УДК 519.7 DOI: 10.31114/2078-7707-2021-3-26-33

Клеточно-автоматный алгоритм пермутации матриц с колебательной схемой движения элемента

И.В. Матюшкин, П.Д. Рубис

Национальный исследовательский университет «Московский институт электронной техники», Москва, Зеленоград, Россия, rubisiay@gmail.com

***Аннотация* — Описывается алгоритм пермутации (перестановки элементов) квадратных матриц, основанный на циклических сдвигах строк и столбцов. Область применения алгоритма – групповая генерация псевдослучайных чисел. Дается формулировка алгоритма в терминах клеточных автоматов (КА). Приводятся результаты численного расчета, прежде всего для периода повторения исходной матрицы. Для матриц четного порядка период линеен или квадратичен. В противном случае зависимость периода алгоритма от порядка матрицы носит рекуррентный характер, полученный нами в виде связки правил, и не превышает экспоненту (точнее, функцию Ландау). Траектории индивидуальных элементов визуализируются в расширенном поле КА. В качестве параметра глобальной динамики КА анализируются две «метрики перемешанности» на пермутациях матрицы (по сравнению с начальной). Поведение этих метрик представлено на графиках и гистограммах (условно: плотности распределения), показывающих, как часто встречаются по периоду пермутации с заданным интервалом значений метрик.**

***Ключевые слова* — клеточные автоматы, пермутация, случайные числа, криптография, метрика.**

I. Введение

Актуальность поиска эффективных способов псевдослучайной пермутации матрицы обусловлена применением этой операции в различных областях. Генерация тестовых последовательностей в логических блоках сверхбольших интегральных схем может требовать случайной перестановки данных [1-2]. Помимо генерации случайных чисел перестановка матричных элементов непосредственно используется в криптографии [3-4]. Обычно вводится псевдослучайная функция, где первая часть декартова произведения относится к k-битному ключу, а вторая – к n-битному сообщению. При составлении стойких криптографических алгоритмов по сей день руководствуются утверждением Шеннона о том, что хорошее перемешивающее преобразование часто является результатом многократного применения произведения двух простых некоммутирующих операций. Такими операциями являются сдвиги строк и столбцов матрицы [5]; там же дана более строгая формализация и точную оценка периода пермутации для алгоритма, предлагаемого нами базовым, через функцию Ландау в теории групп. В [6] показано решение задачи пермутации близкое, будучи дискретизированным аналогом непрерывного двухпараметрического преобразования; оно применялось для кодирования серого 8-битного рисунка.

Отметим, что клеточно-автоматная (КА) формализация алгоритмов облегчает вычислительный параллелизм [7,8]. Поставленный в конце статьи вопрос о максимальном периоде пермутаций, на наш взгляд, связан напрямую с решением задачи о достижимости конфигураций (см., например, [9]). Генерация псевдослучайных чисел может быть эффективно проведена также с помощью аппаратного КА-блока [10]. Базовый алгоритм и доказательство формулы его периода впервые был рассмотрен нами в [11], а результаты исследований смежных с ним алгоритмов описаны нами в [12].

II. КА-формализация алгоритма

Разработанный алгоритм является модификацией ранее представленной [11] схемы «1:1». Поэтому, в терминах КА, алгоритм представляет собой автомат с квадратным полем, размера $n \times n$, где n - порядок входной матрицы (каждому элементу матрицы соответствует ячейка КА). Для всех клеток поля задана окрестность фон-Неймана радиуса 1, определено множество компонент состояния и локальная функция перехода (ЛФП). Границы поля цикличны (тороидальная топология), время дискретно (единица времени соответствует одной итерации). Новая схема получила название «sin» из-за волнообразных сдвигов элементов матрицы, а также из-за смещения направлений перестановок во времени. Реализация движения элемента матрицы по функции $sin\,x$, в данном случае, состоит из четырех периодически повторяющихся смещений, представляющих собой сдвиги в условно положительном и отрицательном направлениях, а также сохранение текущей позиции (отсутствие перемещения). Шаблон движения здесь «0 1 0 -1», что дает 8 итераций, формирующих во времени один проход (рис. 1), далее повторяющийся заново. Первый проход перемешивания матрицы размера 6×6 схемой «sin» представлен в табл. 1.

