Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики
и информационных технологий

Уравновешенные неполные блок-схемы

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ (МАГИСТЕРСКОЙ, ДИПЛОМНОЙ)
РАБОТЫ

Студента 4 курса 421 группы
направления (специальности)
09.03.01 «Информатика и вычислительная техника» факультета КНиИТ
Люкшина Ильи Андреевича

Научный руководитель
д.ф.-м.н., профессор ____________________ В.А. Молчанов
должность, уч. степень, уч. звание дата, подпись инициалы, фамилия

Заведующий кафедрой
к.ф.-м.н., доцент ____________________ Л.Б. Тяпаев
должность, уч. степень, уч. звание дата, подпись инициалы, фамилия

Саратов 2018

Введение. Часто ученым приходится сталкиваться с необходимостью проведения экспериментов. Для эффективной постановки эксперимента необходимо тщательно его спланировать. Цель планирования эксперимента – достижение максимальной точности измерений при минимальном количестве проведенных опытов и сохранении достоверности результатов. Приведем пример. Необходимо протестировать на человеке эффекты шести разных препаратов. Есть 10 человек, которые будут употреблять лекарства в течение шести дней, по одному препарату каждый день. Помочь созданию наиболее адекватного распределения препаратов должна блок-схема.

Целью данной дипломной работы является создание программы, генерирующей два латинских квадрата и проверяющей их ортогональность. Для выполнения этой цели поставлены следующие задачи: изучение блок-схем, подробное рассмотрение их свойств, а также изучение методов генерации блок-схем.

В главе 1 «Основные определения» приводятся основные определения, дается обзор свойств блок-схем, рассматриваются приложения блок-схем, таких, как системы Штейнера, матрицы Адамара и латинские квадраты. В этой главе также рассматриваются методы генерации блок-схем.

В главе 2 «Практическая часть» описан алгоритм работы программы генерации двух латинских квадратов и проверки их ортогональности, проведено тестирование программы на ряде примеров. Также описаны две возможные программы.

Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и приложения. Общий объём работы – 55 страницы, из них 45 страниц – основное содержание, включая 12 рисунков, список использованных источников из 12 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

1 Основные определения.

1.1 Блок-схема. Блок-схема — это множество вместе с семейством подмножеств, члены которого удовлетворяют некоторым свойствам, которые считаются полезными для конкретного приложения. Эти приложения из разных областей науки, включая планирование эксперимента, конечную геометрию, тестирование программного обеспечения, криптографию и алгебраическую геометрию. Наиболее интенсивно изучались уравновешенные неполные блок-схемы[3].

Определение 1. Уравновешенная неполная блок-схема с параметрами (v, b, r, k, λ) – это пара (P, B) со следующими свойствами:

1. P – множество из v элементов;
2. $B = \{B_1, \ldots, B_b\}$ – подмножество в $\mathcal{P}(P)$ из b элементов;
3. Каждый элемент $p \in P$ встречается ровно в r множествах из B;
4. Каждое B_i содержит ровно k элементов, $k < v$;
5. Каждая неупорядоченная пара $\{p, q\}$, где $p, q \in P$, $p \neq q$, встречается ровно в λ множествах из B.

Существует также другой вид блок-схем:

Определение 2. Блок-схема с параметрами v, b, k, r называется частично уравновешенной неполной блок-схемой с m классами, если выполняются следующие условия:

1) всякий блок содержит лишь различные элементы;

2) между различными элементами установлено отношение i-связанности $(i = 1, 2, \ldots, m)$ такое, что:

а) все $\binom{v}{2}$ неупорядоченных пар разделены на m классов, причем каждая пара принадлежит одному и только одному классу; два элемента, образующие пару из i-го класса, называются i-связанными; отношение i-связанности симметрично;

б) для любого элемента существует точно n_i элементов, i-связанных с ним $(i = 1, 2, \dots , m)$;

в) для всякой пары i-связанных элементов α и β число элементов, j-связанных с α и одновременно k-связанных с β, не зависит от исходной пары и равно p^i_{jk} в силу симметричности отношения i-связанности

$$p^i_{jk} = p^i_{kj}\ (i, j, k = 1,2, \dots , m)$$

3) всякая пара i-связанных элементов принадлежит точно λ_i блокам $(i = 1, 2, \dots , m)$[2].

