Гибридный алгоритм решения задачи 3-ВЫПОЛНИМОСТЬ ассоциированной с задачей факторизации

Ю.Ю. Огородников, Р.Т. Файзуллин

Омский государственный технический университет, г. Омск

В данной статье рассматривается гибридный алгоритм поиска приближённого решения задачи ВЫПОЛНИМОСТЬ. Предлагается комплексный подход – сегментный генетический алгоритм и метод последовательных приближений.

1. Введение

Задача ВЫПОЛНИМОСТЬ – одна из наиболее интересных задач теории сложности [1]. Широкое применение эта задача находит в таких сферах, как искусственный интеллект, проектирование компьютерных систем, криптография. В частности, в криптографии часть выполняющего набора для КНФ и 3-КНФ, ассоциированных с задачами криптографического анализа можно рассматривать как ключ шифрования.

Одним из перспективных направлением в поиске алгоритмов решения задачи ВЫПОЛНИМОСТЬ представляется сведение КНФ к непрерывному аналогу [2], т.е. к задаче поиска точек глобального минимума ассоциированной функции и гибридизация с дискретными методами.

Другим перспективным направлением поиска решений являются генетические алгоритмы - стохастические, эвристические оптимизационные методы, впервые предложенные Холландом[3]. Они основываются на идее эволюции с помощью естественного отбора. На основе генетических алгоритмов разработаны такие методы, как, например, VEGAS[4].

В поисках более эффективных методов решения происходит разработка комплексных подходов к решению данной задачи. Одним из таких подходов является гибридный алгоритм поиска приближённого решения, состоящий из двух стадий: сегментного генетического алгоритма и метода последовательных приближений. Этот метод и рассматривается в данной статье.

2. Этап работы сегментного генетического алгоритма

Эволюционные (или генетические) алгоритмы – это эвристические методы поиска, используемые для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию [3], [5].

Данные алгоритмы оперируют такими «биологическими» понятиями, как наследование, отбор, мутация, кроссинговер, индивид. Под индивидом понимается строка битов – геном. Для каждого индивида в каждом поколении вычисляется значение фитнесс-функции – его приспособленность к ареалу.

Перед непосредственным применением генетического алгоритма для задачи ВЫПОЛНИМОСТЬ следует описать задачу в терминах, соответствующих генетическому методу. Исходную 3-ДНФ можно представить в виде среды обитания (ареала). Геномом будет служить строка размерности N (где N – число переменных, задействованных в 3-ДНФ), состоящая из чисел 0 и 1. Исходной популяцией будет служить набор из L случайно сформированных геномов. Приспособленностью генома будем называть число скобок, которые при подстановке данного генома в 3-КНФ обращаются в значение ИСТИНА. В целях повышения эффективности процесса поиска решения будем использовать несколько параллельно работающих генетических алгоритмов. В основе такого подхода лежит разделение множества скобок исходной 3-КНФ на подмножества с последовательным распределением по генетическим алгоритмам с различными функциями приспособленности, т.е. один и тот же геном длины N будет иметь различную приспособленность в разных подмножествах. Для удобства дальнейшего изложения будем также

называть данные подмножества скобок *подзадачами* или *сегментами.* Формирование подзадач следует проводить в соответствии с разбиением *множества индексов* исходной 3-КНФ. Каждой подзадаче соответствует своё подмножество индексов, на основании которого происходит распределение скобок. Подмножества индексов попарно не пересекаются. Также отметим, что разбиение, полученное для 3-КНФ, также применимо и для эквивалентной ей 3-ДНФ.

Таким образом, задача ВЫПОЛНИМОСТЬ сводится к задаче минимизации функционала вида

$$F(x) = \sum_{i=1}^{M} \sum_{j=1}^{Qi} C_j(x), \text{ где } C_j \text{ - произведения вида } C_j(x) = \prod_{k=1}^{N} q_{j,k}(x), \quad (1.5.1)$$

$$q_{j,k}(x) = \begin{cases} x_k, \textit{если переменная } x_k \textit{ входит в } j-\textit{й конъюнкт непосредственно}, \\ \overline{x_k}, \textit{если переменная } x_k \textit{ входит в } j-\textit{й конъюнкт с отрицанием}, \\ 1, \textit{иначе} \end{cases}$$

Здесь M – число подзадач, Q_i – число конъюнктов в подзадаче i, x_k - литерал, N – общее число индексов.

