrand, która umożliwia korzystanie ze sprzętowego generatora liczb losowych.

Liczby losowe
Serwis Random.org

- Źródło liczb losowych bazujące na szumie atmosferycznym.
- Ponieważ źródło jest bardzo skomplikowane, nie jest możliwe udowodnienie, że liczby są losowe.
- To że uzyskane liczby *wydają się* losowe wynika z faktu, że nie jesteśmy w stanie odtworzyć pełnej ewolucji układu.

Liczby losowe
Serwis Random.org



https://play.google.com/store/apps/details?id=org.random.randomapp

Liczby losowe
Kwantowe generatory liczb losowych

Opis probabilistyczny

Mechanika kwantowa mówi, że nie jest możliwe podanie deterministycznego opisu układu – możemy jedynie przewidywać z jakim prawdopodobieństwem wystąpi dany wynik.

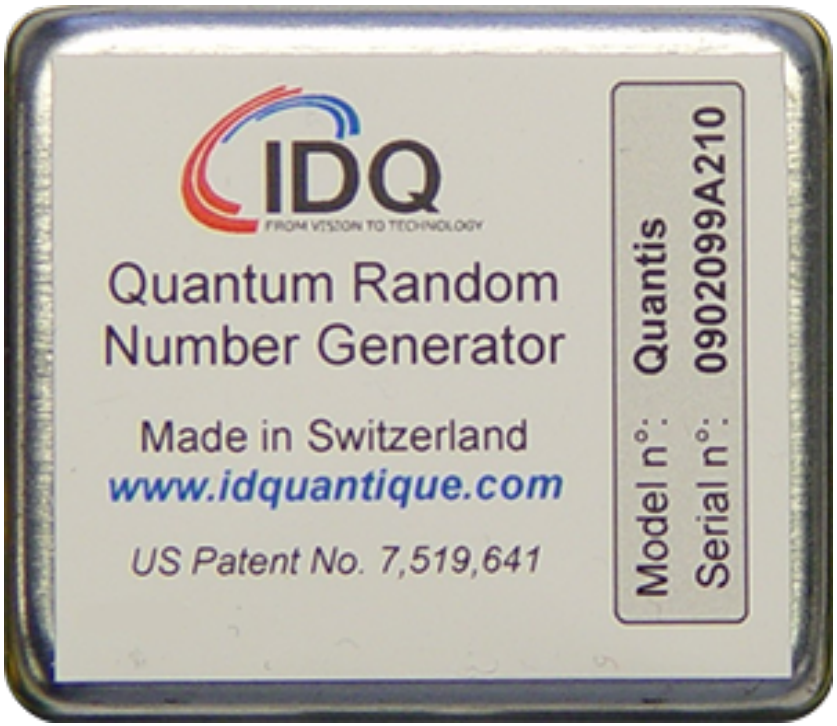
Liczby losowe
Kwantowe generatory liczb losowych

- Stan kwantowy $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ opisuje układ w superpozycji stanów bazowych.
- Stany bazowe są określone przez pomiary które możemy wykonać na układzie.
- Wykonanie pomiaru powoduje, że otrzymamy $|0\rangle$ lub $|1\rangle$.

Liczby losowe
Kwantowe generatory liczb losowych

- W mechanice klasycznej stan układu może być przewidziany z dowolną dokładnością – o ile znamy dokładnie prawa rządzące układem i warunki początkowe – mechanika klasyczna jest *deteministyczna*.
- Z powodu dużej liczby cząstek (np. w filiżance kawy) do opisu wielu układów fizycznych stosujemy podejście statystyczne.
- W mechanice kwantowej **nawet w przypadku jednej cząstki** nie jest zawsze możliwe deterministyczne podanie wyniku pomiaru.

