

Эффективные алгоритмы и программные средства реализации линейных диофантовых моделей сетей ЭВМ

Кирилл Александрович Кулаков

Математика и информатика: тенденции и перспективы

Петрозаводск — 2011

Актуальность

- Задача восстановления соединений в сети MPLS (алгоритм SLSP)
- Линейная диофантова модель
(система одНЛДУ $Ax = \mathbb{O}$, $A \in \mathbb{Z}$, $x \in \mathbb{Z}_+$)
- Эффективные алгоритмы нахождения базиса Гильберта (*НБГ*) и генерации систем с известным базисом Гильберта (*ИБГ систем*)
(сложность, частные классы, системы одАНЛДУ, метод исключения уравнений и неизвестных)
- Комплекс программ (реализации алгоритмов, автоматизированное тестирование и экспериментальное исследование, доступ через Интернет)
- Экспериментальное исследование (на системах одАНЛДУ, структурно соответствующих реальным сетям MPLS, M-системы)

Используемый частный класс систем одАНЛДУ

$$\text{одАНЛДУ: } \sum_{i \in I_k} x_i = \sum_{i=1}^m a_{ki} x_i, \quad k = 1, 2, \dots, n;$$

$$a_{ki} \in \mathbb{Z}_+; \quad \bigcup_{k=0}^n I_k = \mathbb{N}_m; \quad I_k \cap I_j = \emptyset \quad k \neq j; \quad I_k \neq \emptyset \quad \forall k \neq 0.$$

Любое x_i встречается в левой части не более 1 раза.

$$\text{Базис Гильберта: } \mathcal{H} = \{h^{(1)}, \dots, h^{(q)}\}, \quad x = \sum_{s=1}^q \alpha_s h^{(s)}, \quad \alpha_s \in \mathbb{Z}_+.$$

$$\text{1одАНЛДУ: } \sum_{i \in I_1} x_i = \sum_{j \in I_0} a_j x_j.$$

$$\text{содАНЛДУ: } \begin{cases} \sum_{i \in I_k \setminus J_k} x_i = \sum_{i \in J_k \setminus I_k} x_i, & k = 1, \dots, n \\ \sum_{i \in Z} x_i = 0, & Z \subseteq \mathbb{N}_m \\ \sum_{i \in F} x_i = \sum_{i \in F} x_i, & F \subseteq \mathbb{N}_m. \end{cases}$$

Преобразование к трапециевидной форме

$$k\text{-е уравнение: } \sum_{i \in L_k} x_i + \sum_{i \in I_k \setminus L_k} x_i = \sum_{i \notin I_k} a_{ki} x_i, \quad k = 1, \dots, n.$$

1. Строим $L_k \neq \emptyset$ для уравнения системы $S^{(k)}$.

$$2. S^{(k)} = \begin{cases} \sum_{i \in L_k} x_i + \sum_{i \in I_k \setminus L_k} x_i = \sum_{i \notin I_k} a_{ki} x_i, \\ S^{(k+1)}. \end{cases}$$

3. $S^{(k)} \rightarrow S^{(k+1)}$

Трапециевидная форма системы одАНЛДУ:

$$\begin{cases} \sum_{i \in L_k} x_i + \sum_{i \in I_k \setminus L_k} x_i = \sum_{i \notin I_k} a_{ki} x_i, & k = 1, 2, \dots, r-1, \\ S^{(r)}. \end{cases} \quad (1)$$

Теорема(1.2). Система $S^{(r)}$ равносильна либо 1одАНЛДУ (при $r = n$), либо системе содАНЛДУ (при $r < n$).

Обратная подстановка базиса Гильберта

1. Подставляем $\mathcal{H}^{(k+1)}$ в первое уравнение $S^{(k)}$ получая

$$\sum_{i \in L_k} x_i + \sum_{i \in I_k \setminus L_k} \sum_{h \in \mathcal{H}^{(k+1)}} h_i \alpha_h = \sum_{i \notin I_k} a_{ki} \sum_{h \in \mathcal{H}^{(k+1)}} h_i \alpha_h. \quad (2)$$

2. Находим базис $\tilde{\mathcal{H}}$ решая (2) относительно $(x_i)_{i \in L_k}$ и $(\alpha_h)_{h \in \mathcal{H}^{(k+1)}}$

