

СРАВНЕНИЕ СПОСОБОВ ЗАПОЛНЕНИЯ ДИАГОНАЛЬНЫХ ЛАТИНСКИХ КВАДРАТОВ ДЛЯ ИТЕРАЦИОННОЙ РЕАЛИЗАЦИИ МЕТОДА ПОЛНОГО ПЕРЕБОРА

Аннотация. В работе приведены описания различных способов заполнения диагональных латинских квадратов, примеры получившихся на их основе порядков заполнения диагональных латинских квадратов, их сравнение по темпу генерации диагональных латинских квадратов.

Ключевые слова: диагональные латинские квадраты, добровольные распределенные вычисления, BOINC, порядок заполнения элементов диагональных латинских квадратов, итерационная реализация метода полного перебора.

Латинским квадратом (рис. 1) (ЛК) N -ого порядка называется квадратная матрица $L = \|l_{ij}\|, i, j = \overline{1, N}$, каждая ячейка которой заполнена элементом некоторого множества $U, \forall l_{ij} \in U, |U| = N$, таким образом, что в каждой строке и в каждом столбце элемент множества U встречается ровно один раз $\forall l_{ij}, l_{ik}, j \neq k: (l_{ij} \neq l_{ik}) \wedge (l_{ji} \neq l_{ki}), i, j, k = \overline{1, N}$ [1]. Обычно элементами множества U являются целые числа $0, 1, 2, \dots, N - 1$.

4	7	2	0	6	5	1	3
7	4	0	2	5	6	3	1
2	0	4	7	1	3	6	5
0	2	7	4	3	1	5	6
6	5	1	3	4	7	2	0
5	6	3	1	7	4	0	2
1	3	6	5	2	0	4	7
3	1	5	6	0	2	7	4

а)

0	1	2	3	4	5	6	7
1	0	3	2	5	4	7	6
2	3	0	1	6	7	4	5
3	2	1	0	7	6	5	4
4	5	6	7	0	1	2	3
5	4	7	6	1	0	3	2
6	7	4	5	2	3	0	1
7	6	5	4	3	2	1	0

б)

Рисунок 1 – Примеры денормализованного (а) и нормализованного (б) ЛК

Для диагональных латинских квадратов (рис. 2) (ДЛК) вводится дополнительное ограничение на уникальность элементов на главной и побочной диагоналях $\forall l_{ii}, l_{jj}, i \neq j: (l_{ii} \neq l_{jj}) \wedge (l_{i, N-i+1} \neq l_{j, N-j+1}), i, j = \overline{1, N}$.

7	5	0	2	3	1	6	4
6	4	3	1	0	2	7	5
2	0	5	7	4	6	1	3
1	3	4	6	5	7	2	0
4	6	1	3	2	0	5	7
5	7	2	0	1	3	4	6
3	1	6	4	7	5	0	2
0	2	7	5	6	4	3	1

а)

0	1	2	3	4	5	6	7
6	7	4	5	2	3	0	1
3	2	1	0	7	6	5	4
5	4	7	6	1	0	3	2
7	6	5	4	3	2	1	0
1	0	3	2	5	4	7	6
4	5	6	7	0	1	2	3
2	3	0	1	6	7	4	5

б)

Рисунок 2 – Примеры денормализованного ДЛК (а) и ДЛК с упорядоченной первой строкой (б)

Нормализованным ЛК называется ЛК, у которого первая строка и первый столбец упорядочены по возрастанию (рис. 1б). Любой ЛК можно привести к нормализованному виду путем перенумерации элементов множества U и последующей перестановки строк. Несложно посчитать, что $N! \times (N - 1)!$ денормализованному ЛК соответствует один и тот же нормализованный ЛК: множитель $N!$ – соответствует числу возможных перенумераций множества U , а множитель $(N - 1)!$ – числу перестановок строк квадрата (со второй по N -ую).

Для ДЛК описанная выше процедура нормализации может нарушить условие уникальности диагональных элементов за счёт перестановки строк. При этом упорядочивание по возрастанию первой строки ДЛК путем перестановки элементов множества U не оказывает влияние на данное условие. Имеет смысл генерировать только такие ДЛК с фиксированной первой строкой (рис. 2б), потому что, имея один ДЛК данного вида, можно тривиально получить остальные $N! - 1$ ДЛК, входящие с ним в один класс эквивалентности.