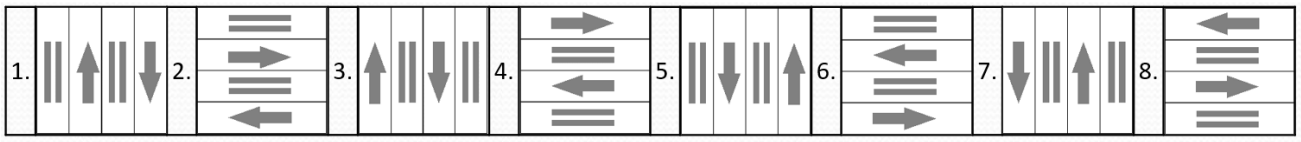


Рис. 1. Визуализация прохода схемы «sin». Указаны направления перемещений для 4-х соседних строк или столбцов по времени (знак «=» означает отсутствие движения)

Таблица 1

Проход схемы «sin» для матрицы 6×6)

Номер итерации	Слой данных
0 (исходная матрица)	1 7 13 19 25 31 2 8 14 20 26 32 3 9 15 21 27 33 4 10 16 22 28 34 5 11 17 23 29 35 6 12 18 24 30 36
1	1 8 13 24 25 32 2 9 14 19 26 33 3 10 15 20 27 34 4 11 16 21 28 35 5 12 17 22 29 36 6 7 18 23 30 31
2	1 8 13 24 25 32 33 2 9 14 19 26 3 10 15 20 27 34 11 16 21 28 35 4 5 12 17 22 29 36 31 6 7 18 23 30
3	33 8 7 24 19 32 3 2 13 14 27 26 11 10 9 20 35 34 5 16 15 28 29 4 31 12 21 22 23 36 1 6 17 18 25 30
4	32 33 8 7 24 19 3 2 13 14 27 26 10 9 20 35 34 11 5 16 15 28 29 4 36 31 12 21 22 23 1 6 17 18 25 30
5	32 6 8 14 24 30 3 33 13 35 27 19 10 2 20 28 34 26 5 9 15 21 29 11 36 16 12 18 22 4 1 31 17 7 25 23
6	32 6 8 14 24 30 33 13 35 27 19 3 10 2 20 28 34 26 11 5 9 15 21 29 36 16 12 18 22 4 31 17 7 25 23 1
7	31 6 35 14 23 30 32 13 20 27 24 3 33 2 9 28 19 26 10 5 12 15 34 29 11 16 7 18 21 4 36 17 8 25 22 1
8	6 35 14 23 30 31 32 13 20 27 24 3 26 33 2 9 28 19 10 5 12 15 34 29 16 7 18 21 4 11 36 17 8 25 22 1

В табл. 2 представлены компоненты, определяющие состояния ячеек в схеме «sin».

Флаг перемещения m однозначно задает сдвиги элементов, в то время как компонента d обеспечивает работу алгоритма как по столбцам, так и по строкам. Оба флага меняют значения друг друга на каждой итерации, и для их корректного взаимоопределения вводится компонентаit. ЛФП схемы приводится в табл. 3.

Таблица 2

Описание компонент клетки $\langle m, d, it, s\rangle$ схемы «sin»

Название:	Флаг перемещения	Флаг направления движения	Флаг итераций	Регистр данных
Обозначение:	m	d	it	s
Множество значений:	$m = \{-1; 0; 1\}$	$d = \{1; 2; 3\}$	$it = \{0; 1; 2; 3\}$	Произвольное
Описание:	Определяет направление сдвигов элементов на текущей итерации.	Обеспечивает сдвиги как по столбцам, так и по строкам.	Определяет итерацию смены основных условий изменения двух других флагов	Хранит элементы матрицы M=$\|\|M_{ij}\|\|$.

