НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

«На правах рукопису»
УДК 510.52

«До захисту допущено»
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2024 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Математичні методи криптографічного захисту інформації»

зі спеціальності: 113 Прикладна математика
на тему: **«Застосування алгоритму QAOA для розв'язання задачі SVP»**

Виконав:
студент IV курсу, групи ФІ-03
Кістаєв Матвій Андрійович _____

Керівник:
к.ф.-м.н., ст. викладач
Фесенко Андрій В'ячеславович _____

Рецензент:
к.т.н., доцент
Стьопочкіна Ірина Валеріївна _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ — 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти — перший (бакалаврський)
Спеціальність — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2024 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Кістаєв Матвій Андрійович

1. Тема роботи: «*Застосування алгоритму QAOA для розв'язання задачі SVP*», науковий керівник дипломної роботи: к.ф.-м.н., ст. викладач Фесенко Андрій В'ячеславович,

затверджені наказом по університету №__ від «__» _____ 2024 р.

2. Термін подання студентом роботи: «__» _____ 2024 р.

3. Об'єкт дослідження: *процеси перетворення інформації в квантових обчисленнях*

4. Предмет дослідження: *застосування алгоритму QAOA до задачі SVP*

5. Перелік завдань:

1) провести огляд наявних результатів, щодо застосування квантових алгоритмів для розв'язання задачі SVP;

2) провести пошук та вивчення наявних результатів стосовно зведення задачі SVP до задачі пошуку базового стану гамільтоніану;

3) провести аналіз складності квантової схеми алгоритму QAOA для відомого зведення;

4) дослідити можливість побудови ефективнішого зведення та побудувати його;

5) побудувати точну оцінки складності квантової схеми алгоритму QAOA для нового зведення;

6) провести порівняльний аналіз нового та відомого зведень, дослідити складність алгоритму QAOA для практичних екземплярів задачі SVP.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: *Презентація доповіді*

7. Орієнтовний перелік публікацій: ПОБУДОВА ЗВЕДЕННЯ ЗАДАЧІ SVP ДО ЗАДАЧІ ПОШУКУ ОСНОВНОГО СТАНУ ГАМІЛЬТОНІАНУ ДЛЯ АЛГОРИТМУ QAOA.
XXII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (Україна, м. Київ, 17 травня 2024 р.) : с. 209-212

8. Дата видачі завдання: 10 вересня 2023 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2023 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2023 р.	Виконано
3	Ознайомлення з літературою про зведення задачі SVP до задачі пошуку основного стану гамільтоніану	Листопад-грудень 2023 р.	Виконано
4	Дослідження алгоритму QAOA та відповідної літератури. Побудова загальних оцінок для квантової схеми в алгоритмі	Січень-лютий 2024 р.	Виконано
5	Розробка нового методу зведення задачі SVP до задачі пошуку основного стану	Березень 2024 р.	Виконано
6	Аналіз та побудова оцінок складності для запропонованого методу зведення	Квітень 2024 р.	Виконано
7	Порівняння запропонованого методу із іншими наявними результатами. Формулювання результатів дослідження.	02-15 травня 2024 р.	Виконано
8	Оформлення дипломної роботи, підготовка до захисту та створення презентації	Кінець травня 2024 р.	Виконано

Студент _____ Матвій КІСТАЄВ

Керівник _____ Андрій ФЕСЕНКО

РЕФЕРАТ

Наразі вже оголошено перші 3 переможці конкурсу постквантових асиметричних криптографічних алгоритмів NIST та опубліковано попередні версії відповідних державних стандартів постквантових цифрового підпису та інкапсуляції ключа. Стійкість цих криптосистем-переможців ґрунтується на складності певних задач на решітках. Задача SVP є базовою задачею, яка визначає верхню межу складності для більшості задач на решітках, що використовуються в постквантовій криптографії.

В той же час, поки сучасний рівень розвитку технологій не дозволяє створити завадостійкий масштабований квантовий комп'ютер, який зміг би ефективно застосовувати повноцінні квантові алгоритми для задач практичних розмірів. В таких умовах активно відбувається дослідження та розробка квантових алгоритмів, які дозволяли б застосовувати наявні NISQ-комп'ютери для розв'язання практичних задач. Одним із основних напрямків побудови таких алгоритмів стали варіаційні квантові алгоритми, найбільш сучасним та універсальним із яких наразі є QAOA. Застосування таких алгоритмів вимагає зведення досліджуваної задачі до задачі пошуку основного стану гамільтоніану.

Метою роботи є дослідження складності застосування алгоритму QAOA до задачі SVP, а також можливих шляхів покращення цієї складності. Предметом дослідження є процес застосування алгоритму QAOA до задачі SVP та його складність. У роботі побудовано та доведено оцінки складності алгоритму QAOA для наявного зведення. Також запропоноване новий метод зведення, для якого побудовано аналогічні оцінки складності. Виконано порівняльний аналіз двох описаних методів зведення для практично значимих екземплярів задачі SVP.