3. Вычисляем $\mathcal{H}^{(k)} = \min \left\{ \sum_{i \in L_k} x_i \mathbf{e}_i + \sum_{h \in \mathcal{H}^{(k+1)}} \alpha_h h \mid (x, \alpha) \in \tilde{\mathcal{H}} \right\}$

4. $S^{(k)} \rightarrow S^{(k-1)}$

Уравнение (2): $\sum_{i \in L_k} x_i = \sum_{h \in \mathcal{H}^{(k+1)}} d_{kh} \alpha_h$, где $d_{kh} = \sum_{i \notin I_k} a_{ki} h_i - \sum_{i \in I_k \setminus L_k} h_i$.

Теорема(1.3). Пусть система одАНЛДУ преобразована к (1) и

$R_k = \bigcup_{j=0}^{k-1} I_j$. Тогда для любых $k = 1, 2, \dots, r$ и $h \in \mathcal{H}^{(k)}$ выполняется:

(а) $\sum_{i \in R_k} h_i \leq 1$; (б) если $k \neq 1$, то $d_{k-1,h} \geq -1$.

Алгоритм SLSP и линейные диофантовы модели

■ Алгоритм SLSP

- Сеть MPLS представлена в виде графа $\Omega(N, L)$
- Разбиение маршрута r на перекрывающиеся домены

1 Построение множества CY простых циклов из $\Omega(N, L)$

2 Для каждого домена s отобрать простые циклы $CY_{r,s} \subset CY$ покрывающие домен — маршруты-кандидаты

3 Выбор оптимального маршрута из $CY_{r,s}$ (задача ЦЛП)

■ Линейные диофантовы модели

- Система одАНЛДУ $E(I^{n,m})x = Ax$
- Уравнения — маршрутизаторы сети
- Неизвестные — линии связи
- Коэффициенты матрицы A — характеристики пересылки пакетов
- Базис Гильберта — маршруты-кандидаты

Модель топологии

- Система содАНЛДУ

$$\sum_{i \in I_k} x_i = \sum_{i \in J_k} x_i, \quad k = 1, \dots, n$$
- Матрица инцидентности орграфа $E(I^{n,m}) - E(J^{n,m})$
- Каждой линии связи соответствуют 2 дуги (направления)
- Базисное решение — контур орграфа
- содержательно эквивалентна SLSP-модели

Модель с фиксированным соединением

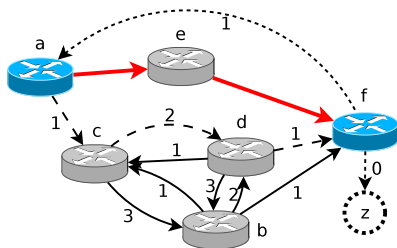
- Построение маршрута для домена $s = (u, v)$
- Удаление дуг входящих в u и исходящих из v
- Удаление внутренних дуг и вершин маршрута
- Добавление фиктивной дуги (v, u)
- Поиск контуров, содержащих дугу (v, u)

Модель с характеристиками линий связи

- Характеристики линий связи (загруженность, приоритет, число промежуточных устройств)
- Добавление фиктивных вершины z и дуги (v, z)
- Вес дуг орграфа $A = (a_{wi})_{w \in N, i \in L}$, а дуг (v, u) и (v, z) равен 1 и 0

■ Система одАНЛДУ
$$\begin{cases} x_{vu} + x_{vz} = \sum_{i \notin I_v} a_{vi} x_i, \\ \sum_{i \in I_w} x_i = \sum_{i \notin I_w} a_{wi} x_i, \quad w \in N \setminus \{v\}. \end{cases}$$

- Решение — маршрут из u в v



$$\begin{cases} a: & x_{ac} = x_{fa} \\ b: & x_{bc} + x_{bd} + x_{bf} = 3x_{cb} + 3x_{db} \\ c: & x_{cb} + x_{cd} = x_{ac} + x_{bc} + x_{dc} \\ d: & x_{db} + x_{dc} + x_{df} = 2x_{bd} + 2x_{cd} \\ f: & x_{fa} + x_{fz} = x_{bf} + x_{df} \end{cases}$$

Базис Гильберта содержит 32 решения,
для 7 из которых $x_{fa} = 1$.

