Министерство образования Российской Федерации
Петрозаводский государственный университет
Математический факультет
Кафедра Информатики и математического обеспечения

Кулаков Кирилл Александрович

Тестирование и экспериментальный анализ алгоритмов решения неотрицательных линейных диофантовых уравнений

510200 — Прикладная математика и информатика

Выпускная квалификационная работа бакалавра

Научные руководители:
к.т.н., доцент,
Богоявленский Ю. А.
к.ф.-м.н., ст. преподаватель,
Корзун Д. Ж.

Петрозаводск — 2003

Оглавление

Перечень сокращений и условных обозначений

ПО — программное обеспечение.

АНЛДУ система — ассоциированная с грамматикой система НЛДУ.

НЛДУ — неотрицательное линейное диофантово уравнение.

$\mathbb{O}$ — нулевой вектор, $\mathbb{O} = (0, 0, \dots, 0)$.

e_k — стандартный единичный вектор, $e_k = (0, \dots, 0, 1, 0, \dots, 0)$.

$\mathcal{H}$ — базис Гильберта однородной системы НЛДУ, $\mathcal{H} = \{h^{(1)}, h^{(2)}, \dots, h^{(q)}\}$.

$\varnothing$ — пустое множество.

I — разбиение множества $\mathbb{N}_m$, $I = \{I_1, \dots, I_n\}$.

$\mathbb{N}$ — натуральный ряд, $\mathbb{N} = \{1, 2, \dots\}$.

$\mathbb{N}_m$ — отрезок натурального ряда длины m, $\mathbb{N}_m = \{1, 2, \dots, m\}$.

$\mathbb{Z}$ — множество целых чисел, $\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$.

$\mathbb{Z}_+$ — множество неотрицательных целых чисел, $\mathbb{Z}_+ = \{0, 1, 2, \dots\}$.

$\mathrm{E}(I)$ — булева $(n \times m)$-матрица, коэффициенты которой определяются как $\mathrm{E}_{kj}(I) = \mathrm{E}_{kj}(I_1, \dots, I_n) = 1 \iff \overrightarrow{\mathrm{I}}_j = k$ для $k = \overline{1, n}$, $j = \overline{1, m}$.

$\vec{I}$ — линейное представление матрицы $\mathrm{E}(I)$ в виде вектора из $\mathbb{N}_n^m$.

Введение

При создании математического программного обеспечения (ПО) важным требованием является обоснование корректности его реализации. Этого можно достичь на основе комплексного тестирования. При этом исследуемое ПО необходимо проверить на разнообразном наборе классов тестов, определяющих как типичные условия работы ПО, так и нестандартные (вырожденные) случаи. Для обоснованного подтверждения корректности классы таких тестов должны охватывать если не все, то бо́льшую часть возможных вариантов. Кроме того, число используемых тестов для каждого класса должно быть достаточно велико, чтобы гарантировать с приемлемым уровнем значимости отсутствие ошибок реализации.

В силу этого, выполнение подобного тестирования практически невозможно без использования специальных разработанных программных средств, автоматизирующих процесс генерации и проверки тестов. Именно такой подход позволяет детально и всесторонне (комплексно) проверить корректность реализации заданного ПО.

Другой важной характеристикой реализованного ПО является оценка потребляемых ресурсов. В отличии от обоснования корректности, где проверяется, может ли ПО правильно решать поставленные задачи, здесь оценивается, насколько эффективно работает ПО. В частности, эта оценка может быть использована для сравнения данного ПО с альтернативным. В свою очередь, последняя задача часто бывает исключительно важной при выборе наиболее приемлемого ПО для решения возникающей на практике проблемы.

Для решения такой задачи разумно использовать подход аналогичный комплексному тестированию. С помощью специальных программных средств генерируется большие выборки тестов из разнообразного набора классов и выполняется измерения ресурсов, потребляемых исследуемым ПО. На основании этих выборок для каждого класса статистическими методами строятся оценки объема потребления ресурсов (оценки эффективности).

В данной работе в качестве исследуемого ПО рассматривается математическое обеспечение решения систем неотрицательных линейных диофантовых уравнений (система НЛДУ). Основными задачами решения систем НЛДУ являются: 1) определение совместности, 2) нахождение некоторого решения, 3) нахождение базиса Гильберта. В общем случае эти задачи чрезвычайно трудоемки (NP, overNP) [5].

Одними из перспективных алгоритмов для эффективного решения частных классов систем НЛДУ являются синтаксические алгоритмы, предложенные Д. Ж. Корзуном [4, 5]. Они позволяют решать ассоциированные с КС-грамматиками системы НЛДУ (системы АНЛДУ) [1, 3, 5]. В качестве альтернативного для синтаксических алгоритмов рассматривается алгоритм нахождения базиса Гильберта однородных систем НЛДУ, предложенный португальскими математиками M. Filgueiras, A.-P. Tomás [8–10].

Целью данной работы является разработка технологии комплексного тестирования, экспериментального и сравнительного анализа алгоритмов решения систем НЛДУ. Объектом исследования в работе являются алгоритмы решения систем НЛДУ, а предметом исследования — методы генерации тестовых систем НЛДУ для исследуемых алгоритмов. Прикладные возможности разработанной технологии применяются для тестирования и экспериментального анализа вычислительной сложности синтаксического алгоритма нахождения базиса Гильберта произвольной однородной системы АНЛДУ.

Для достижения поставленной цели необходимо решить следующие задачи.

1) Разработать алгоритмы генерации тестовых систем и соответствующих им базисов Гильберта для различных классов однородных систем АНЛДУ.

2) Разработать и реализовать программную систему, которая использует предложенные алгоритмы генерации для выполнения комплексного тестирования, экспериментального и сравнительного анализа различных алгоритмов решения систем НЛДУ. Для обеспечения гибкости системы построить ее интерфейс с пользователем на основе web технологий.

3) Выполнить тестирование и экспериментальный анализ синтаксических алгоритмов решения однородных систем АНЛДУ, а также провести сравнительный анализ этого алгоритма с альтернативным алгоритмом M. Filgueiras, A.-P. Tomás.

Выпускная работа состоит из введения, четырех глав, заключения, библиографического списка использованной литературы (11 наименований), имеет общий объем 47 машинописных страниц, содержит 20 рисунков и 2 таблицы.

В первой главе представлен понятийный аппарат и вводятся основные обозначения, используемые в работе. Во второй главе представлен ряд теоретических свойств однородных систем АНЛДУ, описана постановка задачи генерации и приведены 4 алгоритма генерации систем АНЛДУ и базиса Гильберта к ним. В третьей главе рассмотрено программное обеспечение для выполнения тестирования и экспериментального анализа алгоритмов решения систем АНЛДУ. В четвертой главе приведены результаты экспериментов.

Результаты выпускной работы докладывались и обсуждались на 54 и 55 научных студенческих конференциях ПетрГУ, а также на рабочих семинарах кафедры информатики и математического обеспечения ПетрГУ. По результатам работы опубликованы тезисы [6].

1. Обозначения и понятийный аппарат

В настоящей главе представлен понятийный аппарат, необходимый для построения алгоритмов генерации тестовых систем, а также вводятся основные обозначения. Для случая однородных систем АНЛДУ обозначения предлагаются нами впервые и дают более компактную и прозрачную запись.

Определение 1.1. *Однородной системой неотрицательных линейных диофантовых уравнений (НЛДУ)* называется система уравнений, все коэффициенты которой являются произвольными целыми числами, а компоненты решений принимают неотрицательные целые значения:

$$Ax = \mathbb{O}, \qquad A \in \mathbb{Z}^{n\times m}, \qquad x \in \mathbb{Z}_+^m, \qquad \mathbb{O} \in \mathbb{Z}^n, \tag{1.1}$$

где n — число уравнений системы, m — число неизвестных. При этом матрица A называется матрицей коэффициентов системы, а вектор $\mathbb{O}$ — нулевой вектор из $\mathbb{Z}^n$.

Определение 1.2. Ненулевое решение $h \in \mathbb{Z}_+^m$ однородной системы НЛДУ (1.1) называется *неразложимым*, если оно не может быть представлено в виде суммы двух ненулевых решений этой же системы.

Определение 1.3. *Базисом Гильберта* однородной системы НЛДУ (1.1) называется множество всех неразложимых решений однородной системы НЛДУ:

$$\mathcal{H} = \{h^{(1)}, \dots h^{(q)}\}, \qquad h \in \mathbb{Z}_+^m,$$

где q — число базисных решений.

Отметим, что согласно теореме о базисе Гильберта [5] число неразложимых решений всегда конечно и любое решение x системы (1.1) может быть выражено через базис Гильберта следующим образом:

$$x = \sum_{s=1} \alpha_s h^{(s)}, \qquad \alpha_1, \alpha_2, \dots, \alpha_q \in \mathbb{Z}_+. \tag{1.2}$$

Неразложимость элементов базиса Гильберта эквивалентна их *минимальности*, т. е. для любого решения x неоднородной системы (любого решения h однородной системы) и произвольного базисного элемента $x' \in \mathcal{N}$, $x' \neq x$ ($h' \in \mathcal{H}$, $h' \neq h$) неверно, что $x \leqslant x'$ ($h \leqslant h'$). При этом сравнение здесь покомпонентное:

$$(u_1, u_2, \ldots, u_m) \leqslant (v_1, v_2, \ldots, v_m) \quad \overset{\text{def}}{\Longleftrightarrow} \quad u_j \leqslant v_j \quad \forall\, j = 1, 2, \ldots, m\ .$$

Таким образом, базисное решение покомпонентно не превосходит любое другое решение системы НЛДУ. Более того, базисные решения несравнимые друг с другом в смысле покомпонентного сравнения векторов.

Пример 1.1. Рассмотрим однородную систему НЛДУ:

$$\begin{cases} x_1 + x_3 = 2x_2 + x_4\,, \\ x_2 + x_4 = 3x_1\,. \end{cases}$$

Ее базис Гильберта следующий:

$$\mathcal{H} = \left\{ \begin{pmatrix} 1 \\ 3 \\ 5 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 4 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 2 \\ 3 \end{pmatrix} \right\}.$$

Например, решение $x = (3, 1, 7, 8)^{\mathsf{T}}$ может быть выражено через этот базис как

$$x = 0h^{(1)} + 0h^{(2)} + 1h^{(3)} + 2h^{(4)} = h^{(3)} + 2h^{(4)}.$$

Определение 1.4. *Разбиением конечного отрезка натурального ряда* $\mathbb{N}_m = \{1, \ldots, m\}$ называется множество непустых подмножеств $I = \{I_1, \ldots, I_n\}$, где

$$I_k \cap I_j = \varnothing \text{ для } k \neq j, \qquad \bigcup_{k=1}^{n} I_k = \mathbb{N}_m.$$

Определение 1.5. *Ассоциированной с контекстно-свободной грамматикой однородной системой НЛДУ* (системой АНЛДУ) называется система вида

$$\mathrm{E}(I_1, \ldots, I_n)x = Ax,$$

где $I = \{I_1, \ldots, I_n\}$ — разбиение $\mathbb{N}_m$ (), n — число уравнений системы, m — число неизвестных. Матрица $\mathrm{E}(I) = \mathrm{E}(I_1, \ldots, I_n) \in \{0, 1\}^{n \times m}$ и $\mathrm{E}_{k,i}(I_1, \ldots, I_n) = 1$ тогда и только тогда, когда $i \in I_k$, т. е. I_k определяет множество индексов неизвестных, коэффициенты при которых в левой части уравнения k равны 1.

В частности, для примера 1.1 матрица $\mathrm{E}(I)$ будет следующего вида:

$$\mathrm{E}(I_1, I_2) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \qquad I_1 = \{1, 3\}, \;\; I_2 = \{2, 4\}.$$

Специфика матрицы $\mathrm{E}(I)$ допускает линейное описание в виде вектора $\overrightarrow{\mathrm{I}} \in \mathbb{N}_n^m$. А именно, $\overrightarrow{\mathrm{I}}_i = l \Leftrightarrow \mathrm{E}_{l,i}(I) = 1$. Для матрицы $\mathrm{E}(I)$ из примера 1.1 вектор $\overrightarrow{\mathrm{I}} = (1, 2, 1, 2)$.

