ОПРЕДЕЛЕНИЕ НУЛЕВЫХ БИТ ЗАДАЧИ 3-ВЫПОЛНИМОСТЬ, АССОЦИИРОВАННОЙ С ЗАДАЧЕЙ ФАКТОРИЗАЦИИ

Огородников Ю.Ю., Файзуллин Р.Т.
Омский государственный технический университет

Аннотация

В работе приведены два эвристических способа распознавания нулевых бит задачи ВЫПОЛНИМОСТЬ, ассоциированной с задачей факторизации. Первый основывается на сведении задачи ВЫПОЛНИМОСТЬ к эквивалентной задаче минимизации непрерывной гладкой функции методом последовательных приближений. В свою очередь, данный метод расширяется путём изменения порядка вычисления переменных. Другой способ заключается в сведении к системе линейных алгебраических уравнений с симметричной матрицей диагонального преобладания.

Ключевые слова: ВЫПОЛНИМОСТЬ, метод последовательных приближений, изменение порядка обхода переменных, диагональный способ, факторизация.

Введение

Задача выполнимости булевых формул (SAT или ВЫП) занимает центральное место в теории вычислительной сложности, важнейшем разделе математики. В 1971 году С. Кук доказал, что задача SAT, записанная в конъюнктивной нормальной форме, является NP-полной, т. е. другие задачи, лежащие в классе NP, за полиномиальное время сводятся к задаче SAT [1]. В частности, среди таких задач стоит выделить задачи факторизации, дискретного логарифмирования и логарифмирования на эллиптической кривой [2]. В силу того, что в настоящее время неизвестны эффективные алгоритмы решения данных проблем, вышеупомянутые задачи имеют огромное значение в криптографии.

На вычислительной сложности задачи факторизации основан алгоритм RSA, используемый как для шифрования, так и для цифровой подписи. Дискретное логарифмирование лежит в основе протокола Диффи–Хеллмана, позволяющего выработать двум сторонам общий секретный ключ, используя не защищённый от прослушивания канал связи. Также на данной задаче основана электронная подпись Эль-Гамаля и криптосистема Мэсси–Омуры [3].

В настоящее время известно и практикуется сведение данных задач к эквивалентным экземплярам задачи SAT [2]. В частности, задача факторизации сводится к 3-SAT. Таким образом, выполняющий набор для 3-SAT будет соответствовать решению эквивалентной ей задачи факторизации, что позволит, в частности, провести атаку на алгоритм RSA [4]. Проблема заключается в том, что в настоящее время неизвестен полиномиальный алгоритм, обеспечивающий поиск решения задачи SAT в приемлемые временные рамки. Экземпляры задачи SAT небольших размерностей могут быть решены переборными методами, однако с увеличением размерности время решения растёт экспоненциально.

Существует множество подходов к разработке алгоритма поиска решения задачи SAT. Один из них основан на усовершенствовании переборных методов, в частности, алгоритм Дэвиса–Патнема–Логемана–Лавленда (DPLL) [5], являющийся модификацией более раннего алгоритма Дэвиса–Патнема, основанного на методе резолюций.

Другой базируется на сведении задачи поиска выполняющего набора истинности к задаче минимизации непрерывной (не обязательно выпуклой) гладкой функции с последующим применением к ней методов непрерывной оптимизации. В частности, в работе [6] к получившейся функции применяется метод последовательных приближений.

К сожалению, далеко не всегда удаётся получить точное решение, однако существует возможность накопления статистических данных с последующим их использованием. В работе [7] приведены результаты накопления и анализа статистических данных, полученных в результате работы гибридного алгоритма поиска приближённого решения задачи SAT.

В данной статье рассматриваются два **эвристических** способа определения нулевых бит задачи 3-SAT, ассоциированной с задачей факторизации целых чисел. Первый способ является модификацией одного из компонентов гибридного алгоритма – метода последовательных приближений. Второй метод основан на построении симметричной матрицы с диагональным преобладанием, исходя из информации о дизъюнктах, входящих в 3-КНФ.

Полученная в результате тестирования статистика позволяет составить тест для отдельных битов приближённого решения. Так, если для некоторого бита y_i частота совпадения с соответствующим битом точного решения довольно велика, то мы можем утверждать, что в результате выполнения предложенных методов y_i принимает верное значение, в противном же случае про бит y_i ничего не известно. К сожалению, в силу особенностей выполнения методов речь может идти только об определении нулевых бит.

1. Предварительные сведения

В работе [7] описывается гибридный алгоритм, состоящий из двух стадий: сегментного генетического алгоритма и метода последовательных приближений. На стадии работы сегментного генетического алгоритма проводится аналогия с живой природой и поиск решения задачи ВЫПОЛНИМОСТЬ производится в соответствии с законами эволюции. На следующей же стадии происходит переход от задачи SAT к эквивалентной непрерывной функции и запус-

кается процесс её минимизации. Рассмотрим эту стадию более подробно.

