ПРОГРАММИРОВАНИЕ, 2025, № 2, с. 3–14

КОМПЬЮТЕРНАЯ АЛГЕБРА

УДК 004.421.6+512.644

О ДВОИЧНЫХ РЕШЕНИЯХ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ ПО МОДУЛЮ ТРИ

© 2025 г. **О. А. Зверков** *, **А. В. Селиверстов** **

Институт проблем передачи информации им. А.А. Харкевича Российской академии наук

127051 Москва, Большой Каретный пер., д. 19, стр. 1, Россия

* *e-mail: zverkov@iitp.ru*, ORCID: 0000-0002-8546-364X

** *e-mail: slvstv@iitp.ru*, ORCID: 0000-0003-4746-6396

Поступила в редакцию 30.07.2024

Рассмотрена задача поиска двоичного решения для системы линейных уравнений по модулю три. В случае, когда количество уравнений ограничено сверху достаточно медленно растущей функцией от числа переменных, предложен новый алгоритм полиномиального времени для распознавания существования двоичного решения у такой системы. Алгоритм основан на замечании: если в матрице коэффициентов присутствуют ненулевые пропорциональные друг другу столбцы, то элиминация соответствующих переменных сохраняет свойство отсутствия двоичного решения системы. В частности, каждая система из двух уравнений от пяти переменных допускает элиминацию некоторых переменных, при которой сохраняется свойство отсутствия двоичного решения системы. На основе этих результатов, мы предлагаем безошибочный эвристический алгоритм, который реализован на языке программирования Python. Для представления матриц и выполнения базовых операций используется библиотека NumPy. Входом служит расширенная матрица системы. С использованием этой реализации была рассчитана эмпирическая оценка времени работы. Экспериментально показано, что алгоритм эффективнее для разреженных систем уравнений. Очевидно, метод двоичного поиска позволяет найти двоичное решение системы, когда оно существует. Это открывает возможность применения, в частности, для решения задач математической биологии.

Ключевые слова: конечное поле, система линейных уравнений, система компьютерной алгебры

1. ВВЕДЕНИЕ

Обозначим через $GF(3)$ поле вычетов по модулю три. Элементы поля $GF(3)$ будем обозначать числами из множества $\{0, 1, 2\}$. Операциями в этом поле служат сложение и умножение по модулю три, в частности, оба выражения -1 и $1/2$ равны элементу 2. Мы рассматриваем системы алгебраических уравнений над полем $GF(3)$. В этой работе предложен алгоритм полиномиального времени для частного случая задачи об инцидентности подпространства и точек с координатами из множества $\{0, 1\}$. Эта задача считается вычислительно трудной в худшем случае. В отличие от работ [1, 2] теперь рассмотрены системы из малого числа уравнений.

Обсуждаемый алгоритм реализован на языке Python. Благодаря доступности библиотек, таких как NumPy, используемой в данной реализации для представления матриц и выполнения базовых операций с ними, Python обеспечивает хороший баланс между скоростью разработки и производительностью.

Системы алгебраических уравнений над конечным полем допускают геометрическую интерпретацию. Начиная с XIX века, создано много конечных геометрий, в которых существует лишь конечное число точек и прямых [3, 4]. В каждой из них разрешимо любое утверждение о взаимном расположении точек и прямых, поскольку возможен полный перебор конечного числа вариантов [5, 6]. Однако вычислительная сложность может быть высокой, что объясняет актуальность поиска частных случаев, вычисли-

мых за полиномиальное время. С другой стороны, вычисления над конечными полями часто оказываются менее сложными, чем над полем рациональных чисел, поскольку над конечным полем ограничена длина записи каждого из коэффициентов уравнений, ср. [7].

Над полем $GF(3)$ каждая аффинная прямая содержит ровно три точки. Аффинная плоскость соответствует конфигурации Гессе [8]. На этой плоскости 9 точек и 12 прямых. В 3-мерном аффинном пространстве 27 точек, 117 прямых и 39 плоскостей. Каждая пара различных точек определяет одну прямую, а каждая прямая может быть задана тремя парами точек. Поэтому в n-мерном аффинном пространстве 3^n точек и $3^{n-1}(3^n-1)/2$ прямых. В силу двойственности, n-мерное проективное пространство содержит $(3^{n+1}-1)/2$ точек и столько же гиперплоскостей. Поэтому в аффинном пространстве число гиперплоскостей равно $(3^{n+1}-1)/2-1$.

Решение системы уравнений, в котором значение каждой переменной принадлежит множеству $\{0,1\}$, называется $(0,1)$-решением или двоичным решением. Распознавание существования $(0,1)$-решения у системы линейных уравнений над полем $GF(3)$ служит примером NP-полной задачи. Однако для одного уравнения эта задача решается легко: только линейное уравнение вида $x_k = 2$ не имеет $(0,1)$-решения, поскольку каждое линейное уравнение, нетривиально зависящее от двух или более переменных, имеет $(0,1)$-решение. Более того, задача поиска $(0,1)$-решения также разрешима за полиномиальное время для систем из фиксированного числа линейных уравнений. Например, можно использовать сводимость задачи над $GF(3)$ к аналогичной задаче над полем рациональных чисел, для решения которой известно много алгоритмов. Описание таких алгоритмов можно найти в обзорах [9, 10]. Мы же рассмотрим системы, в которых число уравнений ограничено не константой, а монотонно возрастающей функцией от числа переменных. С геометрической точки зрения, задача состоит в распознавании инцидентности подпространства и вершин куба. Разбиение NP-полной задачи на несколько NP-полных задач и задач низкой сложности согласуется с недавно полученными результатами [11].

