Решетки и приведенные базисы
Лекция

Н.Ю. Золотых

2 ноября 2006

Содержание

1 Решетки 1

2 Ортогонализация Грама–Шмидта 2

3 Кратчайший вектор решетки 2

4 Алгоритм приведения Гаусса ($n = 2$) 3

5 Свойства кратчайших векторов 3

6 Последовательные минимумы решетки 4

7 Базисы Коркина–Золотарева 4

8 c–ортогональные базисы 4

9 c–приведенные базисы 5

10 c–приведенные по Ловасу базисы 6

11 Алгоритм Ловаса приведения базиса решетки 7

12 Пример работы алгоритма LLL-Reduction 10

13 Использование Arageli 11

14 Приложения 14

1 Решетки

Пусть $b_1, \ldots, b_n$ — некоторые линейно независимые векторы в m-мерном евклидовом пространстве. Рассмотрим *решетку*

$$\Lambda = \Lambda(b_1, \ldots, b_n) = \left\{ x : \sum_{j=1}^{n} \alpha_j b_j,\ \alpha_j \in \mathbf{Z}\ (j = 1, 2, \ldots, n) \right\}.$$

Векторы $b_1, \ldots, b_n$ называются ее *базисом*, а число n — *размерностью* (или *рангом*). *Матрицей Грама* называется матрица

$$\Gamma(b_1, \ldots, b_n) = \begin{pmatrix} (b_1, b_1) & (b_1, b_2) & \ldots & (b_1, b_n) \\ (b_2, b_1) & (b_2, b_2) & \ldots & (b_2, b_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ (b_n, b_1) & (b_n, b_2) & \ldots & (b_n, b_n) \end{pmatrix}.$$

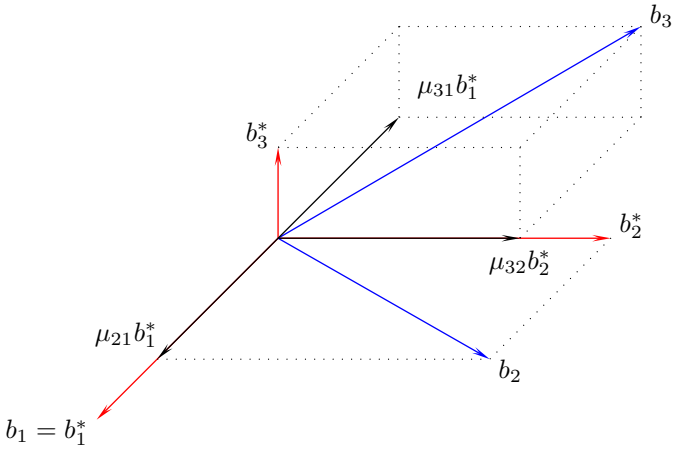

Рис. 1: Процесс ортогонализации Грама–Шмидта

Число

$$\det \Lambda = \sqrt{\det \Gamma(b_1, \ldots, b_n)}$$

называется *определителем* решетки. Определитель решетки равен объему *фундаментального параллелепипеда*, натянутого на векторы $b_1, \ldots, b_n$. Заметим, что если $m = n$, то $\det \Lambda = \det(b_1, \ldots, b_n)$, где $(b_1, \ldots, b_n)$ — матрица, составленная из координат векторов $b_1, \ldots, b_n$ в некотором ортонормированном базисе. Базис решетки не единственен, но легко видеть, что матрица перехода от одного базиса решетки к произвольному другому *унимодулярна*, т. е. ее определитель равен ± 1, поэтому $\det \Lambda$ не зависит от выбора базиса.

2 Ортогонализация Грама–Шмидта

Применим к векторам $b_1, \ldots, b_n$ *процесс ортогонализации Грама–Шмидта*:

$$b_k^* = b_k - \sum_{l=1}^{k-1} \mu_{kl} b_l^* \quad (k = 1, 2, \ldots, n), \tag{1}$$

где

$$\mu_{kl} = \frac{(b_k, b_l)}{(b_l, b_l)} \qquad \text{при } k > l. \tag{2}$$

Заметим, что b_k^* есть проекция вектора b_k на ортогональное дополнение к подпространству $L(b_1, \ldots, b_{k-1}) = L(b_1^*, \ldots, b_{k-1}^*)$. Отсюда, в частности, следует, что $b_1^*, \ldots, b_n^*$ составляют ортогональный базис подпространства $L(b_1^*, \ldots, b_n^*)$ (но не базис решетки Λ, так как векторы b_k^* могут ей не принадлежать).

Лемма 1 (Неравенство Адамара)**.** *Пусть $b_1, \ldots, b_n$ — некоторый базис решетки Λ, а $b_1^*, \ldots, b_n^*$ — векторы, полученные из него в результате ортогонализации Грама–Шмидта* (1), (2)*. Тогда*

$$\det \Lambda = \prod_{i=1}^{n} |b_i^*| \le \prod_{i=1}^{n} |b_i|.$$

3 Кратчайший вектор решетки

Ненулевой вектор решетки минимальной длины называется ее *кратчайшим* вектором.

Лемма 2. *Пусть $b_1, \ldots, b_n$ — некоторый базис решетки Λ, а $b_1^*, \ldots, b_n^*$ — векторы, полученные из него в результате ортогонализации Грама–Шмидта (1), (2). Тогда*

$$\ell \geq \min\{|b_1^*|, \ldots, |b_n^*|\},$$

где ℓ — длина кратчайшего вектора решетки Λ.