1.2 Свойства блок-схем. К свойствам блок-схем можно отнести:

1) Двойственность.
2) Основное матричное соотношение.
3) Дополнительная, остаточная и производные блок-схемы.
4) Разрешимость и аффинная разрешимость блок-схем[2].

1.3 Приложения блок-схем.

1.3.1 Система Штейнера. Система Штейнера с параметрами t, k, v — это v - элементное множество S вместе с набором k-элементных подмножеств множества S (называемых блоками) со свойством, что каждое t-элементное подмножество S содержится ровно в одном блоке. [4]

Определение системы Штейнера, данное выше, включает в себя классическое определение системы Штейнера, в котором дополнительно требуется, чтобы $k = t + 1$.

1.3.2 Матрица Адамара. Матрица H порядка n, элементами которой являются +1 и -1, называется матрицей Адамара, если

$$HH^T = nI,$$

где H^T — транспонированная матрица H, I — единичная матрица порядка n[7]. Из этого равенства следует, что

$$|H|^2 = n^n$$

и

$$H^{-1} = n^{-1}H^{T},$$

следовательно,

$$HH^{T} = H^{T}H = nI.$$

1.3.3 Латинские квадраты. Квадратная $(n \times n)$-матрица $L = (a_{ij})$ над множеством $A = \{a_1, \ldots, a_n\}$ называется латинским квадратом порядка n, если каждая её строка и каждый столбец содержит ровно один элемент из A ровно один раз. Два латинских квадрата одного и того же порядка называются ортогональными, если при наложении одного квадрата на другой каждый символ первого квадрата встречается один и только один раз с каждым символом второго квадрата. Иначе говоря, два латинских квадрата (a_{ij}) и (b_{ij}) порядка n над A ортогональны, если все $\upsilon = n^2$ упорядоченных пар (a_{ij}, b_{ij}) различны. Множество латинских квадратов одного и того же порядка называется множеством взаимно ортогональных латинских квадратов, если латинские квадраты в этом множестве попарно ортогональны. Множество из $n - 1$ взаимно ортогональных латинских квадратов порядка n называется полным[1].

1.4 Методы генерации блок-схем и приложений блок-схем

1.4.1 Алгоритм локального поиска для сбалансированных неполных блочных конструкций. Локальный поиск оказался очень успешным для многих комбинаторных задач. Естественными способами применения локального поиска к задачам оптимизации являются изучение возможных решений и минимизация затрат. Для этих задач локальный поиск обычно применяется к нахождению всех возможных вариантов, а целевая функция является количеством нарушений ограничений; если оно становится равным нулю, то решение найдено. В качестве примера рассмотрим проблему N ферзей. Ограничения для этой задачи заключаются в том, что никакие два ферзя не могут атаковать друг друга. Обычный подход к локальному поиску состоит в том, чтобы исследовать все возможные варианты, то есть все возможные размещения ферзей[8].

Существует также совершенно другая форма локального поиска N ферзей: беспорядочное размещение ферзей в неатакуемых позициях; когда он не может быть помещен, произвольно удаляем один или несколько помещенных ферзей; будем так делать до тех пор, пока все ферзи не будут помещены. Преимущество этого типа локального поиска: ограничения никогда не нарушаются.

Генерация блок-схем является стандартной комбинаторной проблемой. Один из способов определения блок-схемы заключается в ее матрице инцидентности, которая представляет собой двоичную матрицу с v строками, b столбцы, r единиц в строке, k единиц в столбце и скалярным произведением λ между любой парой различных строк. Можно доказать, что для существования блок-схемы его параметры должны удовлетворять условиям $rv = bk$, $\lambda(v - 1) = r(k - 1)$ и $b \geq v$, что, однако, не является достаточными условиями. Одним из источников неразрешимости является большое количество симметрий: при любом решении любые две строки или столбца могут поменяться местами для получения другого решения[8].

Самая простая модель для генерации BIBD представляет каждый элемент матрицы двоичной переменной $m_{ij} \in \{0,1\}$. Существует три типа ограничений:

1) v b-ичных ограничений для каждой строки,

2) b v-ичных ограничений для каждого столбца и

3) $v(v\text{-}1)$ $2b$-ичных ограничений для λ совпадений в каждой паре строк.