По окончании работы всех генетических алгоритмов из каждого подмножества скобок будет браться геном с наилучшей из найденных приспособленностей. Данные результаты объединяются в главный геном. Отметим, что каждая подзадача оперирует геномами длины N, но при объединении достигнутых результатов в главный геном будут включаться только биты с индексами, на которых основана подзадача.

Существует несколько способов формирования подзадач. Ниже представлен один из таких методов. Пусть изначально имеется множество индексов $\Xi_0 = \{1,2,..., N\}$. Разделим множество индексов Ξ_0 = (1, 2,…, N) на два равных по мощности множества Ξ_1 = (1, 2,…,N/2), Ξ_2 = (N/2 + 1,…,N). Выберем произвольный индекс i из Ξ_1 для которого существует слагаемое вида $x_i \vee x_j \vee y$ (индекс j, также как и индекс i, принадлежит Ξ_1, y – произвольный литерал) в рассматриваемой 3-КНФ. Если такого слагаемого не существует, то мы переносим индекс i в множество индексов Ξ_2. В этом множестве искомое слагаемое по необходимости найдётся, иначе x_i не присутствует ни в одном из слагаемых. Перебирая все i в Ξ_1, мы получим два непересекающихся подмножеств индексов, в каждом из которых любые два индекса литералов (с отрицаниями или без) входят в Ξ_1 либо в Ξ_2.

Разделив далее большее по мощности из Ξ_s (или оба, если мощности сравнимы) на два подмножества Ξ_{s1} и Ξ_{s2} повторим процедуру и т.д.

Полностью выполнить такое требование для всех скобок невозможно, однако можно значительно уменьшить число скобок вида $x_i \vee x_j \vee x_k$, где i, j, k принадлежат разным подмножествам индексов. Такие слагаемые следует равномерно распределить по имеющимся подзадачам.

Рассмотрим пример разбиения на ниши:

Имеется следующая 3-КНФ:

$$(x_1 \vee \overline{x_9} \vee x_{10})(\overline{x_8} \vee x_9 \vee x_{10})(x_2 \vee \overline{x_3} \vee x_8)(x_4 \vee x_6 \vee \overline{x_{10}})(\overline{x_2} \vee x_5 \vee \overline{x_7})(x_4 \vee x_8 \vee \overline{x_9})$$

Литералом называется переменная x_1, индексом литерала называется в данном случае 1.

Стартовое множество индексов выглядит следующим образом:

$$\Xi_0 = \{1,2,3,4,5,6,7,8,9,10\}$$

I. Разобьём множество индексов Ξ_0 на 2 подмножества:

$$\Xi_1 = \{1,2,3,4,5\} \quad \text{и} \quad \Xi_2 = \{6,7,8,9,10\}$$

В соответствии с алгоритмом берём множество наибольшей мощности.

Если таких множеств несколько, то берём множество с наименьшим номером – в данном случае выбираем множество Ξ_1.

II. Перебирая множество индексов из Ξ_1 замечаем, что для индексов $i = 1$ и $i = 4$ не существует слагаемых вида $x_i \vee x_j \vee y$ в рассматриваемой 3-КНФ (x_i - литерал или его отрицание, индекс j принадлежит множеству Ξ_1, y – произвольный литерал). Для индексов $i = 2, 3, 5$ такие слагаемые существуют (для $i = 2$ это будет индекс $j = 3$ и дизъюнкт $x_2 \vee \overline{x_3} \vee x_8$, для $i = 3$ индекс j = 2 и дизъюнкт $x_2 \vee \overline{x_3} \vee x_8$, для $i = 5$ индекс $j = 2$ и дизъюнкт $\overline{x_2} \vee x_5 \vee \overline{x_7}$). Переносим индексы $i = 2$ и $i = 4$ во множество Ξ_2. Переходим на следующий шаг алгоритма разбиения.