Доказательство. Представим кратчайший вектор x в виде

$$x = \sum_{i=1}^{n} \alpha_i b_i = \sum_{i=1}^{n} \lambda_i b_i^*, \qquad \text{где } \alpha_i \in \mathbf{Z},\ \lambda_i \in \mathbf{R}. \tag{3}$$

Пусть j — наибольший индекс, для которого $\alpha_j \neq 0$. Умножая все части равенства (3) скалярно на b_j^*, получаем $\alpha_j(b_j, b_j^*) = \lambda_j(b_j^*, b_j^*)$. Так как $(b_j, b_j^*) = (b_j^*, b_j^*)$, то $\alpha_j = \lambda_j \geq 1$. Поэтому

$$|x|^2 \geq \lambda_i^2 |b_j^*|^2 = \alpha_j^2 |b_j^*|^2 \geq |b_j^*|^2.$$

□

4 Алгоритм приведения Гаусса ($n = 2$)

Рассмотрим алгоритм Гаусса построения кратчайшего вектора двумерной решетки.

procedure GAUSS-REDUCTION (var b_1, b_2)
    repeat
        if $|b_1| > |b_2|$
            $b_1 \leftrightarrow b_2$;
        end
        найти такое q, что $b_2 - qb_1$ имеет минимальную длину;
        if $q = 0$
            return
        end
        $b_2 \leftarrow b_2 - qb_1$;
    end
end.

Очевидно, что на выходе алгоритма векторы b_1, b_2 составляют базис решетки. Простые геометрические соображения показывают, что при этом b_1 — кратчайший вектор решетки, а b_2 составляет с b_1 угол θ, такой, что

$$\frac{\pi}{2} - \frac{\pi}{6} \leq \theta \leq \frac{\pi}{2} + \frac{\pi}{6}.$$

5 Свойства кратчайших векторов

Замечание 3. Задача отыскания кратчайшего вектора решетки является NP-трудной [7][1].

Теорема 4 (Ш. Эрмит, см. [2])**.** *Пусть ℓ — длина кратчайшего вектора решетки Λ. Тогда*

$$\ell \leq \gamma_n^{1/2} (\det \Lambda)^{1/n}, \tag{4}$$

где γ_n — некоторая величина, зависящая только от n.

Замечание 5. Значение γ_n, при котором оценка (4) достигается, называется *константой Эрмита*. Известно, что для нее справедлива асимптотическая оценка

$$\frac{1}{2\pi e} + o(1) \leq \frac{\gamma_n}{n} \leq \frac{1}{\pi e} + o(1) \qquad \text{при } n \to \infty$$

[1] Я не проверял эту ссылку. Хорошо известно, что задача отыскания кратчайшего вектора решетки для норм L_1, L_∞ являются NP-трудными. Для нормы L_2 (о которой в тексте идет речь) проблема долгое время была открытой.

и точное неравенство

$$\gamma_n \le \left(\frac{4}{3}\right)^{(n-1)/2}, \tag{5}$$

поэтому

$$\ell \le \left(\frac{4}{3}\right)^{(n-1)/4} (\det \Lambda)^{1/n}. \tag{6}$$

Точное значение γ_n известно только для небольших n:

$$\gamma_1 = 1, \quad \gamma_2^2 = \frac{4}{3}, \quad \gamma_3^3 = 2, \quad \gamma_4^4 = 4, \quad \gamma_5^5 = 8, \quad \gamma_6^6 = \frac{64}{3}, \quad \gamma_7^7 = 64, \quad \gamma_8^8 = 256.$$

6 Последовательные минимумы решетки

Пусть $b_1, \dots, b_n$ — векторы из Λ, удовлетворяющие следующему свойству. Среди всех векторов, принадлежащих Λ, но не принадлежащих $L(b_1, \dots, b_{k-1})$, вектор b_k имеет минимальную длину $(k = 1, \dots, n)$ (считаем, что $L(\emptyset) = \{0\}$). Очевидно, что тогда векторы $b_1, \dots, b_n$ составляют базис решетки Λ. Этот базис называется *редуцированным по Минковскому*, при этом b_k называется *k-м последовательным минимумом* решетки.

Теорема 6 (см. [2]). *Для редуцированного по Минковскому базиса $b_1, \dots, b_n$ справедливо*

$$|b_1| \cdot \ldots \cdot |b_n| \le \gamma_n^{n/2} \det \Lambda, \tag{7}$$

где γ_n — константа Эрмита. В частности, из (5) следует

$$|b_1| \cdot \ldots \cdot |b_n| \le \left(\frac{4}{3}\right)^{n(n-1)/4} \det \Lambda, \tag{8}$$

Замечание 7. Неравенство (7) указывает на то, что приведенный по Минковскому базис близок к ортогональному.

7 Базисы Коркина–Золотарева

Пусть векторы $b_1^*, \dots, b_n^*$ получены из базиса $b_1, \dots, b_n$ решетки Λ в результате ортогонализации (1), (2). Базис $b_1, \dots, b_n$ решетки Λ называется приведенным по Коркину–Золотареву[2], если

b_1 — кратчайший вектор решетки;

b_k — ненулевой вектор решетки, такой, что проекция b_k на орт. доп. к $L(b_1, \dots, b_{k-1})$ минимальна $(k = 2, \dots, n)$;

$$|\mu_{kl}| \le \frac{1}{2} \qquad (1 \le l < k \le n).$$

8 c–ортогональные базисы

Неравенство (8) обосновывает введение нового понятия. Базис $b_1, \dots, b_n$ называется *c-ортогональным*, или *c-редуцированным по Ленстре*, (первый термин взят из [5]), если

$$|b_1| \cdot \ldots \cdot |b_n| \le c^{n(n-1)/4} \det \Lambda.$$

Из неравенства Адамара следует, что при $c < 1$ c-ортогонального базиса не существует, а при $c = 1$ это понятие превращается в ортогональность. Из теоремы 6 следует, что для любой решетки существует c-ортогональный базис при $c \ge 4/3$. Первый полиномиальный при фиксированном n алгоритм нахождения c-ортогонального базиса для $c > 4/3$ предложил Х. Ленстра [9], см. также [5, § 5.3 п. 2]. Первый полиномиальный алгоритм построения c-ортогонального базиса предложил Л. Ловас (см. ниже).

[2] Александр Николаевич Коркин (1837–1908) и Егор Иванович Золотарёв (1847–1878) — русские математики.