Таблица 3

ЛФП схемы «sin»

№ п/п	Условие перехода				Формула перехода
1	$d = 1$				$m := 1$
	$d = 2$				$m := 0$
	$d = 3$				$m := -1$
2	$(mov = -1) \vee (mov = 1)$				$d := 2$
	$mov = 0$	$(mov_{\downarrow} = mov_{\uparrow}) \wedge (mov_{\uparrow} = mov)$	$it = 1$	$mov_{\rightarrow} \neq -1$	$d := 1$
				$mov_{\rightarrow} = -1$	$d := 3$
			$it\ mod\ 2 = 0$	$mov_{\leftarrow} = -1$	$d := 1$
				$mov_{\leftarrow} = 1$	$d := 3$
			$it = 3$	$mov_{\rightarrow} = 1$	$d := 1$
				$mov_{\rightarrow} \neq 1$	$d := 3$
		$(mov_{\leftarrow} = mov_{\rightarrow}) \wedge (mov_{\rightarrow} = mov)$	$it = 1$	$mov_{\downarrow} \neq -1$	$d := 1$
				$mov_{\downarrow} = -1$	$d := 3$
			$it\ mod\ 2 = 0$	$mov_{\uparrow} = -1$	$d := 1$
				$mov_{\uparrow} = 1$	$d := 3$
			$it = 3$	$mov_{\downarrow} = 1$	$d := 1$
				$mov_{\downarrow} \neq 1$	$d := 3$
3	$d_{\uparrow} = d_{\downarrow}$				$it := (it + 1) \bmod 4$
4	$m = -1$	$(m_{\uparrow} = -1) \wedge (m_{\downarrow} = -1)$			$s := s_{\uparrow}$
		$(m_{\leftarrow} = -1) \wedge (m_{\rightarrow} = -1)$			$s := s_{\rightarrow}$
	$m = 0$				$s := s$
	$m = 1$	$(m_{\uparrow} = 1) \wedge (m_{\downarrow} = 1)$			$s := s_{\downarrow}$
		$(m_{\leftarrow} = 1) \wedge (m_{\rightarrow} = 1)$			$s := s_{\leftarrow}$

III. Период алгоритма

Эмпирически установлено, что через некоторое число проходов алгоритмы переводят матрицу в исходное состояние, что принято нами за условие останова алгоритма. Это следует теоретически из конечности числа перестановок, которое для матрицы $n \times n$ для попарно различных элементов не превосходит $(n^2)!$. Уточним, что период алгоритма N вводится не только для компоненты данных s, а также и для других компонент состояния. То есть пермутация совпадает с исходной матрицей, если обе компоненты всех её элементов принимают начальные значения. Если же для флагов совпадения нет, то такую ситуацию назовем *самопересечением*. Чем длиннее период, тем лучше с точки зрения криптографии; и здесь, что несколько необычно, мы должны не сокращать, а увеличивать сложность алгоритма.

Данный алгоритм имеет запутанную структуру зависимостей $N(n)$, однако все n, как и в схеме «1:1», могут быть также разделены на четные и нечетные. N

для всех матриц с четным n в синусоидальной схеме структурно разбивается на две ветви, одна из которых состоит из трех подветвей (табл. 4). Сравнение периодов схем «1:1» и «sin» для нечетных значений n представлено в табл.5.

Таблица 4

$N(n)$ в схеме «sin» для классов $n = 4k$ и $n = 4k + 2$

$n(k)$	$k(m)$	$N(n)$	
		множитель	формула
$4k$	–	1	$(9/4)n$
$4k + 2$	$2m + 1$	1	$\frac{9}{16}(n + 6)(3n + 2)$
	$4m + 2$	$\frac{1}{4}$	
	$4m$	$\frac{1}{2}$	

