НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
сентябрь–октябрь 2018 Том 18 № 5 ISSN 2226-1494 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
September–October 2018 Vol. 18 No 5 ISSN 2226-1494 http://ntv.ifmo.ru/en

УДК 004.056

ГЕНЕРАТОР ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ НА ОСНОВЕ КЛЕТОЧНЫХ АВТОМАТОВ

Д.Д. Мухамеджанов[a], А.Б. Левина[a]

[a] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
Адрес для переписки: levina@cit.ifmo.ru

Информация о статье
Поступила в редакцию 05.06.18, принята к печати 15.07.18
doi: 10.17586/2226-1494-2018-18-5-894-900
Язык статьи – русский

Ссылка для цитирования: Мухамеджанов Д.Д., Левина А.Б. Генератор псевдослучайных чисел на основе клеточных автоматов // Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18. № 5. С. 894–900. doi: 10.17586/2226-1494-2018-18-5-894-900

Аннотация

Предмет исследования. Разработан алгоритм генерации псевдослучайных чисел, основанный на свойствах клеточных автоматов. Клеточные автоматы имеют большой потенциал, обладают высокой скоростью вычислений, особенно при реализации в параллельной архитектуре. **Метод.** В представленном алгоритме псевдослучайные числа генерируются с помощью правил переходов в ячейках клеточного автомата в зависимости от шаблонов соседства и выходных данных ячеек «соседей». Через несколько переходов по выбору методики генерирования получается последовательность псевдослучайных чисел из нулей и единиц. **Основные результаты.** Разработанный алгоритм протестирован на NIST-тестах. Результаты тестирования показали, что алгоритм производит последовательность с равномерным распределением с вероятностью 99–100%. На NIST-тестах проведено сравнение предложенного алгоритма с линейно конгруэнтным методом – основным методом генерации псевдослучайных чисел в настоящее время. По всем тестам разработанный генератор псевдослучайных чисел показал лучшие результаты. Алгоритм обладает высокой скоростью и легкостью реализации, а также возможностью масштабирования. **Практическая значимость.** Генератор может использоваться в различных приложениях, таких как теория кодирования или легковесная криптография. Достигается криптографическая стойкость при испытаниях по стандартным методикам оценивания качества генератора псевдослучайных чисел.

Ключевые слова
криптография, генератор псевдослучайных чисел, клеточные автоматы, однородные структуры, NIST, случайные числа

PSEUDORANDOM NUMBER GENERATOR ON CELLULAR AUTOMATA

D.D. Mukhamedjanov[a], A.B. Levina[a]

[a] ITMO University, Saint Petersburg, 197101, Russian Federation
Corresponding author: levina@cit.ifmo.ru

Article info
Received 05.06.18, accepted 15.07.18
doi: 10.17586/2226-1494-2018-18-5-894-900
Article in Russian
For citation: Mukhamedjanov D.D., Levina A.B. Pseudorandom number generator on cellular automata. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2018, vol. 18, no. 5, pp. 894–900 (in Russian). doi: 10.17586/2226-1494-2018-18-5-894-900

Abstract

Subject of Research. The paper presents an algorithm for pseudorandom number generation based on properties of cellular automata. Cellular automata have high potential, high speed of calculations, especially at realization in parallel architecture. **Method.** In the presented algorithm pseudorandom numbers are generated by means of rules of transitions in cells of the cellular automaton depending on templates of the neighborhood and the output data of cells of "neighbors". Through several transitions at the choice of a generation technique the sequence of pseudorandom numbers turns out from zeroes and units. **Main Results.** The developed algorithm is tested on NIST-tests. The results of testing have shown that the algorithm makes the sequence with uniform distribution with probability of 99-100%. Comparison of the proposed algorithm with linearly congruent method, the main up-to-date method of generation of pseudorandom numbers, is carried out on NIST-tests. According to all tests the developed generator of pseudorandom numbers has shown the best results. The algorithm has the high speed, easy realization and also scaling possibility. **Practical Relevance.** The generator can be used in various

applications, such as the theory of coding or lightweight cryptography. The cryptographic firmness is reached at tests by standard quality estimation techniques for the generator of pseudorandom numbers.