9 c–приведенные базисы

Теперь введем понятие c-приведенного базиса. Пусть векторы $b_1^*, \ldots, b_n^*$ получены из базиса $b_1, \ldots, b_n$ решетки Λ в результате ортогонализации (1), (2). Базис $b_1, \ldots, b_n$ называется *c-приведенным* (по Зигелю) (термин взят из [5]), если

$$|\mu_{kl}| \le \frac{1}{2} \qquad (1 \le l < k \le n); \tag{9}$$

$$|b_{k-1}^*|^2 \le c \cdot |b_k^*|^2 \qquad (k = 2, 3, \ldots, n). \tag{10}$$

Далее мы увидим, что c-приведенный базис является c-ортогональным. Обратное, вообще говоря, не верно. Например, базис $b_1 = (1,1)^\top$, $b_2 = (0,1)^\top$ при $c = 2$ является c-ортогональным, но не является c-приведенным. Действительно, для него $\mu_{21} = 1/2$, $b_1^* = (1,1)^\top$, $b_2^* = (-1/2, 1/2)^\top$, $|b_1|^2 = 2$, $|b_2|^2 = 1$, $|b_1^*|^2 = 2$, $|b_2^*|^2 = 1/2$. Неравенство $|b_1|^2 \cdot |b_2|^2 \le 2 \cdot |b_1^*|^2 \cdot |b_2^*|^2$ выполнено, а $|b_1^*|^2 \le 2 \cdot |b_2^*|^2$ — нет.

Известно [2], что при $c \ge 4/3$ для любой решетки c-приведенный базис существует. Его существование при $c > 4/3$ будет следовать также из приведенного далее алгоритма построения более специального базиса.

Сначала приведем некоторые неформальные соображения, указывающие на то, что c-приведенный базис состоит из коротких, почти ортогональных друг к другу векторов.

Условие (9) означает, что длина проекции вектора b_k на вектор b_l^* не должна быть слишком большой по сравнению с вектором b_l^*. Заметим, что это условие будет выполнено, если b_k почти ортогонален b_l^* или если длина вектора b_k мала по сравнению с длиной вектора b_l^*. Таким образом, условия (9) могут не означать, что базис состоит из почти ортогональных векторов. Например, возможна ситуация, когда b_1 — очень длинный вектор, b_2 мал по сравнению с b_1, b_3 мал по сравнению с b_2 и т. д. Однако реализоваться этой ситуации препятствуют условия (10), означающие, что длины векторов в c-приведенном базисе «почти не уменьшаются» (длина вектора b_k^* не слишком мала по сравнению с длиной вектора b_{k-1}^*).

Из следующих теорем следует, что c-приведенный базис является c-ортогональным, а вектор b_1 близок к кратчайшему вектору решетки.

Теорема 8. *Пусть $b_1, \ldots, b_n$ — c-приведенный базис решетки Λ, $c \ge 3/4$, тогда*

$$\det \Lambda \le |b_1| \cdot \ldots \cdot |b_n| \le c^{n(n-1)/4} \det \Lambda. \tag{11}$$

$$|b_1| \le c^{(n-1)/4} (\det \Lambda)^{1/n}. \tag{12}$$

(Ср. с (4), (6) и (7), (8) соответственно).

Доказательство. Первое неравенство в (11) — это неравенство Адамара. Докажем второе неравенство. Пусть $b_1^*, \ldots, b_n^*$ — векторы, полученные из $b_1, \ldots, b_n$ в результате ортогонализации Грама–Шмидта (1), (2). Из (10) следует, что при $l \le k$

$$|b_l^*|^2 \le c^{k-l} |b_k^*|^2. \tag{13}$$

Поэтому, ввиду $|\mu_{kl}| \le 1/2$ и $c \ge 3/4$, имеем

$$|b_k|^2 = |b_k^*|^2 + \sum_{l=1}^{k-1} \mu_{kl}^2 |b_l^*|^2 \le |b_k^*|^2 \cdot \left(1 + \frac{1}{4} \sum_{l=1}^{k-1} c^l\right) = |b_k^*|^2 \cdot \left(\frac{3}{4} + \frac{c^k - 1}{4(c-1)}\right) \le \frac{3}{4} \cdot c^k \le |b_k^*|^2 \cdot c^{k-1}. \tag{14}$$

Следовательно,

$$\prod_{k=1}^{n} |b_k|^2 \le 1 \cdot c \cdot \ldots \cdot c^{n-1} \cdot (\det \Lambda)^2 = c^{n(n-1)/2} \cdot (\det \Lambda)^2.$$

Теперь докажем (12). Из (13) и (14) следует $|b_l| \le c^{(k-1)/2} |b_k^*|$ при $1 \le l < k \le n$. В частности, $|b_1|^2 \le |b_1^*|^2$, $|b_1|^2 \le c|b_2^*|^2$, …, $|b_1|^2 \le c^{n-1} |b_n^*|^2$. Перемножив эти неравенства, получим

$$|b_1|^{2n} \le 1 \cdot c \cdot \ldots \cdot c^{n-1} \cdot \prod_{i=1}^{n} |b_i^*|^2 = c^{(n-1)/2} (\det \Lambda)^2,$$

откуда вытекает (12). □

Замечание 9. Из (11) следует, что c-приведенный базис является c-ортогональным. Обратное, вообще говоря, не верно.

Замечание 10. Сравнение формул (11) и (12) с (4), (6) и (7), (8) соответственно показывает, что c-приведенный базис обладает некоторыми важными свойствами, которыми удовлетворяет базис, приведенный по Минковскому.

Теорема 11. *Пусть $b_1, \ldots, b_n$ — c-приведенный базис решетки Λ, ℓ — длина кратчайшего вектора. Тогда*

$$|b_1| \leq c^{(n-1)/2}\ell. \tag{15}$$

Доказательство. По лемме 2

$$\ell \geq \min\{|b_1^*|, \ldots, |b_n^*|\},$$

но $|b_1|^2 = |b_1^*|^2 \leq 2|b_2^*|^2 \leq \ldots \leq 2^{n-1}|b_n^*|^2$, поэтому $\min\left\{|b_1^*|^2, \ldots, |b_n^*|^2\right\} \geq 2^{1-n}|b_1|^2$, откуда получаем требуемое. □

Замечание 12. Неравенство (15) показывает, что b_1 — один из самых коротких векторов решетки. Заметим, что при доказательстве этой теоремы мы не пользовались свойством (9).