Кумулятивная характеристика маршрута

Маршрут из вершин: $u = w_0, w_1, w_2, \dots, w_l = v$.

Кумулятивная характеристика:

$$C_u^u = 1, \quad C_u^{w_k} = \sum_{i \notin I_{w_k}} a_{w_k i} P_i(C_u^{w_t}), \quad k = 1, \dots, l, \quad t < k$$

$P_i(C_u^{w_t}) = x_i$ — часть кумулятивной характеристики маршрута (u, w_t) , проходящего через дугу $i \in I_{w_t}$.

Влияние дуги i на маршрут \Rightarrow к C_u^w добавляется величина $a_{wi}x_i \Rightarrow$ принцип “слабого звена”

Вычисление кумулятивной характеристики в модели: $C_u^v = x_{vz} + 1$

Рациональный предварительный отбор кандидатов на шаге 2 алгоритма SLSP

Уменьшение размерности задачи ЦЛП на шаге 3.

Пример решения модельной системы в ПС Web-SynDic

Файл Правка Вид Журнал Зкладки Инструменты Справка

http://websyndic.cs.karelia.ru/process.servlet%3Bsessionid=E54B7945962688E89B13BE17C80A0E37

Process:

- ANLDE System
- Set of ANLDE Systems

Documentation:

- ANLDE Theory
- User Manual

Send Notes

Algorithms configuration

Log In

Nickname:

Password:

Log in Register

(you may also continue working as anonymous user, but your profile and limits information will not be saved)

Server Load

1 active users
0 registered users
0 solver tasks
0 generator tasks

1. Test ANLDE system:

$$\begin{aligned} Xac &= Xfa \\ Xbc + Xbd + Xbf &= 3 \times Xcb + 3 \times Xdb \\ Xcb + Xcd &= Xac + Xbc + Xdc \\ Xdb + Xdc + Xdf &= 2 \times Xbd + 2 \times Xcd \\ Xfa + Xfb &= Xbf + Xdf \end{aligned}$$

Number of solutions: 32

2. Waiting time:

- Solver spooler buffer: 00 min 00 sec

3. Performance metrics of the algorithms:

Algorithm name	Summary CPU work time(sec)	Summary real work time(sec)	Maximum process virtual size(Kb)	Solving result
Syntactic	0.004	0.475	1728	solved
Trans	< 0.001	0.132	3728	solved

4. Solving machine characteristics:

- CPU: Intel(R) Xeon(R) CPU E5420 @ 2.50GHz (2493.748 MHz)
- RAM: 2109984 kB
- Operating system: Linux 2.6.27.21-0.1-xen, amd64
- Environment: Java 1.6.0_11 (Sun Microsystems Inc.); Apache Tomcat/5.0
- Solver process nice value: 15 (-20 - highest priority, 19 - lowest priority)

5. Solutions of test ANLDE system:

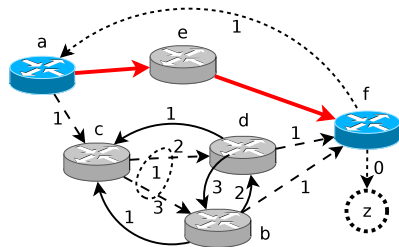
- Syntactic:

	Xac	Xfa	Xbc	Xbd	Xbf	Xcb	Xdb	Xcd	Xdc	Xdf	Xfz
$h_1(1)$	1	1	0	3	0	1	0	0	0	6	5
$h_1(2)$	1	1	0	2	1	1	0	0	0	4	4
$h_1(3)$	1	1	0	1	2	1	0	0	0	2	3
$h_1(4)$	1	1	0	0	6	0	2	1	0	0	5
$h_1(5)$	1	1	0	0	3	0	1	1	0	1	3
$h_1(6)$	1	1	0	0	3	1	0	0	0	0	2
$h_1(7)$	1	1	0	0	0	0	0	1	0	2	1
$h_1(8)$	0	0	1	0	5	0	2	1	0	0	5