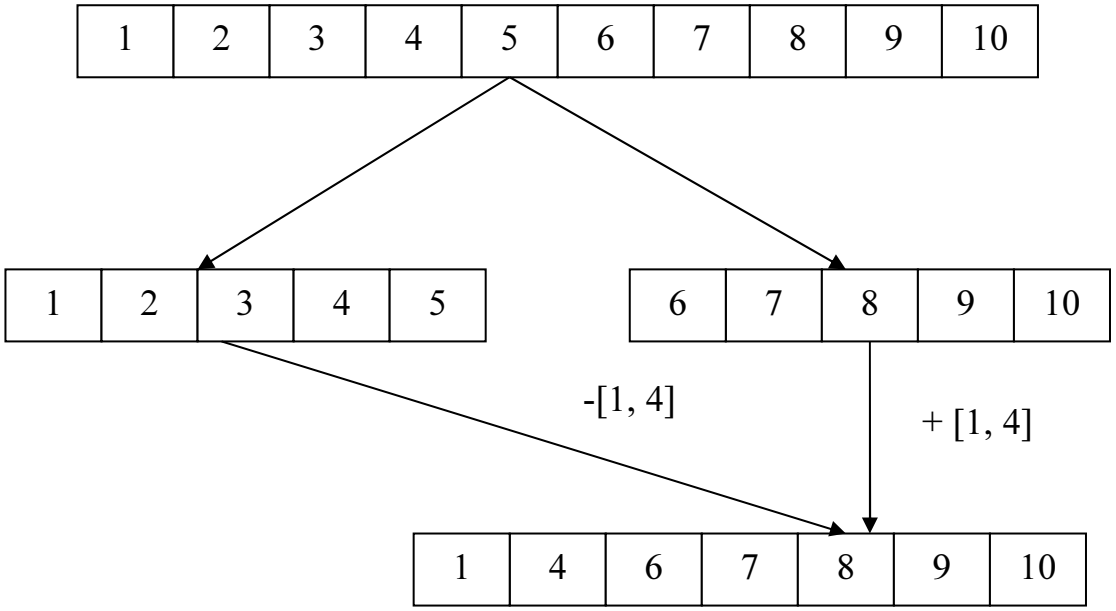

Рис. 1. Первая и вторая итерации разбиения множества индексов

Ш. Рассматриваем большее по мощности множество Ξ_2, которое теперь имеет следующий вид: $\Xi_2 = \{1,4,6,7,8,9,10\}$. Разделим его на два подмножества $\Xi_3 = \{1,4,6\}$ и $\Xi_4 = \{7,8,9,10\}$. Рассмотрим большее по мощности из этих множеств, а именно: Ξ_4. Перебираем все его элементы и замечаем, что только для $i = 7$ нет индекса j во множестве Ξ_4 такого, что $x_i \vee x_j \vee y$ присутствует в рассматриваемой 3-КНФ. Для остальных же индексов условие выполняется (для $i = 8$ индекс $j = 9$ и дизъюнкт $x_4 \vee x_8 \vee \overline{x_9}$, для $i = 9$ индекс $j = 8$ и тот же дизъюнкт $x_4 \vee x_8 \vee \overline{x_9}$, для $i = 9$ индекс $j = 8$ и тот же дизъюнкт $x_4 \vee x_8 \vee \overline{x_9}$, для $i = 10$ индекс $j = 8$ и дизъюнкт $\overline{x_8} \vee x_9 \vee x_{10}$). Для индекса $i = 7$ условие не выполняется, однако в силу того что он единственный такой индекс, то его можно оставить в множестве Ξ_4.

IV. Берём множество Ξ_3 – следующее по мощности за Ξ_4. Замечаем, что для i = 1 рассматриваемая 3-КНФ не содержит дизъюнктов вида $x_i \vee x_j \vee y$. Для i = 4 и i = 6 такие дизъюнкты существуют (для i = 4 индекс j = 6 и дизъюнкт $x_4 \vee x_6 \vee \overline{x_{10}}$, а для i = 6 индекс j = 4 и тот же дизъюнкт $x_4 \vee x_6 \vee \overline{x_{10}}$). Так как индекс i = 1 – единственный, для которого не существует дизъюнкта вида $x_i \vee x_j \vee y$, то его можно оставить в множестве Ξ_3.