Определение 1.6. Две однородные системы АНЛДУ $\mathrm{E}'(I')x = A'x$ и $\mathrm{E}''(I'')x = A''x$ называются *эквивалентными*, если перенумерацией неизвестных их можно привести к системе $\mathrm{E}(I)x = Ax$, где матрица $\mathrm{E}(I)$ имеет следующий стандартный вид:

$$\mathrm{E}(I_1, \ldots, I_n) = \begin{pmatrix} 1 \;\ldots\; 1 \;\; 0 \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\; 0 \\ 0 \;\ldots\; 0 \;\; 1 \;\ldots\; 1 \;\; 0 \;\ldots\ldots\ldots\; 0 \\ \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \\ 0 \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\; 0 \;\; 1 \;\ldots\; 1 \end{pmatrix}$$

где $I_1 = \{1, \ldots, k\}, \; I_2 = \{k+1, \ldots, p\}, \; \ldots, \; I_n = \{q+1, \ldots, m\}$.

Таким образом, в стандартном виде представления системы АНЛДУ ????

Пример 1.2. Для системы из примера 1.1 стандартный вид матрицы $\mathrm{E}(I)$:

$$\mathrm{E}(I_1, I_2) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Стандартный вид системы:

$$\begin{cases} y_1 + y_2 = 2y_3 + y_4\,, \\ y_3 + y_4 = 3y_1\,. \end{cases}$$

Перенумерация неизвестных: $y_1 = x_1$, $y_2 = x_3$, $y_3 = x_2$, $y_4 = x_4$.

Определение 1.7. *Уравнение Фробениуса с единичными коэффициентами* — это уравнение вида

$$\sum_{i=1} x_i = T, \qquad T \in \mathbb{Z}_+, \tag{1.3}$$

где T — константа.

В [7] показано, что уравнение Фробениуса (1.3) имеет конечное множество решений, число которых определяется формулой

$$q = C_{T+n-1}^{n-1} = \frac{(T+n-1)!}{T!(n-1)!}. \tag{1.4}$$

Все решения несравнимы друг с другом в смысле покомпонентного сравнения векторов.

Пример 1.3. Рассмотрим следующее уравнение Фробениуса с единичными коэффициентами:

$$x_1 + x_2 + x_3 = 4.$$

Множество решений этого уравнения состоит из $q = \dfrac{(4+3-1)!}{4!(3-1)!} = \dfrac{6!}{4!2!} = 15$ элементов:

$$\mathbb{S} = \left\{ \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix} \right\}.$$

2. Алгоритмы генерации тестовых систем

2.1. Свойства однородной системы АНЛДУ

В данном параграфе мы получаем ряд теоретических свойств однородных систем АНЛДУ. Эти свойства будут лежать в основе предлагаемых нами алгоритмов генерации тестовых систем (см. пп. 2.3–2.6).

В случае одного уравнения ($n = 1$) однородная система АНЛДУ имеет следующий стандартный вид:

$$\sum_{k=1}^{m} x_k = \sum_{k=m'+1}^{m} (a_k + 1) x_k, \qquad a_k \in \mathbb{Z}_+, \tag{2.1}$$

Здесь $\mathrm{E}(\mathbb{N}_m) = (1, 1, \ldots, 1) \in \mathbb{Z}_+^m$, $A = (0, 0, \ldots, 0, a_{m'+1} + 1, \ldots, a_m + 1) \in \mathbb{Z}_+^m$

Докажем теорему, которая дает способ вычисления базиса Гильберта любого уравнения вида (2.1).

Теорема 2.1. *Базисом Гильберта уравнения (2.1) является множество векторов*

$$\mathcal{H} = \left\{ h \in \mathbb{Z}_+^m \;\middle|\; \begin{array}{l} h = (c_1, \ldots, c_{m'}, 0, \ldots, 0, 1, 0, \ldots, 0)^{\mathsf{T}}, \\ \text{1 стоит на } j\text{-й позиции, } m' < j \leqslant m, \\ \sum_{k=1}^{m'} c_k = a_j, \quad c_k \in \mathbb{Z}_+ \end{array} \right\}, \tag{2.2}$$

Доказательство. В силу условия теоремы множество $\mathcal{H}$ может быть построено как

$$\mathcal{H} = \left\{ \overbrace{\begin{pmatrix} c_{11} \\ \vdots \\ c_{m'1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}^{h^{(1)}}, \overbrace{\begin{pmatrix} c_{12} \\ \vdots \\ c_{m'2} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}^{h^{(2)}}, \ldots, \overbrace{\begin{pmatrix} c_{1q} \\ \vdots \\ c_{m'q} \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}}^{h^{(q)}} \right\},$$

При этом вектора $h^{(s)}$, для которых $h_j^{(s)} = 1$, получаются как все решения уравнения

Фробениуса с единичными коэффициентами ($m' < j \leqslant m$):

$$\sum_{k=1}^{m'} c_{ks} = a_j. \tag{2.3}$$

Очевидно, что любой вектор $h^{(s)} = (c_{*s} \mid e_{j-m'})^{\mathsf{T}}$ является решением уравнения (2.1), т. к. в силу уравнения (2.3) выполняется равенство $\sum_{k=1}^{m'} c_{ks} + 1 = 1 + a_j$.

Любые два решения $h', h'' \in \mathcal{H}$ несравнимы друг с другом: 1) либо $h'_{m'+k} = h''_{m'+l} = 0$ и $h'_{m'+l} = h''_{m'+k} = 1$ (единицы на разных позициях, $1 \leqslant k, l \leqslant m - m'$, $k \neq l$), а значит h' и h'' несравнимы, 2) либо $h'_{m'+k} = h''_{m'+k} = 1$, а тогда h' и h'' несравнимы в силу несравнимости c' и c'' как решений уравнения (2.3) для $j = m' + k$.

Докажем, что произвольное решение $x = (x_1, \ldots, x_m)^{\mathsf{T}}$ уравнения (2.1) может быть выражено через элементы множества $\mathcal{H}$. Тем самым мы докажем, что $\mathcal{H}$ является базисом Гильберта.

Для $x = \mathbb{O}$ это очевидно. Пусть x — ненулевое решение уравнения (2.1). Тогда $\sum_{j=m'+1}^{m} x_j > 0$. Следовательно найдется $x_j > 0$. Найдем $h \in \mathcal{H}$ т. ч. $h_j = 1$ и $x - h > \mathbb{O}$. Представим $x = x' + h$, где $x'_l = x_l$ для $l = m'+1, \ldots, j-1, j+1, \ldots, m$, $x'_j = x_j - 1$. Тогда из уравнения (2.1) получаем

$$\sum_{k=1}^{m'} (x'_k + h_k) = \sum_{l=m'+1}^{m} a_l x_l = \sum_{\substack{l=m'+1\\ l\neq j}}^{m} a_l x_l + a_j x_j,$$

$$\sum_{k=1}^{m'} x'_k + \sum_{k=1}^{m'} h_k = \sum_{\substack{l=m'+1\\ l\neq j}}^{m} a_l x_l + a_j(x_j - 1) + a_j.$$

Последнее равенство будет выполняться для x' и h, удовлетворяющих следующим двум уравнениям:

$$\begin{cases} \sum_{k=1}^{m'} x'_k = \sum_{\substack{l=m'+1\\ l\neq j}}^{m} a_l x_l + a_j(x_j - 1) \\ \sum_{k=1}^{m'} h_k = a_j \end{cases}$$

Первое уравнение разрешимо относительно x' в неотрицательных целых, т. к. является уравнением Фробениуса с неотрицательной правой частью. Второе уравнение в точности совпадает с уравнением (2.3), а значит в $\mathcal{H}$ есть элемент h, являющийся его решением. Следовательно, найдено искомое представление $x = x' + h$.

Очевидно, что вектор x' также является решением уравнения (2.1). Аналогично, можно найти представление $x' = x'' + h$ для некоторого $h \in \mathcal{H}$. Продолжая этот процесс за конечное число шагов получим нулевое решение:

$$x = x' + h' = x'' + h'' + h' = \cdots = \mathbb{O} + \sum_{s=1} \alpha_s h^{(s)}.$$

Итак, получаем разложение произвольного решения x в виде неотрицательной линейной целочисленной комбинации векторов из $\mathcal{H}$, что и завершает доказательство теоремы. $\square$

Отметим, что другое доказательство этой теоремы, основанное на методах теории формальных грамматик, дано в [2].

Алгоритмы генерации тестовых систем АНЛДУ, которые будут представлены далее в этой главе, основаны на построении системы и ее базиса Гильберта, начиная с некоторой простейшей системы. Такое построение является обратным для следующего, предлагаемого нами преобразования произвольной однородной системы АНЛДУ.

Пусть $S^{(0)}(I, A)$ — произвольная однородная система АНЛДУ $\mathrm{E}(I)x = Ax$. Эту систему можно записать как $\mathfrak{A}^{(0)}x = \mathbb{O}$, где $\mathfrak{A}^{(0)} = A - \mathrm{E}(I)$. Без потери общности будем предполагать, что $S^{(0)}$ не содержит тождественных уравнений $0 = 0$ (их всегда можно удалить без изменения множества решений системы).

Преобразование системы $S^{(0)}$ будем выполнять пошагово

$$S^{(0)} \to S^{(1)} \to \cdots \to S^{(r)}$$

до тех пор, пока не получим систему $S^{(r)}$, для которой дальнейшее преобразование невозможно. После каждого шага преобразования очередная система будет иметь на одно уравнение меньше.

Пусть $S^{(l)}$ — текущая система $\mathfrak{A}^{(l)}x = \mathbb{O}$. Сложим все уравнения:

$$\sum_{j=1} c_j x_j = 0, \quad \text{где } c_j = \sum_i \mathfrak{A}^{(l)}_{ij}. \tag{2.4}$$

Суммирование при вычислении c_j идет по всем уравнениям i системы $S^{(l)}$. Преобразование $S^{(l)} \to S^{(l+1)}$ выполняется, если хотя бы один коэффициент $c_j = -1$. В этом случае найдется уравнение i_0 т. ч.

$$K = \left\{ k \mid \mathfrak{A}^{(l)}_{i_0 k} = -1 \text{ и } \mathfrak{A}^{(l)}_{ik} = 0 \text{ для всех } i \neq i_0 \right\} \neq \varnothing.$$

Используя уравнение i_0 выразим неизвестные x_k, $k \in K$:

$$\sum_{k \in K} x_k = \sum_{k \notin K} \mathfrak{A}^{(l)}_{i_0 k} x_k = T_{l+1}(x).$$

Искомая система $S^{(l+1)}$ строится как состоящая из всех уравнений системы $S^{(l)}$ за исключением уравнения i_0.

Преобразование завершается через конечное число шагов, поскольку на каждом шаге из системы удаляется ровно одно уравнение. В результате получаем систему $S^{(r)}$, где $0 \leqslant r < n$, n — число уравнений в исходной системе $S^{(0)}$. В полученной системе будет $n - r$ уравнений.

Если система $S^{(r)}$ состоит из одного уравнения, то она имеет следующий вид:

$$\sum_{j \in I_l} x_j = \sum_{j=1}^{m} a_{lj} x_j \tag{2.5}$$

где l — оставшееся уравнение исходной системы.

Если $p = n - r > 1$, то все коэффициенты $c_j \geqslant 0$. Следовательно при $c_j > 0$ $x_j = 0$. Для коэффициентов $c_j = 0$ система имеет следующий вид:

$$\sum_{i \in I_k} x_i = \sum_{j \in J_k} x_j, \; k = \overline{1, p}, \tag{2.6}$$

где $\bigcup\limits_{k=\overline{1,p}} I_k = \bigcup\limits_{k=\overline{1,p}} J_k$. Разбиение I, J — разбиение для системы $S^{(r)}$ с учетом того, что $x_j = 0$ для $c_j > 0$.

Базис Гильберта $\mathcal{H}^{(0)}$ исходной системы $S^{(0)}$ может быть вычислен по базису Гильберта $\mathcal{H}^{(r)}$ итоговой системы $S^{(r)}$ (уравнение (2.5) или система (2.6)) в порядке, обратном выполненному преобразованию:

$$\mathcal{H}^{(r)} \to \mathcal{H}^{(r-1)} \to \cdots \to \mathcal{H}^{(0)}.$$

Пусть $\mathcal{H}^{(l+1)}$ — текущий базис Гильберта, соответствующий системе $\mathfrak{A}^{(l+1)} x = \mathbb{O}$. Тогда существует уравнение

$$\sum_{k \in K} x_k = \sum_{k \notin K} \mathfrak{A}^{(l)}_{i_0 k} x_k = T_{l+1}(x). \tag{2.7}$$

которое было исключено из системы $\mathfrak{A}^{(l)} x = \mathbb{O}$ для преобразования в систему $\mathfrak{A}^{(l+1)} x = \mathbb{O}$. Компоненты решения x_k, $k \in K$ можно получить подставляя каждый вектор из базиса Гильберта $\mathcal{H}^{(l+1)}$ в уравнение (2.7). Полученный базис является базисом Гильберта $\mathcal{H}^{(l)}$ для системы $\mathfrak{A}^{(l)} x = \mathbb{O}$.