Keywords
cryptography, pseudorandom number generator, cellular automata, homogenous structures, NIST, random numbers

Введение

В работе рассматриваются проблемы существующих генераторов, приведены описания однородных структур и клеточных автоматов (КА), предложен алгоритм генерации псевдослучайных чисел на клеточных автоматах, проведено тестирование предложенного алгоритма и сравнение с существующими. Широкое применение представленного алгоритма обусловливается достаточно легкой интеграцией как на уровне приложений, так и на более низком уровне, а также возможностью распараллеливания для большей скорости порождения псевдослучайных последовательностей чисел.

Последовательности случайных чисел широко используются в различных областях и приложениях, например, в криптографии (теория кодирования, потоковые шифры, ключи шифрования). В данной работе под случайными последовательностями подразумеваются именно последовательности, сгенерированные под воздействием физических процессов или с помощью недетерминированных алгоритмов. Но не всегда нужны абсолютно случайные последовательности, и есть области, где абсолютно случайные последовательности использовать не следует. Довольно часто алгоритмы должны повторять генерацию точно такой же последовательности, как и ранее с теми же входными данными, используя детерминированные функции.

Псевдослучайные числа необходимы в разных приложениях, но поиск генераторов с хорошими показателями критериев статистических тестов является трудной задачей. Известные практические методы получения псевдослучайных чисел основаны на детерминированных алгоритмах, поэтому такие числа и носят название псевдослучайных, поскольку заведомо известно об операциях, порождающих последовательности, разумеется, они отличаются от истинных случайных последовательностей, полученных в результате естественного физического процесса.

Для успешного применения в длинных стохастических моделях, подобных тем, которые используются в вычислительной физике, генераторы псевдослучайных чисел должны обладать рядом свойств. Наиболее важными свойствами с этой точки зрения являются хорошие результаты в стандартных статистических тестах на случайность, вычислительную эффективность, длительный период (минимальное число между повторениями) и воспроизводимость последовательности (например, таковым является набор тестов National Institute of Standards and Technology (NIST) [1]). Рассматривая несколько примеров генераторов псевдослучайных чисел (ГПСЧ), обратим внимание на линейно конгруэнтный метод и регистр сдвига с линейной обратной связью (РСЛОС) как самые известные и распространенные. Они довольно просты, но прекрасно описывают основную идею псевдослучайных чисел.

РСЛОС использует схему сдвигового регистра с входом предыдущего бита, обработанного линейной функцией. В основном используется функция XOR как линейная функция в таких генераторах. Используемая структура описывает простоту алгоритма, но в то же время и его уязвимость. Идея алгоритма заключается в том, что выбирается некоторое начальное значение, а затем это значение смещается на некоторое количество битов с обработкой битов из регистра по линейной функции. Как видно из основной идеи алгоритма, это абсолютно предсказуемо, так что мы можем поэтапно восстановить начальное значение от конца до начала.

Другим ГПСЧ является алгоритм на основе линейно конгруэнтного метода. Лучшее описание идеи этого алгоритма можно увидеть по формуле

$$X_{n+1} = (aX_n + c) \bmod m,\ n \geq 0,\ m > 0,\ 0 < a < m,$$

где $m > 0$ – модуль, a – множитель и c – аддитивная постоянная. Как видно из формулы, она является рекуррентной, а последовательность, порождаемая этим алгоритмом, имеет период m, после чего последовательность будет повторяться [2–5].

Первый генератор случайных чисел, основанный на КА, был предложен Стивеном Вольфрамом [6]. Алгоритм генерации состоял в применении правила «30» к одномерному клеточному автомату с окрестностью радиуса $r = 1$. Было произведено 7 статистических тестов, которые показали, что генератор на КА значительно лучше, чем РСЛОС. В настоящее время существует множество вариаций ГПСЧ на основе КА: на основе генетических алгоритмов, с использованием нелинейной функции, временного интервала и пространственного интервала и т.д.

