Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Курсова робота
освітній ступінь – магістр

на тему: **«АЛГОРИТМИ ЗНАХОДЖЕННЯ РОЗВ'ЯЗКІВ СКЛАДНИХ ЗАДАЧ НА ЦІЛОЧИСЕЛЬНИХ РЕШІТКАХ»**

Виконала: студент 1-го року навчання
освітньої програми «Прикладна математика»,
спеціальності 113 Прикладна математика

Ліхачов Артемій Дмитрович

Керівник: Олійник Б.В.
професор

Рецензент: .

Курсова робота захищена
з оцінкою _____

Секретар ЕК_____
(підпис)

«_____» _____ 20____р.

Київ - 2023

Зміст

Анотація

Курсова робота присвячена вивченню основних алгоритмів розв'язку складних задач на решітках, на яких базуються криптографічні алгоритми та системи цифрового підпису. Вона складається зі вступу, двох розділів, висновків та списку використаної літератури. У вступі розповідається про актуальність тематики та застосування математичного апарату решіток до криптографічних протоколів. У першому розділі вводяться означення решітки, базису решітки, найкоротшого вектора решітки, найближчого вектора решітки, розглядаються властивості решіток. У другому розділі розглядається процес ортогоналізації Грама-Шмідта на решітках, алгоритм LLL, алгоритм Бабаї, числові приклади. У висновках підсумовуються зроблені результати роботи, вказані наступні напрямки досліджень

Ключові слова: цілочисельні решітки, найкоротший вектор, найближчий вектор, алгоритм LLL, алгоритм Бабаї.

Вступ

З розвитком квантових технологій з'являється питання про розвиток та впровадження криптосистем, що базуватимуться на складних задачах для квантових обчислень. Прикладом таких задач, що мають експоненційну складність для квантових обчислень, є задачі на цілочисельних решітках.

Метою даної роботи є дослідження математичного апарату цілочисельних решіток, складних задач на них та основних алгоритмів їх розв'язку.

Робота складається з 2 розділів. У першому розділі наводяться необхідні означення та властивості решіток. Другий розділ присвячений опису алгоритму LLL (Ленстра, Ленстра, Ловас) та алгоритму Бабаї, розглянуто приклади.

Розділ 1

Необхідні визначення

В основі теорії решіток лежать елементи з лінійної алгебри.

Означення 1.1. *[1] Система векторів* $\{b_1, \ldots, b_m\}$ *на просторі* $\mathbb{R}^n$ *є лінійно незалежною, якщо нерівність* $a_1 \cdot b_1 + \ldots + a_m \cdot b_m = 0$ *виконується тоді й тільки тоді, коли* $a_1 = a_2 = \ldots = a_m = 0$.

Означення 1.2. *[1] Векторним підпростіром на* $\mathbb{R}^n$ *розмірності* m *називається лінійна комбінація із* m *лінійно незалежних векторів* $\{b_1, \ldots, b_m\}$ *яка записується наступним чином:*

$$V = \left\{ \sum_{i=1}^{m} a_i \cdot b_i : a_i \in \mathbb{R} \right\},$$

базисом котрого є система векторів $\{b_1, \ldots, b_m\}$.

Означення 1.3. *[1] Якщо ми сформуємо матрицю* B *з* i*-м стовпцем* B*, що дорівнює* b_i *для всіх* i*, тоді ми матимемо*

$$V = \{B \cdot a : a \in \mathbb{R}^m\}.$$

Матриця B *є базисною матрицею.*

Означення 1.4. *[1] Цілочисельною решіткою* L *визначеною системою векторів* $\{b_1, \ldots, b_m\}$ *називається:*

$$L = \left\{ \sum_{i=1}^{m} a_i \cdot b_i : a_i \in \mathbb{Z} \right\} = \{B \cdot a : a \in \mathbb{Z}^m\},$$