Пусть имеется булева функция, определённая на множестве булевых векторов $y=(y_1,...,y_N)$:

$$L^*(y)=\bigwedge_{i=1}^{M} G_i^*(y), \quad (1)$$

где $G_i^*(y)=\bigvee_{j=1}^{N} q_{i,j}^*(y)$,

$$q_{i,j}^*(y)=\begin{cases} y_j, & \text{если переменная } y_j \\ & \text{входит в i-й дизъюнкт непосредственно} \\ \overline{y_j}, & \text{если переменная } y_j \\ & \text{входит в i-й дизъюнкт с отрицанием} \\ \textit{ложь}, \textit{иначе} & \end{cases}.$$

Необходимо отметить, что данная булева формула имеет **единственный** выполняющий набор [5, 6].

Перейдём от 3-КНФ к эквивалентной 3-ДНФ:

$$L(y)=\overline{L^*(y)}=\bigvee_{i=1}^{M} G_i(y), \quad (2)$$

$$G_i(y)=\bigwedge_{j=1}^{N} q_{i,j}(y) \text{ и } q_{i,j}(y)=\overline{q_{i,j}^*(y)}.$$

Вектору булевых переменных y сопоставим вектор вещественных переменных $x=[x_1,...x_N]$

$$x_j=\begin{cases} 1, \text{если } y_j=true \\ 0, \text{иначе} \end{cases}.$$

Форме $L(y)$ сопоставим функцию $F:[0,1]^N\to R_+$ следующего вида:

$$F(x)=\sum_{i=1}^{M} C_i(x), \quad (3)$$

где $C_i(x)=\prod_{j=1}^{N} p_{i,j}(x)$ и

$$p_{i,j}(x)=\begin{cases} x_j^2, & \text{если } q_{i,j}^*(y)=\overline{y_j} \\ (1-x_j)^2, & \text{если } q_{i,j}^*(y)=y_j \\ 1, & \text{иначе} \end{cases}.$$

Дифференцируем $F(x)$, тем самым получая градиент ∇F, и приравниваем к нулю все его компоненты.

$$\Big(\sum_{k\in T(j)}\prod_{l=1,l\neq j}^{N}(p_{k,l}(x))x_j - \sum_{k\in H(j)}\prod_{l=1,l\neq j}^{N} p_{k,l}(x)=0 \quad (j=1,...N), \quad (4)$$

в которой $T(j)=\{k\in\{1,...M\}: p_{k,j}(x)\neq 1\}$ и $H(j)=\{k\in\{1,...M\}: p_{k,j}(x)=(1-x_j)^2\}$.

Приводим полученную систему уравнений (4) к виду

$$A^*x=B^*, \quad (5)$$

где A^* – матрица коэффициентов, B^* – столбец свободных членов.

Применяя к системе (5) метод последовательных приближений [8], получаем приближённое решение $x^*=(x_1,x_2,...,x_n)$. Стоит отметить, что предложенный метод не всегда находит глобальный минимум $F(x)$ из-за наличия точек локального минимума, так как траектория, образованная последовательными приближениями, имеет тенденцию сходиться к ближайшей точке локального минимума [6]. В данном случае сходимость к локальному экстремуму происходит за малое количество итераций (эвристически установлено, что в общем случае число таких итераций равняется 6).

При этом точки локального минимума не являются решением задачи ВЫПОЛНИМОСТЬ. Более того, для SAT, ассоциированной с задачей факторизации, почти все компоненты вектора-приближения оказываются равными 0.

Существуют различные способы преодоления локальных экстремумов, например, инвертирование значений случайно выбранной группы бит, метод смены траектории, рестарт [6]. Все вышеперечисленные методы не всегда преодолевают локальный минимум, поэтому поиск точного решения задачи ВЫПОЛНИМОСТЬ может занимать достаточно большое количество времени.

Более экономным по времени выполнения является остановка «на спуске» в локальный экстремум-ловушку, т.е. за одну итерацию до попадания в «овраг». При этом выдерживается общее направление поиска глобального минимума. Точное решение достигнуто при этом не будет, но, проведя серию испытаний, можно с определённой вероятностью гарантировать точное определение отдельных битов. Эвристически получено, что такая точка в большинстве случаев достигается за 6 итераций.

После проведения серии таких испытаний оказалось, что некоторые биты приближённого решения $x^*=(x_1,x_2,...x_n)$ с достаточно высокой частотой (больше либо равной 0,9) совпадают с битами точного решения.