Генерацию ДЛК можно осуществлять различными способами: SAT-подходом [2], эвристическими методами: случайного, взвешенного случайного перебора, муравьиной колонии [3], методом полного перебора [4].

При генерации ДЛК SAT-подходом данная задача сводится к SAT-задаче. Решение такой задачи заключается в нахождении (если существует) для некоторой булевой формулы такого набора значений булевых переменных, при которых она становится истинной. Таким образом, представив все возможные значения каждой ячейки ДЛК в виде отдельной булевой переменной, можно построить КНФ, описывающую ограничения, накладываемые на строки, столбцы, главную и побочную диагонали ДЛК. Каждое решение SAT-задачи для данной КНФ будет представлять собой некоторый ДЛК.

Обычно алгоритмы решения SAT-задач находят только одно единственное решение. Для поиска заданного количества решений или всех возможных решений в конец булевой формулы необходимо добавить терм, запрещающий повторное нахождение уже ранее полученного решения. Вследствие данного факта темп генерации ДЛК данным способом уменьшается с ростом числа уже найденных ДЛК.

Эвристические методы чаще всего используются для решения задач, у которых существует множество решений, но требуется получить наилучшее, оптимальное или близкое к нему, приближенное, когда поиск оптимального решения затруднен ввиду потребности в огромных затратах вычислительных мощностей. На первый взгляд задача генерации ДЛК не относится к таковым, но это можно исправить, введя в качестве оптимизирующего параметра количество нарушений ограничений ДЛК. Тогда полученные эвристическими методами решения, у которых данный параметр равен нулю, будут являться правильно заполненными ДЛК. К сожалению, эвристические методы не

гарантируют получение всех возможных решений, поэтому для задачи перечисления ДЛК не подходят.

Метод полного перебора, наоборот, рассматривает все возможные потенциальные решения и выбирает только подходящие. Известны две реализации метода полного перебора: рекуррентная и итерационная. Программная реализация рекуррентной версии алгоритма представляет собой подпрограмму, которая явно вызывает сама себя. В итерационной же версии рекурсия заменяется на группу вложенных циклов. Рекуррентная реализация алгоритма является более универсальной. Она позволяет генерировать ДЛК любого размера при минимальных изменениях программного кода, а также модифицировать порядок заполнения ячеек ДЛК по ходу работы. Итерационная реализация лишена данных преимуществ, так как размерность ДЛК и порядок заполнения его ячеек задаются жестко при компиляции программы, но зато обладает более высоким темпом генерации ДЛК благодаря отсутствию команд вызова подпрограммы, сохранения и восстановления локальных переменных подпрограммы с помощью стека, возврата из подпрограммы, на которые тратятся ресурсы процессора. Далее рассматривается именно итерационная реализация метода полного перебора.

Одним из параметров, влияющих на темп генерации ДЛК, является порядок заполнения его элементов. В данной работе была предпринята попытка найти порядок заполнения ДЛК, характеризующийся наивысшим темпом генерации ДЛК.

В ходе исследования было опробовано три способа нерегулярного заполнения ДЛК. Все они имеют общий принцип функционирования. На каждом шаге алгоритма, определяющего порядок заполнения ДЛК, выбирается ячейка, у которой значение некоторого критерия V имеет максимальное значение $V \rightarrow \max$. В связи с тем, что диагональное заполнение имеет наивысший темп генерации ДЛК, для всех способов предварительно заполняются главная и побочная диагонали. При этом единственная незаполненная ячейка в строке, столбце, главной или побочной диагоналях заполняется вне очереди (рис 4д, 4и). Значение такой ячейки можно легко вычислить, потому что ДЛК является частным случаем магического квадрата. Предположим, что в i -строке некоторого ДЛК заполнено $N - 1$ первых элементов, тогда $l_{iN} = R - \sum_{j=1}^{N-1} l_{ij}$, где R – магическое число, при $U = \{0, 1, \dots, N - 1\}$ $R = \frac{N \cdot (N-1)}{2}$. Аналогичные выражения можно получить для столбца и диагоналей. Способы различаются критериями выбора очередной заполняемой ячейки.