1.4.2 Обобщенные матрицы Адамара и их построение. Обобщенная матрица Адамара $GH(qs, G)$ над группой G порядка q есть $(qs \times qs)$-матрица $GH(qs, G) = (h_{ij})$ такая, что

$$h_{ij} \in G \text{ для всех } 1 \leq i, j \leq qs,$$

а также

$$\sum_{k=1}^{qs} h_{ik}h_{jk}^{-1} = \sum_{g\epsilon\, G} sg$$

всякий раз, когда суммирование ведется в групповом кольце $Z[G]$.

Построение обобщенных матриц Адамара состоит из нескольких шагов[10].

1) Пусть элементы $C_{p^r} – e, \chi_1, \dots, \chi_{p^r-1}$.

2) Запишем таблицу умножения группы с первой строкой и столбцом $e, \chi_1, \dots, \chi_{p^r-1}$. Напишем C для ядра этой таблицы умножения (т. е. с удалением первой строки и столбца). Тогда $C = (c_{ij})$ - матрица порядка $p^r - 1$ с тем свойством, что

$$\sum_{k=1}^{p^r-1} c_{ik}c_{jk}^{-1} = (p^r - 1)g,$$

где g – не нейтральный элемент группы.

3) Запишем обобщенную матрицу Адамара $GH(p^r, C_{p^r})$ порядка p^r с первыми строкой и столбцом, состоящих из e, а вторая строка и столбец перестроены как $e, \chi_1, \dots, \chi_{p^r-1}$. Удалим первую строку и столбец, чтобы получить ее ядро $K = (k_{ij})$. Теперь

$$\sum_{h=1}^{p^r-1} k_{ih}k_{jh}^{-1} = C_{p^r}/\{e\},$$

где e - нейтральный элемент, т. е. каждый элемент группы, за исключением нейтрального элемента, встречается ровно один раз.

4) Пусть $Y = (y_{ij})$ – обобщенная матрица Адамара порядка $p^r - 1$, нормированная, как в 3). Напишем A^{ij} для матричного представления y_{ij}.

5) Сформируем блочную матрицу $D = (d_{ij})$, элемент ij которой: ij элемент в C встречается KA^{ij} раз, C, K и A^{ij} определены в 2), 3) и 4).

6) Возьмем матрицу $Y = GH(p^r, C_{p^r})$, полученную в 3). Пусть s – матрица $(1 \times p^r – 1)$, состоящая из единиц. Получим G_a и G_b из Y. Пусть теперь $A = G_a\, x\, s$ и $B = G_b\, x\, s^T$, которые имеют размеры $(p^r - 1)\, x\, (p^r - 1)^2$ и $(p^r - 1)^2\, x\, (p^r - 1)$ соответственно. Пусть E - матрица $(p^r - 1)\, x\, (p^r - 1)$ с каждым элементом e.

Тогда мы утверждаем, что

$$\begin{bmatrix} E & A \\ B & D \end{bmatrix}$$

является необходимым $GH(p^r(p^r\text{-}1), C_{p^r})$[10].

1.4.3. Алгебраические методы построения проективных плоскостей. Для описания данного метода необходимо дать несколько определений.

Определение. Множество N вместе с двумя операциями $+$ и $\circ$ называется почти-кольцом, если

1. $(N, +)$ - группа (не обязательно абелева);
2. $(N, \circ)$ - полугруппа;
3. $\forall\, n, n', n'' \in N : (n + n') \circ n'' = n \circ n'' + n' \circ n''$.

Пусть $(\Gamma, +)$ – (не обязательно абелева) группа с нулем 0. Почти-кольцами будут, например, $\mathrm{M}(\Gamma) := (\Gamma^{\Gamma}, +, \circ), \mathrm{M}_0(\Gamma) := \{f \in \mathrm{M}(\Gamma) | f(0) = 0\}$ и $\mathrm{M}_c(\Gamma) := \{f \in \mathrm{M}(\Gamma) | f$ – константа$\}$.