Теорема 2.2. *Пусть задана произвольная система АНЛДУ из n уравнений с m неизвестными:*

$$\mathrm{E}(I_1, \ldots, I_n) x = Ax. \tag{2.8}$$

Задача нахождения базиса Гильберта системы (2.8) сводится к задаче нахождения базиса Гильберта либо уравнения вида (2.5), либо системы вида (2.6).

Доказательство. Преобразуем систему (2.8) либо к виду (2.5), либо к виду (2.6), как было показано выше. Решаем задачу нахождения базиса Гильберта полученной системы. Вычисляем базис Гильберта исходной системы в порядке, обратном выполненному преобразованию. □

Таким образом, теорема 2.2 сводит задачу генерации системы АНЛДУ и ее базиса Гильберта к генерации более простой системы — уравнения (2.5) или системы (2.6).

Замечание 2.1. Если исходная система (2.8) сводится к уравнению (2.5), то нахождение базиса Гильберта последнего может быть получено на основе теоремы 2.1. В этом случае решение задачи генерации системы (2.8) начинается с генерации уравнения (2.5) и построения его базиса Гильберта (2.2).

Замечание 2.2. Система (2.6) также является однородной системой АНЛДУ. В то же время она имеет более простой вид по сравнению с исходной системой. Во-первых, она содержит меньшее число уравнений. Во-вторых, содержит только нулевые и единичные коэффициенты. Тем самым, нахождение ее базиса Гильберта является, вообще говоря, более простой задачей, чем нахождение базиса исходной системы.

Пример 2.1 *(Приведение однородной системы АНЛДУ к виду (2.5)).* Пусть задана система АНЛДУ:

$$\begin{cases} x_1 + x_3 &= 3x_4 + 2x_5 \\ x_2 &= 5x_1 + 7x_3 + 2x_4 + 3x_5 \\ x_4 + x_5 &= 4x_5 \end{cases}$$

Строим уравнение вида (2.4):

$$4x_1 - x_2 + 6x_3 + 4x_4 + 8x_5 = 0.$$

Применим преобразование системы ($c_2 = -1$). Следовательно получаем правило вычисления значения переменной x_2:

$$x_2 = 5x_1 + 7x_3 + 2x_4 + 3x_5 = T_1(x).$$

Исключаем уравнение 2 из рассмотрения и получаем систему вида:

$$\begin{cases} x_1 + x_3 &= 3x_4 + 2x_5 \\ x_4 + x_5 &= 4x_5 \end{cases}$$

Пересчитываем уравнение (2.4):

$$-x_1 - x_3 + 2x_4 + 5x_5 = 0.$$

Применим преобразование системы ($c_1 = -1$ и $c_3 = -1$). Получаем правило для вычисления переменных x_1 и x_3:

$$x_1 + x_3 = 3x_4 + 2x_5 = T_2(x).$$

Исключаем уравнение 1 из рассмотрения и в итоге у нас остается только одно уравнение. Следовательно мы привели систему АНЛДУ к виду (2.5):

$$\begin{cases} x_4 + x_5 & = & 4x_5 \\ x_1 + x_3 & = & T_2(x) \\ x_2 & = & T_1(x) \end{cases}$$

Нахождение базиса Гильберта для данной системы выглядит так:

$$x_4 = 3x_5$$

Базис Гильберта для данного уравнения:

$$\mathcal{H} = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 3 \\ 1 \end{pmatrix} \right\},$$

Применяем второе правило:

$$x_1 + x_3 = 3 * 3 + 2 * 1 = 11.$$

Получаем 11 решений уравнения. Применяем первое правило:

$$x_2 = 5x_1 + 7x_3 + 2x_4 + 3x_5.$$

Получаем базис Гильберта для однородной системы АНЛДУ.

$$\mathcal{H} = \begin{pmatrix} 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 64 & 66 & 68 & 70 & 72 & 74 & 76 & 78 & 80 & 82 & 84 & 86 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Пример 2.2 *(Приведение однородной системы АНЛДУ к виду* (2.6)*).* Пусть задана система АНЛДУ:

$$\begin{cases} x_1 + x_3 & = & x_1 + 3x_3 \\ x_2 & = & x_2 + 5x_3 \\ x_4 & = & 2x_1 + 3x_2 + 7x_3 \end{cases}$$

Строим уравнение вида (2.4):

$$2x_1 + 3x_2 + 14x_3 - x_4 = 0.$$

Применим преобразование системы ($c_4 = -1$). Получаем правило для вычисления значения переменной x_4:

$$x_4 = 2x_1 + 3x_2 + 7x_3 = T_1(x).$$

Исключаем уравнение 3 из рассмотрения и получаем систему вида:

$$\begin{cases} x_1 + x_3 & = & x_1 + 3x_3 \\ x_2 & = & x_2 + 5x_3 \end{cases}$$

Пересчитываем уравнение (2.4):

$$7x_3 = 0.$$

Так как больше нельзя применять преобразование системы, то полагаем переменную $x_3 = 0$, т.к. $c_3 > 0$. В итоге данная система приведена к виду (2.6):

$$\begin{cases} x_1 & = & x_1 \\ x_2 & = & x_2 \\ x_3 & = & 0 \\ x_4 & = & T_1(x) \end{cases}$$

Нахождение базиса Гильберта выглядит так: решаем систему уравнений

$$\begin{cases} x_1 = x_1 \\ x_2 = x_2 \end{cases}$$

Базис Гильберта для этой системы:

$$\mathcal{H} = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \right\},$$

Применяем правило для нахождения x_4 и получаем базис Гильберта для однородной системы АНЛДУ:

$$\mathcal{H} = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 3 \end{pmatrix} \right\}.$$

2.2. Постановка задачи генерации

Задача заключается в одновременном построении однородной системы АНЛДУ (2.8) и ее базиса Гильберта $\mathcal{H}$. Системы генерируются с точностью до эквивалентной.

В случае произвольной системы эта задача вычислительно эквивалентна задаче нахождения базиса Гильберта. Разумным в такой ситуации является выделение некоторых частных классов систем АНЛДУ, допускающих более простое решение задачи генерации.

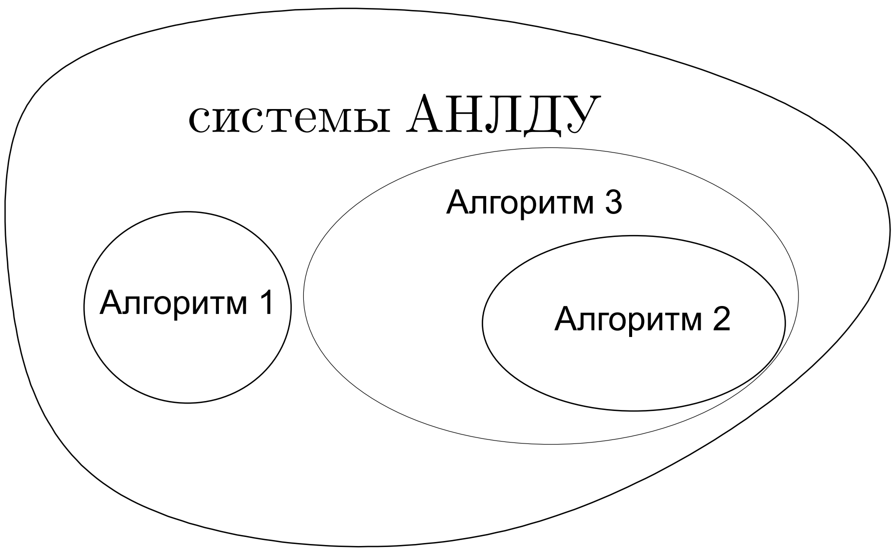


Рис. 1. Разбиение множества однородных систем АНЛДУ на различные классы генерируемых систем.

В общем случае алгоритм генерации систем некоторого класса получает в качестве входных параметров размерности генерируемой системы (число уравнений n, число неизвестных m, $0 < n \leqslant m$). Выходными параметрами являются разбиение множества неизвестных I, матрица коэффициентов правой части A, матрица коэффициентов левой части $\mathrm{E}(I)$ и базис Гильберта $\mathcal{H}$.

Работа алгоритмов нахождения базиса Гильберта зависит от размерностей генерируемой системы, количества элементов базиса Гильберта, величин коэффициентов матрицы A. Поэтому, для получения определенных характеристик генерируемых систем требуется управление параметрами алгоритмов: ограничение на количество элементов базиса Гильберта, на величины коэффициентов матрицы A.

Результаты генерации могут использоваться, во-первых, для тестирования программных реализаций алгоритмов решения систем НЛДУ. В этом случае после генерации тройки $(\mathrm{E}(I), A, \mathcal{H})$ можно дополнительно выполнить перенумерацию неизвестных и/или уравнений. Во-вторых, сгенерированные системы могут использоваться для проведения экспериментального анализа и сравнения алгоритмов решения систем НЛДУ.

2.3. Алгоритм 1 (аналог преобразования Гаусса — Жордано)

Пусть n и m зафиксированны, причем $0 < n \leqslant m$. Будем строить систему АНЛДУ вида

$$\mathrm{E}x = Ax \tag{2.9}$$

с базисом Гильберта, содержащим стандартные единичные вектора из $\mathbb{Z}^m$:

$$\mathcal{H} = \{e_1, e_2, \ldots, e_n\}. \tag{2.10}$$

Матрицу E коэффициентов левой части строим как $\mathrm{E} = (E \mid B)$, где E — стандартная единичная $(n \times n)$-матрица ($E = \mathrm{E}(\{1\}, \{2\}, \ldots, \{n\})$), а $B \geqslant \mathbb{O}$ — произвольная $n \times (m-n)$-матрица. Матрицу A коэффициентов правой части строим как $A = (E \mid C)$, где E — стандартная единичная $(n \times n)$-матрица, $C = B + \Delta$ и матрица $\Delta \in \mathbb{Z}_+^{n \times (m-n)}$ удовлетворяет следующему условию:

$$\forall\, i \text{ уравнение } \quad \sum_{j=1}^{m-n} \Delta_{ij} x_{j+n} = 0 \quad \text{имеет только нулевое решение.} \tag{2.11}$$

Теорема 2.3. *Базис Гильберта системы* (2.9) *есть* (2.10).

Доказательство. Переписывая систему (2.9) получаем

$$(E \mid B)x = (E \mid (B + \Delta))x \quad \Rightarrow \quad (\mathbb{O} \mid \Delta)x = \mathbb{O}$$

Тем самым, $\mathbb{O}x' + \Delta x'' = \mathbb{O}$, где $x' = (x_1, \ldots, x_n)^\mathsf{T}$, $x'' = (x_{n+1}, \ldots, x_m)^\mathsf{T}$. Система $\Delta x'' = \mathbb{O}$ имеет только нулевое решение $x'' = \mathbb{O} \in \mathbb{Z}^{m-n}$ в силу условия (2.11). Решениями системы

$\mathbb{O}x' = \mathbb{O}$ очевидно являются любые вектора из Z_+^n, а значит минимальные решения определяются базисом (2.10). $\square$

На основе вышесказанного получаем алгоритм 1 для генерации систем вида (2.9).

Алгоритм 1 Генерация системы АНЛДУ и базиса Гильберта к ней

Вход: n — число уравнений, m — число неизвестных, $n \leqslant m$.

Выход: $(\mathrm{E}, A, \mathcal{H})$.

1: $\mathcal{H} = \{e_1, e_2, \ldots, e_n\}$;

2: генерируем случайным образом матрицу $B \in \mathbb{Z}_+^{n\times(m-n)}$;

3: генерируем случайным образом матрицу $\Delta \in \mathbb{Z}^{n\times(m-n)}$, удовлетворяющую (2.11);

4: $\mathrm{E} := (E \mid B)$; $\quad A := (E \mid B + \Delta)$; $\quad$ {$E \in \mathbb{Z}^{n\times n}$ — единичная матрица}

Согласно теоремы 2.2 задача нахождения базиса Гильберта для системы (2.9) сводится к задаче нахождения базиса Гильберта для системы вида (2.6). Данный алгоритм является аналогом преобразования Гаусса — Жордано в конечном представлении матриц коэффициентов в виде $A = (E \mid C)$.