В первом способе в качестве критерия выбора очередной заполняемой ячейки ДЛК выступает нижняя оценка числа ограничений $V = V_{ij}^- \rightarrow \max$ (рис. 3), накладываемых заполненными ячейками без учета их фактических значений (т.е. рассматривается только факт заполнения ячейки в связи с использованием итерационной реализации метода полного перебора) на ещё незаполненную, а во втором – верхняя оценка числа ограничений $V = V_{ij}^+ \rightarrow \max$ (рис. 4). Пусть r_i – число заполненных ячеек в i -ой строке; c_j – число заполненных ячеек в i -ом столбце; d_1 – число заполненных ячеек на главной диагонали; d_2 – число заполненных ячеек на побочной диагонали. Тогда $V_{ij}^- = \max(r_i, c_j, d_1 \cdot g(i, j), d_2 \cdot h(i, j))$ и $V_{ij}^+ = r_i + c_j + d_1 \cdot g(i, j) + d_2 \cdot h(i, j)$, где $g(i, j) = \begin{cases} 1, & \text{если } i = j; \\ 0, & \text{если } i \neq j. \end{cases}$ $h(i, j) = \begin{cases} 1, & \text{если } i = N - j + 1; \\ 0, & \text{если } i \neq N - j + 1. \end{cases}$