2. Метод последовательных приближений с изменением порядка вычислений переменных

Рассмотрим модификацию данного способа с целью повысить число верно определяемых бит.

Возьмём перестановку

$$\sigma=\begin{pmatrix} 1 & 2 & ... & n \\ \sigma(1) & \sigma(2) & ... & \sigma(n) \end{pmatrix}$$

порядка n, где n – размерность матрицы A^*. Используем эту перестановку для изменения порядка вычисления переменных $x_1,x_2,....x_n$ (также будем называть этот приём *изменением пути обхода*).

Очевидно, что подобная операция не изменит множества решений системы (5). Выше было отмечено, что из-за наличия точек локального минимума и тенденции сходимости метода последовательных приближений к таким точкам не всегда удаётся полу-

чить точное решение системы (5). Изменение порядка вычислений переменных также ведёт в «овраг», однако если мы снова остановимся в шаге от итерации, ведущей в локальный минимум, то сможем зафиксировать новую точку, которую можно использовать для получения новых данных о совпадении бит приближённого решения с битами точного решения.

Алгоритм 1.1. Определение частот совпадения бит методом последовательных приближений с изменением порядка вычисления переменных.

Входные данные: M – количество экземпляров задачи ВЫПОЛНИМОСТЬ, для которых будет применяться метод последовательных приближений.

Шаг 1. Положить $i=1$.

Шаг 2. Задать перестановку σ, пользуясь одним из известных алгоритмов [9].

Шаг 3. Произвести переход от задачи ВЫПОЛНИМОСТЬ к задаче минимизации непрерывной функции [3]. На выходе получается система уравнений (5).

Шаг 4. Изменить порядок вычисления переменных в соответствии с перестановкой σ.

Шаг 5. Решить систему (5) методом последовательных приближений, результатом будет вектор-приближение $x^*_\sigma = (x_1, x_2, ..., x_n)$.

Шаг 6. Сравнить вектор x^*_σ с точным решением x^T и сформировать множество

$$\lambda_i = \{ j \mid x^*_\sigma(j) = x^T(j) \}.$$

Шаг 7. Положить $i = i+1$. Если $i \le M$, то произвести переход к шагу 1.

Шаг 8. Вычислить значения $\nu_k, k = 1..n$ следующим образом:

$$\nu_k = \frac{1}{M}\sum_{j=1}^{M} Ind(\lambda_j, k),$$

где $Ind(\lambda_j, k) = \begin{cases} 0, k \notin \lambda_j \\ 1, k \in \lambda_j \end{cases}$.

Выходные данные: вектор частот $\nu = (\nu_1, \nu_2, ..., \nu_n)$.

Конец алгоритма 1.1.

Логичной представляется мысль по увеличению числа путей обхода переменных.

Алгоритм 1.2. Определение частот совпадения бит путём метода последовательных приближений с использованием заданного числа путей обхода.

Входные данные: N_π – число перестановок.

Шаг 1. Положить $i=1$.

Шаг 2. Выполнить действия алгоритма 1.

Шаг 3. Добавить результат работы алгоритма 1 в множество векторов частот $V = \{\nu_i\}$.

Шаг 4. Положить $i = i+1$. Если $i \le N_\pi$, то перейти к шагу 3.

Выходные данные: множество векторов частот $V = \{\nu_i\}$.

Конец алгоритма 1.2.

Пользуясь результатами работы алгоритма 1.2, модифицируем применение метода последовательных приближений к задаче ВЫПОЛНИМОСТЬ.

Алгоритм 2. Поиск приближённого решения системы уравнений, эквивалентных задаче ВЫПОЛНИМОСТЬ, с использованием множества векторов частот совпадения бит.

Входные данные: множество векторов частот $V = \{\nu_i\}$, полученное в результате выполнения алгоритма 1.2.

Шаг 1. Произвести переход от задачи ВЫПОЛНИМОСТЬ к задаче минимизации непрерывной функции.

Шаг 2. Применить к системе уравнений, полученной на предыдущем шаге, метод последовательных приближений. Обозначить результат как $x = (x_1, x_2, ...x_n)$.

Шаг 3. Определить функции

$$Q^0(\nu_i, x_i) = \begin{Bmatrix} \nu_i, & x_i = 0 \\ 1-\nu_i, & x_i = 1 \end{Bmatrix}$$

и $Q^1(\nu_i, x_i) = \begin{Bmatrix} \nu_i, & x_i = 1 \\ 1-\nu_i, & x_i = 0 \end{Bmatrix}$.

Шаг 4. Вычислить значения

$$z_j^0 = \frac{1}{N_\pi}\sum_{i=1}^{N_\pi} Q^0(\nu_{i_j}, x_j) \text{ и } z_j^1 = \frac{1}{N_\pi}\sum_{i=1}^{N_\pi} Q^1(\nu_{i_j}, x_j),$$

где $j = 1..n$.