Пример 2.3. Вход: $n = 3, \quad m = 4$

Шаг 1:

$$\mathcal{H} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Шаг 2:

$$B = (1, 0, 0)^{\mathsf{T}}.$$

Шаг 3:

$$\Delta = (2, 2, 3)^{\mathsf{T}}.$$

Шаг 4:

$$\mathrm{E} = (E \mid B) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad A = (E \mid B + \Delta) = \begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{pmatrix}.$$

2.4. Алгоритм 2 (аналог преобразования Гаусса)

Пусть n и m зафиксированны, причем $0 < n < m$. Будем строить систему АНЛДУ вида

$$\mathrm{E}x = Ax \tag{2.12}$$

с базисом Гильберта:

$$\mathcal{H} = \{F \mid E\}^{\mathsf{T}}, \tag{2.13}$$

где E — стандартная единичная $((m-n)\times(m-n))$-матрица ($E = \mathrm{E}(\{1\},\{2\},\ldots,\{m-n\})$), а $F \geqslant \mathbb{O}$ — $(m-n)\times n$-матрица.

Матрицу E коэффициентов левой части строим как $\mathrm{E} = (E \mid B)$, где E — стандартная единичная $(n \times n)$-матрица ($E = \mathrm{E}(\{1\},\{2\},\ldots,\{n\})$), а $B \geqslant \mathbb{O}$ — произвольная $n \times (m-n)$-матрица. Матрицу A коэффициентов правой части строим как $A = (D^+ \mid C)$, где $D^+ \in \mathbb{Z}_+^{n\times n}$, т. ч.

$$D^+_{ij} = \begin{cases} 0 & \text{, если } i \leqslant j \\ d_{ij} \in \mathbb{Z}_+ & \text{, если } i > j \end{cases} \tag{2.14}$$

$C = B + \Delta$ и матрица $\Delta \in \mathbb{Z}_+^{n\times(m-n)}$ удовлетворяет следующему условию:

$$\forall\, i \text{ уравнение } \sum_{j=1}^{m-n} \Delta_{ij} x_{j+n} = 0 \quad \text{имеет только нулевое решение.} \tag{2.15}$$

Теорема 2.4. *Базис Гильберта системы* (2.12) *есть* (2.13).

Доказательство. Согласно теореме 2.2 произведем преобразование системы (2.12).

$$(E \mid B)x = (D^+ \mid (B+\Delta))x \quad \Rightarrow \quad ((D^+ - E) \mid \Delta)x = \mathbb{O}$$

Тем самым, $(D^+ - E)x' + \Delta x'' = \mathbb{O}$, где $x' = (x_1,\ldots,x_n)^{\mathsf{T}}$, $x'' = (x_{n+1},\ldots,x_m)^{\mathsf{T}}$. В силу специфики построения матрицы $(D^+ - E)$ система будет преобразована к уравнению вида (2.5):

$$x_n = \sum_{j=1}^{m-n} \Delta_{nj} x_{j+n}$$

Согласно теореме 2.1 базис Гильберта данного уравнения есть

$$\mathcal{H} = \{F' \mid E\}^{\mathsf{T}}.$$

Вычисляя базис Гильберта в обратном порядке получаем (2.13). □

Алгоритм 2 Генерация системы АНЛДУ и базиса Гильберта к ней

Вход: n — число уравнений, m — число неизвестных, $n < m$.

Выход: $(\mathrm{E}, A, \mathcal{H})$.

1: генерируем случайным образом матрицу $D^+ \in \mathbb{Z}_+^{n\times n}$, удовлетворяющую (2.14);

2: генерируем случайным образом матрицу $B \in \mathbb{Z}_+^{n\times(m-n)}$;

3: генерируем случайным образом матрицу $\Delta \in \mathbb{Z}^{n\times(m-n)}$, удовлетворяющую (2.15);

4: вычисляем матрицу $F \in \mathbb{Z}_+^{(m-n)\times n}$;

5: $\mathrm{E} := (E \mid B)$; $A := (D^+ \mid B + \Delta)$; $\mathcal{H} := \{F \mid E\}^\mathsf{T}$ $\triangleright$ $\{E \in \mathbb{Z}^{n\times n}$ — единичная матрица$\}$

На основе вышесказанного получаем алгоритм 2 для генерации систем вида (2.12).

Согласно теоремы 2.2 задача нахождения базиса Гильберта для системы (2.12) сводится к задаче нахождения базиса Гильберта для системы вида (2.5). Данный алгоритм является аналогом преобразования Гаусса в конечном представлении матриц коэффициентов в виде $A = (D^+ \mid C)$.

Пример 2.4. Вход: $n = 3, \quad m = 5$

Шаг 1:

$$D^+ = \begin{pmatrix} 0 & 2 & 3 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}$$

Шаг 2:

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Шаг 3:

$$\Delta = \begin{pmatrix} 0 & 0 \\ 4 & 2 \\ 3 & 4 \end{pmatrix}$$

Шаг 4:

$$F = \begin{pmatrix} 47 & 19 & 3 \\ 52 & 20 & 4 \end{pmatrix}$$

Шаг 5:

$$\mathrm{E} = (E \mid B) = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad A = (D^+ \mid B + \Delta) = \begin{pmatrix} 0 & 2 & 3 & 1 & 0 \\ 0 & 0 & 5 & 4 & 3 \\ 0 & 0 & 0 & 3 & 4 \end{pmatrix}$$

$$\mathcal{H} = (F \mid E)^{\mathsf{T}} = \begin{pmatrix} 47 & 52 \\ 19 & 20 \\ 3 & 4 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

2.5. Алгоритм 3 (расширенный аналог преобразования Гаусса)

Пусть n и m зафиксированны, причем $0 < n < m$. Данный метод строит систему АНЛДУ, которая может быть преобразована с помощью теоремы 2.2 к уравнению вида (2.5), т. е. систему вида

$$\mathrm{E}x = Ax. \tag{2.16}$$

Матрицу E коэффициентов левой части строим как

$$\mathrm{E} = \left(\begin{array}{cccccccccc|c} e_{11} & \ldots & e_{1(k_1-1)} & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 & \\ 0 & \ldots & 0 & e_{2k_1} & \ldots & e_{2(k_2-1)} & 0 & \ldots & \ldots & 0 & \\ & & & & & & & & & & B \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \\ 0 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 & e_{nk_{n-1}} & \ldots & e_{n(k_n-1)} & \end{array} \right), \tag{2.17}$$

где $B \geqslant \mathbb{O}$ — произвольная $n \times (m-n-p)$-матрица, $\{k_1, k_2, \ldots, k_{n-1}, k_n = n+p+1\}$ произвольная последовательность возрастающих чисел: $0 < k_1 < k_2 < \cdots < k_n = n+p+1$, $e_{ij} = 1$, $p \geqslant 0$ — произвольная константа.

Матрицу A коэффициентов правой части строим как

$$A = \left(\begin{array}{cccccc|c} 0 & \ldots & 0 & a_{1k_1} & \ldots & a_{1(n+p)} & \\ 0 & \ldots & \ldots & 0 \;\; a_{2k_2} & \ldots & a_{2(n+p)} & C \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \\ 0 & \ldots & \ldots & \ldots & \ldots & 0 & \end{array} \right), \tag{2.18}$$

где $a_{ij} \leqslant 0$,

$C = B + \Delta$ и матрица $\Delta \in \mathbb{Z}_+^{n\times(m-n-p)}$ удовлетворяет следующему условию:

$$\forall\, i \text{ уравнение } \sum_{j=1}^{m-n-p} \Delta_{ij} x_{j+n+p} = 0 \quad \text{имеет только нулевое решение.} \tag{2.19}$$

На основе вышесказанного получаем алгоритм 3 для генерации систем вида (2.16).

Алгоритм 3 Генерация системы АНЛДУ и базиса Гильберта к ней

Вход: n — число уравнений, m — число неизвестных, $n < m$.

Выход: $(\mathrm{E}, A, \mathcal{H})$.

1: генерируем случайным образом число p и последовательность $\{k_1, \ldots, k_n\}$;
2: генерируем случайным образом матрицу $B \in \mathbb{Z}_+^{n\times(m-n-p)}$;
3: генерируем случайным образом матрицу $\Delta \in \mathbb{Z}^{n\times(m-n-p)}$, удовлетворяющую (2.19);
4: строим матрицу E вида (2.17);
5: генерируем случайным образом матрицу A вида (2.18);
6: с помощью преобразования теоремы 2.2 строим матрицу $\mathcal{H}$;

Согласно теоремы 2.2 задача нахождения базиса Гильберта для системы (2.16) сводится к задаче нахождения базиса Гильберта для системы вида (2.5). Данный алгоритм является расширенным аналогом преобразования Гаусса в конечном представлении матриц коэффициентов в виде (2.18).

Пример 2.5. Вход: $n = 3, \quad m = 6$

Шаг 1:

$$p = 1 \quad k_1 = 3, k_2 = 4, k_3 = 5$$

Шаг 2:

$$B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Шаг 3:

$$\Delta = \begin{pmatrix} 3 & 2 \\ 3 & 3 \\ 3 & 3 \end{pmatrix}$$

Шаг 4:

$$\mathrm{E} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Шаг 5:

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 3 & 2 \\ 0 & 0 & 0 & 5 & 4 & 3 \\ 0 & 0 & 0 & 0 & 3 & 4 \end{pmatrix}$$

Шаг 6:

$$\mathcal{H} = \begin{pmatrix} 6 & 5 & 4 & 3 & 2 & 1 & 0 & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 & 1 & 2 & 3 & 4 & 5 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

2.6. Алгоритм генерации полного класса систем АНЛДУ

Пусть n и m зафиксированны, причем $0 < n \leqslant m$. Данный метод строит систему АНЛДУ вида

$$\mathrm{E}x = Ax. \tag{2.20}$$

Матрицу E коэффициентов левой части строим как

$$\mathrm{E} = \left(\begin{array}{c|c} E' & B \\ \hline \mathbb{O} & B' \end{array}\right), \tag{2.21}$$

$$E' = \begin{pmatrix} e_{11} & \dots & e_{1(k_1-1)} & 0 & \dots\dots\dots\dots\dots\dots & & & 0 \\ 0 & \dots & 0 & e_{2k_1} & \dots & e_{2(k_2-1)} & 0 & \dots\dots\dots & 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ 0 & \dots\dots\dots\dots\dots\dots & & & 0 & e_{qk_{q-1}} & \dots & e_{q(k_q-1)} \end{pmatrix} \tag{2.22}$$

где $B \geqslant \mathbb{O}$ — произвольная $q \times (m-q-p)$-матрица, $B' \geqslant \mathbb{O}$ — произвольная $(n-q) \times (m-q-p)$-матрица, $\{k_1, k_2, \dots, k_{q-1}, k_q = q+p+1\}$ произвольная последовательность возрастающих чисел: $0 < k_1 < k_2 < \dots < k_q = q+p+1$, $e_{ij} = 1$, $0 \leqslant p < m-q$, $0 \leqslant q \leqslant n$ — произвольные константы.

Матрицу A коэффициентов правой части строим как

$$A = \left(\begin{array}{c|c} A' & C \\ \hline \mathbb{O} & C' \end{array}\right), \tag{2.23}$$

$$A' = \begin{pmatrix} 0 & \ldots & 0 & a_{1k_1} & \ldots\ldots\ldots\ldots\ldots & & a_{1(q+p)} \\ 0 & \ldots\ldots\ldots\ldots\ldots & & 0 & a_{2k_2} & \ldots & a_{2(q+p)} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & & & & & & \\ 0 & \ldots\ldots\ldots\ldots\ldots\ldots\ldots & & & & & 0 \end{pmatrix} \qquad (2.24)$$

где $a_{ij} \leqslant 0$,

$C = B + \Delta$ и матрица $\Delta \in \mathbb{Z}_+^{q\times(m-q-p)}$ удовлетворяет следующему условию:

$$\forall\, i \text{ уравнение } \sum_{j=1}^{m-q-p} \Delta_{ij} x_{j+q+p} = 0 \quad \text{имеет только нулевое решение,} \qquad (2.25)$$

$C' \in \mathbb{Z}_+$ — булева $(n-q) \times (m-q-p)$-матрица.