де b_i *можуть бути лише цілими лінійними комбінаціями.*

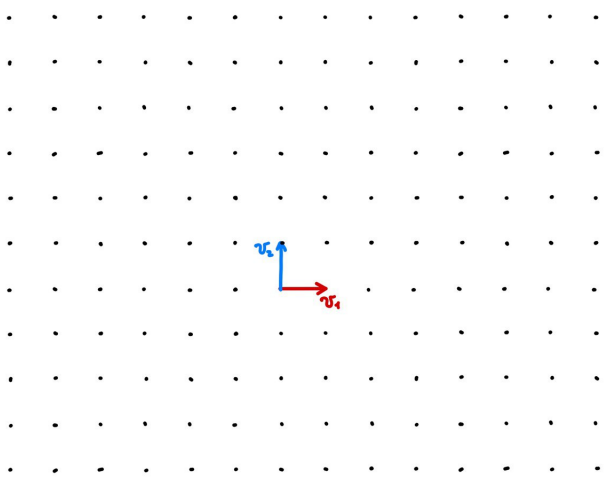

Рис. 1.1: Двовимірна решітка L із базисними векторами $v_1 = (0, 1)$ та $v_2 = (1, 0)$

Отже, якщо взяти базисні вектори $v_1 = (0, 1)$ та $v_2 = (1, 0)$, то решітка L, яку вони описують буде мати вигляд як на Рис. 1.1.

Означення 1.5. *[2] Нехай F буде фундаментальним регіоном на $\mathbb{Z}^n$ та B буде базисом решітки. Тоді, $B \cdot F = \{Bx : x \in F\}$ є фундаментальним регіоном $L = L(B)$. Отже, $P(B)$ буде фундаментальним регіоном L.*

На Рис 1.2 синім зображено фундаментальний регіон решітки L із базисними векторами $\{b_1, b_2\}$.

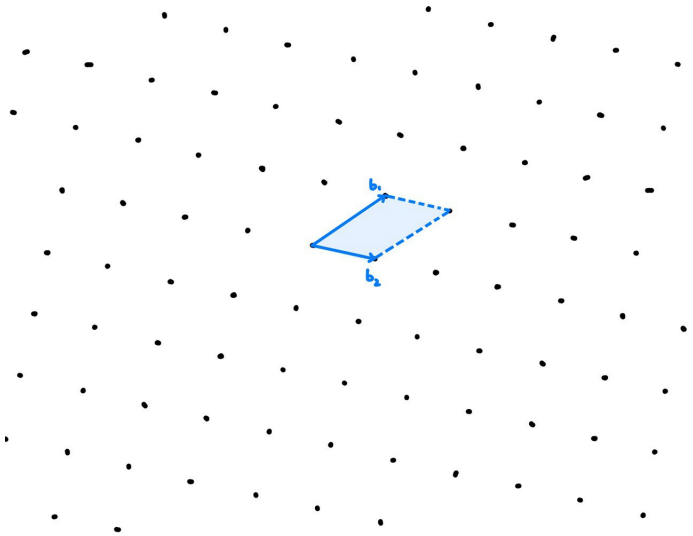

Рис. 1.2: Двовимірна решітка L із базисними векторами $\{b_1, b_2\}$ та фундаментальним регіоном F

Означення 1.6. *[1] Умовно поганим базисом решітки L називають менш ортогональний базис решітки.*

На Рис 1.3 синім кольором зображено умовно поганий базис $\{w_1, w_2\}$, а червоним кольором - умовно хороший базис $\{v_1, v_2\}$.

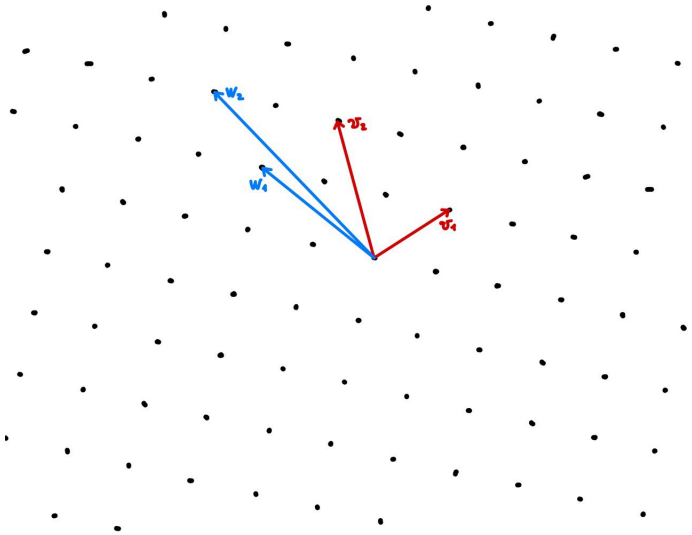

Рис. 1.3: Двовимірна решітка L із базисними векторами $\{w_1, w_2\}$ та $\{v_1, v_2\}$