ПОСТКВАНТОВА КРИПТОГРАФІЯ, ЗАДАЧА SVP, КВАНТОВІ ВАРІАЦІЙНІ АЛГОРИТМИ, АЛГОРИТМ QAOA

ABSTRACT

The first 3 winners of the NIST Post-Quantum Cryptographic Algorithms competition have already been announced, and public draft versions of the corresponding USA federal standards for post-quantum digital signature and key encapsulation have been published. The security of the corresponding cryptosystems is based on the complexity of certain lattice problems. The Shortest Vector Problem (SVP) serves as a fundamental problem that defines the upper bound of complexity for most lattice problems used in post-quantum cryptography.

At the same time, current technological development has not yet enabled the creation of a fault-tolerant scalable quantum computer capable of effectively applying full-fledged quantum algorithms to practical-sized problems. In such conditions, research and development of quantum algorithms are actively underway to enable existing NISQ computers to tackle practical problems. One of the main directions in building such algorithms has been Variational Quantum Algorithms, with QAOA being the most advanced and versatile one today. Applying such algorithms requires reducing the studied problem to the problem of finding the ground state of a Hamiltonian.

The purpose of this work is to study the complexity of applying the QAOA to the SVP and possible ways to improve this complexity. The subject of the research is the application of the QAOA to the SVP and its complexity. The work constructs and proves quantum complexity bounds for the QAOA for the existing reduction. Additionally, a new reduction method is proposed, for which similar complexity bounds are constructed. A comparison of the two described reduction methods is performed for practically significant instances of the SVP problem.

POST-QUANTUM CRYPTOGRAPHY, SHORTEST VECTOR PROBLEM, VARIATIONAL QUANTUM ALGORITHMS, QAOA

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

NISQ, від англ. Noisy Intermediate-Scale Quantum (комп'ютери), – недосконалі квантові комп'ютери малих та середніх розмірів, такі, які наявні зараз та в найближчому майбутньому.

SVP, від англ. Shortest Vector Problem, – задача пошуку найкоротшого ненульового вектора решітки.

QUBO, від англ. Quadratic Unconstrained Binary Optimization, – задача оптимізації квадратичної форми матриці по всім двійковим векторам.

QAOA, від англ. Quantum Approximate Optimization Algorithm, – квантово-класичний алгоритм комбінаторної оптимізації.

NIST, від англ. National Institute of Standards and Technology, – Національний інститут стандартів та технологій США

$\mathcal{H}_2$ – двовимірний гільбертів простір, що описує стан 1 кубіту.

$\hat{H}$ – гамільтоніан, ермітовий оператор

`CNOT` – двокубітний квантовий вентиль контрольованого оператора заперечення.

$|\psi\rangle$ – **кет-вектор** лінійного простору в нотації Дірака (або, як її ще називають, бра-кет нотації).

Бра-вектором $\langle\psi|$ позначають вектор дуальний до $|\psi\rangle$ із дуального лінійного простору.

$\mathcal{O}(n), \omega(n)$ і т.д. – асимптотична нотація Ландау.

ВСТУП

Актуальність дослідження. В кінці XX ст. була формалізована квантова модель обчислень та тоді ж з'явились перші найважливіші алгоритми, які продемонстрували її теоретичну перевагу над класичною: алгоритм Шора [22], який дозволяє ефективно розв'язувати задачу факторизації та дискретного логарифмування, та алгоритм Гровера [9], який квадратично покращує складність пошуку по невпорядкованому масиву даних.

Потенційна можливість реалізації достатньо потужного квантового комп'ютера, який буде здатний «ламати» сучасні асиметричні криптосистеми, використовуючи вищезгадані алгоритми, підштовхнула дослідників до розробки нових криптосистем, які були б стійкими навіть в квантовій моделі обчислень. Одним із ключових об'єктів в цьому напрямку стали [15] решітки та обчислювальні задачі, з ними пов'язані, які вважаються складними і в квантовій моделі обчислень. Двоє із трьох перших переможців конкурсу постквантових асиметричних криптографічних алгоритмів NIST використовують структури решіток та відповідні задачі для обгрунтування стійкості.

В той же час, сучасний рівень розвитку відповідних технологій поки не дозволяє створити завадостійкий масштабований квантовий комп'ютер, який зміг би ефективно застосовувати вищезгадані алгоритми для задач практичних розмірів. В таких умовах останніми роками активно відбувається дослідження та розробка квантових алгоритмів, які дозволяли б застосовувати наявні недосконалі квантові комп'ютери невеликих розмірів для розв'язання практичних задач. Основними двома принципами побудови таких алгоритмів стали: адіабатичні [8] обчислення та варіаційний [4] принцип. Алгоритм QAOA [6] – квантово-класичний алгоритм комбінаторної оптимізації, що активно досліжується [3] та набув чималої популярності в останній час, він поєднує в собі ідеї обох

принципів. Проте його застосування вимагає побудови зведення досліджуваної задачі до задачі пошуку основного стану гамільтоніану.