Теорема 2.5. *С помощью преобразования теоремы* 2.2 *система* (2.20) *может быть приведена либо к виду* (2.5)*, если* $q = n$*; либо к виду* (2.6)*, если* $q \neq n$*.*

Доказательство. 1). Пусть $q = n$. Следовательно мы получаем систему

$$(E' \mid B)x = (A' \mid B + \Delta)x \quad \Rightarrow \quad (A' - E')x' + \Delta x'' = \mathbb{O}$$

$x' = (x_1, \ldots, x_n)^{\mathsf{T}}$, $x'' = (x_{n+1}, \ldots, x_m)^{\mathsf{T}}$. В силу специфики построения матрицы $(A' - E')$ с помощью преобразования теоремы 2.2 система может быть преобразована к уравнению вида (2.5). Согласно теореме 2.1 находим базис Гильберта данного уравнения. Вычисляя базис Гильберта в обратном порядке получаем базис Гильберта для системы (2.20).

2). Пусть $q \neq n$. Следовательно получаем систему

$$\begin{pmatrix} E' & B \\ \hline \mathbb{O} & B' \end{pmatrix} x = \begin{pmatrix} A' & B + \Delta \\ \hline \mathbb{O} & C' \end{pmatrix} x \quad \Rightarrow \quad \begin{pmatrix} A' - E' \\ \hline \mathbb{O} \end{pmatrix} x' + \begin{pmatrix} \Delta \\ \hline C' - B' \end{pmatrix} x'' = \mathbb{O}$$

С помощью преобразования теоремы 2.2 данная система приводится к системе

$$(C' - B')x'' = \mathbb{O},$$

что соответствует системе вида (2.6). Находим базис Гильберта данной системы, вычисляем базис в обратном порядке и получаем базис Гильберта для системы (2.20). $\square$

На основе вышесказанного получаем алгоритм 4 для генерации систем вида (2.20).

Согласно теоремы 2.2 задача нахождения базиса Гильберта для системы (2.20) сводится к задаче нахождения базиса Гильберта либо для системы вида (2.5), либо для системы вида (2.6).

Алгоритм 4 Генерация системы АНЛДУ и базиса Гильберта к ней

Вход: n — число уравнений, m — число неизвестных, $n \leqslant m$.

Выход: $(\mathrm{E}, A, \mathcal{H})$.

1: генерируем случайным образом числа p, q и последовательность $\{k_1, \dots, k_q\}$;

2: генерируем случайным образом матрицу $B \in \mathbb{Z}_+^{q\times(m-q-p)}$, $B' \in \mathbb{Z}_+^{(n-q)\times(m-q-p)}$;

3: генерируем случайным образом матрицу $\Delta \in \mathbb{Z}^{n\times(m-q-p)}$, удовлетворяющую (2.25);

4: генерируем случайным образом матрицу $A' \in \mathbb{Z}^{q\times(q+p)}$, $E' \in \mathbb{Z}^{q\times(q+p)}$;

5: генерируем случайным образом матрицу $C' \in \mathbb{Z}_+^{(n-q)\times(m-q-p)}$;

6: строим матрицу E вида (2.21);

7: строим матрицу A вида (2.23);

8: с помощью преобразования теоремы 2.2 строим матрицу $\mathcal{H}$;

Пример 2.6. Вход: $n = 5, \quad m = 7$

Шаг 1:

$$p = 0, \quad q = 2, \quad k_1 = 2, k_2 = 3;$$

Шаг 2:

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Шаг 3:

$$\Delta = \begin{pmatrix} 3 & 2 & 1 & 4 & 2 \\ 2 & 3 & 5 & 4 & 1 \end{pmatrix}$$

Шаг 4:

$$E' = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad A' = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Шаг 5:

$$C' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Шаг 6:

$$\mathrm{E} = \left(\begin{array}{c|c} E' & B \\ \hline \mathbb{O} & B' \end{array}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Шаг 7:

$$A = \left(\begin{array}{c|c} A' & B+\Delta \\ \hline \mathbb{O} & C' \end{array}\right) = \begin{pmatrix} 0 & 1 & 3 & 2 & 1 & 4 & 2 \\ 0 & 0 & 2 & 3 & 5 & 4 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Шаг 8:

$$\mathcal{H} = \begin{pmatrix} 11 & 3 & 5 \\ 8 & 1 & 2 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

3. Программная система для выполнения тестирования, экспериментального и сравнительного анализа алгоритмов решения систем АНЛДУ

Экспериментальный анализ вычислительной сложности алгоритма решения предназначен для исследования эффективности алгоритма. При этом оценивается объем потребляемых вычислительных ресурсов. В основным из таких ресурсов относят время работы программы и используемый объем памяти. Такая оценка предназначена для решения следующих задач:

1. Построение экспериментальной зависимости объема потребляемых ресурсов от размерности входных данных. В случае систем НЛДУ размерности входных данных могут определяться числом уравнений n, числом неизвестных m, абсолютными величинами коэффициентов (норма системы) и числом элементов в искомом базисе Гильберта.

2. Сравнительный анализ эффективности альтернативных алгоритмов. В случае систем НЛДУ в рамках данной задачи может быть определено, на каких классах систем тот или иной алгоритм решения превосходит по эффективности другие решатели, а на каких классах алгоритм можно считать сопоставимым по эффективности решения.

3. Построение карты эффективности исследуемого алгоритма. В случае систем НЛДУ множество всех возможных систем разбивается на классы, для каждого из которых экспериментально строится оценка эффективности. Тем самым получаем детальное описание эффективности в зависимости от вида решаемых систем, выявляя наихудшие и наилучшие случаи для работы алгоритма.

В силу специфики решаемых задач разработанное программное обеспечение (ПО) должно отвечать следующим требованиям:

1. Модульность. Это необходимо для гибкого подключения новых алгоритмов генерации тестовых систем, исследуемых программ-решателей, используемых методов оценивания потребления ресурсов.

2. Автономность. Возможность работы ПО в автономном режиме на выделенном сервере в течении длительного времени. Это вызвано большим объемом тестов, необходимых для проведения комплексного и детального анализа.

3. Независимость от формата исследуемого алгоритма. Этот алгоритм может быть доступен без исходных кодов, а лишь как непосредственно исполняемая программа. В любом случае разработанное ПО должно осуществлять измерение потребляемых ресурсов.

3.1. Модульная структура ПО

Общая схема работы разработанного ПО представлена на рис.2. При этом

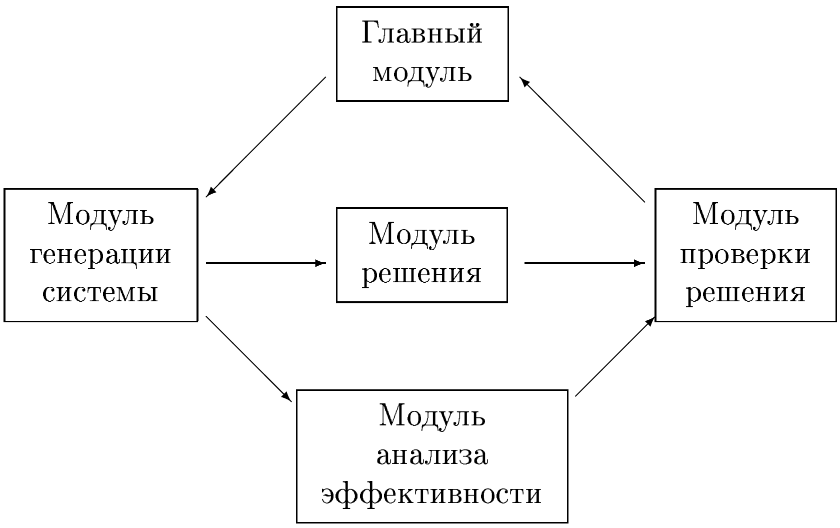


Рис. 2. Схема работы ПО

выделяются следующие модули:

Главный модуль. Главный модуль координирует работу всех остальных модулей. В его задачу входит выбор и последовательный запуск этих модулей, распределение и передача информации модулям. При возникновении критической ситуации (ошибки) он выбирает один из двух вариантов действий:

- зафиксировать возникновение фатальной ошибки путем записи в log-файл и вывода на экран соответствующих сообщений, а затем аварийно завершить работу;
- вывести сообщение о возникновении критической ситуации, записать вызвавшие ошибку данные на диск и продолжить работу (восстановление после ошибки).

При этом приоритет отдается второму направлению действия, т. к. в этом случае впоследствии можно установить факт существования ошибки и вызвавшие ее причины.

Кроме этого главный модуль ведет сбор и вывод первоначальной статистической информации о текущем тесте, ведет подсчет пройденных тестов и задает исходные параметры для модуля генерации системы АНЛДУ.

Модуль генерации системы АНЛДУ. Данный модуль по задаваемым исходным параметрам генерирует систему АНЛДУ и соответствующий ей базис Гильберта. В текущей версии в состав модуля входят три алгоритма генерации (см. 2.3 - 2.5). Исходными параметрами являются: 1) номер алгоритма, 2) количество уравнений n, 3) количество неизвестных m.

Модуль решения. Данный модуль решает систему АНЛДУ и вычисляя ее базис Гильберта. Так как он является внешним, то запускается в виде отдельного процесса и работа с ним идет на уровне входных-выходных файлов.

Модуль проверки решения. Данный модуль проверяет, удалось ли исследуемой программе-решателю получить решение[1] и сравнивает найденный базис Гильберта с тестовым вариантом.

Модуль анализа эффективности. Данный модуль запускается одновременно с запуском модуля решения и снимает статистическую информацию об эффективности работы программы-решателя: время работы, занимаемая память. По завершении работы модуля решения статистическая информация передается главному модулю.

На рис.2 показана последовательность работы модулей ПО, при этом, как было сказано выше, в случае возникновения критической ситуации управление передается главному модулю, независимо от того, какой модуль в данный момент работает, и, в случае успешной обработки критической ситуации, последовательность работы начинается с главного модуля.

3.2. Способы оценки потребления ресурсов

Далее описаны используемые нами способы оценивания объема потребляемых вычислительных ресурсов.

[1]Работа программы-решателя может быть завершена без получения решения, например, из-за нехватки памяти, по истечении отведенного на решение времени, из-за численного переполнения при вычислениях и т. п.

3.2.1. Время

1. Получение времени по системным данным работы процесса. Данная информация находится в папке `\proc\pid`, где `pid` — идентификационный номер процесса. Такой метод обеспечивает получение оценки с точностью до 0.01 секунды, при этом не учитываются время чтения/записи, запуска и завершения работы программы-решателя, время ожидания нового кванта времени и т.д. Недостатком данного метода является то, что при малых вычислительных затратах время работы программы-решателя может быть меньше 0.01 секунды. Следовательно невозможно корректно оценить маштаб вычислительных затрат.

2. Получение времени как разности между временем окончания работы и временем начала работы процесса. Такой метод дает хорошую оценку работы процесса, если время работы настолько мало, что сопоставимо с точностью предыдущего способа. В то же время, отметим следующие его недостатки:

— на значение полученной оценки могут сильно влиять параллельно работающие на сервере процессы. Эти процессы занимают часть ресурсов сервера, увеличивая, тем самым, время работы исследуемого алгоритма решения.

— полученная оценка времени работы включает в себя не только собственно время решения, но и время, требуемое для запуска процесса и т. п.

3. Получение времени как средней разности между временем окончания и временем начала работы процесса, запущенного несколько раз. Данный способ позволяет частично устранить первый из недостатков предыдущего способа. Однако, второй недостаток остается неустраненным.

3.2.2. Память

Объем занимаемой памяти в данный момент при работе программы можно определить по системным данным, которые находятся в папке `\proc\pid`, где `pid` — идентификационный номер процесса. Для получения более детальной и точной информации о работе программы требуется регулярный опрос системных данных. Этот опрос можно организовать с помощью специального процесса, работающего параллельно с программой решения системы.