Пусть $(\Gamma, +)$ - группа и $S \subseteq End(\Gamma)$. Тогда $\mathrm{M}_S(\Gamma) := \{f \in \mathrm{M}_0(\Gamma) |\ \forall\, s \in S : f \circ s = s \circ f\}$ является почти-кольцом (которое будет играть в дальнейшем важную роль). Если, например, $(\Gamma, +) = (R, +)$ *и* $S = \{s_\lambda | \lambda \in R\}$, где $s_\lambda : R \to R,\ s \to \lambda x$, то $M_S(\Gamma)$ состоит из однородных функций:

$$M_S(\Gamma) = \{f \in R^R | \forall x, \lambda \in R : f(\lambda x) = \lambda f(x)\}.$$

Пусть V — векторное пространство $M_{aff}(V) := \{f \in V^V |\ f$ — аффинное отображение, т. е. сумма линейного и постоянного отображений$\}$ - еще одно важное почти-кольцо[1].

Конечно, любое кольцо является почти-кольцом. Следовательно, понятие почти-кольца обобщает понятие кольца: опущены две из кольцевых аксиом — коммутативность сложения и (что более существенно) второй закон дистрибутивности.

Определение. Пусть N — почти-кольцо и n_1, $n_2 \in N$. Положим

$$n_1 \equiv n_2 \Leftrightarrow \forall n \in N : n \circ n_1 = n \circ n_2.$$

Отношение ≡ является отношением эквивалентности на множестве N.

Определение. Почти-кольцо N называется планарным, если $|N/\equiv| \geq 3$ и если каждое управление $x \circ a = x \circ b + c \; (a, b, c \in N, a \neq b)$ имеет единственное решение $x \in N$.

Планарные почти-кольца позволяют строить блок-схемы с параметрами (v, b, r, k, λ), имеющие превосходную эффективность $E := \lambda v/rk$. Число E лежит между 0 и 1 и оценивает, с точки зрения статистического анализа экспериментов, «качество» схемы; блок-схемы с эффективностью $E \geq 0,75$ обычно считаются «хорошими»[1].

2.1 Алгоритм. Созданная программа работает по следующему алгоритму:

1) Вводим в программу n – значение порядка латинского квадрата.
2) Для каждой ячейки латинского квадрата генерируется значение, отвечающее определению латинского квадрата. Способ генерации описан в следующем разделе.
3) Когда n^2 ячеек заполнены, мы получаем латинский квадрат.
4) Аналогичным образом создаём второй латинский квадрат такого же порядка.
5) Совмещаем эти квадраты и проверяем их ортогональность.
6) Выносим вердикт об ортогональности двух созданных латинских квадратов.

2.2 Описание программы. Остановимся подробнее на самых важных аспектах программы: генерация значений для ячеек и проверка ортогональности.

1) Генерация ячеек

Для каждой ячейки существует набор допустимых значений от 1 до n, где n – порядок латинского квадрата. Случайным образом выбирается число, также от 1 до n. Это число сравнивается с набором возможных значений: если в этом наборе есть данное случайно выбранное число, то это число записывается в ячейку, а для ячеек, стоящих в одной строке и в одном столбце с проверяемой ячейкой, это число удаляется из списка. В противном случае, мы подбираем другое число.

2) Проверка ортогональности латинских квадратов

Наша задача на этом этапе – подсчитать количество вхождений ячеек с одинаковым значением. Для порядков больше 10 может возникнуть следующая ситуация: к примеру, для порядка 13 возможна комбинация 113, которая может обозначать две различные пары: 1 и 13 или 11 и 3. Теоретически ортогональный латинский квадрат из-за этой ситуации может быть ошибочно определен. Поэтому при подсчете вхождений было решено разделить два значения разных латинских квадратов.

2.3 Внешний вид программы. Вывод программы выглядит следующим образом. При запуске программы появляется экран вывода, который просит ввести порядок латинских квадратов. Это число должно быть целым положительным числом. Когда программа закончила свою работу, на экране появится сообщение о завершении работы. После того, как программа закончила свою работу, мы можем посмотреть полученный результат в отдельном файле. Путь к файлу задается в коде программы. В этом файле находятся:

1) Два сгенерированных латинских квадрата порядка n.
2) Третий квадрат, полученный наложением первого латинского квадрата на второй.
3) Количество вхождений существующих значений ячеек, полученных в результате наложения.

2.4 Примеры работы. Приведем примеры работы программы для некоторых различных значений порядка латинских квадратов.

Начнем с простых примеров, порядков 1 и 2. Для порядка 1 выведутся два одинаковых квадрата, поэтому для этого порядка выведется специальное сообщение. Для порядка 2 возможны два варианта: либо два возможных выведенных квадрата одинаковы, либо они отличаются. В обоих случаях полученные квадраты не являются ортогональными.