Упражнение 13. Доказать, что если векторы $x_1, \ldots, x_t$ из Λ линейно независимы, то $|b_j| \leq 2^{(n-1)/2} \max\{|x_1|, \ldots, |x_t|\}$ при $1 \leq j \leq t$.

Полиномиальный алгоритм построения c-приведенного базиса при $c > 4/3$ описан в [4], см. [5, § 5.4 п. 1].

10 c–приведенные по Ловасу базисы

Пусть векторы $b_1^*, \ldots, b_n^*$ получены из базиса $b_1, \ldots, b_n$ решетки Λ в результате ортогонализации (1), (2). Базис $b_1, \ldots, b_n$ называется *c-приведенным по Ловасу*, $c \geq 4/3$, если

$$|\mu_{kl}| \leq \frac{1}{2} \qquad (1 \leq l < k \leq n); \tag{16}$$

$$|b_k^* + \mu_{k,k-1}b_{k-1}^*|^2 \geq y\,|b_{k-1}^*|^2, \qquad y = \frac{4+c}{4c} \qquad (k = 2, 3, \ldots, n). \tag{17}$$

Таким образом, определение c-приведенного по Ловасу базиса получается из определения c-приведенного базиса заменой условия (10) условием (17). Далее мы увидим, что (17) сильнее (10). В литературе такие базисы часто (особенно при $c = 2$, $y = 3/4$) называют *LLL-приведенными* по первым буквам авторов работы [10], где впервые они появились. Заметим, что если $c > 4/3$, то $1/4 < y < 1$.

Так как $b_k^* + \mu_{k,k-1}b_{k-1}^*$ есть проекция вектора b_k на ортогональное дополнение к $L(b_1, \ldots, b_{k-2})$, то (17) означает, что эта проекция не слишком мала по сравнению с проекцией b_{k-1} на это ортогональное дополнение. Условие (17) эквивалентно

$$|b_k^*|^2 \geq \left(y - \mu_{k,k-1}^2\right)|b_{k-1}^*|^2 \qquad (k = 2, 3, \ldots, n). \tag{18}$$

Заметим, что из (16) и (18) легко получить неравенство (10):

$$|b_k|^2 \geq |b_k^*|^2 \geq \left(y - \mu_{k,k-1}^2\right)|b_{k-1}^*|^2 \geq \frac{1}{c}\,|b_{k-1}^*|^2.$$

Таким образом, базис, c-приведенный по Ловасу, является c-приведенным. Обратное, вообще говоря, не верно. Например, базис $b_1 = b_1^* = (4, 0)^\top$, $b_2 = b_2^* = (0, 3)^\top$ (он ортогональный) для $c = 2$ является c-приведенным, но не является c-приведенным по Ловасу, так как в данном случае $|b_1^*|^2 = 16$, $|b_2^*|^2 = 9$, $\mu_{21} = 0$. Неравенство $\mu_{11} \leq 2\mu_{22}$ выполнено, а $|b_2^*|^2 \geq (3/4 - \mu_{21}^2)|b_1^*|^2$ — нет.

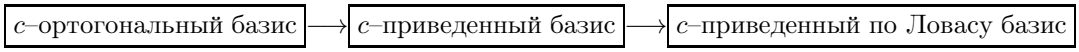

Рис. 2: Приведенные базисы решетки

11 Алгоритм Ловаса приведения базиса решетки

Рассмотрим алгоритм построения приведенного по Ловасу базиса. Как и само понятие такого базиса, алгоритм приведения был предложен Л. Ловасом [10]. Название LLL-алгоритм получил по первым буквам авторов работы [10], где впервые он был опубликован. При $c > 4/3$ алгоритм является полиномиальным. Он основан на двух идеях:

- Если при некоторых k и l условие (16) не выполнено, то b_k можно заменить на $b_k - \lceil \mu_{kl} \rfloor b_l$. При этом μ_{kl} необходимо заменить на $\mu_{kl} - \lceil \mu_{kl} \rfloor$ (и обновить значения μ_{kj} при $j < l$). Очевидно, что после этого условие (16) будет выполнено. Данная процедура называется Size-Reduction.
- Если при некотором k условие (17) не выполнено, то b_k можно поменять местами с b_{k-1}. При этом новым значением b^*_{k-1} станет $b^*_k + \mu_{k,k-1} b^*_{k-1}$ (кроме этого изменятся вектор b^*_k и величины μ_{ij}). В результате величина $|b^*_{k-1}|^2$ уменьшится (более чем в c раз). Эта процедура называется Interchange.

Сначала рассмотрим простейший вариант LLL-алгоритма. Далее $\beta_i = |b^*_i|^2$ $(i = 1, 2, \ldots, n)$.

procedure LLL-Reduction-0 (var $b_1, \ldots, b_n$)
    найти числа μ_{ij} $(i > j)$, β_i и построить систему $b^*_1, \ldots, b^*_n$;
    $k \leftarrow 2$;
    while $k \le n$
        for $l \leftarrow k-1, k-2, \ldots, 1$
            if $|\mu_{k,l}| > 1/2$
                Size-Reduction(k, l);
            end;
        end;
        if $|b^*_k|^2 < \left(y - \mu^2_{k,k-1}\right) |b^*_{k-1}|^2$
            Interchange (k);
            $k \leftarrow \max\{2, k-1\}$;
        else
            $k \leftarrow k+1$;
        end;
    end;
end.

procedure Size-Reduction (k, l)
    $q \leftarrow \lceil \mu_{kl} \rfloor$;
    $b_k \leftarrow b_k - q b_l$;
    $\mu_{kl} \leftarrow \mu_{kl} - q$;
    for $j \leftarrow 1, \ldots, l-1$
        $\mu_{kj} \leftarrow \mu_{kj} - q\mu_{lj}$;
    end
end.