Таблица 5

Сравнение периодов схем «1:1» и «sin» для нечетных n

n	$N(n)$ (в итерациях)			$N(n)$ (в проходах)	
	Схема «1:1»	Схема «sin»	$\frac{N(n)_{sin}}{N(n)_{1:1}} \cdot 100\%$	Схема «1:1»	Схема «sin»
3	60	24	40	15	3
5	1260	1,120	88.(8)	315	140
7	180180	6,336	3.516484	45,045	792
9	3063060	2,688	0.087755	765,765	336
11	58198140	15,840	0.027217	14,549,535	1,980
13	6692786100	9,853,200	0.147221	1,673,196,525	1,231,650
15	582,272,390,700	–	–	145,568,097,675	
17	18,050,444,111,700	2,545,920	0.0000017	4,512,611,027,925	318,240
19	667,866,432,132,900	–	–	166,966,608,033,225	–
21	27,382,523,717,448,900	–	–	6,845,630,929,362,225	–
23	1,177,448,519,850,302,700	–	–	294,362,129,962,575,675	–

Примечание: прочерки означают отсутствие непосредственного численного расчета из-за сверхбольших чисел

IV. Индивидуальные траектории элементов матрицы

Для лучшего понимания формул для периода рассмотрим теперь траектории движения отдельных элементов матрицы (или, для краткости, индивидуальные траектории – УИТ, пример для матрицы размера 6×6 представлен на рис. 2) и их временные длины в проходах (ДИТ). Поле КА с замыканием удобно представить как бесконечную плоскую периодическую решётку (рис. 3). Такое представление будем называть *расширенным*.

Все ДИТ для матриц четных порядков ($n = 4k$и $n = 4k + 2$) на исследованном натуральном линейно зависят от *n*. Также, множество длин траекторий класса $n = 4k + 2$ всегда содержит 1 и 3.

В классах нечетных порядков мощность множества ДИТ линейна по n (табл. 6). Для обнаружения закономерностей нами были угаданы рекуррентные соотношения. При переходе $k \to k + 1$ в множество ДИТ добавляются два элемента, на 9 большие двух наибольших элементов прежнего множества ДИТ. Уже имевшиеся ДИТ увеличиваются обычно на единицу, кроме некоторых, инкремент возрастания которых равняется двум или трем. Легко видеть, что для такого нарушения характерна периодичность (табл. 7)

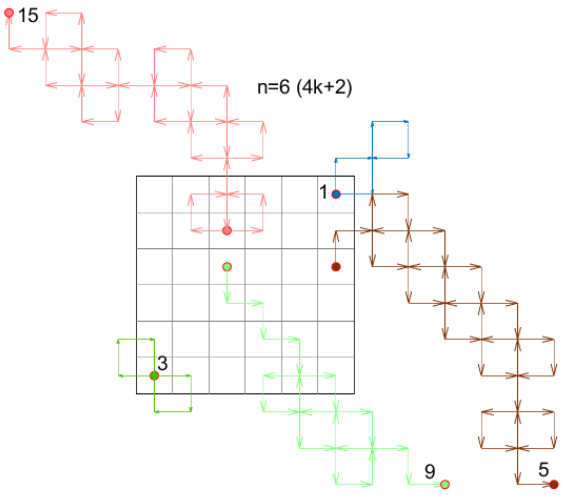


Рис. 2. УИТ для матрицы 6×6 ($n = 4k + 2$) в схеме «sin»

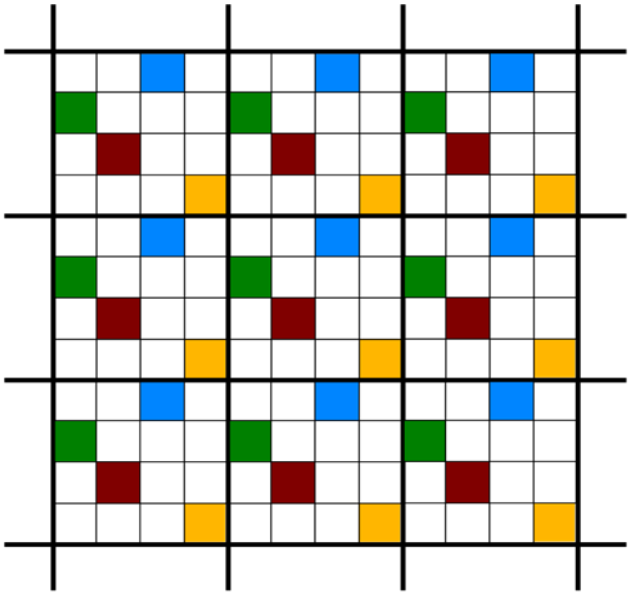


Рис. 3. Расширенное представление для матрицы 4×4

Таблица 6

Эмпирически полученные формулы ДИТ для $n = 4k$ и $n = 4k + 2$ в схеме «sin»