Также в ходе работы были сгенерированы ортогональные квадраты порядков 3 и 4.

Данная программа при определенных усовершенствованиях может решать еще две задачи:

1) Задача о выводе всех латинских квадратов порядка n.

Программа, реализующая данную задачу, должна выводить все латинские квадраты порядка n.

2) Нахождение двух ортогональных квадратов порядка n.

Программа, реализующая данную задачу, является дополнением исходной программы. К одному генерируемому латинскому квадрату мы подбираем ортогональный ему второй латинский квадрат с помощью той же генерации.

Заключение. В процессе выполнения данной дипломной работы разработана программа по генерации двух латинских квадратов и проверке их ортогональности. Выполнены все поставленные задачи: изучены блок-схемы и рассмотрены их свойства, изучены методы генерации блок-схем и получена программная реализация метода генерации латинских квадратов.

Список использованных материалов

[1] Лидл Р., Пильц Г. Прикладная абстрактная алгебра: Учебное пособие – перевод с английского, 1996 г. ISBN 5-7525-0483-X.

[2] Широкова С. А. Блок-схемы, УМН, 1968, том 23. [Электронный ресурс] сайт. URL: http://www.mathnet.ru/links/bc34a28c70e57868509d144c35a3d339/rm5669.pdf Дата обращения 20.04.2018.

[3] Блок-дизайн. [Электронный ресурс] сайт. URL: https://ru.wikipedia.org/wiki/Блок-дизайн Дата обращения 25.04.2018.

[4] Система Штейнера. [Электронный ресурс] сайт. URL: https://ru.wikipedia.org/wiki/Система_Штейнера Дата обращения 25.04.2018.

[5] Латинский квадрат. [Электронный ресурс] сайт. URL: https://ru.wikipedia.org/wiki/Латинский_квадрат Дата обращения 01.05.2018.

[6] Joan Cooper, Diane Donovan, Jennifer Seberry. Joan Cooper, Diane Donovan and Jennifer Seberry, Latin squares and critical sets of minimal size, Australasian Journal of Combinatorics, 4, (1991), 113-120. [Электронный ресурс] сайт. URL: https://www.researchgate.net/profile/J_Cooper/publication/246657614_Latin_squares_and_critical_sets_of_minimal_size/links/0f31753b3ea500b45b000000/Latin-squares-and-critical-sets-of-minimal-size.pdf Дата обращения 01.05.2018.

[7] Матрица Адамара. [Электронный ресурс] сайт. URL: https://ru.wikipedia.org/wiki/Матрица_Адамара Дата обращения 11.05.2018.

[8] Steven Prestwich. A Local Search Algorithm for Balanced Incomplete Block Designs, Cork Constraint Computation Centre, Department of Computer Science, University College, Cork, Ireland, 2002, 132-143. [Электронный ресурс] сайт. URL: https://pdfs.semanticscholar.org/4622/aea4b5963edfc3a8c39186286e230407c6cb.pdf Дата обращения 13.05.2018.

[9] S. Georgiou, C. Koukouvinos, J. Seberry. Hadamard matrices, orthogonal designs and construction algorithm, in Wallis, WD (ed), Designs 2002: Further Combinatorial and Constructive Design Theory, Kluwer Academic Publishers, Norwell, Massachusetts, 2002, 133-205. [Электронный ресурс] сайт. URL: http://www.academia.edu/20806250/Hadamard_matrices_orthogonal_designs_and_construction_algorithms Дата обращения 16.05.2018.
[10] Jennifer Seberry. A construction for generalized Hadamard matrices, Journal of Statistical Planning and Inference, 4, 1980, 365-368. [Электронный ресурс] сайт. URL: https://www.sciencedirect.com/science/article/pii/037837588090021X Дата обращения 18.05.2018.
[11] Bing Zhou. BLOCK DESIGN AND APPLICATIONS, 2012, 87-108. [Электронный ресурс] сайт. URL: http://www.trentu.ca/academic/math/bz/block.pdf. Дата обращения 21.05.2018.
[12] Маршалл Холл. Комбинаторика – перевод с английского, Издательство «Мир», Москва. 1970 г. [Электронный ресурс] сайт. URL: http://mathscinet.ru/files/HallM.pdf Дата обращения 21.05.2018.