Для решения близких комбинаторных задач применяются также вероятностные алгоритмы [12]. Распределение значений сумм случайных $(0,1)$-величин рассмотрено Яшунским [13]. Также изучались как распределение элементов в матрицах данного ранга над конечным полем [14], так и распределение рангов случайных матриц с данным количеством ненулевых элементов [15, 16]. Отношение числа невырожденных матриц к числу всех матриц над полем $GF(3)$ монотонно убывает с ростом порядка n, но выше асимптотического значения [17], равного произведению

$$\prod_{k=1}^{\infty}\left(1-\frac{1}{3^k}\right)=0.560126\ldots$$

С другой стороны, обсуждаются так называемые генерические алгоритмы или безошибочные эвристики, дающие правильный ответ на большой доле входов среди входов данной длины и удовлетворяющих легко проверяемым ограничениям. Но такой алгоритм может выдать уведомление о неприменимости алгоритма. В любом случае алгоритм не делает ошибок. Формальные определения можно найти в статьях Рыбалова [18, 19].

В разделе 2 дано теоретическое обоснование алгоритма и близкие результаты. В разделе 3 обсуждаются реализация алгоритма и результаты вычислений. В разделе 4 дано краткое заключение.

2. ТЕОРЕТИЧЕСКИЕ РЕЗУЛЬТАТЫ

Пусть система линейных уравнений от переменных $x_1,\ldots,x_n$ содержит больше одного уравнения, и некоторое уравнение нетривиально зависит от переменной x_k. Будем говорить, что новая система линейных уравнений получена из исходной системы элиминацией переменной x_k, если новая система не зависит от переменной x_k, а исходная система эквивалентна объединению новой системы и ровно одного уравнения (зависящего от x_k), равного линейной комбинации уравнений исходной системы. Элиминация переменной соответствует проекции на координатное подпространство. Рассмотрим пример:

$$\begin{cases} x_1+x_2 & = & 1 \\ x_1-x_2+x_3+x_4 & = & 0. \end{cases}$$

Элиминация переменной x_3 (или x_4) даст одно уравнение $x_1 + x_2 = 1$. При этом каждое его $(0,1)$-решение продолжается до $(0,1)$-решения системы двух уравнений. Действительно, если две переменные x_3 и x_4 независимо пробегают значения из множества $\{0,1\}$, то их сумма x_3+x_4 принимает все три значения из поля $GF(3)$. Это также выполняется и для разности $x_3 - x_4$. Поэтому одновременная элиминация двух переменных, когда возможно, позволяет уменьшить число уравнений, не нарушая существования $(0,1)$-решения.

Элиминация переменной может приводить к системе, имеющей большее число $(0,1)$-решений, чем было у исходной системы. Следующий результат справедлив лишь над полем $GF(3)$, но не над полями с большим числом элементов.

Теорема 1. *Даны натуральные числа n и m, удовлетворяющие неравенствам $n \geqslant 5$, $m \geqslant 2$ и $m \leqslant \log_3(2n-1)$, и система из m линейных уравнений от n переменных над полем $GF(3)$. Пусть для каждого индекса $1 \leqslant k \leqslant n$ существует уравнение, нетривиально зависящее от переменной x_k. Если у этой системы нет $(0,1)$-решения, то существует такой индекс $k \leqslant n$, что элиминация переменной x_k вновь приводит к системе, у которой нет $(0,1)$-решения. Более того, эта система может быть найдена за полиномиальное время $O(mn\log_2(n+1))$.*

Доказательство. Систему уравнений можно записать в матричном виде $A\mathbf{x} = \mathbf{b}$, где через A обозначена $m \times n$ матрица коэффициентов линейных членов уравнений, а через $\mathbf{x}$ и $\mathbf{b}$ — столбцы из n переменных и m чисел, соответственно. По условию теоремы, в матрице A нет нулевых столбцов.

При условии $2 \leqslant m \leqslant \log_3(2n-1)$ в матрице A найдутся два линейно зависимых столбца. Действительно, число возможных различных ненулевых столбцов равно $3^m - 1$. Это множество разбивается на $(3^m-1)/2$ пар линейно зависимых столбцов. Поэтому выполнение условия $n \geqslant (3^m+1)/2$ обеспечивает, что в матрице A найдутся два линейно зависимых столбца. Обозначим номера этих столбцов через j и k. Найти номера j и k можно перебирая $n(n-1)/2$ вариантов и проверяя линейную зависимость соответствующих столбцов.

Исходная система $A\mathbf{x} = \mathbf{b}$ эквивалентна системе $B\mathbf{x} = \mathbf{c}$, где в $m \times n$ матрице B в столбцах с номерами j и k ненулевые элементы расположены лишь в одной строке, номер которой обозначим через ℓ. Здесь матрица B получается из матрицы A элементарными операциями над строками, а элемент столбца $\mathbf{c}$ равен соответствующей линейной комбинации элементов столбца $\mathbf{b}$. Если система уравнений, полученная удалением ℓ-го уравнения из этой системы уравнений, имеет $(0,1)$-решение, то она имеет $(0,1)$-решение при некоторых $(0,1)$-значениях переменных x_j и x_k. Следовательно, вся система тоже имеет $(0,1)$-решение, поскольку выбор значений переменных x_j и x_k позволяет выполнить ℓ-ое уравнение при любой оценке остальных переменных. Удаление ℓ-го уравнения соответствует элиминации каждой из переменных x_j и x_k. □

Отметим следствие. Пусть в 5-мерном пространстве над полем $GF(3)$ дано 3-мерное подпространство, которое не проходит ни через одну $(0,1)$-точку. Образом проекции на некоторую координатную гиперплоскость, которая соответствует элиминации переменной, служит подпространство, которое также не проходит ни через одну $(0,1)$-точку.