$n(k)$	$l_1(n)$	$l_2(n)$	$l_3(n)$	$l_4(n)$	$l_5(n)$	$l_5(n)$
$4k$	$\frac{n}{4}$	$\frac{3}{4}n$	$\frac{9}{4}n$	-	-	-
$4k+2$	1	3	$\frac{1}{4}n+2$	$\frac{3}{4}n+2$	$\frac{3}{4}n+6$	$\frac{9}{4}n+6$

Таблица 7

Особенности изменения множества ДИТ с увеличением k в классе $n = 4k + 1$, $k \in \mathbb{N}$

k	Новые ДИТ	Количество ДИТ	Число нарушений возрастания ДИТ	ДИТ, нарушающие закон возрастания	Инкремент нарушения
1	1;4;5;7	4	0		
2	14;16	6	2	6;7	2;2
3	23;25	8	2	9;10	2;2
4	32;34	10	1	13	3
5	41;43	12	1	16	3
6	50;52	14	2	18;19	2;2
7	59;61	16	2	21;22	2;2
8	68;70	18	1	25	3
9	77;79	20	1	28	3
10	86;88	22	2	30;31	2;2
11	95;97	24	2	33;34	2;2
12	104;106	26	1	37	3
13	113;115	28	1	40	3
14	122;124	30	2	42;43	2;2
15	131;133	32	2	45;46	2;2
16	140;142	34	1	49	3

Опишем рекурсию более строго. Пусть для некоторого k дано множество всех ДИТ:

$$\Lambda_k = \{l \in \mathbb{N}\}, L_k = card(\Lambda_k)$$

Тогда для класса $n = 4k + 1, k > 0$ имеют место следующие свойства:

- $\Lambda_1 = \{1,4,5,7\}$ - начальное множество
- $L_0 = 2, L_{k+1} = L_k + 2$ - рекурсия числа ДИТ
- $1 \in \Lambda_k$ - всегда есть ДИТ=1
- $\{9k - 2, 9k - 4\} \subset \Lambda_{k+1}$ - порождение новых элементов
- $\exists \Phi_k \subset \Lambda(k): |\Phi_k| = 1 \vee 2$ - наличие нарушителей
- $(k = 4m) \vee (k = 4m + 1) \Rightarrow \big(\Phi_k = \{3k + 1\} \wedge (3k + 4 \in \Lambda_{k+1})\big)$ $(k = 4m - 2) \vee (k = 4m - 1) \Rightarrow \big(\Phi_k = \{3k, 3k + 1\} \wedge (\{3k + 2, 3k + 3\} \in \Lambda_{k+1})\big)$- правило на нарушения
- $(\forall k: l \in \Lambda_k)\Big(\big(l \notin (\Phi_k \cup \{1\})\big) \Rightarrow \big((l + 1) \in \Lambda_{k+1}\big)\Big)$ - регулярное правило

Для класса $n = 4k + 3$ особенности изменения элементов множества ДИТ полностью идентичны классу $n = 4k + 1$. Различие только в начальном множестве Λ_1 и в том, что неизменны первые два ДИТ $\{1,3\}$, а не только ДИТ=1.