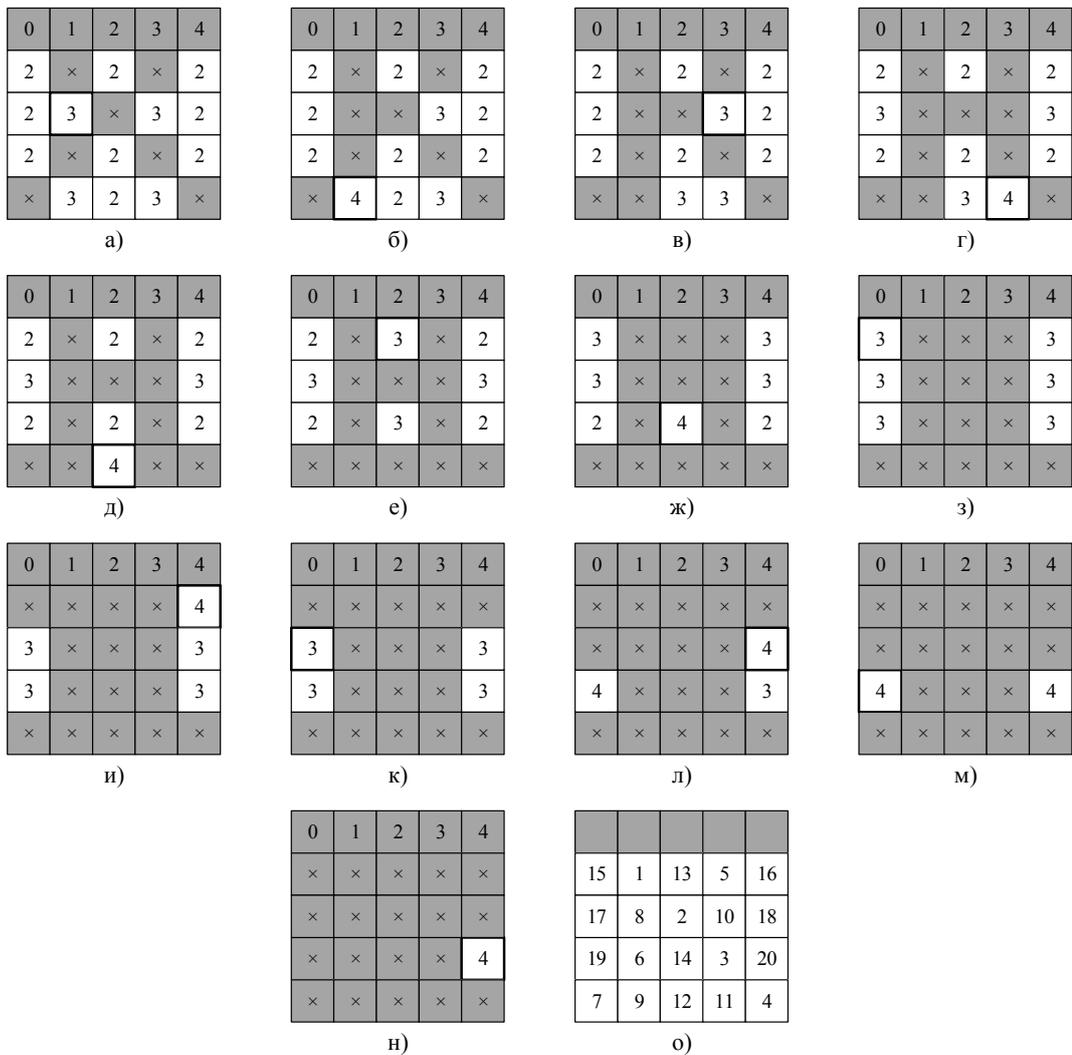


Рисунок 3 – Порядок заполнения ДЛК порядка 5 с помощью нижней оценки числа ограничений V_{ij}^- : серым цветом отмечены ранее заполненные ячейки, цифры на сером фоне обозначают фактические значения заполненные ячеек, кресты – факт их заполнения, цифры на белом фоне соответствуют значениям критерия V_{ij}^- ;

- а) – предварительное диагональное заполнение ДЛК,
- б), в), ..., н) – заполнение недиагональных элементов ДЛК,
- о) – результирующий порядок заполнения ДЛК.