Следующий результат позволяет легко проверить, проходит ли подпространство малой коразмерности через некоторую $(0,1)$-точку.

Теорема 2. *Существует алгоритм полиномиального времени, который получает на вход систему из m линейных уравнений от n переменных над полем $GF(3)$ и при выполнении неравенства $m \leqslant \log_3\log_3(2n-1)$ принимает вход тогда и только тогда, когда система имеет $(0,1)$-решение.*

Доказательство. Алгоритм в цикле делает попытки элиминировать переменные в соответствии с теоремой 1. Также удаляются из рассмотрения все переменные, которые не входят в новую систему. В случае успеха на очередном шаге будет получена система уравнений, состоящая из меньшего числа уравнений. При этом новая система имеет $(0,1)$-решение тогда и только тогда, когда исходная система имеет $(0,1)$-решение. После выполнения менее m шагов этот процесс останавливается в одном из двух возможных случаев: либо осталось одно уравнение,

либо полученная система зависит от малого числа переменных.

Если осталось одно уравнение, то для уравнения вида $x_k = 2$ вход отвергается, а для уравнения другого вида вход принимается.

Если осталось k переменных, а система содержит несколько уравнений и не может быть уменьшена, то происходит разбор 2^k случаев. Оценим сверху число k. Поскольку оставшееся число уравнений не превышает числа m, выполнено неравенство $\log_3(2k-1) < m$. Но по условию применимости алгоритма выполнено $m \leqslant \log_3 \log_3(2n-1)$. Следовательно, выполнены неравенства $(2k-1) < \log_3(2n-1)$ и $k \leqslant 0.5 \log_3(2n-1) < 0.3155 \log_2(2n-1)$. Поэтому число различных $(0,1)$-оценок оставшихся k переменных меньше числа $(2n-1)^{0.3155}$. □

Если условие $m \leqslant \log_3 \log_3(2n-1)$ из теоремы 2 нарушено, то алгоритм всегда даст правильный ответ, но время работы может быть большим. Однако для большой доли случаев среди входов с данными значениями m и n время работы алгоритма будет маленьким даже при более слабом ограничении $m \leqslant \log_3(2n-1)$.

При данных значениях m и n, удовлетворяющих неравенству $m \leqslant \log_3(2n-1)$, почти любая система из m уравнений от n переменных над полем $GF(3)$ будет иметь много $(0,1)$-решений. Это другой эвристический вероятностный алгоритм проверки разрешимости, когда проверяются случайно выбранные $(0,1)$-точки. Если $(0,1)$-решение найдено, то оно действительно существует, а если $(0,1)$-решения не обнаружено, то алгоритм выдаст лишь предупреждение о неудаче. С другой стороны, при этом условии существование $(0,1)$-решения всегда может быть проверено детерминированным алгоритмом за квазиполиномиальное время $n^{O(\log n)}$.

Теорема 3. *Существует алгоритм, который получает на вход систему из m линейных уравнений от n переменных над полем $GF(3)$ и за время $n^{O(m)}$ принимает вход тогда и только тогда, когда система имеет $(0,1)$-решение.*

Доказательство. Каждое $(0,1)$-решение для системы уравнений от n переменных над полем $GF(3)$ продолжается до $(0,1)$-решения для системы уравнений от $n + m\lceil \log_3 n \rceil$ переменных над полем рациональных чисел. И обратно, каждое решение новой системы укорачивается до решения исходной системы. Каждая система имеет по m уравнений. Здесь j-му уравнению вида $a_{j0} + a_{j1}x_1 + \cdots + a_{jn}x_n = 0$ над полем $GF(3)$ соответствует уравнение $a_{j0} + a_{j1}x_1 + \cdots + a_{jn}x_n = 3y_{j1} + 9y_{j2} + \cdots + 3^k y_{jk}$ над полем рациональных чисел, где $k = \lceil \log_3 n \rceil$ и каждая новая переменная $y_{j1}, \ldots, y_{jk}$ встречается только один раз. Коэффициентами новой системы служат целые числа, абсолютные величины которых не превосходят числа $3n$. В свою очередь, $(0,1)$-решения этой системы совпадают с $(0,1)$-решениями одного уравнения, равного линейной комбинации уравнений системы, в которой коэффициенты ограничены сверху величиной $n^{O(m)}$. Дальше поиск $(0,1)$-решения можно выполнить методом динамического программирования [9] за время не более $n^{O(m)}$. □

Можно дополнительно менять систему уравнений, сохраняя число переменных и свойство иметь $(0,1)$-решение. Это позволяет расширить область применения нашего алгоритма. Однако в этом случае $(0,1)$-решение для новой системы, вообще говоря, не будет решением для исходной системы. Поэтому такой подход удобен для проверки существования, но не для поиска $(0,1)$-решения.

Теорема 4. *Дана система линейных уравнений от n переменных над полем $GF(3)$*

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ \cdots & \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m \end{cases}$$

и целое число $1 \leqslant s \leqslant n$. Эта система имеет $(0,1)$-решение тогда и только тогда, когда $(0,1)$-решение имеет новая система, в которой для каждого $1 \leqslant j \leqslant m$ в j-ом уравнении коэффициент a_{js} при переменной x_s заменяется линейной комбинацией коэффициентов

$$c_j = 2b_j - \sum_{k=1}^{n} a_{jk}.$$

Доказательство. Рассмотрим вспомогательную систему, в которой в каждом уравнении добавлен

линейный член от новой переменной y:

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n + c_1 y &= b_1 \\ \cdots & \\ a_{m1}x_1 + \cdots + a_{mn}x_n + c_m y &= b_m, \end{cases}$$

где коэффициенты c_j определены в условии теоремы. Одно из решений этой системы получается, когда все переменные равны элементу 2. При $y = 0$ каждое $(0,1)$-решение продолжает некоторое $(0,1)$-решение исходной системы. При $y = 1$ каждое $(0,1)$-решение получается из некоторого $(0,1)$-решения для $y = 0$ одновременной заменой прочих переменных $x_k \to 1 - x_k$. Поэтому вспомогательная система имеет $(0,1)$-решение тогда и только тогда, когда у исходной системы существует $(0,1)$-решение. Более того, множество $(0,1)$-решений вспомогательной системы разбивается на пары антиподальных решений, переходящих друг в друга при одновременной инверсии значений всех переменных. Следовательно, если $(0,1)$-решение существует, то вспомогательная система имеет пару антиподальных $(0,1)$-решений и для $x_s = 0$ и для $x_s = 1$.