Дальнейшее проведение итераций не имеет смысла, так как далее в полученных подмножествах не будет существовать дизъюнктов вида $x_i \vee x_j \vee y$.

РАЗБИЕНИЕ ОКОНЧЕНО.

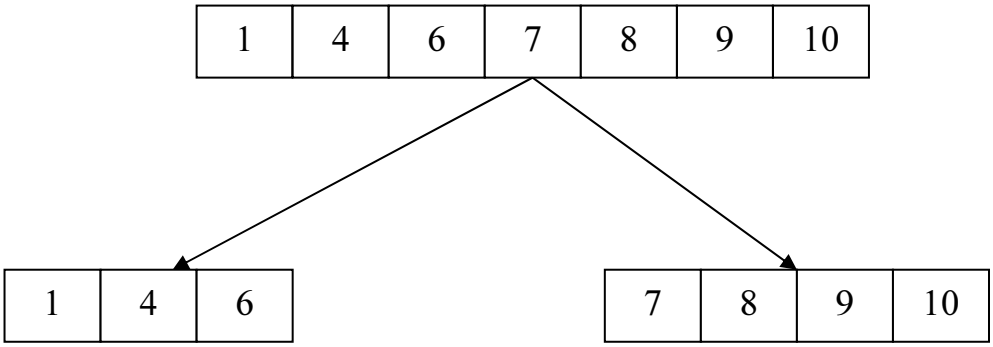

Рис. 2. Третья и четвёртая итерации разбиения множества индексов

Таким образом, скобки будут разделены следующим образом:

- на основании Ξ_1: $(x_2 \vee \overline{x_3} \vee x_8)(\overline{x_2} \vee x_5 \vee \overline{x_7})$
- на основании Ξ_3: $(x_4 \vee x_6 \vee \overline{x_{10}})$
- на основании Ξ_4: $(x_1 \vee \overline{x_9} \vee x_{10})(\overline{x_8} \vee x_9 \vee x_{10})(x_4 \vee x_8 \vee \overline{x_9})$

Множества Ξ_0 и Ξ_2 не учитывались при разбиении, так как из них были получены меньшие по мощности множества: Ξ_1 и Ξ_2 из Ξ_0, Ξ_3 и Ξ_4 из Ξ_2 соответственно.

Родители выбираются с близкими значениями функции приспособленности, и передаются функции скрещивания. В качестве оператора скрещивания выбирается случайная точка в геноме, так называемая точка разрыва. Геном потомка состоит двух частей: части генома первого родителя до точки разрыва и части генома второго родителя после точки разрыва. После создания генома производится его мутация: значения нескольких бит заменяются противоположными[6].

После выполнения процедуры создания следующего поколения вычисляются значения фитнесс для всех потомков, и новая популяция строится путём объединения старого и нового поколения, причём размер популяции остаётся тем же, но туда вбираются индивиды с наилучшей приспособленностью.

По окончании работы всех генетических алгоритмов из каждого подмножества скобок будет браться геном с наилучшей из найденных приспособленностей. Данные результаты объединяются в главный геном. Отметим, что каждая подзадача оперирует геномами длины *N*, но при объединении достигнутых результатов в главный геном будут включаться только биты с индексами, на которых основана подзадача.

3. Этап работы метода последовательных приближений

Осуществим переход от задачи ВЫПОЛНИМОСТЬ к задаче поиска экстремума в подмножестве $[0, 1]^N$ евклидова пространства. Переход основан на построении функционала специального вида, глобальный минимум которого соответствует решению исходной задачи[2].