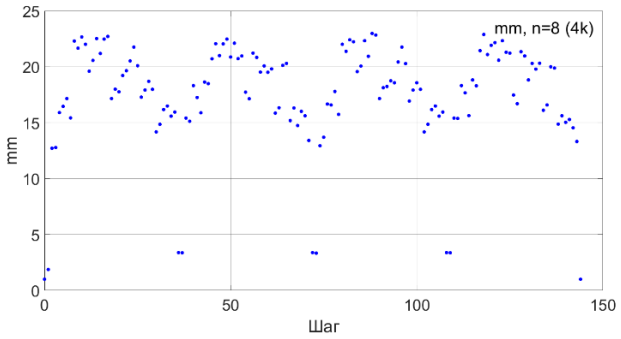
V. Анализ динамики алгоритма с помощью «метрик перемешанности»

Ранее авторами уже были введены специальные «метрики перемешанности» *mm* (matrix mixedness) и *lm* (linear mixedness), численно показывающие степень перемешанности пермутации относительно начальной матрицы. Характер гистограмм данных метрик также показывает перемешивающие свойства алгоритма.

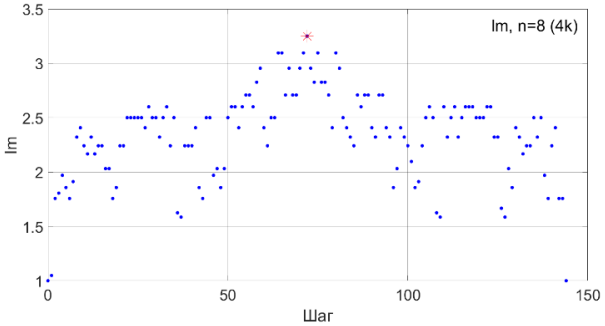
Характеристики перемешанности в группах (n=4k n=4k+2) довольно хаотичны и не описываются аналитическими выражениями. Для класса n=4k на половине периода алгоритма присутствует конфигурация блочной перестановки, как в базовом алгоритме, соответствующая максимуму *lm*-метрики (рис. 4а, б).

Метрики нечетных классов имеют выделяющиеся глобальные максимумы и локальные минимумы. Однако для обеих групп на периоде отсутствуют особые конфигурации. Также для *lm*-метрики характерна большая дискретизация по значениям, что особенно видно на гистограммах (рис. 5а, б).

Общий вид гистограмм *mm*-перемешанности напоминает нормальное распределение с асимметрией (рис. 5а), а на гистограммах *lm*-метрики заметны области значений, в которых отсутствуют какие-либо пермутации (рис. 5б). При этом следует скорее говорить не о спектральности, а о прореженности распределения.

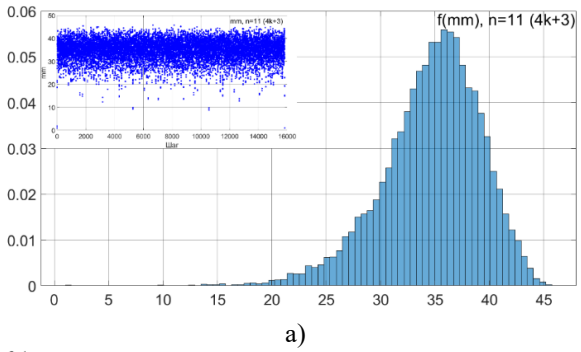


а)

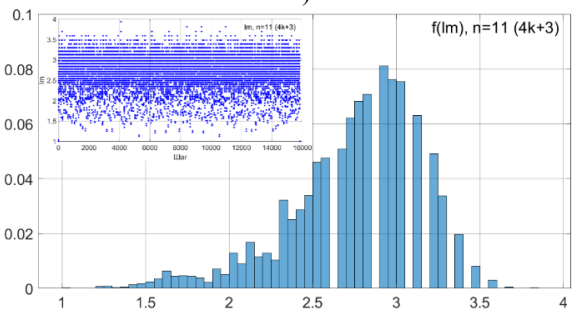


б)

Рис. 4. Перемешанности последовательных пермутаций матрицы 8×8 ($n = 4k$) в схеме «sin»: а) матричная перемешанность; б) линейная перемешанность. Звездочкой обозначено значение метрики для блочной перестановки



а)



б)

Рис. 5. Распределения последовательных конфигураций матрицы 11×11 ($n = 4k + 3$) в схеме «sin» по значениям метрик: а) матричной; б) лиейной