Далее, фиксируем значение $x_s = 0$ и получаем систему от n переменных x_1, …, x_{s-1}, y, x_{s+1}, …, x_n. Заменим имя переменной y на x_s. Новая система от n переменных имеет $(0,1)$-решение тогда и только тогда, когда исходная система имеет $(0,1)$-решение, хотя решение для одной системы не будет, вообще говоря, решением для другой. □

Отметим, что геометрический смысл преобразования из теоремы 4 в проективном преобразовании, при котором гиперплоскость, проходящая через точку в аффинной части с координатами $(2, \cdots, 2)$ и не инцидентная никакой $(0,1)$-точке, становится несобственной гиперплоскостью. Такое преобразование служит инволюцией. Композиция нескольких таких преобразований не приводит к существенно новым системам.

Рассмотрим ещё одно легко проверяемое условие существования $(0,1)$-решения у системы линейных уравнений над $GF(3)$.

Теорема 5. *Дана система линейных уравнений над полем $GF(3)$. Пусть эта система имеет решение, в котором все переменные, кроме одной, равны 2. Если у системы нет $(0,1)$-решения, то в результате элиминации некоторой переменной получается новая система, которая также не имеет $(0,1)$-решения.*

Доказательство. Без ограничения общности можно считать, что исходная система уравнений имеет решение $(0, 2, \cdots, 2)$, где все переменные, кроме первой, равны 2. Предположим, что решением системы служит точка с координатами $(2, a_2, \cdots, a_n)$, где для $j \geqslant 2$ все значения a_j принадлежат множеству $\{0,1\}$. Тогда $(0,1)$-решением системы служит точка с координатами $(1, 1 - a_2, \cdots, 1 - a_n)$. □

3. РЕАЛИЗАЦИЯ И ОБСУЖДЕНИЕ

Реализована следующая функция `has01solution`(M). Входом служит непустая матрица M с m строками над полем $GF(3)$. Обозначим через $\mathbf{b}$ последний столбец этой матрицы, называемый дополнительным. Обозначим через A подматрицу, расположенную в первых n столбцах, кроме последнего. Столбец $\mathbf{b}$ содержит свободные члены уравнений, а матрица A состоит из коэффициентов линейных членов. Если матрица A пустая, то линейные части всех уравнений равны нулю, а сами уравнения превращаются в тождества $0 = 0$ или ложные формулы $0 = 1$ или $0 = 2$. Матрица M служит расширенной матрицей системы уравнений. Матрица M изменяется так, что последний столбец всегда содержит свободные члены, а другие столбцы содержат коэффициенты линейных членов уравнений новой системы линейных уравнений, которая имеет $(0,1)$-решение тогда и только тогда, когда исходная система уравнений имеет $(0,1)$-решение. При этом числа строк и столбцов никогда не возрастают. В цикле выполняются следующие шаги, пока матрица M не стабилизируется или не будет выполнено дополнительное условие остановки, при котором существование или отсутствие $(0,1)$-решения легко проверяется.

1. Удаление из матрицы A нулевых столбцов.
2. Если в некоторой строке матрицы A один ненулевой элемент, который расположен в j-ом столбце, то происходит разбор случаев.
   (a) Если в этой строке элемент столбца $\mathbf{b}$ нулевой, то удаление j-го столбца.

(b) Если в этой строке элементы в j-ом столбце и столбце $\mathbf{b}$ ненулевые и совпадают, то замена столбца $\mathbf{b}$ на разность $\mathbf{b} - \mathbf{a}$, где через $\mathbf{a}$ обозначен j-й столбец в матрице A, а затем удаление j-го столбца.

(c) Если в этой строке элементы в j-ом столбце и столбце $\mathbf{b}$ оба ненулевые и различные, то завершение работы, когда $(0,1)$-решений нет.

Этот шаг повторяется, пока матрица не стабилизируется.

3. Удаление из матрицы M нулевых строк.

4. Если матрица M пустая или дополнительный столбец $\mathbf{b}$ в матрице M нулевой, то завершение работы, когда существует $(0,1)$-решение.

5. Если матрица A пустая или в матрице A присутствует нулевая строка, то завершение работы, когда нет $(0,1)$-решения.

6. Поиск двух линейно зависимых столбцов в матрице A. Если эти столбцы нашлись, то обозначим через j номер одного из них. Иначе вычисляем столбец $\mathbf{c} = 2\mathbf{b} - \sum A_i$, где через A_i обозначен i-й столбец в матрице A и суммируются все столбцы. Если столбец $\mathbf{c}$ ненулевой, то поиск в матрице A столбца, линейно зависимого со столбцом $\mathbf{c}$. Если такой столбец в матрице A найден, то обозначим через j его номер и заменим в матрице A другой столбец на столбец $\mathbf{c}$. Иначе завершение работы.