Означення 1.7. *[1] Враховуючи, що решітка L - це дискретна версія векторного підпростору, у ній існує тривіальний найменший елемент: нульовий вектор. Це дозволяє визначити ненульовий мінімум будь-якої решітки:*

$$\lambda_1(L) = \min\{\|x\| : x \in L, x \neq 0\}$$

Означення 1.8 (Нерівність Адамара)**.** *[3] Нехай L – решітка з базисом $\{b_1, \ldots, b_m\}$ та нехай F – фундаментальний регіон для L, тоді:*

$$\det(L) = \mathrm{Vol}(F) \leq \|b_1\| \cdot \|b_2\| \cdot \ldots \cdot \|b_n\|.$$

Тобто, чим ближче базис буде до ортогональним, тим ближче нерівність Адамара ставатиме рівністю.

Теорема 1.0.1 (Теорема Ерміта)**.** *[3] Будь-яка решітка L розмірності n містить ненульовий вектор $v \in L$, який задовольняє умові:*

$$\|v\| \leq \sqrt{n} \det(L)^{1/n}$$

Існують версії теореми Ерміта для більш ніж одного вектору[3], наприклад, що решітка розмірності L завжди матиме базис $\{b_1, \ldots, b_m\}$, що задовольнятиме умові:

$$\|b_1\| \cdot \|b_2\| \cdot \ldots \cdot \|b_n\| \leq n^{n/2} \det(L),$$

яка доповнює нерівність Адамара. Звідки можна вивести наступне означення[8]

Означення 1.9 (Коефіцієнт Адамара). *[3] Нехай L – решітка з базисом $B = \{b_1, \ldots, b_m\}$, тоді коефіцієнт Адамара дорівнюватиме:*

$$H(B) = \left(\frac{\det(L)}{\|b_1\| \cdot \|b_2\| \cdot \ldots \cdot \|b_n\|} \right)^{1/n}.$$

Отже, $0 < H(B) \leq 1$, та, чим більше значення $H(B)$ наближається до 1, тим більше базиси є ортогональними.

Теорема 1.0.2 (Теорема Мінковського). *[2] Будь-яке опукле центрально-симетричне тіло S об'ємом $vol(S) > 2^n \cdot det(L)$ містить ненульову точку решітки.*

Наслідок 1.0.1 (Перша теорема Мінковського). *[2] Для будь-якої решітки L справедливе наступне твердження:*

$$\lambda_1(L) \leq \sqrt{n} \cdot \det(L)^{1/n}$$

Далі будуть наведені визначення основних складних задач на решітках: пошук найкоротшого вектору та пошук найближчого вектору. Обидві ці задачі експоненційну складність та, на практиці, важко знайти точну відповідь. Тому в означеннях наведено визначення знаходження апроксимованих розв'язків даних задач, які найчастіше використовуються на практиці. У загальному випадку, задачу пошуку найближчого вектору розглядають як NP-повну , а задачу пошуку найкоротшого вектору – як NP-повну за певних умов "гіпотези про рандомізованого скорочення"[4].

Означення 1.10. *[1] Проблему пошуку найкоротшого вектору (Shortest Vector Problem, SVP) у решітці L можна описати одним із трьох наступних способів:*

- *знайти ненульовий вектор x у решітці L, для котрого*

$$\|x\| \leq \|y\|$$

для всіх ненульових $y \in L$, тобто $\|x\| = \lambda_1(L)$.