procedure Interchange (k)
    $b_k \leftrightarrow b_{k-1}$;
    for $j \leftarrow 1, \ldots, k-2$
        $\mu_{kj} \leftrightarrow \mu_{k-1,j}$;
    end
    $\mu \leftarrow \mu_{k,k-1}$;
    $\beta \leftarrow \beta_k - \mu^2 \beta_{k-1}$;

$\mu_{k,k-1} \leftarrow \mu\beta_{k-1}/\beta$
$b \leftarrow b^*_{k-1}$;
$b^*_{k-1} \leftarrow b^*_k + \mu b$;
$b^*_k \leftarrow -\mu_{k,k-1} b^*_k + (\beta_k/\beta)b$;
$\beta_k \leftarrow \beta_{k-1}\beta_k/\beta$;
$\beta_{k-1} \leftarrow \beta$;
for $i \leftarrow k+1, \ldots, n$
$t \leftarrow \mu_{ik}$;
$\mu_{ik} \leftarrow \mu i, k-1-\mu t$;
$\mu_{i,k-1} \leftarrow t + \mu k, k-1\mu_{ik}$;
end
end.

Из предыдущего ясно, что если алгоритм LLL-Reduction-0 остановит свою работу, то в результате система $b_1, \ldots, b_n$ будет составлять LLL-приведенный базис решетки.

Докажем конечность алгоритма.

Рассмотрим, какие из величин $\beta_i = |b^*_i|^2$ изменяются в процессе работы алгоритма Interchange. Изменяются только β_{k-1} и β_k. При этом β_{k-1} заменяется на $\beta = \beta_k - \mu^2_{k,k-1}\beta_{k-1}$ — как мы уже видели, это приводит к ее уменьшению более чем в c раз. А β_k заменяется на $\beta_{k-1}\beta_k/\beta$. Рассмотрим как при этом меняется величина

$$B = \big(\det \Lambda(b_1, \ldots, b_i)\big)^2 = \beta_1^n \cdot \beta_2^{n-1} \cdot \ldots \cdot \beta_n.$$

Сейчас меняются только следующие множители:

$$\beta_{k-1}^{n-k+2} \cdot \beta_k^{n-k+1} = \beta_{k-1} \cdot \underbrace{(\beta_{k-1} \cdot \beta_k)^{n-k+1}}_{\text{не изменяется}}.$$

Таким образом, при каждом вызове процедуры Interchange величина B уменьшается во столько раз, во сколько уменьшается β_{k-1}, т. е. более, чем в c раз. Этот процесс не может продолжаться бесконечно, так как из теоремы 4 следует, что $\det \Lambda(b_1, \ldots, b_i)$, а, следовательно, и B ограничены снизу[3].

Теорема 14. [10] *Пусть $\Lambda = \Lambda(b_1, \ldots, b_n)$ — решетка в $\mathbf{Z}^n$, причем $|b_i| \le 2 \le \alpha$ $(i = 1, 2, \ldots, n)$, $c > 3/4$. Тогда алгоритм* LLL-Reduction-0 *построения c-приведенного по Ловасу базиса выполнит $O(n^4 \log \alpha)$ арифметических операций над целыми числами длины $O(n \log \alpha)$ битов.*

При $n = 2$ алгоритм LLL-Reduction-0 превращается в Gauss-Reduction.

Рассмотрим некоторые усовершенствования алгоритма LLL-Reduction-0. Во-первых, проверку условия $|\mu_{k,l}| > 1/2$ при $l < k-1$ можно отложить, так как при этих значениях l величины μ_{kl} не участвуют в проверке перед вызовом процедуры Interchange. Это замечание приводит нас к следующему варианту алгоритма:

procedure LLL-Reduction-1 (var $b_1, \ldots, b_n$)
найти числа μ_{ij} $(i > j)$, β_i и построить систему $b^*_1, \ldots, b^*_n$;
$k \leftarrow 2$;
while $k \le n$
if $|\mu_{k,k-1}| > 1/2$
Size-Reduction$(k, k-1)$;
end;
if $|b^*_k|^2 < \left(y - \mu^2_{k,k-1}\right)|b^*_{k-1}|^2$
Interchange(k, l);
$k \leftarrow \max\{2, k-1\}$;
else
for $l \leftarrow k-2, k-3, \ldots, 1$

[3] Если скалярное произведение задано матрицей Грама с целыми коэффициентами, то ограниченность снизу величины $\det \Lambda(b_1, \ldots, b_i) \in \mathbf{Z}$ очевидна: $|\det \Lambda(b_1, \ldots, b_i)| \ge 1$. К этому случаю можно также свести случай матрицы Грама с рациональными коэффициентами.

```
                    if |μ_{k,l}| > 1/2
                        SIZE-REDUCTION(k, l);
                    end;
                end;
                k ← k + 1;
            end;
        end;
    end.
```

Во время выполнения процедуры INTERCHANGE перевычисляется большое количество коэффициентов μ_{ij} и векторов b_j^*. Многие них изменят несколько раз свое значение, перед тем, как быть использованными. Объем вычислений можно уменьшить, если вычислять μ_{ij} и b_j^* по мере необходимости. В следующем варианте алгоритма $k_{\max}$ равно максимальному достигнутому ранее значению k. В процедуре INTERCHANGE необходимо заменить n на $k_{\max}$.

```
procedure LLL-REDUCTION (var b_1, ..., b_n)
    b_1^* ← b_1;
    β_i = |b_1^*|^2;
    k_max ← 1;
    k ← 2;
    while k ≤ n
        if k > k_max
            k_max ← k;
            найти числа μ_{kj} (j < k), β_k и построить систему b_k^*;
        end;

        if |μ_{k,k-1}| > 1/2
            SIZE-REDUCTION(k, k − 1);
        end;

        if |b_k^*|^2 < (y − μ_{k,k-1}^2) |b_{k-1}^*|^2
            INTERCHANGE(k);
            k ← max{2, k − 1};
        else
            for l ← k − 2, k − 3, ..., 1
                if |μ_{k,l}| > 1/2
                    SIZE-REDUCTION(k, l);
                end;
            end;
            k ← k + 1;
        end;
    end;
end.
```

Приведенный здесь алгоритм LLL-REDUCTION по существу совпадает с алгоритмом 2.6.3 из [12].