Имеется КНФ на множестве переменных $(y_1, \ldots. y_n) \in B^N$:

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

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

$$q_{i,j}^*(y) = \begin{cases} y_j, \text{если} \quad y_j \in \{G_i^*\} \\ \overline{y}_j, \text{если} \quad \overline{y}_j \in \{G_i^*\}, \\ 0, \text{иначе} \end{cases}$$

N - число переменных , M - число дизъюнктов

Преобразуем исходную КНФ к ДНФ следующим образом:

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

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

$$q_{i,j}(y) = \overline{q_{i,j}^*(y)}$$

Сопоставим булевым переменным $y = (y_1, \ldots y_N) \in B^N$ исходной КНФ переменные пространства вещественных чисел $x = (x_1, \ldots x_N) \in R^N$. Для этого рассмотрим функционал, на множестве переменных $x \in R^N$:

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

C_i - произведения вида $C_i(x) = \prod_{j=1}^{N} p_{i,j}(x),$

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

Суммирование ведётся по всем M конъюнктам ДНФ, эквивалентной исходной КНФ.

Соответствие между булевыми (y_i) и вещественными (x_i) переменными следующее:

$$\begin{cases} y_i = 1 \text{ (ИСТИНА)} \quad \Rightarrow x_i = 1 \\ y_i = 0 \text{ (ЛОЖЬ)} \quad \Rightarrow x_i = 0 \end{cases}$$

В обратную сторону

$$\begin{cases} x_i = 1 \Rightarrow y_i = 1 \text{ (ИСТИНА)} \\ x_i \neq 1 \Rightarrow y_i = 0 \text{ (ЛОЖЬ)} \end{cases}$$

Переход от булевой формулы к вещественной основан на использовании литерной функции вида:

$$\begin{cases} T(y_i \vee y_j) = x_i + x_j \\ T(y_i \wedge y_j) = x_i^2 \cdot x_j^2 \quad \text{, где } y_i \in B,\ x_i \in B,\ i = 1 \ldots N \\ T(\overline{y}_i) = 1 - x_i \end{cases}$$

Функционал F имеет единственный глобальный экстремум, соответствующий решению исходной задачи. Но вместо минимизации F будем производить поиск стационарных точек. Для этого дифференцируем функционал по всем переменным и приравниваем к нулю полученные выражения. В результате получается система уравнений с n неизвестными. Применяя к полученной системе метод последовательных приближений, получим приближённое решение функционала. В качестве начального приближения следует брать значение, полученное в результате выполнения сегментного генетического алгоритма[2].

После окончания работы этапа полученное приближение добавляется в стартовую популяцию генетического алгоритма.

4. Интеграция генетического алгоритма и метода последовательных приближений

С целью получения комплексной процедуры решения задачи ВЫПОЛНИМОСТЬ, объединим этапы работы генетического алгоритма и метода последовательных приближений.

Связь между двумя этапами будет осуществляться подстановкой в выходные данные одного этапа входные данные другого. Так, геном, соответствующий минимуму в части генетического алгоритма, служит начальным приближением для метода последовательных приближений. В свою очередь, конечный вектор $x(W)$, где W – число итераций в методе последовательных приближений, добавляется в популяцию для генетического алгоритма.

Полученная процедура повторяется заданное число итераций C.