- *SVP_γ: знаходження апроксимованого найменшого вектора x у решітці L, який*

$$\|x\| \leq \gamma \cdot \lambda_1(L),$$

для малої константи γ.

- *$uSVP_\gamma$: для константи $\gamma > 1$ та решітки L таких, що $\lambda_2(L) > \gamma \cdot \lambda_1(L)$, знайти такий ненульовий вектор $x \in L$ довжини $\lambda_1(L)$.*

Графічний приклад знаходження SVP зображений на Рис 1.4 зеленим кольором.

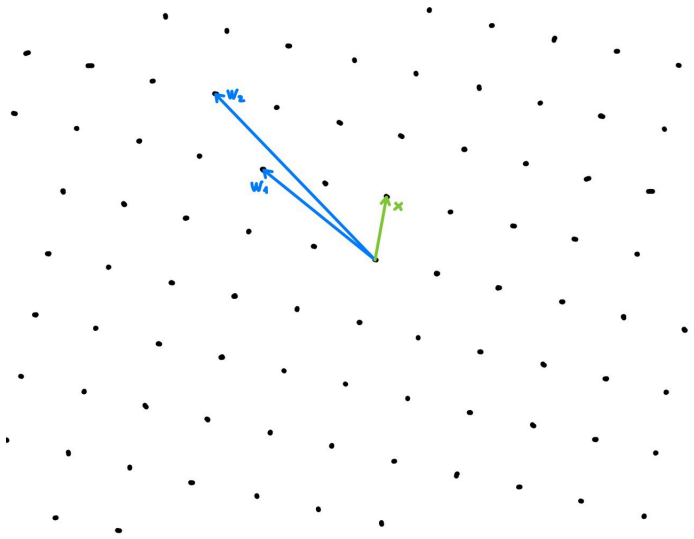

Рис. 1.4: Двовимірна решітка L із базисним вектором $\{w_1, w_2\}$ та найкоротшим вектором x

Означення 1.11. *[1] Маючи решітку L у n-вимірному дійсному просторі та $x \in \mathbb{R}^n$, $x \notin L$, проблему пошуку найближчого вектору (Closest Vector Problem, CVP) можна описати таким чином:*

- *знайти y у решітці, $y \in L$ такий, щоб*

$$\|x - y\| \leq \|x - z\|$$

для всіх $z \in L$.

- *CVP_γ: знайти такий y, щоб*

$$\|x - y\| \leq \gamma \cdot \|x - z\|$$

для всіх $z \in L$ та малої константи γ.

Графічний приклад знаходження CVP зображений на Рис 1.5 зеленим кольором.

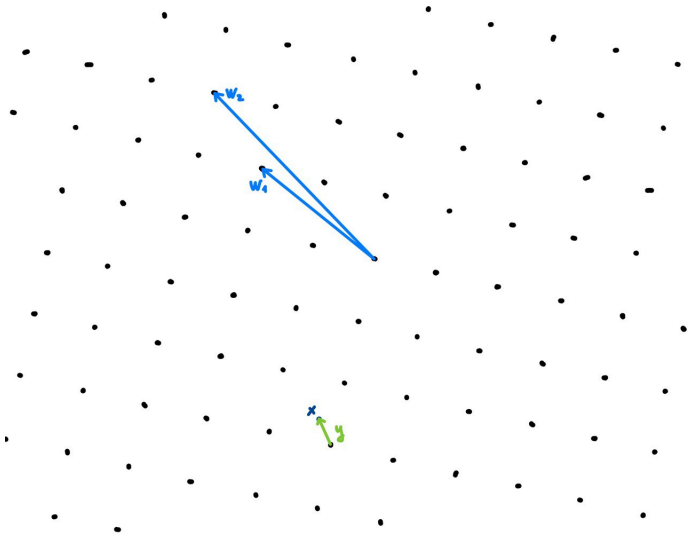

Рис. 1.5: Двовимірна решітка L із базисним вектором $\{w_1, w_2\}$, x та найближчим вектором y

У наступному розділі будуть наведені основні алгоритми пошуку рішення проблем пошуку найкоротшого вектора (Shortest Vector Problem, SVP) та пошуку найближчого вектору (Closest Vector Problem, CVP).

