УДК 681.3
Ватутин Э.И.[1], Заикин О.С.[2], Журавлев А.Д.[3], Манзюк М.О.[3], Титов В.С.[1]
e-mail: evatutin@rambler.ru
[1] Юго-Западный государственный университет
[2] Институт динамики систем и теории управления им. В.М. Матросова СО РАН
[3] Интернет-портал BOINC.RU

О ВЛИЯНИИ ПОРЯДКА ЗАПОЛНЕНИЯ ЯЧЕЕК НА ТЕМП ГЕНЕРАЦИИ ДИАГОНАЛЬНЫХ ЛАТИНСКИХ КВАДРАТОВ

Рассмотрена возможность использования латинских квадратов в диагностических целях. Обоснована необходимость установки в диагональном положении при одиночном использовании и в ортогональном – при использовании в паре.

Одной из интересных комбинаторных структур в области дискретной математики являются латинские квадраты (ЛК) заданного порядка N [1]. ЛК представляет собой квадратную таблицу $A=\left\|a_{ij}\right\|,\ i,j=\overline{1,N}$, заполненную элементами некоторого множества $U=\{u_1,u_2,...,u_N\}$, $|U|=N$, $\forall a_{ij}\in U$ (в простейшем случае в качестве элементов u_i выступают целые числа 0, 1, …, $N-1$), причем в каждой строке и столбце не должно быть повторений элементов: $\forall a_{ij},a_{ik},j\neq k:\left(a_{ij}\neq a_{ik}\right)\wedge\left(a_{ji}\neq a_{ki}\right),i,j,k=\overline{1,N}$ (фактически строки и столбцы ЛК представляют собой перестановки элементов множества U). ЛК находят применение при планировании эксперимента, в криптографии, имеют тесную связь с теорией групп [2]. Частным случаем ЛК являются диагональные латинские квадраты (ДЛК), для которых вводится дополнительное ограничение на отсутствие повторяющихся элементов на главной и побочной диагоналях: $\forall a_{ii},a_{jj},i,j=\overline{1,N},i\neq j:\left(a_{ii}\neq a_{jj}\right)\wedge\left(a_{i,N-i+1}\neq a_{j,N-j}+1\right)$.

Каждому ЛК можно поставить в соответствие нормализованный (или редуцированный) ЛК, у которого первая строка и первый столбец заполнены в соответствии с порядком, заданном на множестве U. Отметим, что любой ДЛК элементарными преобразованиями может быть сведен к ДЛК, в котором первая строка равна любому из $N!$ возможных вариантов (перестановок), а указанное множество квадратов образуют класс изоморфизма. Это осуществляется по аналогии с процедурой нормализации для ЛК, в которой требуется упорядочить первую строку и первый столбец по возрастанию. Далее, не умаляя общности, будем рассматриваем ДЛК с фиксированной первой строкой, равной «0, 1, …, $N-1$». Процесс получения ЛК и ДЛК имеет ярко выраженный комбинаторный характер, а благодаря указанной особенности при формировании множества ДЛК можно исключить из переборного процесса первую строку, уменьшая тем самым размерность задачи.

Одной из категорий задач, имеющих место в дискретной математике и дискретной комбинаторной оптимизации, являются задачи на пересчет заданных комбинаторных объектов, к которым относятся ЛК, ДЛК и многие другие объекты [3, 4]. Аналитические выражения для числа ЛК и ДЛК заданного порядка N неизвестны, известны лишь верхние и нижние оценки, поэтому для определения их точного числа на практике необходима организация переборного вычислительного процесса. Стремительный рост числа квадратов с ростом порядка N не позволяет напрямую подсчитать их число даже для сравнительно небольших N, в том числе с использованием параллельных вычислительных средств, таких как вычислительные кластеры, суперкомпьютеры и/или грид-системы, что вынуждает выявлять алгоритмические особенности задачи. Указанная переборная задача построения квадратов и подсчета их числа является слабо связанной, что допускает ее эффективную реализацию как на вычислительных системах с быстрым интерконнектом, так и на грид-системах, где возможность обмена данными между подзадачами невозможна или существенно затруднена.

Число ЛК известно до $N \leq 11$ [5] (последовательности A000315, A000479 и A002860 в OEIS [6]), в то время как аналогичные оценки для ДЛК не известны. Для их построения необходима эффективная программная реализация, которая обеспечивает получение всех ДЛК заданного порядка с минимальными затратами на получение очередного ДЛК.