Изображения: 7/7 Загружено: 14 KB Скорости: 3,40 KB/c Время: 4,131 Готово

Модель с множественной пересылкой

- Дополнение матрицы A столбцами, соответствующими множественной пересылке
- Маршруты, описываемые в предыдущей модели, и маршруты с множественной пересылкой
- Значение C_u^v зависит от порядка следования линий связи в маршруте



$$\left\{ \begin{array}{l} a : x_{ac} = x_{fa} \\ b : x_{bc} + x_{bd} + x_{bf} = 3x_{cb} + 3x_{db} + x_{(c;b,d)} \\ c : x_{cb} + x_{cd} + x_{(c;b,d)} = x_{ac} + x_{bc} + x_{dc} \\ d : x_{db} + x_{dc} + x_{df} = 2x_{bd} + 2x_{cd} + x_{(c;b,d)} \\ f : x_{fa} + x_{fz} = x_{bf} + x_{df} \end{array} \right.$$

Базис Гильберта содержит 48 решений,
для 10 из которых $x_{fa} = 1$.

Алгоритм НБГ TransSol

- I. Выполнить алгоритм преобразования к трапециевидной форме, получая последовательность систем $S^{(1)}, S^{(2)}, \dots, S^{(r)}$.
- II. Найти базис Гильберта $\mathcal{H}^{(r)}$ системы $S^{(r)}$. Для случая 1одАНЛДУ используется алгоритм НБГ 1одАНЛДУ, для случая системы содАНЛДУ — алгоритм НБГ системы содАНЛДУ.
- III. Выполнить алгоритм обратной подстановки базиса Гильберта для систем $S^{(r-1)}, \dots, S^{(1)}$, получая последовательность базисов $\mathcal{H}^{(r-1)}, \dots, \mathcal{H}^{(1)}$.

Сложность алгоритма $T = O(mnQ^2)$ и $V = O(mQ)$, $Q = \max_{k=1, \dots, r} \{|\tilde{\mathcal{H}}^{(k)}|\}$.

Теорема(2.3, частный случай). Пусть $\mathcal{H}_-^{(k+1)} = \emptyset$ для $k = r - 1, r - 2, \dots, 1$. Тогда сложность алгоритма подстановки базиса Гильберта в наихудшем случае составляет $T = O(mnq)$ и $V = O(mq)$.

Алгоритмы генерации ИБГ систем

■ Алгоритм JordanGen

Система одАНЛДУ: $(E(I^{n,p})|B)x = (E(I^{n,p})|B + \Delta)x$

Базис Гильберта: $\mathcal{H} = \{e_i\}_{i=1}^p$

■ Алгоритм GaussGen

Система одАНЛДУ: $(I|B)x = (D|B + \Delta)x$

Базис Гильберта: $\mathcal{H} = \{h^{(s)}\}_{s=1}^{m-n}$

■ Алгоритм ExtGaussGen

Система одАНЛДУ: $(E(I^{n,p})|B)x = (D|B + \Delta)x$

Базис Гильберта: $\mathcal{H} = \{h^{(s)}\}_{s=1}^q$

■ Алгоритм SymGen

Система одАНЛДУ: $E(I^{n,m})x = E(J^{n,m})x$

Базис Гильберта: $\mathcal{H} = \{h^{(s)} \in \{0, 1\}^m\}_{s=1}^q$

■ Алгоритм ExtSymGen

Система одАНЛДУ: $\left(\begin{array}{c|c} E(I^{b,p}) & \mathbb{O} \\ \hline \mathbb{O} & E(I^{n-b,m-p}) \end{array} \right) x = \left(\begin{array}{c|c} D & \Delta \\ \hline \mathbb{O} & E(J^{n-b,m-p}) \end{array} \right) x$