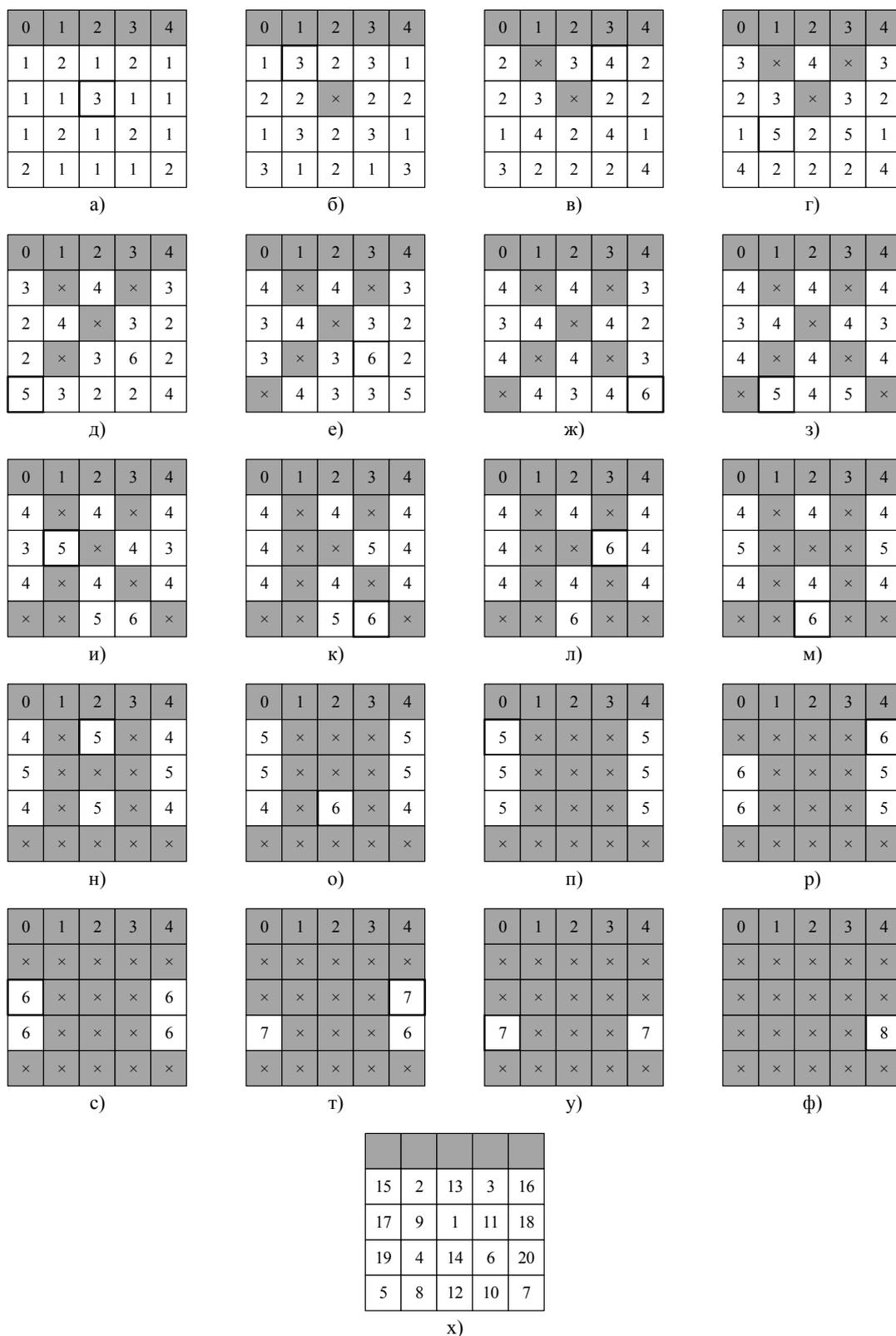


Рисунок 4 – Порядок заполнения ДЛК порядка 5 с помощью верхней оценки числа ограничений V_{ij}^+ : цифры на белом фоне соответствуют значениям критерия V_{ij}^+ ;
 а), б), ..., ж) – диагональное заполнение ДЛК,
 з), и), ..., ф) – заполнение недиагональных элементов ДЛК,
 х) – результирующий порядок заполнения ДЛК.

В третьем способе порядок заполнения ячеек ДЛК зависит от математического ожидания количества ограничений, накладываемых на ещё незаполненную ячейку,

$V = M_{ij} \rightarrow \max$ (рис. 5). Для определения данного параметра для каждой ячейки рассчитывалась вероятность $P_{ij}(C)$ того, что ячейка не сможет принять C значений, а затем вычислялось математическое ожидание $M_{ij} = \sum_{k=1}^N P_{ij}(k) \cdot k$. Расчет вероятностей осуществлялся программным способом путем генерации для i -ой строки всех возможных сочетаний из N по r_i ; для j -ого столбца – сочетаний из $N - 1$ по $c_j - 1$ и добавления элемента со значением j (для учета фактических значений первой строки); для главной диагонали (при $i = j$) – сочетаний из $N - 1$ по $d_1 - 1$ и добавления элемента со значением 0 ; для побочной диагонали (при $i = N - j + 1$) – сочетаний из $N - 1$ по $d_2 - 1$ и добавления элемента со значением $N - 1$. Полученные сочетания всеми возможными способами объединялись в группы, для каждой группы подсчитывалось количество заполненных ячеек C , после чего для каждого из значений $C \in [1, N]$ определялась вероятность, как $P_{ij}(C) = \frac{m}{n}$, где m – количество групп с C заполненными ячейками, n – общее количество групп.

0	1	2	3	4
1	2	1	2	1
1	1	3	1	1
1	2	1	2	1
2	1	1	1	2

а)

0	1	2	3	4
1	2,8	2	2,8	1
1,8	1,8	×	1,8	1,8
1	2,8	2	2,8	1
2,8	1	2	1	2,8

б)

0	1	2	3	4
1,8	×	2,6	3,2	1,8
1,8	2,6	×	1,8	1,8
1	3,3	2	3,5	1
2,8	2	2	1	3,5

в)

0	1	2	3	4
1,8	×	2,6	3,7	1,8
1,8	2,6	×	2,6	1,8
1,8	3,7	2,6	×	1,8
2,8	2	2	2	4,3

г)

0	1	2	3	4
1,8	×	2,6	3,7	2,6
1,8	2,6	×	2,6	2,6
1,8	3,7	2,6	×	2,6
3,2	2,6	2,6	2,6	×

д)

0	1	2	3	4
2,6	×	3,2	×	3,2
1,8	2,6	×	3,4	2,6
1,8	4,1	2,6	×	2,6
3,8	2,6	2,6	3,4	×

е)

0	1	2	3	4
2,6	×	3,2	×	3,2
1,8	3,4	×	3,4	2,6
2,6	×	3,2	×	3,2
4,4	3,4	2,6	3,4	×

ж)