Для генерации ДЛК могут быть применены различные подходы. В [4], например, эта задача решалась при помощи SAT-подхода, т.е. исходная задача была сведена к проблеме булевой выполнимости и решалась с задействованием SAT-решателей. В работе [7] для получения ДЛК были опробованы методы полного перебора, случайного, взвешенного случайного перебора и метод муравьиной колонии, а исходная задача на существование решения сводилась к задаче дискретной комбинаторной оптимизации. Опробованные эвристические методы пригодны для получения ДЛК с более-менее равномерным распределением в дискретном пространстве, образованном возможными значениями элементов квадрата, однако для задачи подсчета ДЛК они не подходят, т.к. не гарантируют получения всех возможных решений и не спасают от возможных повторов, в отличие от метода полного перебора. Средний темп получения ДЛК порядка 10 для метода полного перебора в указанной задаче составлял 6 ДЛК/с. В работе [8] было предложено использовать диагональное заполнение элементов ДЛК (сперва заполняются диагонали, затем – построчно все остальные элементы), что обеспечило повышение темпа генерации приблизительно до 5000 ДЛК/с. В работе [9] был выполнен ряд алгоритмических и высокоуровневых оптимизаций метода полного перебора на базе раннего заполнения элементов формируемого квадрата, для которых на текущем шаге возможно использование единственного элемента из множества U без нарушения ограничений, раннего отсечения неперспективных квадратов и раннего заполнения элементов квадрата с минимальной мощностью множества возможных элементов $S_{ij} \subseteq U$, что при рекуррентной программной реализации позволило увеличить темп генерации приблизительно до 200 000 ДЛК/с. Уход от использования рекурсии к итеративной реализации с N^2 вложенными циклами позволил дополнительно увеличить темп до 340 000 ДЛК/с.

Достоинством рекуррентной реализации является возможность изменения порядка заполнения недиагональных ячеек по критерию $\left|S_{ij}\right| \to \min$, однако наличие команд CALL и RET в коде программы снижает скоростные характеристики программной реализации ввиду появления получения косвенных возвратов. Итеративная реализация лишена подобного недостатка, однако в ней порядок следования циклов задается жестко при компиляции и не может быть изменен, что с легкостью решается с использованием механизма передачи параметров в рекуррентной реализации. При этом итеративная реализация обладает в 1,7 раза более высоким темпом, а значит является более перспективной для дальнейшей доработки.

В данной работе была осуществлена попытка нахождения такого порядка следования вложенных циклов, который характеризуется еще большим темпом генерации. В ходе исследования были опробованы различные порядки заполнения элементов ДЛК. Наивысший темп был получен для диагонального заполнения, последующая вариация заполнения недиагональных ячеек по строкам, по столбцам, с чередованием строк/столбцов или в шахматном порядке не привела к увеличению полученного в [9] темпа. Вместо этого вычислительный эксперимент показал, что темп генерации можно повысить, если перейти к нерегулярному порядку заполнения элементов. Для этого в качестве критерия, определяющего координаты следующего заполняемого элемента ДЛК, было выбрано число ограничений $R_{ij} \to \max$, накладываемое на еще не заполненный элемент со стороны уже заполненных (рассматривается сам факт заполнения элемента, а не его конкретное значение, как при рекуррентной реализации). На рисунке изображен пример полученного порядка заполнения элементов ДЛК порядка 5.