Базис Гильберта: $\mathcal{H} = \{h^{(s)}\}_{s=1}^q$

Сложность алгоритмов

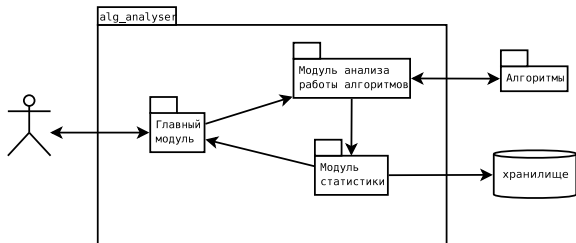
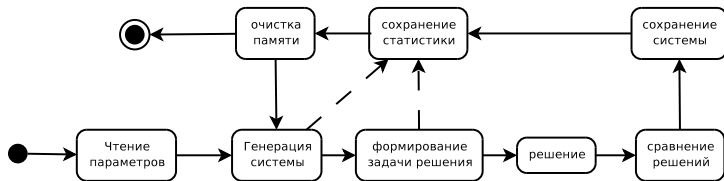
Алгоритм	Временная сложность	Емкостная сложность
TransSol	$O(mnQ^2)$	$O(mQ)$
TransSol (т.2.3)	$O(mnq)$	$O(mq)$
Syntactic (Д. Ж. Корзун)	$O(m^2n^2Q_s^3)$	$O(mn^2Q_s)$
JordanGen	$O(m^2)$	$O(mn)$
GaussGen	$O(mn(m - n))$	$O(m^2)$
ExtGaussGen	$O(mnq)$	$O(m(n + q))$
SymGen	$O(m(m + n)q)$	$O(mq)$
ExtSymGen	$O(m(m + n)q)$	$O(m(n + q))$

$$q = |\mathcal{H}|, Q = \max_{k=1, \dots, r} \{|\tilde{\mathcal{H}}^{(k)}|\},$$

Q_s — ограничение на размер промежуточных базисов в Syntactic

ПС alg_analyser

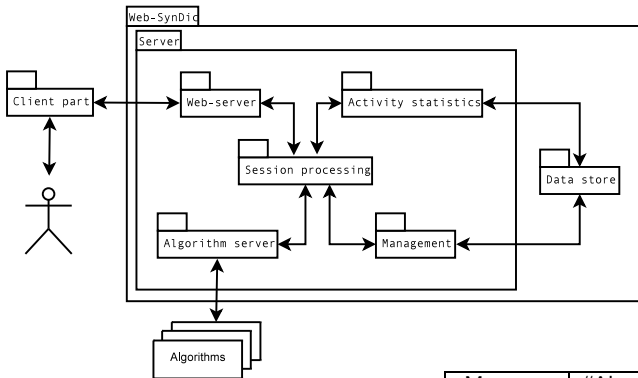
Автоматизированное тестирование, экспериментальный анализ и сравнение алгоритмов



Алгоритм	LOC	COM
JordanGen	588	633
GaussGen	631	637
ExtGaussGen	992	872
SymGen	631	637
TransSol	1144	822
alg_analyser	2660	1825

PC Web-SynDic (<http://websyndic.cs.karelia.ru>)

Удаленная работа с алгоритмами решения и генерации



Документация	363 с
Временные затраты	2243 ч/ч
Тестирование	334 теста

Метрика	"Algorithm server"	PC Web-SynDic
LOC	3805	11907
BLOC	491	1356
CLOC	1275	2757
NCSL	2207	5356
%(NCSL)	58	45

Экспериментальное исследование алгоритмов НБГ

- SlopesSys (А. Р. Томás и М. Filgueiras). Поиск решений с минимальным носителем и построение оставшихся
- Syntactic (Д. Ж. Корзун). Синтаксический метод.
- TransSol. Преобразование к трапециевидной форме.

Структура экспериментального исследования

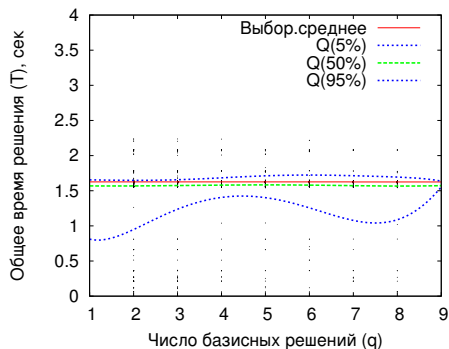
- **Часть I** (2002 г.). Тестирование алгоритма Syntactic
Алгоритмы JordanGen и GaussGen, 7 наборов,
более $1.5 \cdot 10^6$ систем. Ограничения: $n, m \leq 10^3$, $\|A\|_{l_\infty} \leq 10^5$
- **Часть II** (2002 г.). Сравнение алгоритмов SlopesSys и Syntactic
Алгоритмы JordanGen, GaussGen, 2 набора, 10^4 систем.
Ограничения: $n, m \leq 20$, $q \leq 9$, $\|A\|_{l_\infty} \leq 100$
- **Часть III** (2008 г.). Сравнение алгоритмов Syntactic и TransSol при использовании модели с множественной пересылкой
Алгоритмы JordanGen, GaussGen, ExtGaussGen,
6 серий по 3 набора, всего 75800 систем