В работе используется алгоритм, основывающийся на существующих свойствах КА порождать однородные структуры различной сложности (различные конфигурации битов), но с усиленными показателями по критериям статистических тестов NIST, что обеспечивает надежность, усиленные свойства периодичности и случайности конфигураций, оставляя высокую скорость порождения последовательностей.

Теоретические основы клеточных автоматов

КА широко распространены в области моделирования сложных систем и генерации псевдослучайных чисел. КА получили известность благодаря разработанной Конвеем игре «Жизнь» [7], где на решетке «эволюционировали» ячейки, меняя состояния в зависимости от окружающих условий (состояний соседних ячеек), тем самым порождались новые структуры. Именно это послужило началом развития КА как динамических структур.

КА [8–12] можно формально описать как четверку $\sigma = \left(Z^k, E_n, V, \varphi\right)$, где Z^k – множество k-мерных векторов, $E_n = \{0,1,...,n-1\}$ – множество состояний одной ячейки в φ, $V = (\alpha_1,...,\alpha_{h-1})$ – окрестность или шаблон соседства (упорядоченное множество различных k-мерных векторов из Z^k, $\varphi = \varphi(x_0, x_1,..., x_{h-1})$, $\varphi : E_n^h \to E_n$ – локальная функция перехода.

Из определения КА видно, что КА можно сравнить с множеством обычных автоматов Мура [13], если их состояния будут зависеть от состояний и выходов окружающих их автоматов. Шаблоны соседства могут отличаться друг от друга (шаблон соседства Неймана [14] больше похож на символ «+» с изменяемой центральной ячейкой, шаблон соседства Мура больше похож на квадрат 3×3. Существует огромное количество комбинаций и вариаций шаблонов соседства, которые можно использовать.

Существуют также два типа КА, которые отличаются друг от друга в терминах правил. Дело в том, что один тип, называемый однородным КА, определяет одно правило для всех ячеек в сетке, но второй тип, неоднородный [15], может использовать несколько различных правил для ячеек. В то же время оба они имеют одинаковые преимущества в простоте построения, локальности и параллелизме с разницей в реализации (неоднородные требуют больше памяти для описания правил).

Согласно обозначениям Вольфрама [16], сопоставим всем возможным наборам из трех битов значения согласно представлению числа 30 в двоичной форме, недостающие биты заполним нулями. Конфигурацией назовем набор единиц и нулей (КА с двумя состояниями, где последовательность рассматривается как случайное число) в определенный дискретный момент времени. Опишем наиболее известное правило Вольфрама «30» в виде примера (табл. 1) или в виде функции

$$f_0(t+1) = f_{-1}(t) \oplus f_0(t) \vee f_1(t),$$

где – $f_0(t+1)$ – состояние f целевой клетки (0) в следующий момент времени $(t+1)$; $f_{-1}(t)$ – состояние соседней клетки слева от целевой (-1) в настоящий момент времени (t); $f_1(t)$ – состояние соседней клетки справа от целевой (1) в настоящий момент времени (t).

Шифр ячеек	000	001	010	011	100	101	110	111
Следующее состояние центральной ячейки	0	1	1	1	1	0	0	0

Таблица 1. Правило «30»

Алгоритм генерации псевдослучайных чисел

Основная идея представленного алгоритма состоит в том, чтобы использовать преимущества генерации псевдослучайных чисел клеточными автоматами и улучшить ряд свойств для повышения критериев статистических свойств клеточных автоматов.

Сетка. Основной результат представленного алгоритма состоит в том, что он может генерировать псевдослучайные последовательности чисел. Несмотря на то, что КА уже использовался таким образом, мы улучшили некоторые характеристики, добавив предложения по реализации алгоритма и изменению математической модели. Существует сетка, в которой каждая ячейка имеет только два состояния: 0 и 1, а состояния меняются некоторой функцией. Результатом шагов в алгоритме будет двоичное число, которое может быть сформировано из сетки (или ее части). Сетка имеет размеры p и q, которые являются простыми числами (для улучшения периодичности). Изменения начинаются прямо с этого этапа: деление всей сетки на $m \geq 2$ блоков, которые формируются как прямоугольники b_i с размерами l_{b_i} и w_{b_i}, а каждый блок состоит из $l_{b_i} \times w_{b_i}$ ячеек (рис. 1).