7. Поиск ненулевого элемента в j-ом столбце. Обозначим через k номер его строки. Умножение k-й строки расширенной матрицы M на элемент в k-й строке и j-ом столбце. Для каждого индекса i в расширенной матрице M вычитание из i-й строки k-й строки, умноженной на элемент из j-го столбца и i-й строки. В результате j-й столбец и k-я строка становятся нулевыми. Далее обозначения A и $\mathbf{b}$ относятся к полученной расширенной матрице M. Переход к первому шагу.

В результате выполнения функции `has01solution`(M) получается либо пустая матрица, либо расширенная матрица новой системы линейных уравнений, которая имеет $(0,1)$-решение тогда и только тогда, когда исходная система уравнений имеет $(0,1)$-решение. Если получена пустая матрица M, то исходная система уравнений имеет $(0,1)$-решение. Если дополнительный столбец $\mathbf{b}$ в матрице M нулевой, то соответствующая система уравнений однородная, следовательно, существует нулевое решение. Иначе, если получена пустая матрица A или в матрице A появилась нулевая строка, хотя соответствующая строка в расширенной матрице M ненулевая, то новая система содержит невыполнимое уравнение $0 = 1$ или $0 = 2$. Если в системе встретилось уравнение $x_j = 2$ или $2x_j = 1$, то $(0,1)$-решений нет. Иначе требуются дополнительные вычисления, которые не реализованы в этой функции. Например, возможна проверка различных вариантов оценки оставшихся переменных. С другой стороны, такой результат может рассматриваться как неопределённый ответ генерического алгоритма.

Поиск среди n ненулевых столбцов линейно зависимых столбцов требует $O(n\log_2(n+1))$ сравнений столбцов между собой. Такой поиск сводится к сортировке $2n$ столбцов, как исходных, так и умноженных на два. Этот поиск может быть реализован, используя метод `numpy.unique`. Однако более практичной оказывается реализация на основе хеширования с использованием встроенного в Python контейнера `set`, ограничивающаяся в случае достаточно большого числа столбцов рассмотрением лишь небольшой их доли, см. [20, 21]. Полное время выполнения ограничено сверху функцией вида $\mathrm{poly}(m)n\log_2(n+1)$.

Рассмотрим пример. Системе уравнений от двух переменных x_1 и x_2

$$\begin{cases} 2x_1 + 2x_2 &= 1 \\ x_1 + 2x_2 &= 1 \end{cases}$$

соответствует расширенная матрица

$$M = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}.$$

Среди первых двух столбцов нет линейно зависимых. Однако возможно применение шага 6 алгоритма. Столбец

$$\mathbf{c} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

пропорционален первому столбцу в матрице M. Заменяя второй столбец, получаем матрицу

$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix}.$$

Поскольку первые два столбца пропорциональны друг другу, элиминация переменных приводит к новой расширенной матрице из одного элемента 2, которая соответствует невыполнимому равенству $0 = 2$. Следовательно, $(0,1)$-решений нет. В ходе работы программы при элиминации переменных сначала получается матрица

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

из которой потом удаляются нулевые столбцы и нулевые строки. В этом примере алгоритм завершает свою работу за два прохода основного цикла, что соответствует теоретической оценке вычислительной сложности.

Для эмпирической оценки эффективности алгоритма, генерировались матрицы, элементы которых независимо и равномерно распределены над полем $GF(3)$. В таблице 1 для числа m строк указаны медиана и 99-й процентиль для числа n первых случайно сгенерированных ненулевых столбцов, среди которых нет линейно зависимых, но следующий столбец оказался линейно зависимым с каким-то из ранее сгенерированных столбцов. Для каждого числа m использовано по 100000 серий столбцов. Погрешность значения медианы составляет около 1%. В последнем столбце таблицы указана верхняя граница числа попарно независимых столбцов, равная $(3^m - 1)/2$. Медиана наибольшего в серии числа столбцов близка к значению функции $(4/5) \cdot (26/15)^m$.

Из таблицы 1 видно, что даже 99-й процентиль заметно ниже верхней границы для числа попарно независимых столбцов. Поэтому алгоритм остаётся эффективным в среднем для систем, в которых число уравнений заметно превышает границу из теоремы 2. Однако даже теорема 4 не позволяет улучшить границу из теоремы 2 в худшем случае.

Различие вычислительной сложности в худшем случае и в среднем ещё заметнее для разреженных матриц. В следующих экспериментах генерировались серии случайных столбцов, в которых каждый элемент отличен от нулевого с фиксированной вероятностью p, а выбор между двумя ненулевыми значениями равновероятен. На рисунке 1 для столбцов из $m \leqslant 25$ элементов показана экспериментально полученная зависимость от вероятности p числа n первых случайно сгенерированных ненулевых столбцов, среди которых нет линейно зависимых, но следующий столбец оказался линейно зависимым с каким-то из ранее сгенерированных столбцов.

Табл. 1. Для числа m строк указаны медиана и 99-й процентиль для числа случайно сгенерированных ненулевых столбцов, среди которых нет линейно зависимых, но следующий столбец оказался линейно зависимым с каким-то из предыдущих столбцов. Также указана верхняя граница числа попарно независимых столбцов.

m	50%	99%	100%
1	1	1	1
2	2	4	4
3	4	9	13
4	7	18	40
5	13	32	121
6	22	57	364
7	39	99	1093
8	67	172	3280
9	117	300	9841
10	202	520	29524
11	351	902	88573
12	605	1561	265720
13	1050	2708	797161
14	1823	4687	2391484
15	3163	8123	7174453
16	5450	14127	21523360
17	9467	24447	64570081
18	16423	42124	193710244
19	28435	73695	581130733
20	49176	126316	1743392200

В таблице 2 экспериментально полученная зависимость этой медианы от вероятности p показана для больших значений m, но лишь для малых значений p и для $p = 1$. При значении вероятности $p = 0.1$ и $m \leqslant 30$ обсуждаемый метод эффективен в среднем для систем уравнений от малого числа переменных. Поэтому наш метод может быть использован для решения приклад-