Liczby losowe
Kwantowe generatory liczb losowych



Quantis firmy idQuantique

Liczby losowe
Kwantowe generatory liczb losowych

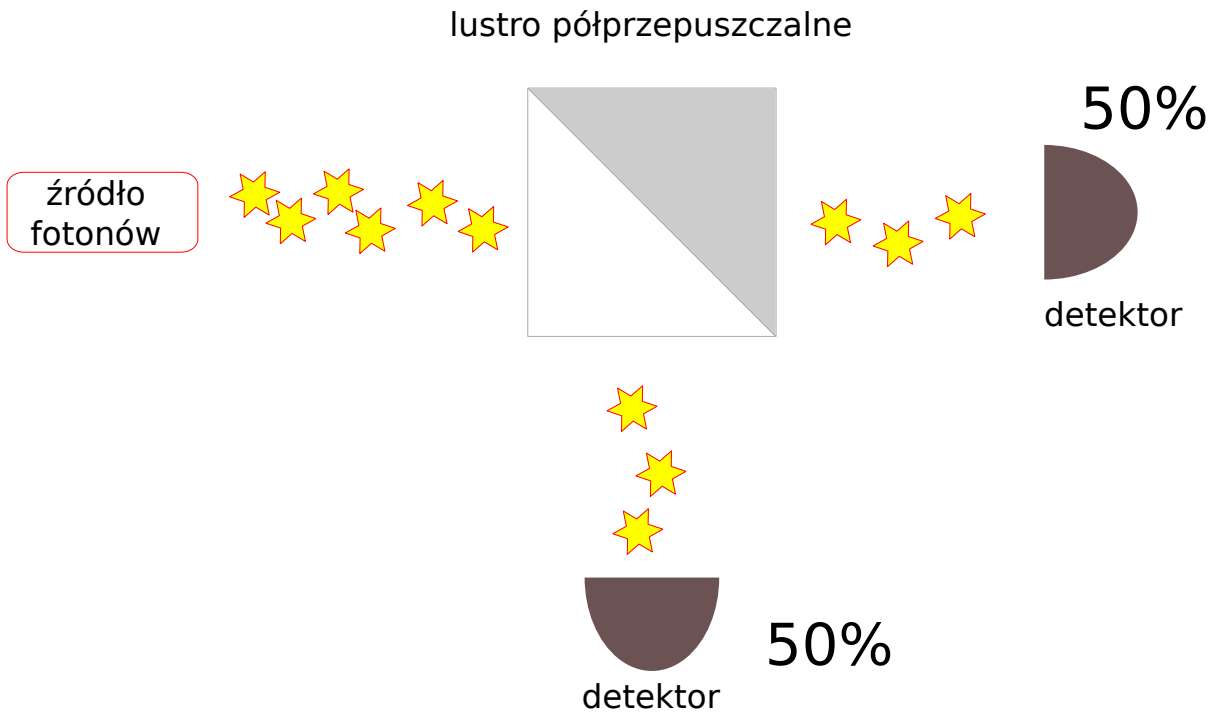
Liczby losowe
Kwantowe generatory liczb losowych

- Zastosowania kryptograficzne.
- Zbyt wolny do celów symulacji komputerowych – do 4Mbits/sec.
- Tani: 1500 EUR – i można go kupić.
- Dostępne jest oprogramowanie do komunikacji z urządzeniem.

Liczby losowe
Kwantowe generatory liczb losowych

- Quantis wykorzystuje do generowania liczb losowych mechanizm lustra półprzepuszczalnego.
- Do jego działania konieczny jest pomiar pojedynczych fotonów.

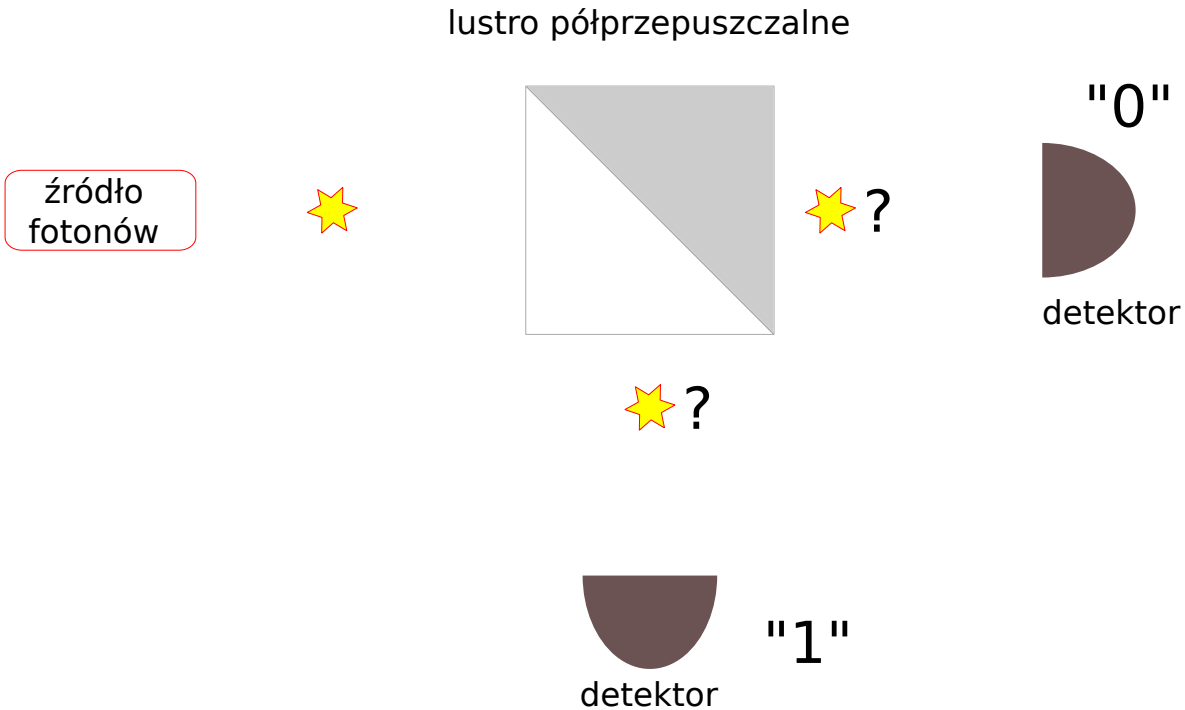
Liczby losowe
Kwantowe generatory liczb losowych



Liczby losowe
Kwantowe generatory liczb losowych

- Lustro półprzepuszczalne przepuszcza 50% fotonów, a resztę odbija.
- Jak zadziała w przypadku jednego fotonu?