VI. Заключение

На начальном участке натурального ряда компьютерное исследование схемы «sin» показало меньший период с базовым алгоритмом. Данное утверждение, однако, требует теоретического доказательства полученных эмпирических формул, в ходе которого также будут проверены утверждения о свойствах периода схемы:

- в схеме «sin» период зависит от класса вычета n по модулю 4: если остаток равен 0, то закон роста линейный, если равен 2, квадратичный.
- В схеме «sin» выделяются сходные классы вычетов с остатком 1 и 3, где период подчинен сложному рекуррентному закону о числах ДИТ. Эти классы отличаются только начальным значением для рекурсии

Идея базовой схемы изначально заключалась в конструкции наиболее простого КА, способного генерировать особые пермутации матриц. На данный момент нам не удалось за счет модификации схемы «1:1» повысить длину периода алгоритма, что было бы полезно с точки зрения криптографии. Однако из этой неудачи возникает вопрос для серьезного математического исследования:

Какое наибольшее число пермутаций матрицы порядка n способен порождать до своего останова детерминированный клеточный автомат, заданный в поле $n \times n$?

Оценка снизу следует из периода базовой схемы – $exp(2n)/n$. Оценка сверху очевидна – $(n^2)! \sim \left(\frac{n^2}{e}\right)^{n^2}$. Проблема алгоритмическая и связана с ограничением клеточного автомата, накладываемого на реализацию алгоритма.

Поддержка

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №20-07-00409.

Литература

[1] Fog A. Pseudo-Random Number Generators for Vector Processors and Multicore Processors. // Journal of modern applied statistical methods. 2015. №14(1) P. 308–334.

[2] Agrawal V, Bushnell M. Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits. / Springer Science & Business Media. 2004. 712 P.

[3] Ключарёв П.Г. Построение случайных графов, предназначенных для применения в криптографических алгоритмах, основанных на обобщенных клеточных автоматах. // Математика и математическое моделирование. 2017. №3. С. 77–90.

[4] Chen, Y. L., Lambooij, E., Mennink, B. How to build pseudorandom functions from public random permutations. // Annual International Cryptology Conference. Springer, Cham. 2019. P. 266–293.

[5] Жуков Д. А. О порядке одной перестановки на элементах квадратной матрицы // Труды десятой международной научно-технической конференции «Безопасные информационные технологии». М.: Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет). 2019. С. 143–145.

[6] Weichuang G., Junqin Z., Ruisong Y. A chaos-based pseudorandom permutation and bilateral diffusion scheme for image encryption // International Journal of Image, Graphics and Signal Processing. 2014. V. 6. №11. P. 50–61.

[7] Матюшкин И. В., Заплетина М.А. Клеточно-автоматный вычислительный параллелизм элементарных матричных операций // Проблемы разработки перспективных микро - и наноэлектронных систем (МЭС). 2018. № 3. С. 132–137.

[8] Dzedzej A. et al. Efficient enumeration of three-state two-dimensional
number-conserving cellular automata // Information and Computation. 2020. V. 270. URL: https://doi.org/10.1016/j.ic.2020.104534 (access date: 09.07.2020)

[9] Adak S., Das S., Mukherjee S. Reachability problem in non-uniform cellular automata // Information Sciences. 2021. V. 543. P. 72–84.

[10] Kyparissas N., Dollas A. Large-scale Cellular Automata on FPGAs: A New Generic Architecture and a Framework // ACM Transactions on Reconfigurable Technology and Systems. 2020. V. 14. № 1. P. 1–32.

[11] Матюшкин И.В., Кожевников В.С. Клеточно-автоматные алгоритмы пермутации матриц // ТРУДЫ МФТИ. 2019. Т. 11, № 1. С. 39–52.

[12] Матюшкин И.В., Рубис П.Д. Четыре клеточно-автоматных алгоритма пермутаций матриц // Математика и математическое моделирование. 2020. №4. С. 1–51.