Схема работы представлена на рис 3. В главном модуле посредством системного вызова *fork()* создается процесс-потомок и возвращается в родительский процессовый

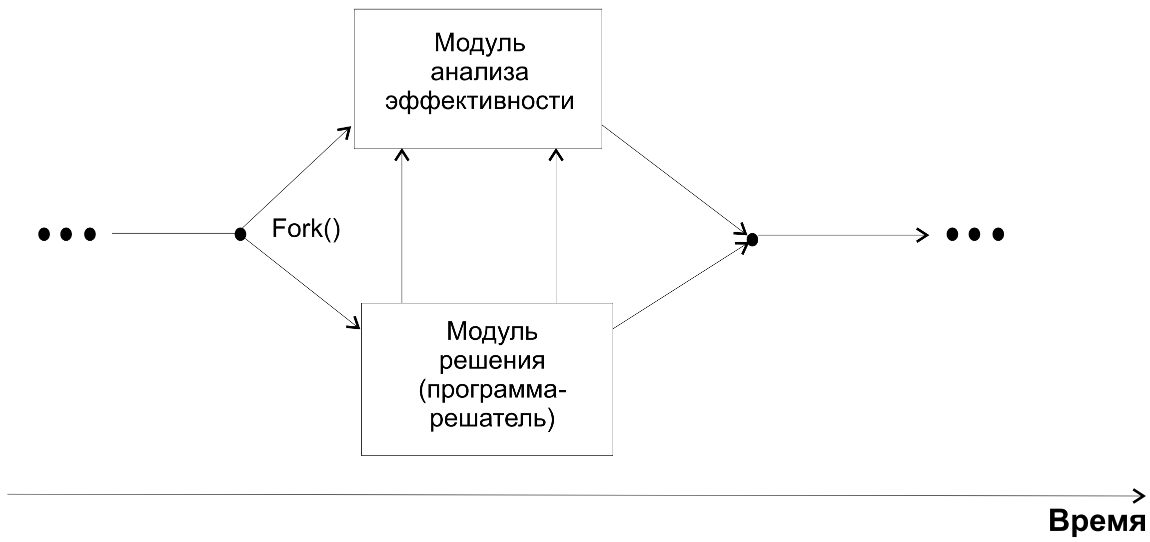


Рис. 3. Схема получения информации об используемой памяти

идентификационный номер. Затем процесс-потомок заменяется на программу-решатель системы АНЛДУ (системный вызов *execl()*), а родительский процесс осуществляет мониторинг и сбор информации о потреблении ресурсов процессом-потомком. Сбор информации останавливается при завершении работы программы-решателя.

Такой способ позволяет следить за изменением количества используемой памяти. В тоже время, недостатком данного способа является то, что часть ресурсов сервера используется для постоянного мониторинга и это замедляет работу программы-решателя.

3.3. Сводные характеристики

В таблице 1 представлены сводные характеристики модулей реализаций ПО. Для реализации используются языки программирования C++ и Perl в среде ОС Linux. Объем каждого модуля указан в строках кода без учета комментариев.

Таблица 1. Характеристики реализованного программного обеспечения

Название модуля	Язык программирования	Количество строк
Главный модуль	C++	739
Модуль генерации:		
Алгоритм 1	C++	55
Алгоритм 2	C++	171
Алгоритм 3	C++	321
Модуль анализа эффективности	C++	145
Модуль проверки решения	Perl	63

4. Эксперименты

Для экспериментального исследования был выбран алгоритм нахождения базиса Гильберта однородных систем АНЛДУ (anlde), разработанный и реализованный Д.Ж.Корзуном [4, 5]. В качестве альтернативного алгоритма был выбран алгоритм нахождения базиса Гильберта систем НЛДУ (slopessys), разработанный и реализованный португальскими математиками A.P.Tomás и M.Filgueiras [9, 10]. Реализации других алгоритмов нам в настоящее время не доступны.

Экспериментальная часть исследований состояла из двух этапов:

1) проверка качества реализации алгоритма на наличие ошибок,

2) проверка эффективности алгоритма anlde и сравнение его с альтернативным алгоритмом slopessys.

На первом этапе было произведено массовое тестирование алгоритма на различных классах тестовых систем для выявления возможных нерешаемых систем. Результаты тестирования представлены в таблице 2. Было сгенерировано и решено более 1.5

Таблица 2. Журнал проведенных тестов

Номер алгоритма	Дата запуска	Дата остановки	Кол-во тестов	Пропу- щено	Ошибочно тестов	Размер матрицы	Размер переменных
1	26.02.02	28.02.02	410786	0	0	100	100000
1	28.02.02	28.02.02	3	0	0	500	100000
2	28.02.02	07.03.02	204308	0	0	1000	100000
1	28.02.02	01.03.02	10	0	0	300	500
1	01.03.02	07.03.02	2837	0	0	1000	10000
1	18.03.02	08.07.02	88640	0	0	1000	10000
1	18.03.02	08.07.02	937234	0	0	1000	10000
Итого	—	—	1643818	0	0	—	—

миллиона систем и ошибок в реализации алгоритма anlde выявлено не было.

На втором этапе была произведена генерация 10 тысяч различных тестовых систем и сравнение на их основе алгоритмов решения anlde и slopessys. Так как португальский алгоритм накладывал существенные ограничения на размерность решаемых систем НЛДУ, то генерация и проверка была произведена на максимально возможной размерности (20). Несмотря на то, что возможности разработанного ПО позволяют измерять используемый объем памяти, для систем столь малых размерностей подобное измерение становилось малоинформативным — размерности систем отличаются друг от друга незначительно и средства операционной системы не позволяют их дифференцировать.

В силу этого, в экспериментальном анализе эффективности мы использовали только измерение затраченного на решение времени.

Описание общих характеристик полученной выборки представлены на рис.6, 9, 12, 15, 18. На основе полученной выборки из 9500 элементов нами строились экспериментальные зависимости потребления времени от ряда параметров, характеризующих систему.

Для каждого графика вычислялись медиана (median), математическое ожидание (E) и два процентеля: 20% и 80%.

На рис.4,5 показано распределение времени решения по номерам систем. Время решения распределено однородно по всей выборке тестовых систем.

На рис.6 показано распределение выборки по числу уравнений n. С ростом n количество соответствующих систем убывает. Число уравнений ограничено сверху числом m и выбиралось с помощью датчика равномерно распределенных чисел от 1 до m.

На рис.7,8 показано распределение времени решения по числу уравнений в системе. С ростом числа уравнений среднее время убывает, т.к. при увеличении числа уравнений количество базисных векторов становится меньше из-за наложения ограничений новыми уравнениями.

На рис.9 показано распределение систем по числу неизвестных в системе. На больших размерностях график убывает, так как нерешенные системы не брались во внимание.

На рис.10,11 показано распределение времени решения по числу неизвестных в системе. Для алгоритма slopessys на графике заметен характерный рост времени при увеличении размерности.

На рис.12 показано распределение систем по числу векторов базиса Гильберта.

На рис.13,14 показано распределение времени решения по векторов базиса Гильберта. Так как алгоритм португальцев содержит переборные методы, то при увеличении количества базисных решений заметен рост времени решения.

На рис.15 показано распределение систем по сумме чисел уравнений и переменных в системе.

На рис.16,17 показано распределение времени решения по сумме чисел уравнений и переменных в системе.

На рис.18 в логарифмической шкале показано распределение систем по произведению чисел уравнений и переменных в системе.

На рис.19,20 в логарифмической шкале показано распределение времени решения по произведению чисел уравнений и переменных в системе.

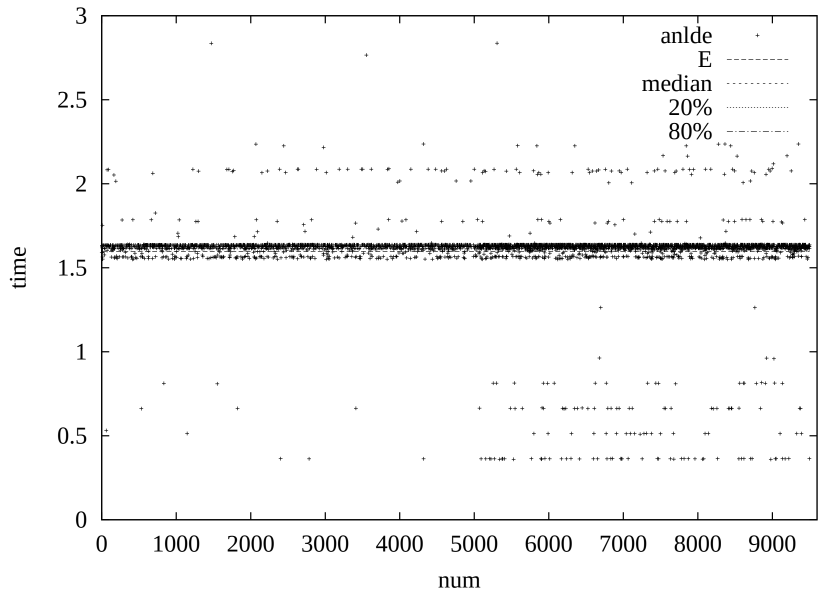


Рис. 4. Распределение времени решения по номерам сгенерированных тестовых систем

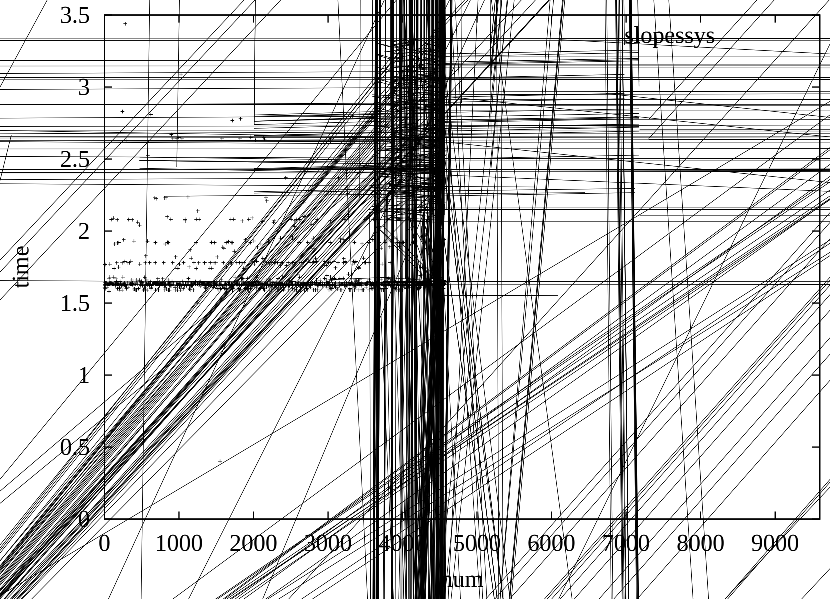
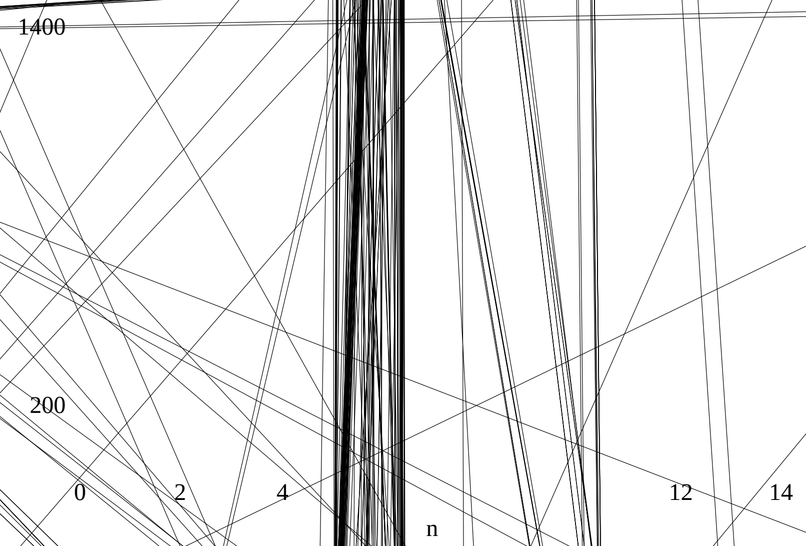
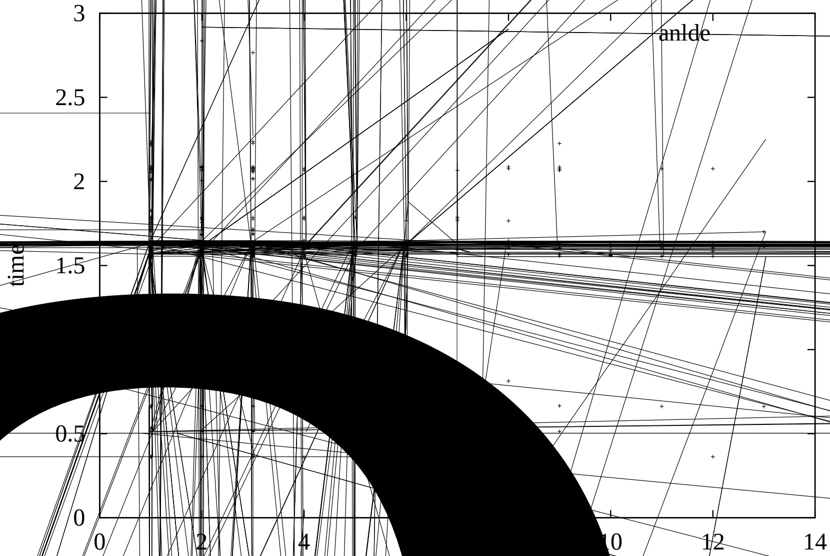