Liczby losowe
Kwantowe generatory liczb losowych



Liczby losowe
Kwantowe generatory liczb losowych

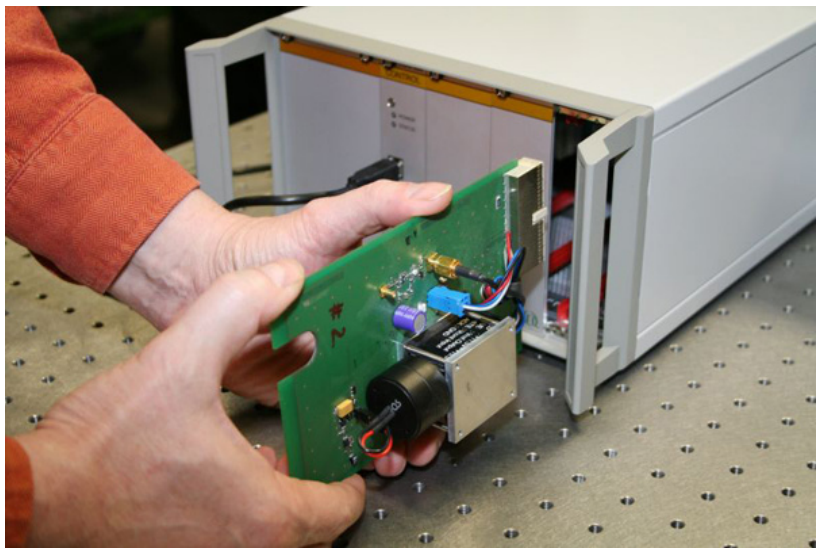
Odpowiada to działaniu bramki Hadamarda

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

Liczby losowe
Kwantowe generatory liczb losowych

- Wadą generatora Quantis jest jego prędkość.
- Istnieją rozwiązanie szybsze i kilka z nich jest dostępnych jako usługi sieciowe.
- Wykorzystują one bardziej subtelne metody i droższy sprzęt.

Liczby losowe
Kwantowe generatory liczb losowych

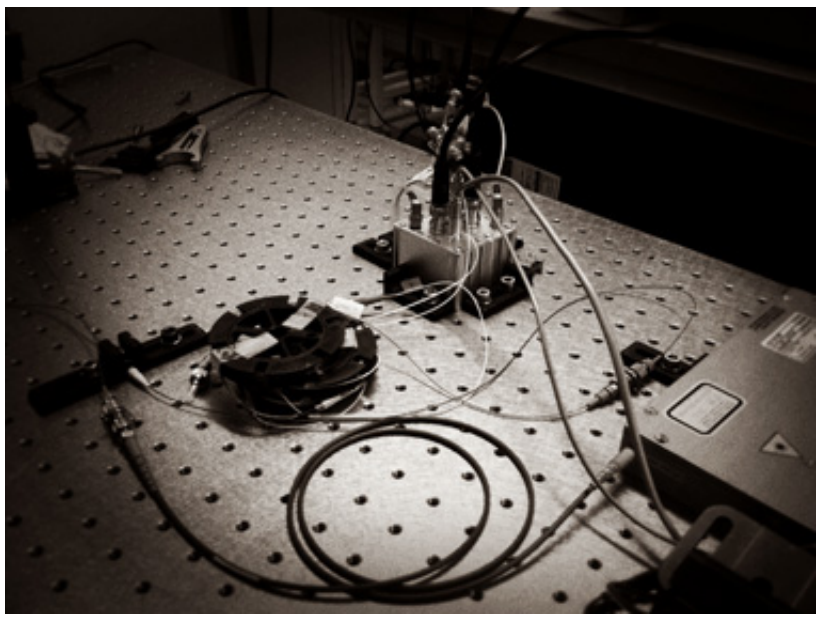


- High Bit Rate Quantum Random Number Generator – PicoQuant GmbH i Wydział Fizyki Uniwersytetu Humboldta
- Metoda: pomiar czasu przyjścia fotonu.
- https://qrng.physik.hu-berlin.de/

Liczby losowe
Kwantowe generatory liczb losowych

- QRBG – Centre for Informatics and Computing, Ruder Bošković Institute, Zagreb, Chorwacja
- http://random.irb.hr/

Liczby losowe
Kwantowe generatory liczb losowych



- ANU Quantum Random Numbers Server – Australian National University
- Metoda: pomiary fluktuacji próżni
- https://qrng.anu.edu.au/

In mathematics you don't understand things. You just get used to them. – John von Neumann