Начальная конфигурация. Следующим шагом алгоритма будет выбор начальной конфигурации. Есть два способа – заполнить каждый из блоков в отдельности или заполнить исходную сетку. Также есть два способа заполнения сеток – источником энтропии или с помощью известных величин размерности каждого блока и исходной сетки.

Если выбирается заполнение исходной сетки, то в какой-нибудь из блоков может попасть большинство нулей, а значит, конфигурации будут эволюционировать неравномерно и с различной скоростью. В

этом случае есть вероятность получения длинной последовательности единиц или нулей в результате. Поэтому выберем заполнение каждого блока в отдельности. Физический источник энтропии, такой как белый шум, например, даст случайное число, но при этом необходимо потратить время и вычислительные ресурсы на получение и использование данных величин. В ином случае, выбирая размерность как источник начальной конфигурации, экономим время и память устройства реализации.

Выбрав произведение сторон прямоугольников блоков $l_{b_i} \times w_{b_i}$ как начальную конфигурацию, необходимо заполнить ячейки таким образом, чтобы в каждой ячейке располагался один бит (состояние 0 или 1).

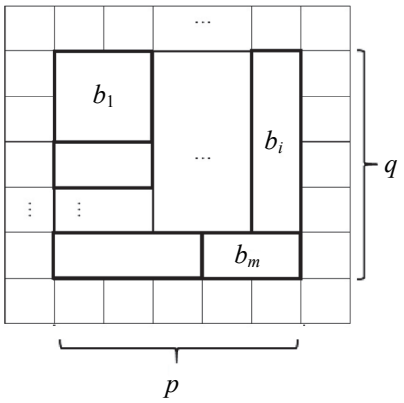

Рис. 1. Разделение сетки на блоки

Применим алгоритм заполнения под названием NESW (North, East, South, West). Метод такого движения носит название по сторонам света (рис. 2). Суть способа в том, чтобы заполнять сетку блока согласно направлениям сторон света, т.е. мы циклически заполняем битами полученного числа сетку, начиная с левого нижнего угла, двигаясь сначала вверх, затем вправо, вниз и влево до конца сетки (или до заполненной ячейки).

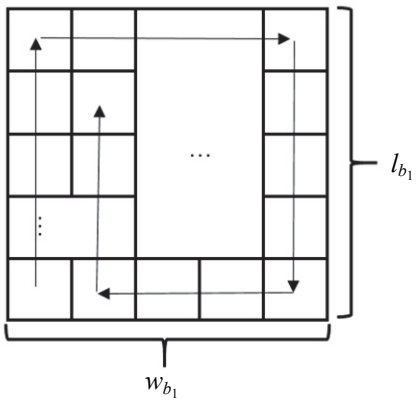

Рис. 2. Метод NESW

Правила и шаблоны соседства. Будем работать только с двумерным пространством, это означает, что мы берем $k = 2, n = 2, Z^2, E_2$. Начальная конфигурация будет формироваться по правилу NESW, так же как и блоки.

Алгоритм обеспечивается двумя типами правил – пронумерованные правила (например, правило «30» в начале этой статьи) и простые операции (их комбинация). Каждый блок может иметь свое собственное правило, что делает КА неоднородным и приводит к лучшим показателям критериев статистических свойств и лучшим периодическим свойствам.

Правило для каждого блока может быть выбрано простой операцией mod из $l_{b_i} \times w_{b_i}$ в случае использования всех возможных правил в блоке. Но для достижения наилучших результатов следует выбрать несколько правил с равным числом единиц и нулей (это условие может использоваться для всей сетки – главное, чтобы совокупное число единиц и нулей во всех правилах было равно). Кроме того, чтобы избежать поколений с последовательностями, не удовлетворяющих условиям критериев, мы должны определить «хорошие» правила для сетки.