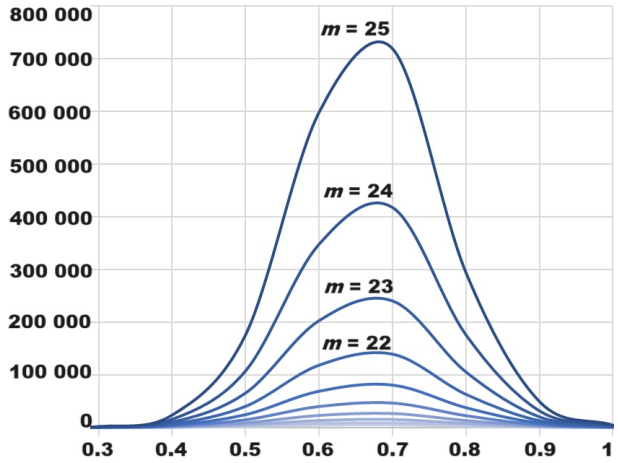

Рис. 1. По оси абсцисс отложена вероятность p, что случайно выбираемый элемент отличен от нулевого. По оси ординат отложена медиана для числа случайно сгенерированных ненулевых столбцов, среди которых нет линейно зависимых, но следующий столбец оказался линейно зависимым с каким-то из предыдущих столбцов. Через m обозначено число элементов в столбце.

ных задач в области математической биологии.

Чтобы эмпирически оценить время работы программы, для различных значений числа строк m и столбцов n случайной матрицы A коэффициентов линейных членов уравнений, вычислялись медианы времени работы при условии получения определённого ответа, когда существование или отсутствие $(0,1)$-решения установлено. Результаты приведены в таблице 3.

Листинг программы и примеры доступны по адресу http://lab6.iitp.ru/-/havoc

4. ЗАКЛЮЧЕНИЕ

Обоснован и реализован алгоритм, позволяющий за полиномиальное время проверить существование $(0,1)$-решения у системы из достаточно малого количества уравнение над полем $GF(3)$. Изложенные результаты согласуются с общепринятой гипотезой о высокой вычислительной сложности задач распознавания $(0,1)$-решений для системы линейных уравнений, поскольку изменение задачи посредством элиминации переменных встречает препятствие в худшем случае. Хотя для системы с малым количеством уравнений над полем $GF(3)$ вычислительная сложность оказывается малой в типичном случае. Экспериментально показано, что алгоритм намного эффективнее для разреженных систем уравнений. Более того, метод двоичного поиска позволяет найти некоторое $(0,1)$-решение системы, когда оно существует, хотя перечисление всех $(0,1)$-решений может быть слишком трудным. Это открывает возможность практического использования для решения тех прикладных задач, которые легко свести к поиску $(0,1)$-решения системы линейных алгебраических уравнений. Например, в работе [22] рассмотрены примеры сводимости комбинаторных задач к задаче о разрешимости системы линейных уравнений над аддитивной полугруппой натуральных чисел.

Системы компьютерной алгебры позволяют

Табл. 2. Для числа m строк и вероятности p, что случайно выбираемый элемент отличен от нулевого, указана медиана для числа случайно сгенерированных ненулевых столбцов, среди которых нет линейно зависимых, но следующий столбец оказался линейно зависимым с каким-то из предыдущих столбцов.

	p				
m	0.1	0.2	0.3	0.4	1.0
10	6	12	25	55	26
11	7	15	34	82	37
12	8	18	45	122	53
13	9	21	60	183	75
14	10	26	81	271	107
15	11	31	109	409	150
16	12	38	147	608	213
17	13	46	198	911	303
18	14	55	267	1371	426
19	16	67	360	2059	603
20	17	80	485	3094	848
21	19	98	658	4659	1204
22	21	118	895	7010	1699
23	23	144	1210	10611	2409
24	25	174	1642	15886	3420
25	27	212	2228	23958	4816
26	30	257	3027	36115	6831
27	33	313	4131	54066	9640
28	36	379	5598	81713	13658
29	40	463	7616	123263	19301
30	44	563	10313	186122	27250

Табл. 3. Для m строк и n столбцов указана медиана в секундах времени работы программы при условии определённого ответа.

	n			
m	10^5	10^6	10^7	10^8
2	0.01	0.1	1	14
3	0.02	0.2	2	23
4	0.03	0.3	3	35
5	0.05	0.5	5	48
6	0.06	0.6	6	62
7	0.08	0.8	8	79
8	0.09	0.9	10	97
9	0.11	1.1	12	117
10	0.13	1.3	14	139
11	0.15	1.6	16	162
12	0.17	1.8	18	187
13	0.2	2	21	214
14	0.23	2.2	24	243
15	0.28	2.5	26	273
16	0.33	2.9	28	304
17	0.4	3.2	32	334
18	0.49	3.6	35	369
19	0.67	4.1	39	405
20	0.9	4.7	43	445
21	1.18	5.5	47	479
22	1.46	6.5	52	518
23	1.8	8.1	58	563
24	2.06	10.4	63	609

проводить вычисления над полем вычетов по простому модулю. Это позволяет встроить новую программу в пайплайн для обработки данных, в частности, для решения задач математической биологии.

5. БЛАГОДАРНОСТИ

Работа выполнена с использованием вычислительных ресурсов Межведомственного суперкомпьютерного центра Российской академии наук (МСЦ РАН).

6. ФИНАНСИРОВАНИЕ

Исследование выполнено за счет гранта Российского научного фонда № 24-44-00099, https://rscf.ru/project/24-44-00099/.

