Field: *Mathematics/Applied Mathematics/Computer Science*

Planning Group Members:

Hiroki Arimura, Hokkaido University
Jörg Roth, University of Duesseldorf

Session Topic:

Randomness in Computations

Randomness is ubiquitous in our world. It plays an important role in science and technology. Originally, the notion of randomness was introduced in order to capture uncertainty and nonregularity in stochastic processes such as gambling or Brownean motion of particles. In this session, we explore various aspects of randomness in computer science and related areas.

Is randomness useful? In engineering, randomness is often considered harmful, since it introduces uncertainty and makes measurements or products inaccurate and imprecise. In computer science, however, randomness helps to cope with computationally hard problems. Randomized algorithms are computer programs that can access a sequence of random bits. They can thus make errors, but they often are simpler and more efficient than the best known deterministic algorithms solving the same problem. Using known probability amplification techniques, the error can be made small enough so as to be negligible. Moreover, it is often the case that a randomized algorithm is found first and can then be transformed into a deterministic algorithm for the same problem by so-called derandomization techniques.

For example, efficient randomized primality tests (i.e., algorithms that test whether a given number is a prime number, which is an important task in many applications) were developed due to the lack of efficient deterministic algorithms for testing primality. Solving a long-standing open problem, Agrawal, Kayal, and Saxena recently found the first polynomial-time deterministic primality test. Their celebrated result created much sensation, not only in computer science and mathematics but also in the daily press such as in the *New York Times*. One can argue that their achievement was possible only because of the previously found randomized primality tests.

Randomization techniques are also useful in applied areas of computer science such as knowledge discovery in databases, machine learning, and financial engineering.

Is randomness really necessary? Interestingly, the answer is both yes and no. On the one hand, randomness is inherently necessary to capture the strongest cryptographic security notions. Cryptography is the art and science of designing secure cryptosystems, which can be used for example to encrypt messages such that unauthorized decryption by an eavesdropper is prevented.

On the other hand, generating truly random sequences is expensive. However, using a (*pseudo-)random number generator*, one can deterministically generate a sequence of numbers that looks just like a truly random sequence in terms of specified statistical tests. For example, the Linear Congruential Generator (LCG) is one of the most popular random number generators. However, some real-world applications require large integral computation in high-dimensional spaces, and the LCG method is not good enough for some applications such as Monte Carlo simulations in financial engineering. In 1998, Makoto Matsumoto invented a random number generator called Mersenne Twister (MT) algorithm, which generates (uniform pseudo) random numbers with the astronomically large period of 2^{19937}-1 and achieves a very high uniformity. The MT algorithm is used in many real-world applications.

Mathematics/Applied Mathematics/Computer Science
Planning Group Members: Hiroki Arimura and Jörg Rothe

Randomness in Computations

Speaker:
Makoto Matsumoto, Hiroshima University

Generating Randomness by deterministic computations: impossibility, compromise, and assurance

1. Introduction: what for?

A random number generator (RNG) is a soft/hardware to generate a sequence of numbers that *appears* to be random. Random numbers are necessary to simulate probabilistic events in computer simulation. Such a simulation is called *Monte Carlo methods.*

There are other important applications: e.g. cryptography and randomized algorithms, which we omit here.

2. How to generate

Requirements to RNG are (1) randomness (2) generation speed, and (3) reproducibility. Physical generators pick up noises from electric devices, and transform them into random numbers. Their shortcomings are cost, speed and irreproducibility. In simulations in nuclear physics, 10^{12} of random numbers are often consumed. To reproduce one simulation (e.g. for optimization, for examinations by other research groups), one needs to record all the generated numbers.

An effective reproducible approach is called a Pseudorandom Number Generator (PRNG).

PRNG generates a sequence of numbers by a recursion formula, which is used as a random

number sequence.

The essential problems of PRNG are (1) how to choose the recursion and its parameter, and (2) how to qualify it to be feasible for the simulation. There is no commonly accepted answers: the answer depends on each researcher of PRNG. This is because: *there is no satisfactory definition of randomness feasible for PRNG.*

3. Gap between definition and practice

It is paradoxical to define randomness of one fixed sequence of finite length. For example, the probabilities to obtain 3141592653 or 1234567890 as a random sample of 10 digits numbers are both 10^{-10}. Which is more random?