Часть I. Тестирование алгоритма Syntactic

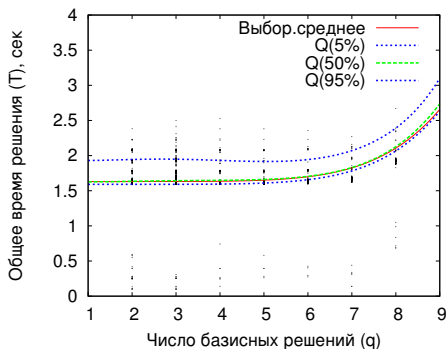
Алгоритм	Дата проведения	Кол-во тестов	Ограничение на m и n	Ограничение на $\ A\ _{l_\infty}$
GaussGen	26–28.02	410786	100	10^5
GaussGen	28.02	3	500	10^5
JordanGen	28.02–07.03	204308	1000	10^5
GaussGen	28.02–01.03	10	300	500
GaussGen	01–07.03	2837	1000	10^4
GaussGen	18.03–08.07	88640	1000	10^4
JordanGen	18.03–08.07	937234	1000	10^4
Итого	26.02–08.07	1643818	—	—

Решение получено для всех тестовых систем

Часть II. Сравнение алгоритмов SlopesSys и Syntactic



Алгоритм Syntactic



Алгоритм SlopesSys

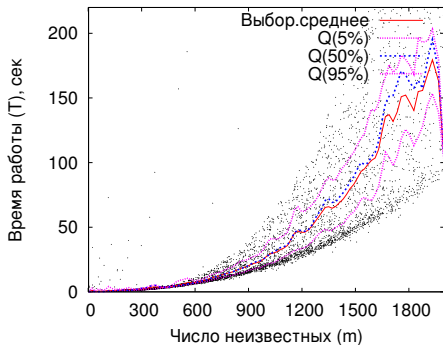
- Более половины систем не решено алгоритмом SlopesSys
- Алгоритм Syntactic обеспечивает лучшее гарантированное время решения
- Для алгоритма SlopesSys наиболее вероятная верхняя граница Q(95%) может существенно превышать типичное значение Q(50%)

Часть III. Сравнение алгоритмов Syntactic и TransSol

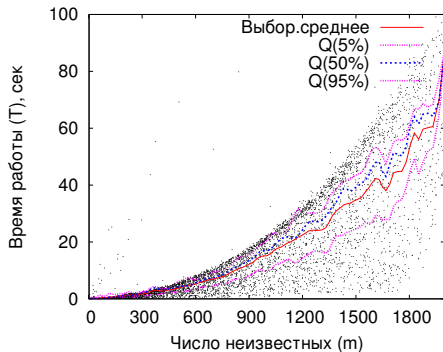
Серия	Ограничения	План	Решено
1. Малые размерности систем с малым базисом Гильберта	$n, m \leq 100,$ $q \leq 500, \ A\ \leq 100$	15000	13439
2. Большие размерности систем с малым базисом Гильберта	$n, m \leq 2000,$ $q \leq 500, \ A\ \leq 100$	15000	14597
3. Малые размерности систем с большим базисом Гильберта	$n, m \leq 100,$ $q \leq 10^4, \ A\ \leq 100$	15000	11452
4. Системы с большими коэффициентами	$n, m \leq 100,$ $q \leq 500, \ A\ \leq 10^4$	15000	13773
5. Большие системы с большим базисом Гильберта	$n, m \leq 2000,$ $q \leq 10^4, \ A\ \leq 10^4$	15000	14884
6. M-системы больших размеров	$n, m \leq 10^4,$ $q \leq 10^6, \ A\ _{l_\infty} \leq 10^4$	800	570

Для частного класса одАНЛДУ оба алгоритма за приемлемое время решают системы существенно больших размерностей, чем известные алгоритмы.

Часть III. Сравнение алгоритмов Syntactic и TransSol



Алгоритм Syntactic



Алгоритм TransSol

- Алгоритм TransSol эффективнее Syntactic ($\sim mn$)
- Часть систем (до 25%) не решена алгоритмом Syntactic
- Для алгоритма TransSol выборочные характеристики подвержены меньшим скачкам, чем для алгоритма Syntactic