Последний вопрос при генерации псевдослучайного числа – это окрестность. Имея блоки, мы можем выбрать несколько выборок для некоторых блоков, а также их количество и разнообразие в зависи-

мости от количества блоков. Невозможно использовать больше правил, чем количество блоков, но можно использовать аналогичные правила для нескольких блоков.

Повторение. Чтобы повторить генерацию того же числа, необходимо воссоздать идентичный КА, но передавать весь КА в виде матрицы небезопасно.

Предложение по некоторой обфускации передаваемых данных – ключ K, который включает в себя следующее:

$$K = p \,|\, q \,|\, l_{b_1} \,|\, w_{b_1} \,|\, ... \,|\, l_{b_m} \,|\, w_{b_m} \,|\, V_1 \,|\, ... \,|\, V_m \,|\, T,$$

где V_i определяется как последовательность из 9 бит: соответственно, если бит равен 1, тогда ячейка входит в выборку, и наоборот.

$(-1,1)$	$(0,1)$	$(1,1)$
$(-1,0)$	$(0,0)$	$(1,0)$
$(-1,-1)$	$(-1,0)$	$(1,-1)$

Таблица 2. Схема расположения и описания координат ячеек

Табл. 2 передается в ключе следующим образом:

$(-1,-1)\,;(-1,0)\,;(1,-1)\,;(-1,0)\,;(0,0)\,;(1,0)\,;(-1,1)\,;(0,1)\,;(1,1)$,

где координаты меняются на значение вектора 0 или 1, которые говорят о наличии ячеек-соседей. Например, в рассматриваемой работе вектор соседства (одномерный с радиусом $r=1$) будет кодироваться как 000111000; T – вспомогательной информационный вектор, который может нести информацию о правилах, использующихся в работе КА, количестве шагов эволюции КА и других данных. Так как используется одномерный случай, то не представляется сложным кодировать правила и разбирать строку.

Тестирование

Для тестирования алгоритма генерации псевдослучайных чисел на основе клеточных автоматов применялись следующие средства: IDE Visual Studio 2015; язык C, C++; NIST test suite.

P-value	**Proportion, %**	**Statistical test (Тест)**
0,744146	99,6	Frequency (Частотный)
0,380537	99,4	BlockFrequency (Блоковый частотный)
0,734146	99,9	CumulativeSums (Кумулятивные суммы)
0,428244	98,5	Runs (Пробег)
0,383827	99,1	LongestRun (Самая длинная последовательность)
0,702458	99,0	Rank (Ранг)
0,650637	98,5	FFT (Быстрое преобразование Фурье)
0,433358	98,7	NonOverlappingTemplate (Неперекрывающий шаблон)
0,632191	100	OverlappingTemplate (Перекрывающий шаблон)
0,414862	98,3	Universal (Универсальный)
0,778903	99,1	ApproximateEntropy (Энтропия)
0,610977	98,8	RandomExcursions (Случайные отклонения)
0,633388	99,2	RandomExcursionsVariant (Вариативные случайные отклонения)
0,651956	99,3	Serial (Серии)
0,730485	100	LinearComplexity (Линейная сложность)

Таблица 3. Результаты тестирования алгоритма на NIST test suite

На языке C++ было написано ядро программы ГПСЧ на основе КА с помощью IDE Visual Studio 2015, результатом работы которой был выходной текстовый файл с последовательностью битов длиной не менее 1000000, так как требования пакета NIST test suite не позволяли проверять последовательность меньшей длины. Далее текстовый файл с последовательностью проверялся программным пакетом NIST test suite и на выходе печатались различные показатели статистических тестов.

Были проверены около 1000 различных вариаций правил и шаблонов соседства и выделены самые устойчивые и эффективные среди них. Данные теста были усреднены по всем показателям. Была использована исходная сетка размером $p=307$, $q=53$. Шаблоном соседства была выбрана одномерная окрестность с радиусом $r=1$.

Для каждого блока было взято свое правило, набор правил был следующим: 45, 75, 89, 101, 135, 86. Результаты тестирования показаны в табл. 3.