Замечание 15. Чтобы получить матрицу перехода из исходного базиса в найденный LLL-приведенный, достаточно в ходе вычислений выполнять следующие действия. В начале работы алгоритма LLL-REDUCTION положить H равной единичной матрице. В алгоритме SIZE-REDUCTION выполнить присваивание $h_k = h_k - qh_l$, где h_j — j-й столбец матрицы H. В алгоритме INTERCHANGE выплнить перестановку столбцов h_k и h_{k-1}.

Также матрица перехода может быть вычислена по окончании работы алгоритма по имеющимся исходному и найденному базисам.

Замечание 16. Если матрица Грама для исходного базиса целочисленна, то алгоритм можно модифицировать так, что все вычисления будут проходить только с целыми числами.

Программные реализации LLL-алгоритма содержатся, например, в библиотеках [11, 15, 14].

Различные модификации описанных здесь алгоритмов, постановок задач и приложения см. в [5, 12].

12 Пример работы алгоритма LLL-Reduction

Пример 17. Найдем c-приведенный по Ловасу базис решетки, натянутой на векторы, чьи столбцы координат в некотором ортонормированном базисе записанны по столбцам матрицы

$$A = \begin{pmatrix} 17 & 0 & 81 \\ 1 & 17 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Выберем $c = 2$, следовательно, $y = 3/4$.

Векторы b_1, b_2, b_3 и b_1^*, b_2^*, b_3^* будем записывать по столбцам в матрицы B и B^* соответственно. Матрица M будет содержать коэффициенты μ_{ij}. На диагонали матрицы M будем записывать величины $|b_i^*|^2$.

$k = 1$

$$B = A \begin{pmatrix} 17 & 0 & 81 \\ 1 & 17 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 17 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad M = \begin{pmatrix} 290 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$k = 2$

$k_{\max} = 2$. После нахождения вектора b_2 с помощью ортогонализации:

$$B = \begin{pmatrix} 17 & 0 & 81 \\ 1 & 17 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 17 & -289/290 & 0 \\ 1 & 4913/290 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad M = \begin{pmatrix} 290 & 0 & 0 \\ 17/290 & 83811/290 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

$\mu_{21} \le 1/2$

$\mu_{22} \ge (3/4 - \mu_{21}^2)\mu_{11}$

$k = 3$

$k_{\max} = 3$. После нахождения вектора b_3^* ортогонализацией:

$$B = \begin{pmatrix} 17 & 0 & 81 \\ 1 & 17 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 17 & -289/290 & 27/27937 \\ 1 & 4913/290 & -459/27937 \\ 0 & 1 & 7803/27937 \end{pmatrix}, \quad M = \begin{pmatrix} 290 & 0 & 0 \\ 17/290 & 83811/290 & 0 \\ 1377/290 & -7803/27937 & 2187/27937 \end{pmatrix}$$

$\mu_{32} \le 1/2$

$\mu_{33} < (3/4 - \mu_{32}^2) * \mu_{22}$. После вызова процедуры Interchange(3):

$$B = \begin{pmatrix} 17 & 81 & 0 \\ 1 & 0 & 17 \\ 0 & 0 & 1 \end{pmatrix}, \quad B^* = \begin{pmatrix} 17 & 81/290 & 0 \\ 1 & -1377/290 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 290 & 0 & 0 \\ 1377/290 & 6561/290 & 0 \\ 17/290 & -289/81 & 1 \end{pmatrix}$$

$k = 2$

$\mu_{21} > 1/2$. После вызова процедуры Size-Reduction(2, 1):

$$B = \begin{pmatrix} 17 & -4 & 0 \\ 1 & -5 & 17 \\ 0 & 0 & 1 \end{pmatrix}, \quad B^* = \begin{pmatrix} 17 & 81/290 & 0 \\ 1 & -1377/290 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 290 & 0 & 0 \\ -73/290 & 6561/290 & 0 \\ 17/290 & -289/81 & 1 \end{pmatrix}$$

$\mu_{22} < (3/4 - \mu_{21}^2) * \mu_{11}$. После вызова процедуры Interchange(2):

$$B = \begin{pmatrix} -4 & 17 & 0 \\ -5 & 1 & 17 \\ 0 & 0 & 1 \end{pmatrix}, \quad B^* = \begin{pmatrix} -4 & 405/41 & 0 \\ -5 & -324/41 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 41 & 0 & 0 \\ -73/41 & 6561/41 & 0 \\ -85/41 & -68/81 & 1 \end{pmatrix}$$

$k = 2$

$\mu_{21} > 1/2$. После вызова процедуры Size-Reduction(2, 1):

$$B = \begin{pmatrix} -4 & 9 & 0 \\ -5 & -9 & 17 \\ 0 & 0 & 1 \end{pmatrix}, \quad B^* = \begin{pmatrix} -4 & 405/41 & 0 \\ -5 & -324/41 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 41 & 0 & 0 \\ 9/41 & 6561/41 & 0 \\ -85/41 & -68/81 & 1 \end{pmatrix}$$

$\mu_{22} \ge (3/4 - \mu_{21}^2) * \mu_{11}$

$k = 3$

$\mu_{32} > 1/2$. После вызова процедуры Size-Reduction(3, 2):

$$B = \begin{pmatrix} -4 & 9 & 9 \\ -5 & -9 & 8 \\ 0 & 0 & 1 \end{pmatrix}, \quad B^* = \begin{pmatrix} -4 & 405/41 & 0 \\ -5 & -324/41 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 41 & 0 & 0 \\ 9/41 & 6561/41 & 0 \\ -76/41 & 13/81 & 1 \end{pmatrix}$$