Шаг 5. Сформировать компоненты нового приближения $\tilde{x}_j = \begin{Bmatrix} 1, z_j^1 \ge z_j^0 \\ 0, \ иначе \end{Bmatrix}$, $j = 1..n$.

Выходные данные: вектор-приближение $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_n)$.

Конец алгоритма 2.

Пояснения к алгоритму 2.

a) На шаге 3 вводятся две функции: $Q^0(\nu_i, x_i)$ и $Q^1(\nu_i, x_i)$. Аргументами у них являются частота совпадения *i*-го бита ν_i и значение бита, стоящего на той же позиции *i*, полученное на шаге 2. В зависимости от значения x_i эти функции возвращают либо значение самой частоты ν_i, либо величину $1-\nu_i$. Объясняется это спецификой задачи ВЫПОЛНИМОСТЬ: к примеру, если значением *i*-го бита оказалась 1, а частота у него 0, то мы с частотой 1 определяем, что данный бит имеет значение 0 (в силу того, что биты могут принимать только два значения – 0 и 1).

b) На шаге 4 функции $Q^0(\nu_i, x_i)$ и $Q^1(\nu_i, x_i)$ используются для подсчёта средней частоты для каждого бита на основании вектора-приближения $x = (x_1, x_2, ...x_n)$. Следует обратить внимание на первый аргумент у данных функций ν_{i_j}. Запись означает, что из множества векторов частот $V = \{\nu_i\}$ выбрали вектор ν_i, соответствующий перестановке σ_i. Индекс *j* означает, что в векторе частот ν_i выбран бит под номером *j*.

c) На шаге 5 происходит формирование нового приближения в зависимости от того, какая величина больше: z_j^0 или z_j^1. Это можно назвать «принципом голосования».

Данное приближение можно использовать как стартовое для других методов.

Существует альтернативный способ определения частоты совпадения нулевых бит, основанный на сведении задачи SAT к системе линейных уравнений [10].

3. Диагональный способ определения частот совпадения бит

Пусть дана КНФ:

$$L(y) = \bigwedge_{i=1}^{M} C_i, \qquad (6)$$

где C_i – дизъюнкции вида $\vee q_{i,j}$. Здесь $q_{i,j} = y_j$ или $q_{i,j} = \overline{y_j}$. Отметим, что каждой дизъюнкции можно поставить в соответствие число f_i, равное количеству литералов, принимающих значение ИСТИНА.

В применении к 3-КНФ, ассоциированной с задачей факторизации целых чисел, число f_i может принимать значения 1, 2, 3. Значение 1 может быть в 3 случаях, значение 2 – также в 3 случаях, а значение 3 возможно только в 1 случае. Значение 0 приниматься не может, т. к. рассматриваемая 3-КНФ всегда имеет выполняющий набор, а значит, все дизъюнкты имеют хотя бы один литерал, принимающий значение ИСТИНА.

Математическое ожидание f_i равняется $\frac{3}{7}\cdot 1 + \frac{3}{7}\cdot 2 + \frac{3}{7}\cdot 3 = \frac{12}{7}$. Это число заключено в интервале $(1{,}5; 2)$, и ближайшее целое число – 2, поэтому сделаем предположение, что в среднем числа f_i принимают значение 2.

На основании 3-КНФ и чисел f_i построим систему линейных уравнений

$$Bx = g. \qquad (7)$$

Алгоритм 3. Построение системы линейных уравнений на основе 3-КНФ, эквивалентной задаче факторизации целых чисел.

Входные данные: 3-КНФ, состоящая из N переменных и M дизъюнктов.

Шаг 1. Инициализировать матрицу B и вектор-столбец g нулями.

Шаг 2. Положить $i = 1$.

Шаг 3. Положить $j = 1$.

Шаг 4. Ввести функцию $sign(q_{i,j})$, отвечающую за знак литерала $q_{i,j}$. Она принимает значение 1, если литерал $q_{i,j}$ входит в дизъюнкцию C_i без отрицания, и -1 в противном случае.

Шаг 5. Обозначить $d_1 = q_{i,j}$.

Шаг 6. Положить $k = 1$.

Шаг 7. Обозначить $d_2 = q_{i,k}$.

Шаг 8. Добавить к элементу $b_{d_1,d2}$ матрицы B значение $sign(d_1) * sign(d_2)$.

Шаг 9. Добавить к элементу g_{d_1} вектор-столбца g значение $sign(d_1) * 2$.

Шаг 10. Положить $k = k + 1$. Если $k \le 3$, то перейти к шагу 7.

Шаг 11. Положить $j = j + 1$. Если $j \le 3$, то перейти к шагу 4.