Розділ 2

Алгоритми знаходження розв'язку задач пошуку SVP та CVP

Алгоритми пошуку зазначених задач, зазвичай, мають складність 2^n або $2^{O(n^2)}$ [5]. Пошук найкоротншого вектору являється найбільш ключовою задачею на решітках, бо результат дозволяє знайти рішення інших задач на решітках шляхом їх зведення [6]. Нижче наведені основні історичні віхи в алгоритмах для SVP та CVP.

2.1 Ортогоналізація Грама-Шмідта

Як було зображено на Рис. 1.3 на сторінці 7, рішітку можуть описувати різні базиси. Визначення набору ортогональних векторів значно полегшить пошук найкоротшого вектору решітки. Цей процес є ітеративним. Отже, нехай є базис $\{b_1, \ldots, b_m\}$, який, із допомогою процесу Грама-Шмідта [1], перетворюється в попарно ортогональний базис $\{b_1^*, \ldots, b_m^*\}$. Елементи b_i^* підраховуються з b_i наступним чином:

$$\mu_{i,j} \leftarrow \frac{\langle b_j, b_i^* \rangle}{\langle b_i^*, b_i^* \rangle}, \text{ for } 1 \leq j < i \leq n,$$

$$b_j^* \leftarrow b_j - \sum_{j=1}^{i-1} \mu_{i,j} \cdot b_i^*$$

На Рис. 2.1 візуалізований цей процес із базовими векторами $\{b_1, b_2\}$, які початково не є ортогональними.

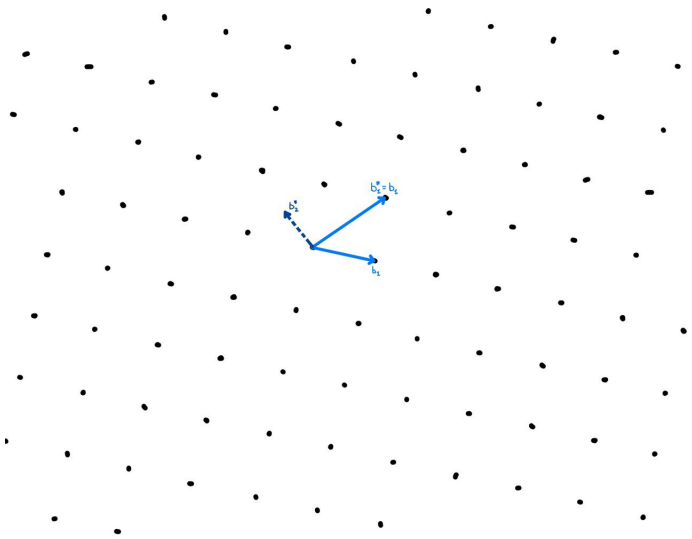

Рис. 2.1: Приклад ортогоналізації Грама-Шмідта

Наприклад, маємо наступні базисні вектори:

$$b_1 = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \text{ та } b_2 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

тоді

$$b_1^* \leftarrow b_1 = \begin{pmatrix} 1 \\ 3 \end{pmatrix},$$

далі підраховуємо

$$\mu_{1,2} = \frac{1 \cdot 0 + 3 \cdot 2}{1 + 9} = \frac{3}{5}$$

та знаходимо b_2^*

$$b_2^* \leftarrow b_2 - \mu_{2,1} \cdot b_1^* = \begin{pmatrix} 0 \\ 2 \end{pmatrix} - \frac{3}{5} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} -3/5 \\ 1/5 \end{pmatrix}$$

Процес ортогоналізації також можна представити у вигляді розкладанні B як $B = B^* \cdot U$ [7]:

$$B = \underbrace{\begin{pmatrix} | & | & | & | \\ b_1^* & b_2^* & \dots & b_n^* \\ | & | & | & | \end{pmatrix}}_{B^*} \cdot \underbrace{\begin{pmatrix} 1 & \mu_{1,2} & \cdots & \mu_{1,n} \\ & 1 & \cdots & \mu_{2,n} \\ & & \ddots & \vdots \\ & & & 1 \end{pmatrix}}_{U}.$$