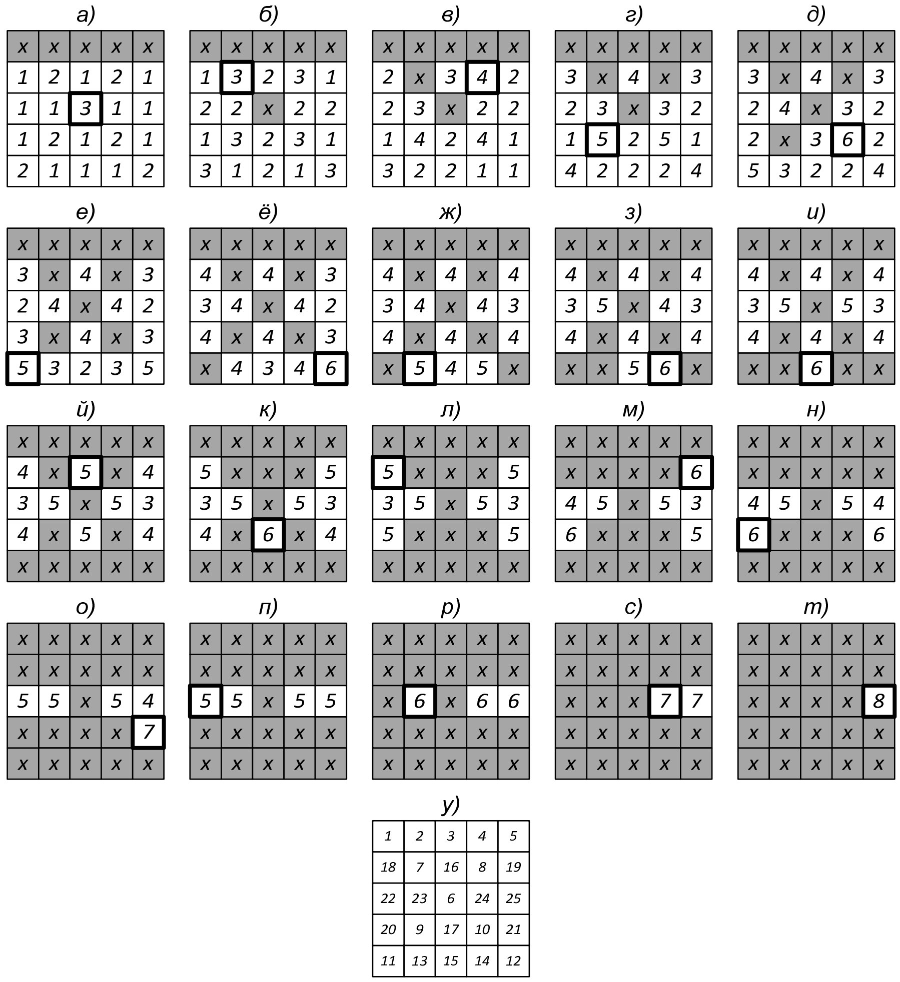

Рис. Порядок заполнения элементов ДЛК порядка 5 отталкиваясь от максимума числа ограничений R_{ij} со стороны заполненных ячеек без учета их значений: крестами отмечены уже заполненные ранее ячейки, цифры в незаполненных ячейках соответствуют числу ограничений; а, б, …, ё – заполнение диагональных элементов; ж, з, …, т – заполнение недиагональных элементов; у – результирующий порядок заполнения элементов ДЛК

Из приведенных результатов видно, что диагональное заполнение по предложенному выше критерию получается автоматически, хотя и в несколько ином порядке (диагонали заполняются поочередно). Указанный порядок заполнения элементов уменьшает арность узлов дерева комбинаторного перебора и, следовательно, – сокращает временные затраты на получение решений, что обеспечивает получение темпа генерации в размере 715 000 ДЛК/с для реализации на Си (790 000 ДЛК/с с использованием PGO-компиляции компилятора Microsoft Visual Studio 2008, процессор Intel Core i7 4770). С использованием данной реализации был реализован подсчет числа ДЛК для $N \le 8$, результаты которого приведены в таблице. При дальнейшем увеличении N необходимо использовать параллельные вычислительные системы, т.к. затраты вычислительного времени с увеличением размерности квадрата на единицу увеличивается на 2–3 порядка и однопоточная организация вычислительного процесса не представляется возможной.

Таблица. Число ДЛК порядка N

N	Число ДЛК с фиксированной первой строкой	$N!$	Общее число ДЛК	Время подсчета
1	1	1	1	< 1 с
2	0	2	0	< 1 с
3	0	6	0	< 1 с
4	2	24	48	< 1 с
5	8	120	960	< 1 с
6	128	720	92 160	< 1 с
7	171 200	5 040	862 848 000	2 с
8	7 447 587 840	40 320	300 286 741 708 800	30 ч

Библиографический список

1. Colbourn C.J., Dinitz J.H. Handbook of Combinatorial Designs. Second Edition. Chapman&Hall, 2006. 984 p.
2. https://ru.wikipedia.org/wiki/Латинский_квадрат
3. Ватутин Э.И., Титов В.С., Емельянов С.Г. Основы дискретной комбинаторной оптимизации. М.: АРГАМАК-МЕДИА, 2016. 270 с.
4. 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.
5. McKay B.D., Wanless I.M. On the number of Latin Squares // Ann. Combin. Vol. 9. 2005. P. 335–344.
6. https://oeis.org
7. 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.
8. Заикин О.С., Ватутин Э.И., Журавлев А.Д., Манзюк М.О. Применение высокопроизводительных вычислений для поиска троек взаимно частично ортогональных диагональных латинских квадратов порядка 10 // Параллельные вычислительные технологии (ПаВТ'2016). Челябинск: издательский центр ЮУрГУ, 2016. С. 155–166.
9. Ватутин Э.И., Журавлев А.Д., Заикин О.С., Титов В.С. Учет алгоритмических особенностей задачи при генерации диагональных латинских квадратов // Известия ЮЗГУ, 2016. Принята к опубликованию