Шаг 12. Положить $i = i + 1$. Если $i \le M$, то перейти к шагу 3.

Выходные данные: матрица B размером $N \times N$ и вектор-столбец g размером $N \times 1$.

Конец алгоритма 3.

Поясним работу алгоритма. В цикле $i = 1..M$ перебираются все дизъюнкты, входящие в 3-КНФ. Для каждого дизъюнкта рассматриваются всевозможные комбинации литералов по два, учитывая порядок. Всего таких пар 9, и для каждой пары литералов (d_1, d_2) мы добавляем к элементу формируемой матрицы $b_{d_1,d2}$ значение $sign(d_1) * sign(d_2)$. Таким образом, элемент $b_{d_1,d2}$ может либо увеличиться, либо уменьшиться на единицу. Заметим, что диагональные элементы матрицы B всегда будут увеличиваться на единицу, в то время как остальные попеременно увеличиваются и уменьшаются на 1. Также данная матрица является симметричной, что видно из алгоритма построения.

После серии экспериментов было установлено, что матрица B обладает диагональным преобладанием и является разреженной.

Например, для 3-КНФ

$$(y_1 \vee \overline{y_2} \vee y_3)(y_1 \vee \overline{y_3} \vee \overline{y_4})(\overline{y_1} \vee \overline{y_2} \vee y_4)(y_1 \vee y_2 \vee \overline{y_4})$$

матрица B имеет следующий вид:

$$B = \begin{pmatrix} 4 & 1 & 0 & -3 \\ 1 & 3 & -1 & -2 \\ 0 & -1 & 2 & 1 \\ -3 & -2 & 1 & 3 \end{pmatrix}.$$

Обратим теперь внимание на правую часть g, которую необходимо построить синхронно с матрицей B. Она получается путём сложения чисел f_i. Выше было показано, что в среднем f_i принимают значение 2, поэтому будем строить правую часть, прибавляя и отнимая число 2 одновременно с построением матрицы B.

К полученной системе уравнений применяется метод Гаусса–Зейделя, и полученное вещественное решение $x = (x_1, x_2, ..., x_N)$ преобразуется в булево $y = (y_1, y_2, ...y_N)$ следующим образом: если $|x_i| \le 0{,}1$, то $y_i = ЛОЖЬ$, и $y_i = ИСТИНА$ иначе.

Численные эксперименты показали, что вещественные компоненты, для которых $|x_i| \geq 0{,}1$, с большой частотой ($\geq 90\%$) совпадают с соответствующими булевыми компонентами. В остальных же случаях частота совпадения – около 50 %, и мы ничего не можем утверждать.

4. Численные эксперименты

В данном разделе приведены результаты тестирования метода последовательных приближений с изменением порядка вычислений переменных и диагонального способа.

Цель тестирования заключалась в выявлении нулевых бит с высокой частотой устойчивости. Эксперименты проводились на экземплярах 3-КНФ, эквивалентных задаче факторизации. При этом гарантируется, что все 3-КНФ различны и имеют единственный выполняющий набор.

В табл. 1–4 приведены результаты тестирования метода последовательных приближений с изменением порядка вычисления переменных (алгоритм 2 данной статьи) для 3-КНФ, эквивалентных задаче факторизации числа размерности 100, 200, 300, 400 бит соответственно.

Табл. 1. Результаты работы алгоритма 2 для 3-КНФ, содержащей N=14700 переменных и M=58200 дизъюнктов. Размер факторизуемого числа – 100 бит

Частота совпадения ν	Число верно определённых нулевых бит N_1, удовлетворяющих заданной частоте	Общее число определённых нулевых бит N_2	Отношение N_1 к N_2
0,85	3476	3822	0,909
0,86	3012	3309	0,91
0,865	2735	3008	0,909
0,866	2382	2618	0,9098
0,867	2382	2618	0,9098
0,868	2382	2618	0,9098
0,869	2382	2618	0,9098
0,87	2382	2618	0,9098
0,875	2043	2246	0,9052
0,88	1674	1844	0,909
0,885	1364	1503	0,9075
0,9	598	646	0,925

Для каждой размерности факторизуемого числа $n = pq$ применялось 100 различных путей обхода. В свою очередь, для каждого пути обхода использовалось 400 случайных постановок задачи SAT, полученной применением качественного генератора случайных чисел.

Как видно из полученных результатов, отношение N_1 к N_2 остаётся постоянным при увеличении размерности задачи.

В табл. 5 приведено сравнение алгоритма 2 с результатами обычного метода последовательных приближений.

Отношение N_1 к N_2 растёт с увеличением требуемой частоты устойчивости ν, однако в целом можно говорить о лучших результатах метода последовательных приближений с изменением порядка обхода по сравнению с предыдущим способом.