Далі ми можемо факторизувати довжини стовпчиків b_i^* у B^* та отримати: [13]

$$B^* = Q \cdot \underbrace{\begin{pmatrix} \|b_1^*\| & & & \\ & \|b_2^*\| & & \\ & & \ddots & \\ & & & \|b_n^*\| \end{pmatrix}}_{D},$$

де Q – це ортогональна матриця, $Q^{'}Q = I$. Це тому, що її стовпці взаємно ортогональні одиничні вектори $b_i^* / \|b_i^*\|$ [7]. Із вище зазначеного можемо записати наступне:

$$B = QDU$$

для ортогональної Q , діагональної D та трикутної матриці U.

Лема 2.1.1. *[7] Для будь-якої решітки $L = L(B)$ маємо* $\det(L) = \prod_{i=1}^{n} \|b_i^*\|$.

Лема 2.1.2. *[7] Для будь-якої решітки $L = L(B)$ маємо* $\lambda_1(L) \geq \min_i \|b_i^*\|$.

Якщо взяти наслідок із теореми Мінковського зі сторінки 7, лему 2.1.1 та лему 2.1.2, то можна виразити наступне:

$$\min_i \|b_i^*\| \leq \lambda_1(L(B)) \leq \sqrt{n} \cdot \left(\prod_{i=1}^{n} \|b_i^*\| \right)^{1/n} = \sqrt{n} \cdot GM\left(\|b_i^*\| \right),$$

де GM – геометричне середнє. Тобто, таким чином можна знайти апроксимований розв'язок задачі SVP_γ.

2.2 Алгоритм LLL

Алгоритм Ленстри – Ленстри – Ловаса також використовується для знаходженя апроксимованого розв'язку задачі SVP_γ. Процес Грама-Шмідта знаходить ортогональні базисні вектори, що належать до того ж векторного простору, що й решітка, але ці вектори не належать власне решітці. Алгоритм LLL дозволяє змінити базис так, що новий буде наближеним до ортогонального із $\gamma = 2^{(n-1)/2}$ за експоненційним часом. Визначають наступні вимоги для базису B решітки, щоб він називався зниженим за LLL [7]:

1. Для кожного $i < j$, маємо $|\mu_{i,j}| \leq \frac{1}{2}$.

2. Для кожного $1 \leq i < n$, маємо $\frac{3}{4} \|b_i^*\|^2 \leq \left\|\mu_{i,i+1} b_i^* + b_{i+1}^*\right\|^2$.

Алгоритм LLL працює наступним чином [8]:

1. З допомогою процесу Грама-Шмідта підраховується B^*.

2. Для кожного $j = 2, ..., n$ та i від $j-1$ до 1, нехай $b_j \leftarrow b_j - [\mu_{i,j}] \cdot b_i$, де $\mu_{i,j} = \langle b_j, b_i^* \rangle \,/\, \langle b_i^*, b_i^* \rangle$ являють собою (i, j) елементи верхньої трикутної матриці U в процесі декомпозиції Грама-Шмідта поточної матриці B. Далі виконуємо операцію $B \leftarrow B \cdot W$ на (i, j) ітерації, де W – одинична діагональна матриця з одним не нульовим елементом поза діагоналлю, який дорівнює $\mu_{i,j}$ на позиції (i, j).

3. Якщо існує $1 \leq n < n$ для якого порушується вимога 2, тобто $\frac{3}{4} \|b_i^*\|^2 > \left\|\mu_{i,i+1} b_i^* + b_{i+1}^*\right\|^2$, тоді змінюємо місцями b_i та b_{i+1} та повертаємось на крок 1. В іншому випадку на вихід іде B.

Умови LLL алгоритму запобігають тому, що довжини векторів не будуть стрімко зменшуватись під час процесу ортогоналізації Грама-Шмідта.

Лема 2.2.1. *[7] У базисі B, зменшеному за алгоритмом LLL маємо наступне:*

$$\|b_{i+1}^*\|^2 \geq \frac{1}{2} \|b_i^*\|^2$$

для всіх $1 \leq i < n$.