Рис. 5. Распределение времени решения по номерам сгенерированных тестовых систем





80

20

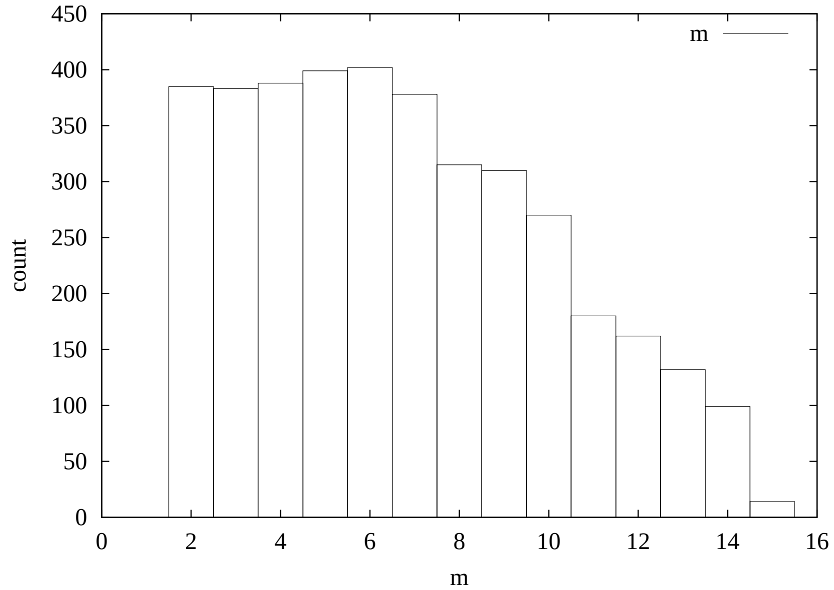


Рис. 9. Распределение систем по числу неизвестных m в системе

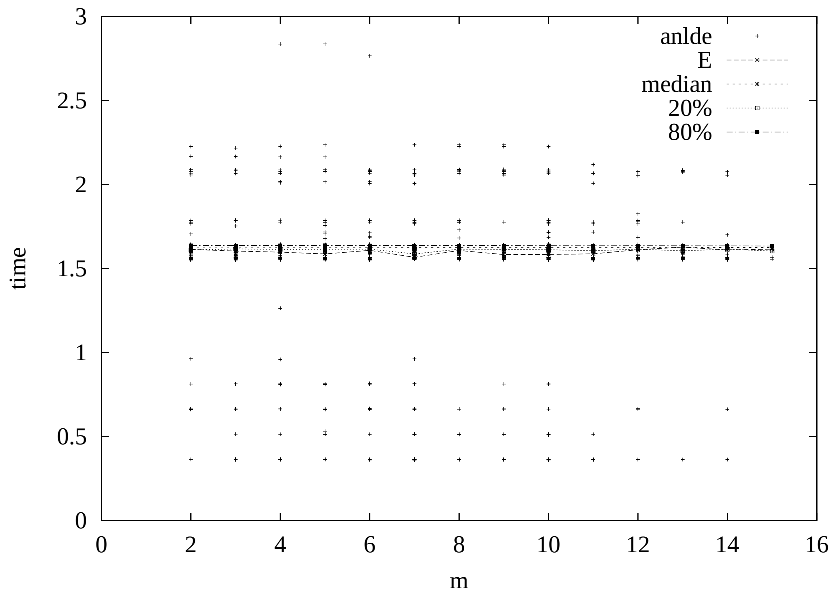


Рис. 10. Распределение времени решения по числу неизвестных m в системе

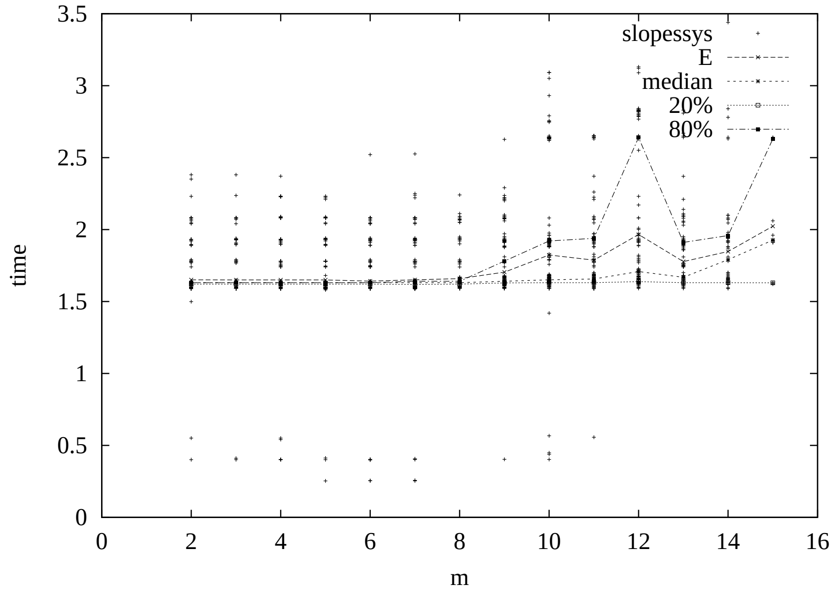


Рис. 11. Распределение времени решения по числу неизвестных m в системе

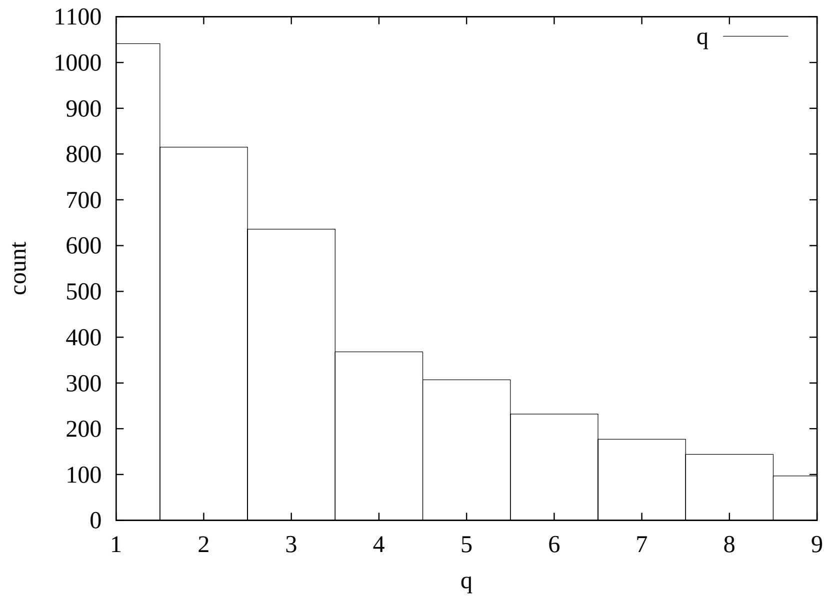
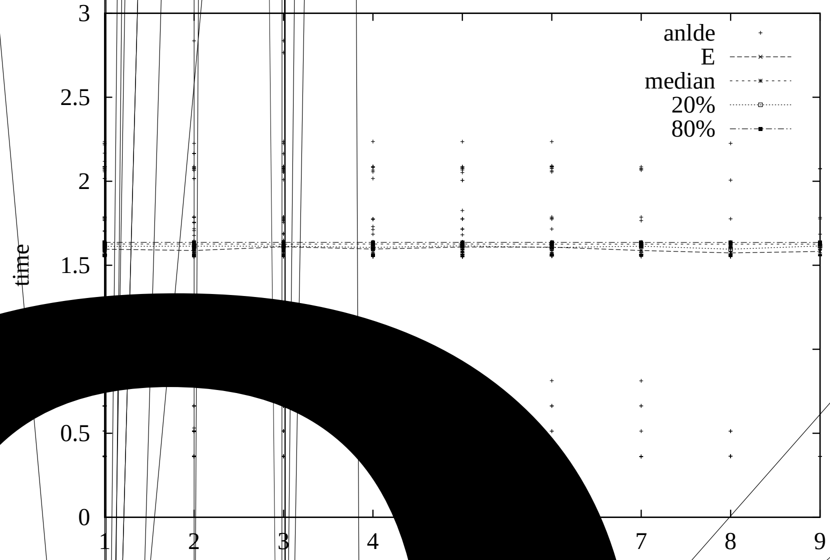
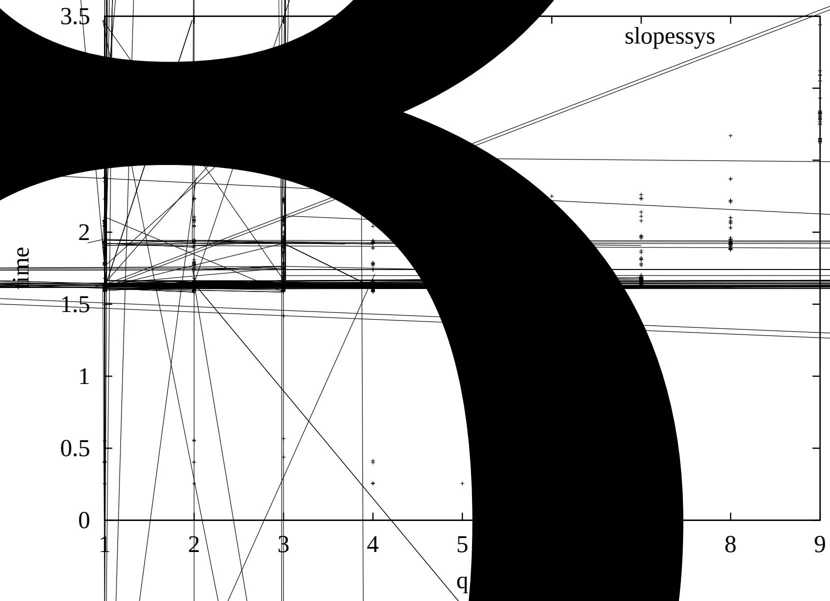


Рис. 12. Распределение систем по числу векторов базиса Гильберта



ис. 13. Распределение времени ре[illegible]у векторов базиса Г[illegible]