$\mu_{33} < (3/4 - \mu_{32}^2) * \mu_{22}$. После вызова процедуры Interchange(3):

$$B = \begin{pmatrix} -4 & 9 & 9 \\ -5 & 8 & -9 \\ 0 & 1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} -4 & 65/41 & 27/14 \\ -5 & -52/41 & -54/35 \\ 0 & 1 & -351/70 \end{pmatrix}, \quad M = \begin{pmatrix} 41 & 0 & 0 \\ -76/41 & 210/41 & 0 \\ 9/41 & 351/70 & 2187/70 \end{pmatrix}$$

$k = 2$

$\mu_{21} > 1/2$. После вызова процедуры Size-Reduction(2, 1):

$$B = \begin{pmatrix} -4 & 1 & 9 \\ -5 & -2 & -9 \\ 0 & 1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} -4 & 65/41 & 27/14 \\ -5 & -52/41 & -54/35 \\ 0 & 1 & -351/70 \end{pmatrix}, \quad M = \begin{pmatrix} 41 & 0 & 0 \\ 6/41 & 210/41 & 0 \\ 9/41 & 351/70 & 2187/70 \end{pmatrix}$$

$\mu_{22} < (3/4 - \mu_{21}^2) * \mu_{11}$. После вызова процедуры INTERCHANGE(2):

$$B = \begin{pmatrix} 1 & -4 & 9 \\ -2 & -5 & -9 \\ 1 & 0 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 1 & -5 & 27/14 \\ -2 & -3 & -54/35 \\ 1 & -1 & -351/70 \end{pmatrix}, \quad M = \begin{pmatrix} 6 & 0 & 0 \\ 1 & 35 & 0 \\ 9/2 & -18/35 & 2187/70 \end{pmatrix}$$

$k = 2$
$\mu_{21} > 1/2$. После вызова процедуры SIZE-REDUCTION(2, 1):

$$B = \begin{pmatrix} 1 & -5 & 9 \\ -2 & -3 & -9 \\ 1 & -1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 1 & -5 & 27/14 \\ -2 & -3 & -54/35 \\ 1 & -1 & -351/70 \end{pmatrix}, \quad M = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 35 & 0 \\ 9/2 & -18/35 & 2187/70 \end{pmatrix}$$

$\mu_{22} \geq (3/4 - \mu_{21}^2) * \mu_{11}$
$k = 3$
$\mu_{32} > 1/2$. После вызова процедуры SIZE-REDUCTION(3, 2):

$$B = \begin{pmatrix} 1 & -5 & 4 \\ -2 & -3 & -12 \\ 1 & -1 & -1 \end{pmatrix}, \quad B^* = \begin{pmatrix} 1 & -5 & 27/14 \\ -2 & -3 & -54/35 \\ 1 & -1 & -351/70 \end{pmatrix}, \quad M = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 35 & 0 \\ 9/2 & 17/35 & 2187/70 \end{pmatrix}$$

$\mu_{33} < (3/4 - \mu_{32}^2) * \mu_{22}$.
После вызова процедуры SIZE-REDUCTION(3, 1):

$$B = \begin{pmatrix} 1 & -5 & -1 \\ -2 & -3 & -2 \\ 1 & -1 & -6 \end{pmatrix}, \quad B^* = \begin{pmatrix} 1 & -5 & 27/14 \\ -2 & -3 & -54/35 \\ 1 & -1 & -351/70 \end{pmatrix}, \quad M = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 35 & 0 \\ -1/2 & 17/35 & 2187/70 \end{pmatrix}$$

Матрица перехода:

$$H = \begin{pmatrix} -19 & 14 & 100 \\ 1 & -1 & -6 \\ 4 & -3 & -21 \end{pmatrix}$$

$AH = B$, $\det H = 1$

13 Использование ARAGELI

Listing LLLReduction.cpp

```
#include <arageli/arageli.hpp>

using namespace std;
using namespace Arageli;

int main(int argc, char *argv[])
{
	typedef rational<> T;
	typedef matrix<T> MT;

	MT A = "((17,0,81),(1,17,0),(0,1,0),(5,1,12))";
	MT B, H;

	cout << endl << "Initial basis (in columns) A =" << endl;
	output_aligned(cout, A);

	B = A;
	if (lll_reduction(B, H))
	{
		cout << endl << "Reduced bases (in columns) B =" << endl;
		output_aligned(cout, B);
```

```
        cout << endl << "Transformation matrix H =" << endl;
        output_aligned(cout, H);

        cout << endl << "A*H = " << endl;
        output_aligned(cout, A*H);

        cout << endl << "Check B == A*H? -> " << boolalpha << (B == A*H) << endl;

        cout << endl << "det(H) = " << det(H) << endl;
    }
    else
    {
        cout << "Columns in A don't form a basis" << endl;
        cout << "(they are linear dependent)" << endl;
    }

    return 0;
}
```

```
Initial basis (in columns) A =
||17 0  81||
||1  17 0 ||
||0  1  0 ||
||5  1  12||

Reduced bases (in columns) B =
||-4  13 0 ||
||-5  -4 17||
|| 0  0  1 ||
||-13 -8 1 ||

Transformation matrix H =
||-5 -4 0||
||0  0  1||
||1  1  0||

A*H =
||-4  13 0 ||
||-5  -4 17||
|| 0  0  1 ||
||-13 -8 1 ||

Check B == A*H? -> true

det(H) = 1
```