Доведення. Виходячи з ортогональності векторів, після процесу Грама-Шмідта, між собою, користуючись теоремою Піфагора, маємо:

$$\frac{3}{4} \|b_i^*\|^2 \leq \|\mu_{i,i+1} b_i^* + b_{i+1}^*\|^2 = \mu \mu_{i,i+1}^2 \cdot \|b_i^*\|^2 + \|b_{i+1}^*\|^2 \leq \frac{1}{4} \|b_i^*\|^2 + \|b_{i+1}^*\|^2$$

□

Наслідок 2.2.1. *[7] Для будь-якої решітки L справедливе наступне твердження:*

$$\lambda_1(L) \leq \sqrt{n} \cdot \det(L)^{1/n}$$

Доведення. Відомо, що $b_1 = b_1^*$, отже й $\|b_1\| = \|b_1^*\|$. Із леми 2.2.1 також маємо $\left\|b_{i+1}^*\right\| \geq \frac{1}{\sqrt{2}} \|b_i^*\|$ для кожного $1 \leq i < n$. Тому,

$$\|b_1\| \leq 2^{(i-1)/2} \cdot \|b_i^*\| \leq 2^{(n\cdot 1)/2} \cdot \|b_i^*\|$$

для кожного i. Із цього та з леми 2.1.2 випливає, що

$$\|b_1\| \leq 2^{(n-1)/2} \cdot \min_i \|b_i^*\| \leq 2^{(n-1)/2} \cdot \lambda_1(L(B)).$$

□

Отже, перший вектор у базисі, скороченому за LLL, буде апроксимованим розв'язком задачі пошуку найкоротшого вектору.

Наприклад маємо базиси двовимірної цілочисельної решітки:

$$b_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, b_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

та базис, отриманий із допомогою процесу Грама-Шмідта:

$$b_1^* = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, b_2^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Але не виконується друга вимога:

$$3 = \frac{3}{4} \|b_1^*\|^2 > \|\mu_{1,2} b_1^* + b_2^*\|^2 = 2,$$

отже, змінюємо місцями базові вектори:

$$b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, b_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

Відповідний базис Грама-Шмідта:

$$b_1^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, b_2^* = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Тепер не виконується друга умова, де $|\mu_{1,2}| = 1$, тому віднімаємо базис b_1 від базису b_2 та отримуємо $|\mu_{1,2}| = 0$:

$$b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, b_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Базис Грама-Шмідта залишається таким самим:

$$b_1^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, b_2^* = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Перевіряючи другу вимогу, отримуємо наступне:

$$\frac{3}{2} = \frac{3}{4} \|b_1^*\|^2 \leq \|\mu_{1,2} b_1^* + b_2^*\|^2 = 2.$$

Отже, оскільки обидві умови задовольняються, ми можемо стверджувати, що зменшеним за LLL базисом будуть наступні вектори:

$$b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, b_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

2.3 Алгоритм Бабаї

Алгоритм найближчої площини, або алгоритм Бабаї[9] вирішує задачу пошуку найближчого вектора з наближенням $\gamma = 2^{(n-1)/2}$ за експоненційним часом.

Нехай $L \subset \mathbb{R}^n$ буде решіткою з базисом $\{b_1, \ldots, b_m\}$ та нехай $x \in \mathbb{R}^n$ буде довільним вектором. Якщо вектори в базисі достатньо ортогональні один одному, то проблему найближчого вектору розв'язує наступний алгоритм[3]:

1. Записати $x = t_1 b_1 + t_2 b_2 + \ldots + t_n b_n$ із $t_1, \ldots, t_n \in \mathbb{R}$.
2. Знайти $t_1, \ldots, t_n$.
3. Обчислити $a_i = [t_i] \quad i = 1, 2, \ldots, n$.
4. Повернути $y = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$.

На практиці, вектори базиса рідко є ортогональними, тому є доречним застосовувати алгоритм LLL перед першим кроком алгоритму Бабаї.