В табл. 3 и табл. 4 показаны результаты тестов по параметру P-value, который характеризует отличие характеристик тестируемой последовательности с показателями последовательности с равномерным распределением, в некотором роде абсолютной для ГПСЧ.

CA		Linear Congruential		Statistical test
P-value	Proportion, %	P-value	Proportion, %	
0,744146	99,6	0,739918	99,8	Frequency
0,380537	99,4	0,122325	99,0	BlockFrequency
0,734146	99,9	0,689918	100	CumulativeSums
0,428244	98,5	0,213309	98,7	Runs
0,383827	99,1	0,122325	98,1	LongestRun
0,702458	99,0	0,213309	99,0	Rank
0,650637	98,5	0,639918	100	FFT
0,433358	98,7	0,578346	98,4	NonOverlappingTemplate
0,632191	100	0,350485	100	OverlappingTemplate
0,414862	98,3	0,213309	98,3	Universal
0,778903	99,1	0,791468	100	ApproximateEntropy
0,610977	98,8	0,534146	99,2	RandomExcursions
0,633388	99,2	0,468312	100	RandomExcursionsVariant
0,651956	99,3	0,615983	100	Serial
0,730485	100	0,213309	100	LinearComplexity

Таблица 4. Сравнение показателей теста с линейно конгруэнтным методом

Заключение

Предложен алгоритм генерации псевдослучайных чисел на основе клеточных автоматов. Результаты тестирования показывают, что описанный генератор обладает высокой эффективностью, простотой реализации и масштабирования, высокой скоростью и улучшенными, относительно существующих алгоритмов, показателями на статистических тестах. Генератор может использоваться в различных приложениях, таких как теория кодирования или легковесная криптография.

Литература

1. Rukhin A. et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22. 2001. 152 p.
2. Tomassini M., Sipper M., Perrenoud M. On the generation of high-quality random numbers by two-dimensional cellular automata // IEEE Transactions on Computers. 2000. V. 49. N 10. P. 1146–1151. doi: 10.1109/12.888056
3. Ilachinski A. Cellular Automata: A Discrete Universe. World Scientific Publishing Company, 2001. 840 p. doi: 10.1142/4702
4. Tomassini M., Perrenoud M. Cryptography with cellular automata // Applied Soft Computing. 2001. V. 1. N 2. P. 151–160. doi: 10.1016/s1568-4946(01)00015-1
5. Tomassini M., Sipper M., Zolla M., Perrenoud M. Generating high-quality random numbers in parallel by cellular automata // Future Generation Computer Systems. 1999. V. 16. N 2-3. P. 291–305. doi: 10.1016/s0167-739x(99)00053-9
6. Wolfram S. Random sequence generation by cellular automata // Advances in Applied Mathematics. 1986. V. 7. N 2. P. 123–169.
7. Gardner M. The fantastic combinations of John Conway's new solitaire game "Life" // Scientific American. 1970. V. 223. P. 120–123.
8. Кудрявцев В.Б., Подколзин А.С. Клеточные автоматы // Интеллектуальные системы. 2006. Т. 10. № 1-4. С. 657–692.
9. Tomassini M. Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time. Springer, 2005. 193 p. doi: 10.1007/3-540-29938-6
10. Наумов Л.А., Шалыто А.А. Клеточные автоматы. Реализация и эксперименты // Мир ПК. 2003. № 8. С. 64–71.
11. Астафьев Г.Б., Короновский А.А., Храмов А.Е. Клеточные автоматы. Саратов: Центр «Колледж», 2003. 24 с.
12. Жуков А.Е. Клеточные автоматы в криптографии. Часть 2 // Вопросы кибербезопасности. 2017. № 4(22). С. 47–66. doi: 10.21681/2311-3456-2017-4-47-66