Табл. 2. Результаты работы алгоритма 2 для 3-КНФ, содержащей N=59400 переменных и M=236400 дизъюнктов. Размер факторизуемого числа – 200 бит

Частота совпадения ν	Число верно определённых нулевых бит N_1, удовлетворяющих заданной частоте	Общее число определённых нулевых бит N_2	Отношение N_1 к N_2
0,85	14158	16056	0,8817
0,86	12506	14195	0,88101
0,865	11435	12991	0,8802
0,866	10195	11584	0,88009
0,867	10195	11584	0,88009
0,868	10195	11584	0,88009
0,869	10195	11584	0,88009
0,87	10195	11584	0,88009
0,875	8876	10089	0,8797
0,88	7512	8539	0,8797
0,885	6167	7027	0,8776
0,9	2677	3075	0,8705

Табл. 3. Результаты работы алгоритма 2 для 3-КНФ, содержащей N=134100 переменных и M=534600. Размер факторизуемого числа – 300 бит

Частота совпадения ν	Число верно определённых нулевых бит N_1, удовлетворяющих заданной частоте	Общее число определённых нулевых бит N_2	Отношение N_1 к N_2
0,85	30738	34801	0,8832
0,86	27139	30744	0,8827
0,865	24755	28025	0,8833
0,866	22105	25018	0,8835
0,867	22105	25018	0,8835
0,868	22105	25018	0,8835
0,869	22105	25018	0,8835
0,87	22105	25018	0,8835
0,875	19276	21834	0,8828
0,88	16433	18596	0,8836
0,885	13494	15277	0,8832
0,9	6215	7042	0,8825

Табл. 4. Результаты работы алгоритма 2 для 3-КНФ, содержащей N=238800 переменных и M=952802. Размер факторизуемого числа – 400 бит

Частота совпадения ν	Число верно определённых нулевых бит N_1, удовлетворяющих заданной частоте	Общее число определённых нулевых бит N_2	Отношение N_1 к N_2
0,85	52346	59222	0,8839
0,86	46051	52118	0,8836
0,865	44245	50074	0,8836
0,866	41363	46807	0,8837
0,867	41363	46807	0,8837
0,868	41363	46807	0,8837
0,869	41363	46807	0,8837
0,87	41363	46807	0,8837
0,875	37390	42340	0,8831
0,88	35249	39907	0,8833
0,885	32307	36713	0,88
0,9	10961	12442	0,881

Табл. 5. Сравнение метода последовательных приближений и алгоритма 2

Частота совпадения ν	Число верных нулевых бит N_1 для обычного метода последовательных приближений	Число верных нулевых бит N_2 для метода последовательных приближений с изменением порядка обхода	Отношение N_1 к N_2
0,85	3697	3893	0,9496
0,86	2931	3086	0,9497
0,87	556	577	0,9636
0,88	556	577	0,9636
0,9	556	577	0,9636
0,95	4	4	1

В табл. 6 приведены результаты тестирования диагонального способа для КНФ, эквивалентных задаче факторизации. Здесь под размерностью задачи подразумевается размер факторизуемого числа $n = pq$.

Табл. 6. Результаты тестирования для диагонального способа

Размерность задачи, бит	Число предположительно нулевых бит N_1	Число верных нулевых бит N_1	Отношение N_1 к N_2
100	4947	4277	0,8645
200	19894	17136	0,8613
300	44828	38942	0,8686
400	79786	68849	0.8629

Данный способ определяет достаточно большое число верных нулевых бит. Сильной стороной данного способа является то, что отношение N_2 к N_1 остаётся постоянным с увеличением размерности задачи.

Стоит отметить, что предложенные методы для различных экземпляров SAT определяют с высокой частотой различные нулевые биты. Другими словами, для нулевых бит приближённого решения, полученного вышерассмотренными способами, мы можем утверждать, что они с высокой частотой совпадают с битами точного решения. Про единичные биты мы ничего сказать не можем: они могут как совпадать с точным решением, так и принимать противоположное значение.

Не менее важным вопросом является определение конкретных бит, которые будут совпадать с точным решением для ***любого*** экземпляра задачи SAT. Это можно выявить, объединяя данные, полученные для различных постановок SAT. В табл. 7 приведены данные о «пересечении» множеств нулевых бит для различных экземпляров SAT, ассоциированной с задачей факторизации числа размерностью 100 бит. Частота совпадения равняется 0,9, число верно определённых нулевых бит для каждой отдельно взятой задачи равняется 598.

С увеличением числа экземпляров задачи SAT сокращается число общих верно определённых бит, однако в дальнейшем общие биты можно использовать для элиминации (сокращения) 3-КНФ и последующего применения других методов (например, алгоритма DPLL). Рассмотрим ещё одно эвристическое применение полученных результатов. Обозначим через Θ множество общих бит с высокой частотой совпадения.