СПИСОК ЛИТЕРАТУРЫ

1. *Селиверстов А.В.* Обобщение задачи о сумме подмножеств и кубические формы // Журнал вычислительной математики и математической физики. 2023. Т. 63. № 1. С. 51–60.

2. *Бойков А.А., Селиверстов А.В.* О кубе и проекциях подпространства // Вестн. Удмуртск. ун-та. Матем. Мех. Компьют. науки. 2023. Т. 33. № 3. С. 402–415. DOI: 10.35634/vm230302

3. *Kárteszi F.* Introduction to Finite Geometries. Budapest: Akadémian Kiadó, 1976.

4. *Feng T., Lu J.* New families of flag-transitive linear spaces // Finite Fields and Their Applications. 2023. Vol. 87. No. 102156. P. 1–10. DOI: 10.1016/j.ffa.2022.102156

5. *Stoichev S.D., Gezek M.* Unitals in projective planes of order 25 // Mathematics in Computer Science. 2023. Vol. 17. No. 5. P. 1–19. DOI: 10.1007/s11786-023-00556-9

6. *Когабаев Н.Т.* О системах диофантовых уравнений над конечными конфигурациями // Сибирский математический журнал. 2023. Т. 64. № 2. С. 321–338.

7. *Байрамов Р.Э., Блинков Ю.А., Левичев И.В., Малых М.Д., Мележик В.С.* Аналитическое исследование кубатурных формул на сфере в системах компьютерной алгебры // Журнал вычислительной математики и математической физики. 2023. Т. 63. № 1. С. 93–101.

8. *Hesse O.* Über die Elimination der Variabeln aus drei algebraischen Gleichungen vom zweiten Grade mit zwei Variabeln // Journal für die Reine und Angewandte Mathematik. 1844. Vol. 28. P. 68–96. DOI: 10.1515/crll.1844.28.68

9. *Cacchiani V., Iori M., Locatelli A., Martello S.* Knapsack problems — An overview of recent advances. Part I: Single knapsack problems // Computers and Operations Research. 2022. Vol. 143. No. 105692. P. 1–13. DOI: 10.1016/j.cor.2021.105692

10. *Cacchiani V., Iori M., Locatelli A., Martello S.* Knapsack problems — An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems // Computers and Operations Research. 2022. Vol. 143. No. 105693. P. 1–14. DOI: 10.1016/j.cor.2021.105693

11. *Zhang L., Quweider M., Khan F., Lei H.* Splitting NP-complete sets infinitely // Information Processing Letters. 2024. Vol. 186. No. 106472. P. 1–7. DOI: 10.1016/j.ipl.2024.106472

12. *Vyalyi M.N.* Testing the satisfiability of algebraic formulas over the field of two elements // Probl. Inf. Transm. 2023. Vol. 59. P. 57–62. DOI: 10.1134/S0032946023010052

13. *Яшунский А.Д.* О суммах бернуллиевских случайных величин по модулю 3 // Математические заметки. 2022. Т. 111. № 1. С. 154–157. DOI: 10.4213/mzm13214

14. *Sanna C.* On the distribution of the entries of a fixed-rank random matrix over a finite field // Finite Fields and Their Applications. 2024. Vol. 93. No 102333. P. 1–15. DOI: 10.1016/j.ffa.2023.102333

15. *Балакин Г.В.* Распределение ранга случайных матриц над конечным полем // Теория вероятностей и ее применения. 1968. Т. 13. № 4. С. 631–641.

16. *Круглов В.И., Михайлов В.Г.* О ранге случайной матрицы над простым полем, состоящей из независимых строк с заданными числами ненулевых элементов // Математические вопросы криптографии. 2020. Т. 11. № 3. С. 41–52. DOI: 10.4213/mvk331

17. *Cooper C.* On the distribution of rank of a random matrix over a finite field // Random Structures and Algorithms. 2000. Vol. 17. No. 3-4. P. 197–212. DOI: 10.1002/1098-2418(200010/12)17:3/4<197::AID-RSA2>3.0.CO;2-K

18. *Рыбалов А.* Генерические полиномиальные алгоритмы для проблемы о рюкзаке в некоторых матричных полугруппах // Сибирские электронные математические известия. 2023. Т. 20. № 1. С. 100–109.

19. *Рыбалов А.Н.* Генерически неразрешимые и трудноразрешимые проблемы // Прикладная дискретная математика. 2024. № 63. С. 109–116.

20. *Nayak S., Patgiri R.* A review on role of Bloom filter on DNA assembly // IEEE Access. 2019. Vol. 7. P. 66939–66954. DOI: 10.1109/ACCESS.2019.2910180

21. *Bille P., Gørtz I.L., Stordalen T.* Predecessor on the ultra-wide word RAM // Algorithmica. 2024. Vol. 86. P. 1578–1599. DOI: 10.1007/s00453-023-01193-1

22. *Рыбалов А.Н.* О генерической сложности решения уравнений над натуральными числами со сложением // Прикладная дискретная математика. 2024. № 64. С. 72–78.

On binary solutions to a system of linear equations modulo three

O. A. Zverkov *, A. V. Seliverstov **

Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Bolshoy Karetny per. 19, build. 1, Moscow 127051 Russia

* *e-mail: zverkov@iitp.ru*,
ORCID: 0000-0002-8546-364X

** *e-mail: slvstv@iitp.ru*,
ORCID: 0000-0003-4746-6396

We consider the problem of finding a binary solution to a system of linear equations modulo three. In case the number of equations is less than a sufficiently slowly growing function of the number of variables, a new polynomial-time algorithm is proposed to recognize the existence of a binary solution to such a system. The algorithm is based on the note that if the coefficient matrix contains non-zero columns proportional to each other, then the elimination of the corresponding variables preserves the property of having no binary solution to the system. In particular, every system of two equations in five variables allows the elimination of some variables that preserves the property of having no binary solution to the system. Based on these results, we propose an errorless heuristic algorithm, which is implemented using the Python programming language. The NumPy library is used to represent matrices and perform basic operations. The input is the augmented matrix. An empirical running time estimate has been calculated using the implementation. It has been experimentally shown that the algorithm is more efficient for sparse systems of equations. Obviously, the binary search method allows finding a binary solution to the system when one exists. This observation opens up the possibility of practical use, in particular, for solving problems of mathematical biology.