One of the reasonable definitions is due to Kolmogorov: “A number sequence of finite length is random if there is no computer program shorter than the sequence itself that generates the sequence.” In other words, if a sequence admits no shorter description than recording the sequence itself, then it is random. This is brilliant, but no computer can generate such a sequence.

A more practical definition is called “computationally secure PRNG.” If one gives an initial seed to the PRNG, then the PRNG effectively generates a sequence. On the other hand, without knowing the initial seed, one can not make any guess about the next number in a feasible time, from the sequence generated so far. Such a PRNG is satisfactory, since no computer program can reveal its non-randomness. Such a PRNG is proved to exist, under a big hypothesis that “factorization of big numbers does not have an effective algorithm.”

4. Theoretical evaluation and Mersenne Twister

A more classical and practical approach is to focus on some mathematical structures of a PRNG sequence, to score the structures and to select preferable recursion and parameters by the scores. Mersenne Twister PRNG (MT) proposed by the speaker and Nishimura [MT] has excellent scores, and now widely believed to be one of the best generators.

A typical theoretical score is the period of PRNG. Some of commonly used PRNGs have period 2^{31}, which is too short for modern computers (used up in a few minutes). Mersenne Twister has period $2^{19937}-1$.

Another important score is the dimension of equidistribution. A PRNG is said to be *k-dimensionally equidistributed* if the generated numbers are uniformly distributed in a *k*-dimensional cube through a whole period, that is, if every consecutive *k* numbers have no relation. For example, a sequence of bits 0001011100010111... with period 8 is 3-dimensionally equidistributed, since the overlapped 3-tuples 000, 001, 010, 101, 011, 111, 110, 100 constitute all the possible 3-bit patterns in a period.

The dimension of equidistribution of MT is 623, far higher than 20 of some widely used GFSR generator. Most significant bits of MT has even higher dimension of equidistribution, for example, at 3-bit precision it is 6240.

To realize such good scores, MT utilizes an unusual number system introduced by Galois in the 19th century based on 1 + 1 = 0. In the usual binary number system, we have carry and 110 × 11 = 110 + 1100 = 10010. MT is based on the number system without carry, that is 110 × 11 = 110 + 1100 = 1010, called “the polynomial ring over the two element field” in mathematics. MT uses geometry of this number system to compute the dimension and the period. Computations in this number system in a binary computer is much faster, and algorithms are much more efficient than the usual number system.

The recursion formula of MT is carefully selected to be efficient in binary computers, and at the same time to have good scores. MT is even faster than classical PRNGs.

5. Conclusion

Because of the absence of a good definition of pseudorandomness, the evaluation of PRNG is difficult and consequently poor PRNGs are still being used.

Using mathematics of a non-standard number system which fits to the binary computers,

MT realized very fast generation and excellent theoretical assurance.

An open problem is how to define theoretical scores which measure pseudorandomness. Those used in selecting MT are empirically meaningful and useful, but no rigid justifications exist.

6. Reference

[MT] M. Matsumoto and T. Nishimura “Mersenne Twister: a 623-dimensionally equidistributed uniform pseudorandom number generator”
ACM Transactions on Modeling and Computer Simulation 8. (Jan. 1998) 3--30.
http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html

Mathematics/Applied Mathematics/Computer Science
Planning Group Members: Hiroki Arimura and Jörg Rothe

Randomness in Computations

Speaker:
Misako Takayasu, Tokyo Institute of Technology

Random Potential Force and Abnormal Diffusion observed in Market Price Data

In the late 18th century, Adam Smith applied the physics concept of force to market and introduced the concept of forces of demand and supply in economics, or "the invisible hands." This classical mechanics model of market has been the standard base of modern economics; however, it is less known that the price dynamics has never been confirmed scientifically by real market data.