Табл. 7. Количество общих верно определённых нулевых бит в зависимости от числа экземпляров задачи SAT

Число экземпляров задачи SAT	Число общих верно определённых нулевых бит
350	476
360	476
370	134
380	92
390	15
400	15

Инвертируем все биты множества Θ (т. е. установим значения в 1). Следующим шагом распределим инвертированные биты в векторе

$$z = (z_1, ..., z_N), \quad z_i = 0 \ \ \forall i = 1..N . \qquad (8)$$

Таким образом формируется вектор $\tilde{z}$, у которого на некоторых позициях, начиная с номера $i = n...N$, стоят единичные биты, остальные же места по-прежнему занимают нули. Установим ещё несколько бит на позициях $i = n...N$ в значение 1, причём выбор позиций осуществляется случайным **равномерным** образом.

Следующим шагом к вектору применяется метод последовательных приближений. По-прежнему выполняется ограниченное число итераций, но при этом значения бит, выбранных на предыдущем шаге, не будут изменяться. В полученном же приближении нас будут интересовать биты, стоящие на позициях, отвечающих битам сомножителей, т. е. биты с индексами $i = 1..n$.

Для иллюстрации эффективности данного подхода были проведены тестовые вычисления с приближениями, сформированными на основании имеющегося точного решения. При этом не были использованы биты множества Θ. В табл. 8 представлены данные по выявлению зависимости числа верных бит сомножителей от числа бит, точно задаваемых в позициях с индексами $i = 1..n$. Все значения приведены в процентном соотношении.

Табл. 8. Зависимость числа верных бит сомножителей от числа бит, точно задаваемых в матрице умножения (в процентном соотношении)

Процент точно задаваемых бит, исключая биты сомножителя	Процент верных бит, отвечающих за биты сомножителя
0	58
5	65
10	76
12	84
14	87
16	92
20	100

Оказывается, что при числе верно задаваемых бит, равном 20 % от общей длины вектора z, происходит распознавание бит сомножителей с вероятностью 100 %. Важным фактором в работе данного способа является то, что биты распределяются **равномерно** на позициях $i = 1…n$.

На практике же, задавая множество бит Θ, удаётся распознать биты сомножителей с вероятностью 60-65 %. Происходит это из-за того, что биты множества Θ располагаются концентрированно, неравномерно.

В дальнейшем планируется увеличение данного процента путём повышения мощности множества бит Θ, которые при этом располагались бы равномерно.

Заключение

В статье описаны два эвристических подхода к поиску выполняющего набора для задачи SAT. Первый из них заключается в расширении известного способа минимизации функции путём изменения порядка обхода переменных. Другой метод основан на создании матрицы с диагональным преобладанием, используя структуру 3-КНФ.

Проведены численные эксперименты. Предложено эвристическое использование полученных бит с высокой частотой устойчивости для формирования начального приближения для дальнейшего использования с целью распознавания бит сомножителей задачи факторизации.

Благодарности

Работа выполнена при поддержке РФФИ (проект 12-07-00294-а).

Литература

1. **Cook, S.** The complexity of theorem-proving procedures / S. Cook // Proceedings of the Third Annual ACM Symposium on Theory of Computing. – 1971. – P. 151-158.
2. **Дулькейт, В.И**. Сведение задач факторизации, дискретного логарифмирования и логарифмирования на эллиптической кривой к решению ассоциированных задач ВЫПОЛНИМОСТЬ / В.И. Дулькейт // Компьютерная оптика. – 2010. – Т. 34, №1. – С. 118-123. – ISSN 0134-2452.
3. **Menezes, A.** Handbook of applied cryptography / A. Menezes, P. van Oorschot, S. Vanstone. – CRC Press, 2001. – 780 p.
4. **Patsakis, C.** RSA private key reconstruction from random bits using SAT solvers [Electronic resource]. – Mode of access: https://eprint.iacr.org/2013/026. – Access date: 01.03.2013.
5. **Martin, D.** A Machine Program for Theorem Proving / D. Martin, G. Logemann, L. Donald // Communications of the ACM. – 1962. – V. 5(7). – P. 394–397. – ISSN: 0001-0782.
6. **Дулькейт, В.И.** Непрерывные аппроксимации решения задачи ВЫПОЛНИМОСТЬ применительно к криптографическому анализу асимметричных шифров / В.И. Дулькейт, Р.Т. Файзуллин, И.Г. Хныкин // Компьютерная оптика. – 2009. – Т. 33, №1. – С. 68-73. – ISSN 0134-2452.
7. **Дулькейт, В.И.** Гибридный метод поиска приближенного решения задачи 3-ВЫПОЛНИМОСТЬ, ассоциированной с задачей факторизации / В.И. Дулькейт, Ю.Ю. Огородников, Р.Т. Файзуллин // Труды института математики и механики УрО РАН. – 2013. – Т. 33, №2. – С. 285-294.
8. Численные методы / А.А. Самарский, А.В. Гулин. – М.: Физматлит, 1989. – 432 с. – ISSN: 0134-4889.
9. **Липский, В.** Комбинаторика для программистов / В. Липский; пер. с польск. – М.: Мир, 1988. – 200 с.
10. **Файзуллин, Р.Т.** Задачи линейной алгебры, соотнесенные с задачей ВЫПОЛНИМОСТЬ / Р.Т. Файзуллин // Прикладная дискретная математика. – 2009. – Приложение № 1. – С. 90–91. – ISSN: 2071-0410.