Listing LLLReductionInt.cpp

```
#include <arageli/arageli.hpp>

using namespace std;
using namespace Arageli;

int main(int argc, char *argv[])
{
```

```
        typedef big_int T;
        typedef matrix<T> MT;

        MT A = "((17,0,81),(1,17,0),(0,1,0),(5,1,12))";
        MT B, H;

        cout << endl << "Initial basis (in columns) A =" << endl;
        output_aligned(cout, A);

        B = A;
        if (lll_reduction_int(B, H))
        {
                cout << endl << "Reduced bases (in columns) B =" << endl;
                output_aligned(cout, B);

                cout << endl << "Transformation matrix H =" << endl;
                output_aligned(cout, H);

                cout << endl << "A*H = " << endl;
                output_aligned(cout, A*H);

                cout << endl << "Check B == A*H? -> " << boolalpha << (B == A*H) << endl;

                cout << endl << "det(H) = " << det(H) << endl;
        }
        else
        {
                cout << "Columns in A don't form a basis" << endl;
                cout << "(they are linear dependent)" << endl;
        }

        return 0;
}
```

```
Initial basis (in columns) A =
||17 0  81||
||1  17 0 ||
||0  1  0 ||
||5  1  12||

Reduced bases (in columns) B =
||-4  13 0 ||
||-5  -4 17||
|| 0  0  1 ||
||-13 -8 1 ||

Transformation matrix H =
||-5 -4 0||
||0  0  1||
||1  1  0||

A*H =
||-4  13 0 ||
||-5  -4 17||
```

```
|| 0  0  1 ||
||-13 -8 1 ||

Check B == A*H? -> true

det(H) = 1
```

14 Приложения

Приведенные базисы решетки имеют многочисленные приложения в компьютерной алгебре [13], теории чисел [12], целочисленном линейном программировании [3, 5, 6], криптографии [1] и др. В качестве примера рассмотрим задачу поиска рациональной зависимости набора вещественных чисел [10]. Пусть $\alpha_1, \ldots, \alpha_n$ — заданные вещественные числа. Требуется найти такие целые числа $x_1, \ldots, x_n$, что $x_1\alpha_1 + \ldots + x_n\alpha_n = 0$ или доказать, что таких чисел не существует.

Рассмотрим пример [8]. Предположим, мы знаем, что интеграл

$$V = \int_0^\infty \frac{\sqrt{x}\ln^5 x}{(1-x)^5}dx = -16.699473719229070496187243400 73\ldots$$

можно выразить в замкнутой форме как сумму с рациональными коэффициентами четных степеней числа π степени не выше 6. Вычислим с высокой точностью:

$$\pi^2 = 9.869604401089358618834490999 88\ldots, \qquad \pi^4 = 97.409091034002437236440332688 71\ldots,$$

$$\pi^6 = 961.389193575304437030219443652 42\ldots$$

Умножим имеющиеся значения V, π^2, π^4, π^6 на 10^{29} и округлим результаты до ближайших целых. Найдем LLL-приведенный базис решетки, натянутой на столбцы матрицы A:

$$A^\top = \begin{pmatrix} 1 & 0 & 0 & 0 & -166994737192290704961872434007 3 \\ 0 & 1 & 0 & 0 & 986960440108935861883449099988 \\ 0 & 0 & 1 & 0 & 974090910340024372364403326887 1 \\ 0 & 0 & 0 & 1 & 961389193575304437030219443652 42 \end{pmatrix}.$$

Пользуясь соответствующим программным обеспечением, получаем LLL-приведенный базис:

$$B = \begin{pmatrix} -24 & -2935612280 & -4639618523 & -825299928 \\ -120 & -542647822 & 1254353485 & -8895056225 \\ -140 & -2670797711 & 3574983458 & 4530117612 \\ 15 & 225187175 & -455689592 & -382015972 \\ -118 & 4346287373 & -4631394987 & 3790400272 \end{pmatrix}.$$

Матрица перехода получается из B вычеркиванием последней строки. Первый столбец матрицы B дает коэффициенты линейной зависимости:

$$-24V - 120\pi^2 - 140\pi^4 + 15\pi^6 = 0.$$

Итак,

$$V = \int_0^\infty \frac{\sqrt{x}\ln^5 x}{(1-x)^5}dx = \frac{5}{24}\pi^2(3\pi^3 - 28\pi^2 - 24).$$

Список литературы

[1] *Василенко О.Н.* Теоретико-числовые алгоритмы в криптографии. М.: МЦНМО, 2003.

[2] *Касселс Дж.* Введение в геометрию чисел. М.: Мир, 1965.

[3] *Схрейвер А.* Теория линейного и целочисленного программирования. В 2-х тт. М.: Мир, 1991.

[4] *Чирков А.Ю., Шевченко В.Н.* О нахождении последовательных минимумов целочисленной решетки и вектора решетки, ближайшего к данному // Кибернетика. 1987. №4. С. 46–49.

[5] *Шевченко В.Н.* Качественные вопросы целочисленного программирования. М.: Физматлит, 1995.

[6] *Aardal K., Weismantel R., Wolsey L.* Non-standard approaches to integer programming. 1999.

[7] *Ajtai M.* The shortest vector problem in L_2 is NP-hard for randomized reductions // Electronic Colloquium for Computational Complexity. 1997. TR97-047.

[8] *Borwein J.M., Lisoněk P.* Applications of Integer Relation Algorithms. Centre for Experimental and Constructive Mathematics Department of Mathematics and Statistics Simon Fraser University. 1997

[9] *Lenstra H.W.* Integer programming with a fixed numbers of variables. Report 81-03. Dep. Math. Univ., Amsterdam. 1981.

[10] *Lenstra A.K., Lenstra H.W., Lovasz L.* Factoring polynomials with rational coefficients // Math. Ann. 1982. V 261. P.515–534.

[11] Arageli: A library for doing exact computation. Department of Computer Science. University of Nizhni Novgorod. `http://www.unn.ru/cs/arageli`.

[12] *Cohen H.* A course in computational algebraic number theory. Berlin e.a.: Springer, 1993.

[13] *von zur Gathen J., Gerhard J.* Modern Computer Algebra. Cambridge University Press, 1999.

[14] LiDIA — A library for computational number theory. TH Darmstadt/Universität des Saarlandes, Fachbereich Informatik, Institut für Theoretische Informatik.
`http://www.informatik.th-darmstadt.de/pub/TI/LiDIA`.

[15] *Shoup V.* NTL: A library for doing number theory. Department of Computer Science, University of Wisconsin–Medison. `http://www.shoup.net`.