Keywords: finite field, system of linear equations, computer algebra system.

7. REFERENCES

1. *Seliverstov A.V.* Generalization of the subset sum problem and cubic forms. Computational Mathematics and Mathematical Physics, 2023, vol. 63, no. 1, pp. 48–56. DOI: 10.1134/S0965542523010116

2. *Boykov A.A., Seliverstov A.V.* On a cube and subspace projections. Vestn. Udmurtsk. Univ. Mat. Mekh. Komp. Nauki, 2023, vol. 33, no. 3, pp. 402–415. DOI: 10.35634/vm230302

3. *Kárteszi F.* Introduction to Finite Geometries. Budapest: Akadémian Kiadó, 1976.

4. *Feng T., Lu J.* New families of flag-transitive linear spaces. Finite Fields and Their Applications, 2023, vol. 87, no. 102156, pp. 1–10. DOI: 10.1016/j.ffa.2022.102156

5. *Stoichev S.D., Gezek M.* Unitals in projective planes of order 25. Mathematics in Computer Science, 2023, vol. 17, no. 5, pp. 1–19. DOI: 10.1007/s11786-023-00556-9

6. *Kogabaev N.T.* Systems of Diophantine equations over finite configurations. Siberian Math. J., 2023, vol. 64, no. 2, pp. 325–337. DOI: 10.1134/S0037446623020076

7. *Bairamov R.E., Blinkov Yu.A., Levichev I.V., Malykh M.D., Melezhik V.S.* Analytical study of cubature formulas on a sphere in computer algebra systems. Computational Mathematics and Mathematical Physics, 2023, vol. 63, no. 1, pp. 77–85. DOI: 10.1134/S0965542523010050

8. *Hesse O.* Über die Elimination der Variabeln aus drei algebraischen Gleichungen vom zweiten Grade mit zwei Variabeln. Journal für die Reine und Angewandte Mathematik, 1844, vol. 28, pp. 68–96. DOI: 10.1515/crll.1844.28.68

9. *Cacchiani V., Iori M., Locatelli A., Martello S.* Knapsack problems — An overview of recent advances. Part I: Single knapsack problems. Computers and Operations Research, 2022, vol. 143, no. 105692, pp. 1–13. DOI: 10.1016/j.cor.2021.105692

10. *Cacchiani V., Iori M., Locatelli A., Martello S.* Knapsack problems — An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems. Computers and Operations Research, 2022, vol. 143, no. 105693, pp. 1–14. DOI: 10.1016/j.cor.2021.105693

11. *Zhang L., Quweider M., Khan F., Lei H.* Splitting NP-complete sets infinitely. Information Processing Letters, 2024, vol. 186, no. 106472, pp. 1–7. DOI: 10.1016/j.ipl.2024.106472

12. *Vyalyi M.N.* Testing the satisfiability of algebraic formulas over the field of two elements. Probl. Inf. Transm., 2023, vol. 59, pp. 57–62. DOI: 10.1134/S0032946023010052

13. *Yashunskii A.D.* On sums of Bernoulli random variables modulo 3. Mathematical Notes, 2022, vol. 111, no. 1, pp. 166–169. DOI: 10.1134/S0001434622010205

14. *Sanna C.* On the distribution of the entries of a fixed-rank random matrix over a finite field. Finite Fields and Their Applications, 2024, vol. 93, no. 102333, pp. 1–15. DOI: 10.1016/j.ffa.2023.102333

15. *Balakin G.V.* The distribution of the rank of random matrices over a finite field. Theory of Probability and its Applications, 1968, vol. 13, no. 4, pp. 594–605. DOI: 10.1137/1113076

16. *Kruglov V.I., Mikhailov V.G.* On the rank of random matrix over prime field consisting of independent rows with given numbers of nonzero elements. Mat. Vopr. Kriptogr., 2020, vol. 11, no. 3, pp. 41–52. DOI: 10.4213/mvk331

17. *Cooper C.* On the distribution of rank of a random matrix over a finite field. Random Structures and Algorithms, 2000, vol. 17, no. 3-4, pp. 197–212. DOI: 10.1002/1098-2418(200010/12)17:3/4<197::AID-RSA2>3.0.CO;2-K

18. *Rybalov A.N.* Generic polynomial algorithms for the knapsack problem in some matrix semigroups. Siberian Electronic Mathematical Reports, 2023, vol. 20, no. 1, pp. 100–109. (In Russian)

19. *Rybalov A.N.* Generically undecidable and hard problems. Prikl. Diskr. Mat., 2024, no. 63, pp. 109–116. (In Russian)

20. *Nayak S., Patgiri R.* A review on role of Bloom filter on DNA assembly. IEEE Access, 2019, vol. 7, pp. 66939–66954. DOI: 10.1109/ACCESS.2019.2910180

21. *Bille P., Gørtz I.L., Stordalen T.* Predecessor on the ultra-wide word RAM. Algorithmica, 2024, vol. 86, pp. 1578–1599. DOI: 10.1007/s00453-023-01193-1

22. *Rybalov A.N.* On the generic complexity of solving equations over natural numbers with addition. Prikl. Diskr. Mat., 2024, no. 64, pp. 72–78. (In Russian)