References

1. **Cook, S.** The complexity of theorem-proving procedures / S. Cook // Proceedings of the Third Annual ACM Symposium on Theory of Computing. – 1971. – P. 151-158.
2. **Dulkeit, V.I.** Reduction of factorization, discrete logarithm and elliptic curve logarithm problems to solving associated satisfiability problems / V.I.Dulkeit // Computer Optics. – 2010. – V.34(1). – P. 118-123. – ISSN 0134-2452. – (In Russian).
3. **Menezes, A.** Handbook of applied cryptography / A. Menezes, P. van Oorschot, S. Vanstone. – CRC Press, 2001. – 780 p.
4. **Patsakis, C.** RSA private key reconstruction from random bits using SAT solvers [Electronic resource]. – Mode of access: https://eprint.iacr.org/2013/026. – Access date: 01.03.2013.
5. **Martin, D.** A Machine Program for Theorem Proving / D. Martin, G. Logemann, L. Donald // Communications of the ACM. – 1962. – V. 5(7). – P. 394–397. – ISSN: 0001-0782.
6. **Dulkeit, V.I.** Continuous approximation of SAT decision as applied to cryptographic analysis of asymmetric ciphers /V.I. Dulkeit, I.G.Hnikin, R.T.Faizullin // Computer Optics. – 2009. – V. 33(1). – P. 86–91. – ISSN 0134-2452. – (In Russian).
7. **Dulkeit, V.I.** A hybrid method of searching the approximate solution of 3-SAT, associated with factorization problem / V.I. Dulkeit, R.T. Faizullin, Y.Y. Ogorodnikov // Proceedings of the Institute of Mathematics and Mechanics URAN. – 2013. – V. 19(2). – P. 285-294. – ISSN: 0134-4889. – (In Russian).
8. Numerical methods / A.A. Samarski, A.V. Gulin. – M.: “Fismatlit” Publisher, 1989. – 432 p. – (In Russian).
9. Witold Lipski. Kombinatoryka dla programistow. – “Warszawa” Publisher, 1982.
10. **Faizullin, R.T.** Problems of linear algebra, correlated with the satisfiability problem / R.T. Faizullin //Applied Discrete Mathematics. – 2009. – Application № 1. – P. 90-91. – ISSN: 2071-0410. – (In Russian).

RECOGNITION OF ZERO BITS OF 3-SAT PROBLEM BY APPLYING LINEAR ALGEBRA’S METHODS

Y.Y. Ogorodnikov, R.T. Faizullin
Omsk State Technical University

Abstract

The paper presents two heuristic methods of recognizing zero bits satisfiability problem. The first is based on the reduction of the satisfiability problem to an equivalent problem of minimizing a continuous smooth function by method of successive approximations, extended by changing the order of calculation of variables. Another way is to reduce to a system of linear algebraic equations with symmetric diagonally dominant matrix.

Key words: satisfiability, method of successive approximations, the order of calculation of variables, diagonal methods, factorization.

Сведения об авторах

Огородников Юрий Юрьевич, аспирант Омского государственного технического университета кафедры комплексной защиты информации. Область научных интересов: криптография, компьютерное моделирование, математические расчёты, программирование.

E-mail: *yogorodnikov@gmail.com* .

Yuri Yurevich Ogorodnikov, a graduate school student at Omsk State Technical University Complex Protection of Information Department. Research interests: cryptography, computer simulations, mathematical calculations, programming.

Файзуллин Рашит Тагирович, доктор технических наук (1999 г.), профессор Омского государственного технического университета кафедры комплексной защиты информации. В списке научных работ Р.Т. Файзуллина более 100 статей, 2 монографии.

Rashit Tagirovich Faizullin, Doctor of Technical Sciences (1999), Professor of Omsk State Technical University Complex Protection of Information department. He is co-author of more then 100 scientific papers, 2 monographs.

Поступила в редакцию 3 февраля 2014 г.