0	1	2	3	4
3,2	×	3,2	×	3,2
2,6	3,4	×	3,4	2,6
3,2	×	3,2	×	3,2
×	3,8	3,2	3,8	×

з)

0	1	2	3	4
3,2	×	3,2	×	3,2
2,6	4,2	×	3,4	2,6
3,2	×	3,2	×	3,2
×	×	3,8	4,2	×

и)

0	1	2	3	4
3,2	×	3,2	×	3,2
3,2	×	×	3,8	3,2
3,2	×	3,2	×	3,2
×	×	3,8	4,2	×

к)

0	1	2	3	4
3,2	×	3,2	×	3,2
3,2	×	×	4,4	3,2
3,2	×	3,2	×	3,2
×	×	4,4	×	×

л)

0	1	2	3	4
3,2	×	3,2	×	3,2
3,8	×	×	×	3,8
3,2	×	3,2	×	3,2
×	×	4,4	×	×

м)

0	1	2	3	4
3,2	×	3,8	×	3,2
3,8	×	×	×	3,8
3,2	×	3,8	×	3,2
×	×	×	×	×

н)

0	1	2	3	4
3,8	×	×	×	3,8
3,8	×	×	×	3,8
3,2	×	4,4	×	3,2
×	×	×	×	×

о)

0	1	2	3	4
3,8	×	×	×	3,8
3,8	×	×	×	3,8
3,8	×	×	×	3,8
×	×	×	×	×

п)

0	1	2	3	4
×	×	×	×	4,4
4,2	×	×	×	3,8
4,2	×	×	×	3,8
×	×	×	×	×

р)

0	1	2	3	4
×	×	×	×	×
4,2	×	×	×	4,2
4,2	×	×	×	4,2
×	×	×	×	×

с)

0	1	2	3	4
×	×	×	×	×
×	×	×	×	4,6
4,6	×	×	×	4,2
×	×	×	×	×

т)

0	1	2	3	4
×	×	×	×	×
×	×	×	×	×
4,6	×	×	×	4,6
×	×	×	×	×

у)

0	1	2	3	4
×	×	×	×	×
×	×	×	×	×
×	×	×	×	4,8
×	×	×	×	×

ф)

15	2	13	5	16
17	9	1	11	18
19	6	14	3	20
7	8	12	10	4

х)

Рисунок 4 – Порядок заполнения ДЛК порядка 5 с помощью математического ожидания числа ограничений M_{ij} : цифры на белом фоне соответствуют значениям критерия M_{ij} ,

округленным до десятых:

а), б), ..., ж) – диагональное заполнение ДЛК,

з), и), ..., ф) – заполнение недиагональных элементов ДЛК,

х) – результирующий порядок заполнения ДЛК.

Стоит отметить, что диагональное заполнение для способов, использующих верхнюю оценку числа ограничений V_{ij}^+ и математическое ожидание числа ограничений M_{ij} , диагональное заполнение получается автоматически (рис. 4а – 4ж; рис. 5а – 5ж).

С использованием данных стратегий был вычислен темп генерации для некоторых частичных заполнений ДЛК порядка 9. Вычислительный эксперимент производился на одном ядре процессора Intel Core i5-10210U с использованием 64-битного компилятора C++ Microsoft Visual Studio 2019. Для способов с использованием критериев V_{ij}^- (рис. 6а) и V_{ij}^+ (рис. 6б) темп генерации ДЛК в среднем одинаков и составляет 5 200 000 ДЛК/с, для способа с использованием критерия M_{ij} (рис. 6в) – 5 000 000 ДЛК/с. Для сравнения темп генерации ДЛК с построчным заполнением недиагональных элементов (рис. 7а) равняется 2 000 000 ДЛК/с, с заполнением по столбцам – 1 000 000 ДЛК/с (рис. 7б), с чередованием строк и столбцов (рис. 7в) – 650 000 ДЛК/с, со случайным заполнением – 4 000 ДЛК/с (рис. 7г). Наивысший темп известной реализации составляет 5 600 000 ДЛК/с [5].