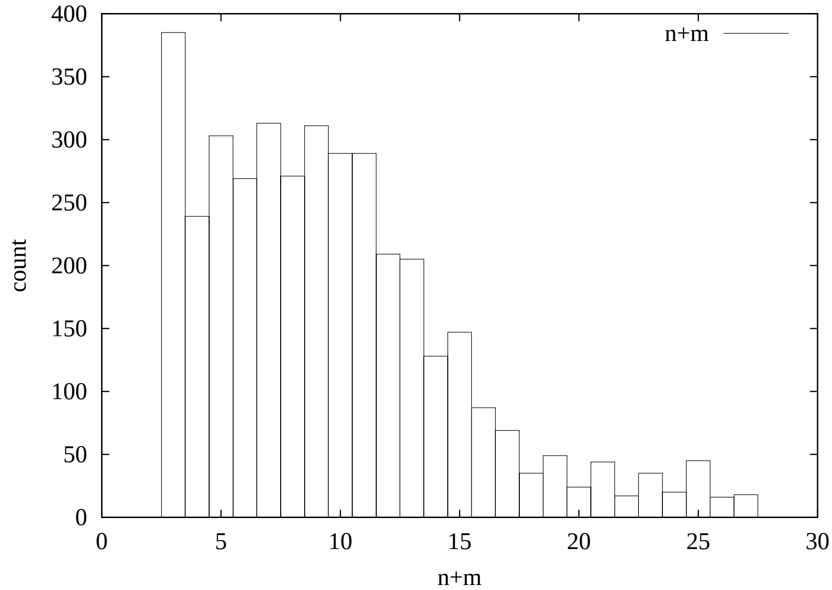


Рис. 15. Распределение систем по сумме чисел уравнений и неизвестных

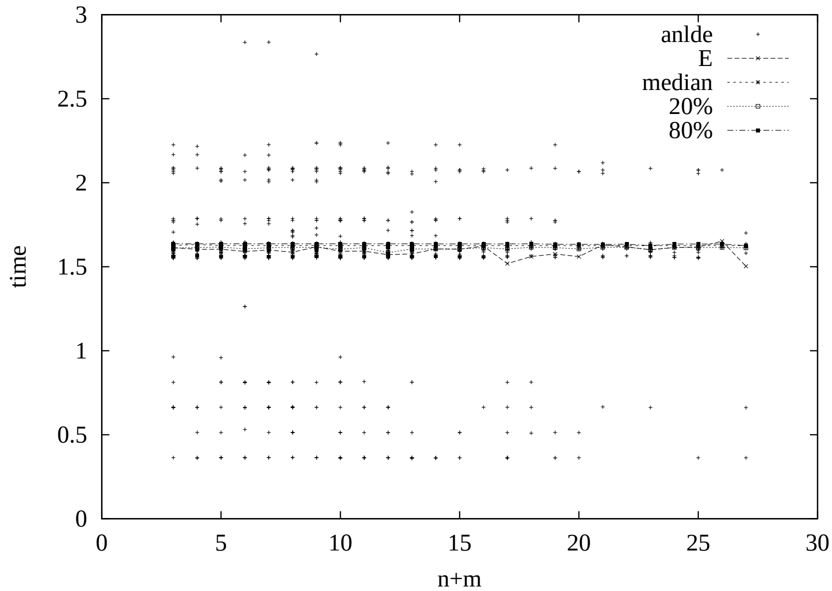


Рис. 16. Распределение времени решения по сумме чисел уравнений и неизвестных

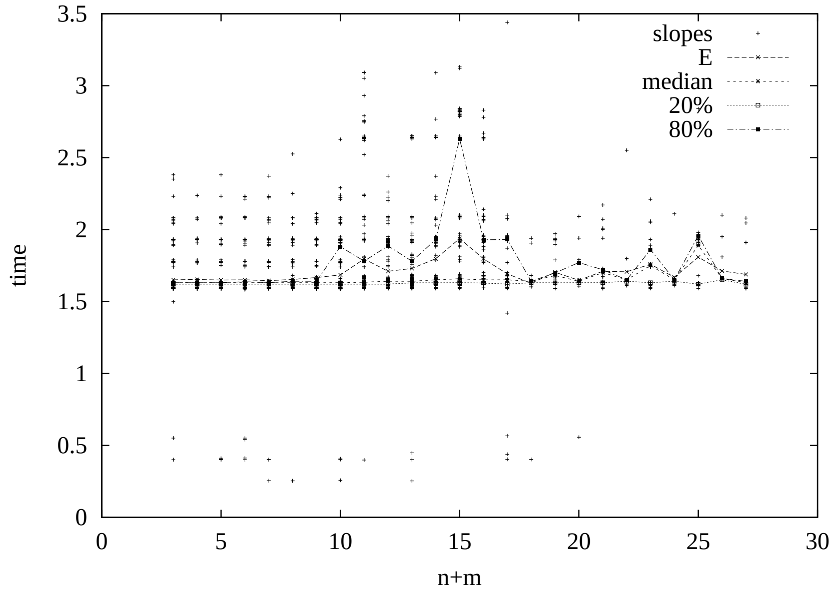


Рис. 17. Распределение времени решения по сумме чисел уравнений и неизвестных

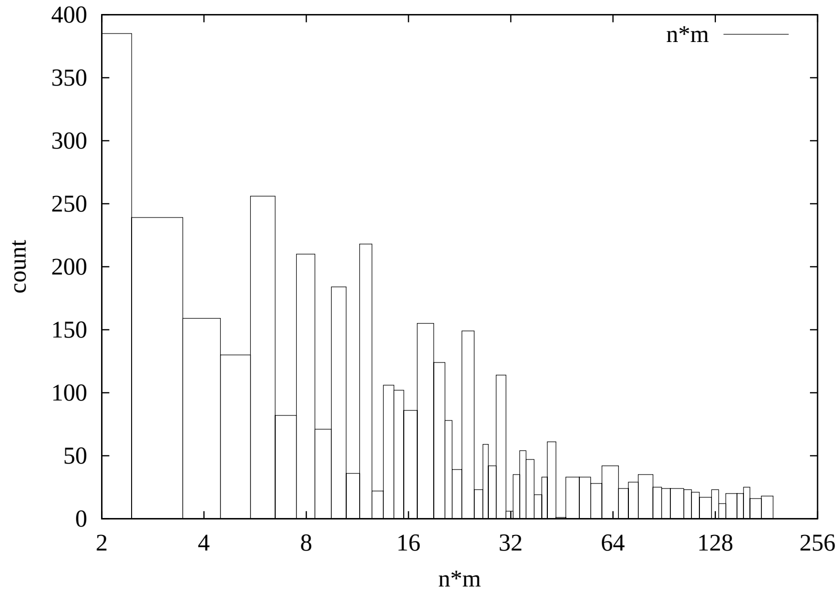


Рис. 18. Распределение систем по произведению чисел уравнений и неизвестных

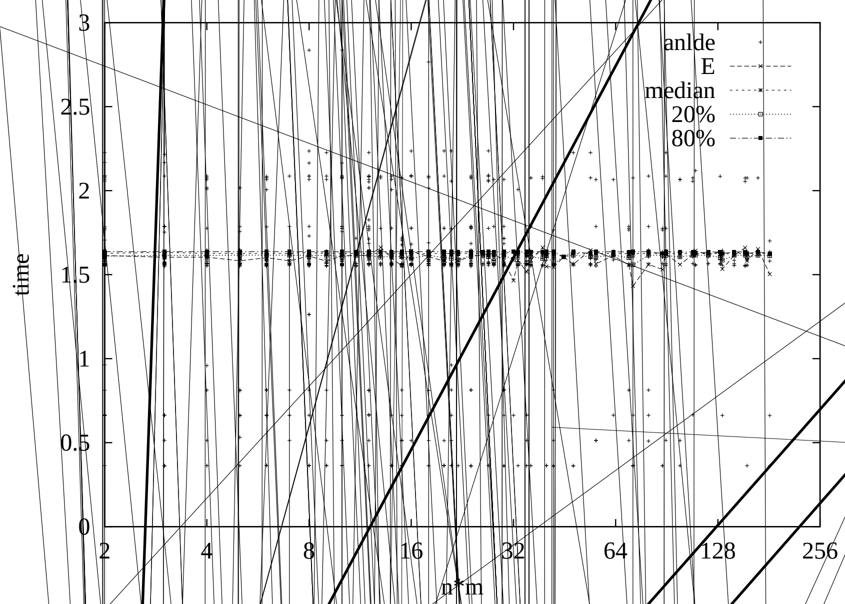
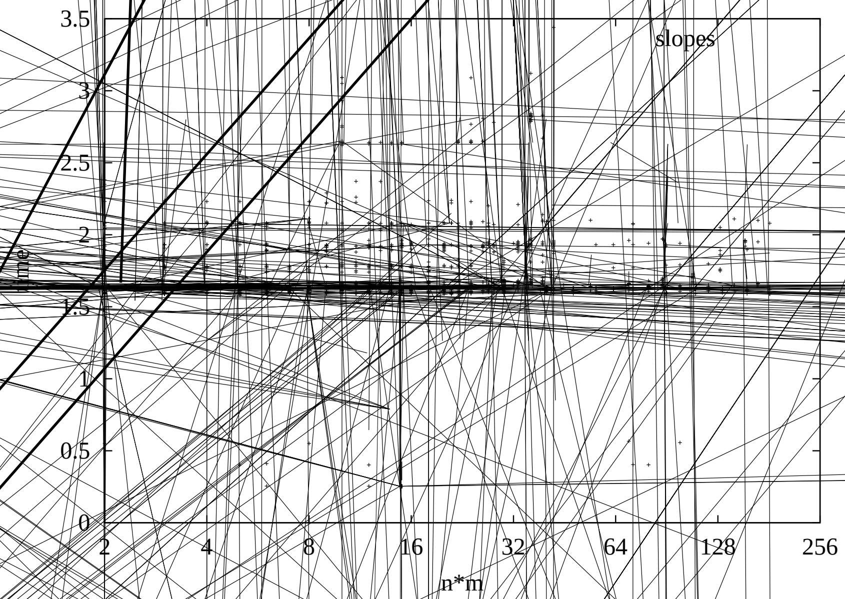


Рис. 19. Распределение времени решения по произведению чисел уравнений и неизвестных



Заключение

В выпускной работе получены следующие результаты.

1. разработан и реализован прототип программной системы для тестирования и экспериментального анализа,

2. разработаны и реализованы алгоритмы генерации специальных классов систем АНЛДУ,

3. разработан алгоритм построения полного класса систем АНЛДУ,

4. проведено тестирование алгоритма нахождения базиса Гильберта систем АНЛДУ. Данное тестирование не обнаружило ошибок в исследуемом алгоритме.

5. проведен экспериментальный анализ и сравнение алгоритмов нахождения базиса Гильберта систем АНЛДУ. В ходе сравнения выяснилось, что алгоритм, предложенный португальскими математиками, по эффективности уступает алгоритму, предложенному Д.Ж.Корзуном.

Полученные результаты требуют дальнейшего развития. Нами предлагается продолжить эти исследования. Наиболее интересными направлениями здесь представляется следующие.

— расширение классов тестовых систем за счет обобщения предложенных алгоритмов генерации и разработки новых.

— развитие других методов тестирования, отличных от основных на сравнении базиса Гильберта. Например, проверка только на частные решения.

— экспериментальное исследование синтаксического алгоритма anlde на различных диапазонах размерностей тестовых систем с целью получить детальную карту его эффективности.

— привлечение различных альтернативных алгоритмов решения систем НЛДУ для проведения обширного сравнительного анализа.

Библиографический список использованной литературы

1. Богоявленский Ю. А., Корзун Д. Ж. *Общий вид решения системы линейных диофантовых уравнений, ассоциированной с контекстно-свободной грамматикой* // Труды Петрозаводского государственного университета. Сер. “Прикладная математика и информатика”. Вып. 6. — Петрозаводск: Изд-во ПетрГУ, 1997. — С. 79–94.
2. Корзун Д. Ж. *Решение одного класса линейных диофантовых уравнений в неотрицательных целых числах методами теории формальных языков* // Труды Петрозаводского государственного университета. Сер. “Прикладная математика и информатика”. Вып. 7. — Петрозаводск: Изд-во ПетрГУ, 1999. — С. 93–116.
3. Корзун Д. Ж. *Об одной взаимосвязи формальных грамматик и систем линейных диофантовых уравнений* // Вестник молодых ученых, 2000. № 3. — СПб: Изд-во СПбГТУ, 2000. — С. 50–56.
4. Корзун Д. Ж. *Grammar-Based Algorithms for Solving Certain Classes of Nonnegative Linear Diophantine Systems* // Труды международного семинара Finnish Data Processing Week at the University of Petrozavodsk (FDPW’2000): Advances in Methods of Modern Information Technology. Vol. 3. — Петрозаводск: Изд-во ПетрГУ, 2001. — С. 52–67.
5. Корзун Д. Ж. *Синтаксические алгоритмы решения неотрицательных линейных диофантовых уравнений и их приложение к моделированию структуры нагрузки канала Интернет.* Дисс. на соиск. канд. физ.-мат. наук. Петрозаводск, ПетрГУ, 2002. 185 с.
6. Кулаков К. А. *Технология тестирования, экспериментального анализа и сравнения алгоритмов решения неотрицательных линейных диофантовых систем* // Материалы 54-й научной студенческой конференции ПетрГУ. Секция «Информатика». — Петрозаводск: Изд-во ПетрГУ, 2002. — С. 173–174.
7. Нефедов В. Н., Осипова В. А. *Курс дискретной математики.* — М.: Изд-во МАИ, 1992. — 264 с.
8. Filgueiras M., Tomás A. P. *Solving Linear Constraints on Finite Domains through Parsing* // In P. Barahona, L. Moniz Pereira, A. Porto (eds.), Proceedings of the 5th Por-

tuguese Conference on Artificial Intelligence, Springer-Verlag, 1991. LNAI 541. — PP. 1–16.

9. Filgueiras M., Tomás A. P. *A fast method for finding the basis of nonnegative solutions to a linear Diophantine equation* // Journal of Symbolic Computation. — 1995. Vol. 19. — PP. 507–526.

10. Tomás A. P., Filgueiras M. *An algorithm for solving systems of linear Diophantine equations in naturals* // In E. Costa, A. Cardoso (eds.), Progress in Artificial Intelligence (EPIA'97). Springer-Verlag, 1997. LNAI 1323. — PP. 73–84.