Cellular-automaton Algorithm of Matrices Permutation with an Oscillatory Scheme of Element Shift

I.V. Matyushkin , P.D. Rubis

National Research University of Electronic Technology, Moscow, Zelenograd, Russia,
rubisiay@gmail.com

***Abstract* — Numerical calculation uses to describe the operation of matrix permutation algorithm based on cyclic shifts of rows and columns. Scope of the algorithm is the group generation of pseudorandom numbers. Algorithm is formulated in terms of cellular automata (CA). Results of numerical calculation are given, primarily for the repetition period of the original matrix. For even-order matrices, the period is linear or quadratic. In case of matrices with odd order, the dependence of the algorithm period from the order of the matrix is recurrent, obtained in the form of list of rules, and does not exceed the Landau function. Visualization of individual paths of matrix elements is presented on the expanded field of the CA. As a parameter of the global dynamics of automata, two "mixing metrics" are analyzed on the permutations of the matrix (compared to the initial one). The behavior of these metrics is shown in graphs and histograms (conditional density distribution) describing how often the permutation period with the specified interval of metrics occurs.**

***Keywords* — cellular automata, permutation, random numbers, cryptography, metric**

REFERENCES

[1] Fog A. Pseudo-Random Number Generators for Vector Processors and Multicore Processors. // Journal of modern applied statistical methods. 2015. №14(1) P. 308–334.

[2] Agrawal V, Bushnell M. Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits. / Springer Science & Business Media. 2004. 712 P.

[3] Klyucharyov P.G. Postroenie sluchajnyh grafov, prednaznachennyh dlya primeneniya v kriptograficheskih algoritmah, osnovannyh na obobshchennyh kletochnyh avtomatah. (Random Graph Construction for Cryptographic Applications) // Matematika i matematicheskoe modelirovanie. 2017. №3. S. 77-90.

[4] Chen, Y. L., Lambooij, E., Mennink, B. How to build pseudorandom functions from public random permutations. // Annual International Cryptology Conference. Springer, Cham. 2019. P. 266–293.

[5] Zhukov D. A. O poryadke odnoj perestanovki na elementah kvadratnoj matricy (On the order of some matrix permutation) // Trudy desyatoj mezhdunarodnoj nauchno-tekhnicheskoj konferencii «Bezopasnye informacionnye tekhnologii». M.: Moskovskij gosudarstvennyj tekhnicheskij universitet imeni N.E. Baumana (nacional'nyj issledovatel'skij universitet). 2019. S. 143-145.

[6] Weichuang G., Junqin Z., Ruisong Y. A chaos-based pseudorandom permutation and bilateral diffusion scheme for image encryption // International Journal of Image, Graphics and Signal Processing. 2014. V. 6. №11. P. 50–61.

[7] Matyushkin I. V., Zapletina M.A. Kletochno-avtomatnyj vychislitel'nyj parallelizm elementarnyh matrichnyh operacij (Cellular-automaton computational parallelism of elementary matrix operations) // Problemy razrabotki perspektivnyh mikro - i nanoelektronnyh sistem (MES). 2018. № 3. S. 132–137.

[8] Dzedzej A. et al. Efficient enumeration of three-state two-dimensional number-conserving cellular automata // Information and Computation. 2020. V. 270. URL: https://doi.org/10.1016/j.ic.2020.104534 (access date: 09.07.2020)

[9] Adak S., Das S., Mukherjee S. Reachability problem in non-uniform cellular automata // Information Sciences. 2021. V. 543. P. 72–84.

[10] Kyparissas N., Dollas A. Large-scale Cellular Automata on FPGAs: A New Generic Architecture and a Framework // ACM Transactions on Reconfigurable Technology and Systems. 2020. V. 14. № 1. P. 1–32.

[11] Matyushkin I.V., Kozhevnikov V.S Kletochno-avtomatnye algoritmy permutacii matric (Cellular automata algorithms for matrix permutations) // TRUDY MFTI. 2019. T. 11, № 1. S. 39–52.

[12] Matyushkin I.V., Rubis P.D. Chetyre kletochno-avtomatnyh algoritma permutacij matric (Four cellular automata algorithms for matrix permutation) // Matematika i matematicheskoe modelirovanie. 2020. №3. S. 1–51.