22	1	16	23	24	25	26	9	27
53	54	2	28	49	33	10	55	56
57	58	17	3	50	11	42	59	60
61	64	18	29	4	34	43	65	66
62	67	19	12	51	5	44	69	70
63	68	13	30	52	35	6	71	72
38	14	20	31	39	36	40	7	41
15	46	21	32	47	37	45	48	8

а)

20	2	16	17	21	18	19	3	22
25	26	6	23	27	24	7	28	29
55	56	57	10	53	11	58	59	60
61	63	65	45	1	47	67	69	70
62	64	66	12	54	14	68	71	72
32	33	8	30	34	31	9	35	36
39	4	42	37	40	38	43	5	41
13	49	50	44	48	46	51	52	15

б)

25	2	16	22	26	23	24	9	27
53	54	3	44	49	47	10	55	56
57	58	17	4	50	11	28	59	60
61	64	21	43	1	46	32	65	66
62	67	18	12	51	5	29	69	70
63	68	13	45	52	48	6	71	72
35	14	19	33	36	34	30	7	37
15	41	20	38	40	39	31	42	8

в)

Рисунок 6 – Порядок заполнения ячеек ДЛК порядка 9:

- а) с помощью нижней оценки числа ограничений V_{ij}^- ;
- б) с помощью верхней оценки числа ограничений V_{ij}^+ ;
- в) с помощью математического ожидания числа ограничений M_{ij} .

16	1	17	18	19	20	21	9	22
23	24	2	25	26	27	10	28	29
30	31	32	3	33	11	34	35	36
37	38	39	40	4	41	42	43	44
45	46	47	12	48	5	49	50	51
52	54	13	56	57	58	6	59	60
53	14	63	65	67	69	70	7	61
15	55	64	66	68	71	72	62	8

а)

61	1	35	43	22	67	66	9	41
57	53	2	18	23	34	10	58	56
38	24	55	3	46	11	54	51	28
21	52	26	42	4	45	29	63	64
60	50	71	12	40	5	59	62	70
30	20	13	25	32	31	6	16	19
44	14	17	37	39	68	48	7	69
15	27	72	33	47	36	65	49	8

б)

16	1	29	35	41	48	54	9	55
17	23	2	36	42	49	10	63	64
18	24	30	3	43	11	56	65	66
19	25	31	37	4	50	57	67	68
20	26	32	12	44	5	58	69	70
21	27	13	38	45	51	6	71	72
22	14	33	39	46	52	59	7	60
15	28	34	40	47	53	61	62	8

в)

16	1	23	24	25	26	27	9	28
17	29	2	35	36	37	10	38	39
18	30	40	3	45	11	46	47	48
19	31	41	49	4	53	54	55	56
20	32	42	12	57	5	61	62	63
21	33	13	50	58	64	6	67	68
22	14	43	51	59	65	70	7	69
15	34	44	52	60	66	71	72	8

г)

Рисунок 7 – Порядок заполнения ДЛК порядка 9:

а) по строкам;

б) по столбцам;

в) с чередованием строк и столбцов;

г) случайное.

Программную реализацию способов заполнения ДЛК, полученную в ходе данной работы, планируется использовать в дальнейшем для решения задачи перечисления ДЛК с использованием грид-систем.

Библиографический список

1. Colbourn C.J., Dinitz J.H. Handbook of Combinatorial Designs, Second Edition. Chapman & Hall/CRC, 2006. 1016 p.
2. Заикин О.С., Кочемазов С.Е. Поиск пар ортогональных диагональных латинских квадратов порядка 10 в проекте добровольных распределенных вычислений SAT@home // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2015. Т. 4, № 3. С. 95–108.
3. Ватутин Э.И., Журавлев А.Д., Заикин О.С., Титов В.С. Особенности использования взвешивающих эвристик в задаче поиска диагональных латинских квадратов // Известия ЮЗГУ. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2015. № 3 (16). С. 18–30.

4. Вату́тин Э.И., Журавле́в А.Д., Заики́н О.С., Титов В.С. Учет алгоритмических особенностей задачи при генерации диагональных латинских квадратов // Известия ЮЗГУ, 2016. № 2 (65). С. 46–59.

5. Kochemazov S., Zaikin O., Vatutin E., Belyshev A. Enumerating Diagonal Latin Squares of Order Up to 9 // Journal of Integer Sequences. Vol. 23. Iss. 1. 2020. Article 20.1.2.