Візьмемо базисні вектори з попереднього розділу

$$b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, b_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Та вектор x, до якого шукатимемо найближчий вектор решітки

$$x = \begin{pmatrix} 4 \\ 7 \end{pmatrix}.$$

Запишемо x у вигляді $x = t_1 b_1 + t_2 b_2$:

$$\begin{pmatrix} 4 \\ 7 \end{pmatrix} = t_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + t_2 \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

розв'язуючи це рівняння отримуємо

$$\begin{matrix} t_1 = \dfrac{11}{2} \approx 6 = a_1 \\ t_2 = -\dfrac{3}{2} \approx -2 = a_2 \end{matrix}.$$

Далі обчислюємо $y = a_1 b_1 + a_2 b_2$:

$$y = 6 \begin{pmatrix} 1 \\ 1 \end{pmatrix} - 2 \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 4 \\ 8 \end{pmatrix}.$$

Перевіримо, чи виконується вимога $\|x - y\| \leq \|x - z\|$, $z \in L$ де $z = b_1$, наприклад.

$$\left\| \begin{pmatrix} 4 \\ 7 \end{pmatrix} - \begin{pmatrix} 4 \\ 8 \end{pmatrix} \right\| = 1$$

$$\left\| \begin{pmatrix} 4 \\ 7 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\| = \sqrt{45}$$

$$1 < \sqrt{45}$$

Вимога виконується, отже, найближчим вектором для $x = \begin{pmatrix} 4 \\ 7 \end{pmatrix}$ у решітці L буде вектор $y = \begin{pmatrix} 4 \\ 8 \end{pmatrix}$.

У цьому розділі були розглянуті алгоритми для розв'язку складних задач на решітках - проблеми пошуку найкоротшого вектору (алгоритм LLL) та проблеми пошуку найближчого вектору (алгоритм Бабаї). Були наведені приклади.

Висновки

Курсову роботу було присвячено вивченню властивостей цілочисельних решіток. Розглянуто складні задачі з пошуку найближчого й найкоротшого векторів решіток. Також було наведено алгоритми LLL та Бабаї пошуку найкоротшого та найближчого векторів, відповідно. Для кожного алгоритму було розглянуто приклади, зокрема для конкретної цілочисельної решітки з визначеними базисами було представлено процес знаходження "хорошого"базису та відстань до довільної точки з площини решітки. Алгоритми, що описані в цій роботі лежать в основі криптосистеми GGH , у якій публічний ключ являє собою "поганий"базис, а приватний ключ – "хороший"базис. У такому випадку, "хороший"базис може знайти правильну найближчу точку з найбільшою ймовірністю, навідміну від "поганого"базису. Отже, передача зашифрованого повідомлення відбудеться безпечно. Планується продовжити дослідження використання алгоритмів на цілочисельних решітках та можливості їх застосування в криптографічних системах.

Список літератури

[1] Smart Nigel P. *Cryptography Made Simple*, Springer International Publishing Switzerland 2016.

[2] Peikert Chris *Lecture 1. Mathematical Background*, Lattices in Cryptography, Georgia Tech, Fall 2013.

[3] Hoffstein J., Pipher J., Silverman J.H.. *An Introduction to Mathematical Cryptography*. New York, NJ, USA: Springer, 2008.

[4] Alford W. R., Granville A., Pomerance C. *There are infinitely many Carmichael numbers* Ann. of Math. (2), 139(3):703–722, 1994.

[5] Peikert Chirs *Lecture 6. Algorithms for SVP, CVP*, Lattices in Cryptography, University of Michigan, Fall 2015.

[6] Zhaofei Tian *GGH Cryptosystem and Lattice Reduction Algorithms*, McMaster University (Computing Software) Hamilton, Ontario, Canada May. 2011.

[7] Peikert Chirs *Lecture 2. SVP, Gram-Schmidt, LLL*, Lattices in Cryptography, University of Michigan, Fall 2015.

[8] Peikert Chirs *Lecture 3. LLL, Coppersmith*, Lattices in Cryptography, University of Michigan, Fall 2015.

[9] Babai L. *On Lovasz' lattice reduction and the nearest lattice point problem.* Combinatorica, 6:1-13, 1986. 10.1007 /BF02579403.