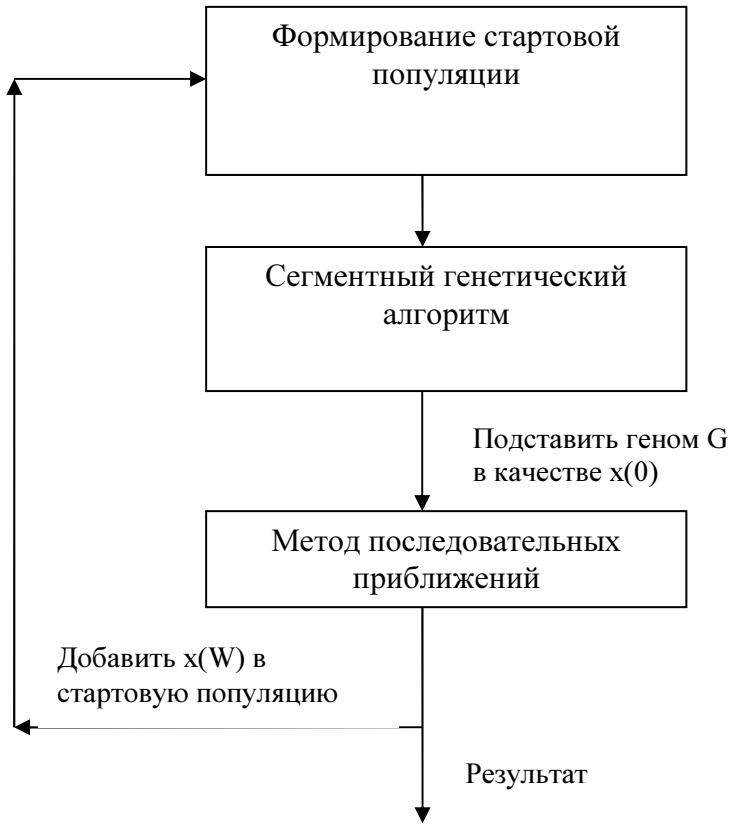

Рис. 3. Графическое представление двухэтапного метода

5. Способы распараллеливания алгоритма

Главный поток выполняет чтение входных данных и инициализацию структур программы. Он же производит запуск вспомогательных потоков для сегментов. Каждому сегменту соответствует отдельный поток. Объединение результатов подзадач происходит также в главном потоке.

6. Промежуточные результаты

В таблице 1 представлены результаты вычислений для числа итераций $C = 100$. Вычисления проводились для разных параметров N и M, где N – число переменных, M – общее число скобок. Остальные параметры фиксированы и равны следующим значениям:

- численность популяции $L = 2000$
- число подзадач $Q = 10$
- число поколений $G = 500$

Под числом верных скобок T подразумевается количество скобок в исходной 3-КНФ, которое обращается в значение ИСТИНА при подстановке решения, полученного в результате выполнения алгоритма.

Таблица 1. Результаты вычислений

Число переменных N	Общее число скобок M	Число верных скобок T	Отношение T / M
20	68	68	1
120	440	440	1
252	952	951	0,9989
540	2080	2075	0,9975
1260	4920	4920	1
3600	14200	14200	1
5220	20640	20640	1
9360	37120	37120	1
14700	58400	58396	0,9999
33300	132600	132591	0,9999
59400	236800	236787	0,9999
97536	389120	389038	0,9999

Из полученных результатов видно, что число найденных верных скобок довольно велико, и к неверным (т.е. принимающим значение ЛОЖЬ) скобкам можно применить различные методы для поиска точного решения. В следующей теореме показано, что существует предположительная область сходимости для общей 3-КНФ с единственным решением при выполнении определённых условий на приближении.

Теорема 1.

Пусть задана 3-КНФ содержащая N литералов, M скобок, и имеющая единственный решающий набор. Предположим, что выполнено следующее условие: пусть задано приближение к решению в виде целочисленной точки с компонентами (0,1) такое, что среди невыполняющихся при подстановке приближения скобках имеются L скобок, невыполнение которых зависит только от одного литерала, а остальные два верные, и $L \geq M/8$. В этом случае сужение функции, ассоциированной с 3-КНФ, на луч, соединяющий решение и данную целочисленную точку, с координатами в вершинах гиперкуба, отличающуюся от решения является строго монотонной убывающей, а если $L \geq M/3$, то и выпуклой функцией. Заметим, что из теоремы не следует выпуклость графика самой функции, но численные эксперименты показали, что данная теорема имеет практическую ценность. Оказывается, что при числе верных бит N (т.е. совпадающих с битами точного решения), большем 6/7 от общего количества, метод последовательных приближений, применённый к задаче ВЫПОЛНИМОСТЬ, всего за несколько итераций сходится к решению. Причём смещение идёт практически прямо по отрезку, соединяющему приближение и точное решение.

Основываясь на результатах теоремы, можно попытаться построить методику поиска наиболее вероятных бит, что позволит нам выбрать приближение из области сходимости к точке глобального экстремума.

Методика может состоять из нескольких независимых тестов. Например, по содержанию ненулевых скобок можно оценить, у скольких бит и примерно каких следует изменить значение.

Другой способ может быть основан на таблице частот устойчивости бит, полученной в результате применения метода последовательных приближений к поиску выполняющего набора для 3-ДНФ, эквивалентной 3-КНФ. Чем частоты более удалены от значения 0.5, тем выше вероятность верно определить бит. К примеру, если частота равна 1, то это означает, что во всех случаях бит точно угадан. Напротив же, если частота равна 0, то, следовательно, во всех случаях бит неверно угадан, и можно гарантированно произвести его инвертирование. Если частота равна 0.5, то ничего определённого сказать нельзя.