About 100 years ago Bachelier introduced a random walk model of market price a little before the Einstein's famous paper on Brownian motion. It took nearly 70 years until his idea underwent a revaluation by the name of financial technology. In the formulation of financial technology the motion of market price is assumed to be a random walk without any market force. Although financial technology is now widely used in practical financial world, scientific validation is insufficient and deviations from real market data are pointed out.

Econophysics is a new frontier of science that is aimed to reconstruct economics by orthodox methodology of physics based on objective data analysis. Owing to the recent development of computer technology huge amount of detail market data are now stored and a lot of new empirical findings are established, such as power law distribution of price changes, long time volatility correlations and short time abnormal diffusions.

Here, we analyze the high-precision data in yen-dollar exchange market for 6 years consisted of about 13 million prices in the period of 1995-2002. In our analysis, we observed potential force hidden in random motion of high precision market data. The center of the potential force is moving, and also its curvature is changing moment to moment. When the curvature is positive the force corresponds to the demand-supply force that acts stabilizing the market as anticipated by Adam Smith. On the other hand negative curvature potential, which has never been predicted, makes the market unstable like balancing a bar on a hand.

We introduce the following noise separation for the raw market price data of a US dollar paid by Japanese yen, $p(t)$, where t denotes a pseudo-time called the “tick time”, that is the count of transaction numbers. Since the original tick data, $p(t)$, include a lot of noise, we need some process to reduce the random fluctuation before applying it to analyze the dynamics of short time scale.

$$p(t) = \overline{p(t)} + f(t). \tag{1}$$

This equation shows the decomposition of tick price into slow dynamics and random noise. The first term in the right hand side is named the **optimal moving average** which is defined by a weighted moving average over n ticks. The weights are calculated so that the second term $f(t)$

becomes an independent random noise. This type of noise separation is successfully applicable for any fragment of the data which contains more than 100 thousand ticks.

After successfully removing the random noise from the original noisy data, we introduce another moving average over past *M* optimal moving averages which is called the **Super moving average** to elucidate the center of the hidden force as

$$\overline{P_M(t)} \equiv \frac{1}{M}\sum_{k=1}^{M}\overline{P(t-k)}\,. \qquad (2)$$

We confirm the following relation between the time difference, $\overline{P(t+1)}-\overline{P(t)}$, and displacement of price from the center at time *t*, $\overline{P(t)}-\overline{P_M(t,m)}$.

$$\overline{P(t+1)}-\overline{P(t)} \propto \overline{P(t)}-\overline{P_M(t)}\,. \qquad (3)$$

The left hand term corresponds to derivative of price by time and the right hand term denotes difference of the price from estimated center of the force.

From the observation of equation (3) the proportionality coefficient changes slowly as time develops and it takes both positive and negative coefficients. It also depends on the value of *M.*

If the market price fluctuation is described by a simple random walk, the slope of equation (3) can be estimated as zero mathematically. In the case of negative slope, the relation describes a random walker in an attractive central force. On the other hand, in the case of positive slope it corresponds to a random walker in a repulsive force with its center at $\overline{P_M(t)}$. By integrating from the center of force $\overline{P_M(t)}$, we estimate the potential functions for attractive force. Each of the potential function can be approximated by a quadratic function with a coefficient, depending on time *t* and the size of super moving average *M.*

Analyzing the whole market data it is found that the *M* dependence of the coefficient is factorized by $1/(M-1)$; therefore, the potential function is characterized by a time dependent coefficient $b(t)$ which is independent of the bin size *M* of taking super-moving average:

$$U(\overline{P(t)}-\overline{P_M(t)}) = \frac{1}{2}\cdot\frac{b(t)}{M-1}\cdot(\overline{P(t)}-\overline{P_M(t)})^2 \qquad (4)$$

Here, when $b(t)$ is positive the potential force is attractive and the market is stable; on the other hand, when it is negative the market is unstable due to the repulsive potential force effect.

In this talk, we show the results of our potential analysis in Yen-Dollar exchange market including the data of terrorist attack on 9.11.2001, and intervention of the Bank of Japan.

We also study the statistics of derived equation (4), and show the abnormal diffusion in such moving potentials.