References

1. Rukhin A. et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. *NIST Special Publication 800-22*, 2001, 152 p.
2. Tomassini M., Sipper M., Perrenoud M. On the generation of high-quality random numbers by two-dimensional cellular automata. *IEEE Transactions on Computers*, 2000, vol. 49, no. 10, pp. 1146–1151. doi: 10.1109/12.888056
3. Ilachinski A. *Cellular Automata: A Discrete Universe*. World Scientific Publishing Company, 2001, 840 p. doi: 10.1142/4702
4. Tomassini M., Perrenoud M. Cryptography with cellular automata. *Applied Soft Computing*, 2001, vol. 1, no. 2, pp. 151–160. doi: 10.1016/s1568-4946(01)00015-1
5. Tomassini M., Sipper M., Zolla M., Perrenoud M. Generating high-quality random numbers in parallel by cellular automata. *Future Generation Computer Systems*, 1999, vol. 16, no. 2-3, pp. 291–305. doi: 10.1016/s0167-739x(99)00053-9
6. Wolfram S. Random sequence generation by cellular automata. *Advances in Applied Mathematics*, 1986, vol. 7, no. 2, pp. 123–169.
7. Gardner M. The fantastic combinations of John Conway's new solitaire game "Life". *Scientific American*, 1970, vol. 223, pp. 120–123.
8. Kudryavtsev V.B., Podkolzin A.S. Cellular automata. *Intellektual'nye Sistemy*, 2006, vol. 10, no. 4, pp. 657–692. (in Russian)
9. Tomassini M. *Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time*. Springer, 2005, 193 p. doi: 10.1007/3-540-29938-6
10. Naumov L.A., Shalyto A.A. Cellular automata. Implementation and experiments. *Mir PK*, 2003, no. 8, pp. 64–71. (in Russian)
11. Astaf'ev G.B., Koronovskii A.A., Khramov A.E. *Cellular Automata*. Saratov, Kolledzh Publ., 2003, 24 p. (in Russian)

13. Moore E.F. Gedanken-experiments on sequential machines / In: Automata Studies. Eds. C.E. Shannon, J. McCarthy. Princeton University Press, 1956. P. 129–154. doi: 10.1515/9781400882618-006
14. Aspray W. John von Neumann and the Origins of Modern Computing. MIT Press, 1990. 376 p.
15. Cattaneo G., Dennunzio A., Formenti E., Provillard E. Non-uniform cellular automata // Lecture Notes in Computer Science. 2009. V. 5457. P. 302–313. doi: 10.1007/978-3-642-00982-2_26
16. Wolfram S. Tables of cellular automaton properties / In: Theory and Applications of Cellular Automata (Including Selected Papers 1983–1986). World Scientific Publ., 1986. P. 485–557.

12. Zhukov A.E. Cellular automata in cryptography. Part 2. *Cybersecurity Issues*, 2017, no. 4, pp. 47–66. doi: 10.21681/2311-3456-2017-4-47-66
13. Moore E.F. Gedanken-experiments on sequential machines. In: *Automata Studies*. Eds. C.E. Shannon, J. McCarthy. Princeton University Press, 1956, pp. 129–154. doi: 10.1515/9781400882618-006
14. Aspray W. *John von Neumann and the Origins of Modern Computing*. MIT Press, 1990, 376 p.
15. Cattaneo G., Dennunzio A., Formenti E., Provillard E. Non-uniform cellular automata. *Lecture Notes in Computer Science*, 2009, vol. 5457, pp. 302–313. doi: 10.1007/978-3-642-00982-2_26
16. Wolfram S. Tables of cellular automaton properties. In *Theory and Applications of Cellular Automata (Including Selected Papers 1983–1986)*. World Scientific Publ., 1986, pp. 485–557.

Авторы

Мухамеджанов Данияр Давлетович – студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, ORCID ID: 0000-0002-4460-1460, danmd.info@gmail.com

Левина Алла Борисовна – кандидат физико-математических наук, доцент, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 56427692900, ORCID ID: 0000-0003-4421-2411, levina@cit.ifmo.ru

Authors

Daniyar D. Mukhamedjanov – student, ITMO University, Saint Petersburg, 197101, Russian Federation, ORCID ID: 0000-0002-4460-1460, danmd.info@gmail.com

Alla B. Levina – PhD, Associate Professor, Associate Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 56427692900, ORCID ID: 0000-0003-4421-2411, levina@cit.ifmo.ru