В таблицах 2 – 4 приведены данные исследования устойчивости бит для задачи ВЫПОЛНИМОСТЬ, эквивалентной задаче ФАКТОРИЗАЦИИ.

Частоты получены на основе исследований 200 различных исходных данных для задачи ФАКТОРИЗАЦИИ, которые получены качественным источником генерации случайных значений[7]. В каждой таблице указано количество бит, для которых частота устойчивости удовлетворяет некоторым условиям, и средняя ошибка.

Также эксперименты показали, что для крайних бит сомножителей (например, для сомножителей размерности 600 это 1, 300, 301 и 600 биты), частота устойчивости равна 1. Это означает, что биты, полученные на таких позициях, гарантированно являются верными.

Таблица 2. Частота устойчивости бит удовлетворяющих условию > 0.7 или < 0.3

Размерность сомножителя	Общее число бит M	Число нулевых бит T	Отношение T / M	Средняя ошибка
100	14700	9331	0,6347	0,1803
200	59400	38912	0,655	0,1871
300	134100	86174	0,6462	0,1821
400	238800	153336	0,6421	0,1836
500	373500	239339	0,6408	0,1832
600	538200	345203	0,6414	0,1834

Таблица 3. Частота устойчивости бит удовлетворяющих условию > 0.8 или < 0.2

Размерность сомножителя	Общее число бит M	Число нулевых бит T	Отношение T / M	Средняя ошибка
100	14700	4924	0,3349	0,1237
200	59400	19329	0,3254	0,1241
300	134100	46081	0,3436	0,1259
400	238800	81710	0,3421	0,1268
500	373500	128226	0,3433	0,127
600	538200	186164	0,3459	0,1275

Таблица 4. Частота устойчивости бит удовлетворяющих условию > 0.9 или < 0.1

Размерность сомножителя	Общее число бит M	Число нулевых бит T	Отношение T / M	Средняя ошибка
100	14700	757	0,0514	0,0844
200	59400	441	0,0074	0,0879
300	134100	6014	0,0448	0,0858
400	238800	9524	0,0398	0,0864
500	373500	14973	0,04008	0,0866
600	538200	27003	0,0501	0,0883

Как видно из полученных результатов, отношение числа нулевых бит к общему числу бит в целом остаётся постоянным при увеличении размерности задачи. Средняя ошибка также стабилизирована возле определённого значения.

7. Заключение

В работе представлен двухэтапный метод поиска решения задачи ВЫПОЛНИМОСТЬ. Первый этап заключается в сегментом генетическом алгоритме, второй – в методе последовательных приближений. Произведено тестирование метода для различных размерностей. Выяснено, что при применении двухэтапного метода число верных скобок достаточно велико. В таком случае для оставшихся скобок возможно применение перебора с использованием таблицы частот устойчивости бит.

Литература

1. Cook S.A., Mitchel D., G. Finding hard instances for the satisfiability problem: A survey. DIMACS series in discrete mathematics and theoretical computer science. V. 5. 1997.
2. Хныкин И.Г. Минимизация функционалов, ассоциированных с задачей ВЫПОЛНИМОСТЬ: Дис. канд. техн. наук // Омск, 2009. – С.38-42.
3. Holland J.H. Adaptation in natural and artificial systems / The University of Michigan Press, 1975.
4. Скворцов Е.С. VEGAS – новый генетический алгоритм для решения задачи ВЫПОЛНИМОСТЬ. Известия Уральского Государственного Университета №74, 2010.
5. Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning. – Reading: Addison Wesley, 1989.
6. Еремеев А.В., Генетические алгоритмы и оптимизация. // Учебное пособие. – Омск: Изд-во Омского гос. университета, 2008. С.16-24.
7. Дулькейт В.И. КНФ представления для задач факторизации, дискретного логарифмирования и логарифмирования на эллиптической кривой: Дис. канд. физ.-мат. наук // Омск, 2010. с.27-71.