Метою дослідження є аналіз складності застосування алгоритму QAOA для розв'язання задачі SVP. Для досягнення мети необхідно вирішити такі **завдання**:

Об'єктом дослідження процеси перетворення інформації в квантових обчисленнях.

Предметом дослідження є застосування алгоритму QAOA до задачі SVP.

Наукова новизна отриманих результатів полягає у тому, що було побудоване нове, ефективніше зведення задачі SVP до задачі пошуку основного стану гамільтоніану, а також вперше отримано оцінки складності застосування алгоритму QAOA до задачі SVP.

Практичне значення. Отримане зведення дозволяє застосовувати алгоритм QAOA до задач більших розмірів, використовуючи ті ж самі обчислювальні ресурси. А отримані точні оцінки складності квантової схеми алгоритму QAOA при застосуванні до задачі SVP дозволять розуміти практичну складність задачі SVP в квантовій моделі обчислень, що, в свою чергу, допоможе в аналізі стійкості багатьох постквантових криптосистем.

Апробація результатів та публікації. Частину результатів цієї роботи представлено на XXII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики». (13 – 17 травня 2024 р., м. Київ, Україна)

1 ВЛАСТИВОСТІ СКЛАДНИХ ЗАДАЧ НА РЕШІТКАХ. ВАРІАЦІЙНІ КВАНТОВІ АЛГОРИТМИ

В цьому розділі описані необхідні теоретичні відомості стосовно решіток, формулювання важливих досліджуваних задач, а також зв'язки між ними. Обгрунтовано актуальність цієї теми в контексті ключової ролі цих задач в постквантовій криптографії.

В ньому також наведено необхідні теоретичні поняття квантової механіки та теорії квантових обчислень, проведено огляд області квантових варіаційних алгоритмів – розглянуто основні ідеї та відповідні алгоритми в контексті сучасних NISQ-комп'ютерів, детально описано алгоритм QAOA.

1.1 Властивості задач на решітках. Важливість задачі SVP

Наразі вже оголошено перші 3 переможці конкурсу постквантових асиметричних криптографічних алгоритмів Національного інституту стандартів та технологій США (англ. NIST) та опубліковано попередні версії відповідних державних стандартів постквантових цифрового підпису та інкапсуляції ключа. Стійкість стандарту [17] для інкапсуляції ключа та одного [16] із стандартів цифрового підпису ґрунтується на складності певних задач на решітках (та пов'язаних задач).

Наведемо основні означення, переважно спираючись на роботу [15].

Означення 1.1. Решіткою $\mathcal{L}$ вимірності n називають дискретну підмножину лінійного простору $\mathbb{R}^n$, яку визначають як:

$$\mathcal{L}(B) = \left\{ \sum_{i=1}^{n} x_i \cdot \vec{b_i} \ : \ x_i \in \mathbb{Z} \right\},$$

де $B = \left\{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\right\}$ – лінійно незалежні вектори з простору $\mathbb{R}^n$.

Для формулювання наступних понять і задач необхідно ввести поняття норми вектору решітки. Теоретично можна розглядати такі задачі для довільної норми, проте в межах цієї роботи нормою $\|\cdot\|$ вектора решітки будемо називати звичайну евклідову норму на просторі $\mathbb{R}^n$.

Означення 1.2. i-им послідовним мінімумом $\lambda_i(\mathcal{L})$ решітки $\mathcal{L}$ називається i-та за зростанням величина із множини:

$$\left\{ \|\vec{v}\| \ : \ \vec{v} \in \mathcal{L}, \|\vec{v}\| \neq 0 \right\}$$

Окремо визначимо 0-ий послідовний мінімум: $\lambda_0(\mathcal{L}) = 0$

Найбільш базовою задачею є задача пошуку найкоротшого ненульового вектора решітки:

Задача 1.1 (SVP)**.** Для довільного заданого базису $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ лінійного простору $\mathbb{R}^n$, знайти найкоротший ненульовий вектор решітки $\mathcal{L}(B)$, тобто знайти вектор:

$$\vec{v} \in \mathcal{L}(B) \ \text{ такий що } \ \|\vec{v}\| = \min_{\vec{w} \in \mathcal{L}(B)} \left\{ \|\vec{w}\| \ : \ \vec{w} \neq \vec{0} \right\},$$

Приклад екземпляру задачі SVP для двовимірної решітки наведено на рисунку 1.1.

Також наведемо дві інших важливих задачі:

Задача 1.2 (CVP)**.** Для довільного заданого базису $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ лінійного простору $\mathbb{R}^n$ та вектору $\vec{t} \in \mathbb{R}^n$, знайти найближчий до $\vec{t}$ вектор решітки $\mathcal{L}(B)$, тобто знайти вектор:

$$\vec{v} \in \mathcal{L}(B) \ \text{ такий що } \ \|\vec{v} - \vec{t}\| = \min_{\vec{w} \in \mathcal{L}(B)} \left\{ \|\vec{v} - \vec{t}\| \ : \ \vec{w} \neq \vec{t} \right\},$$

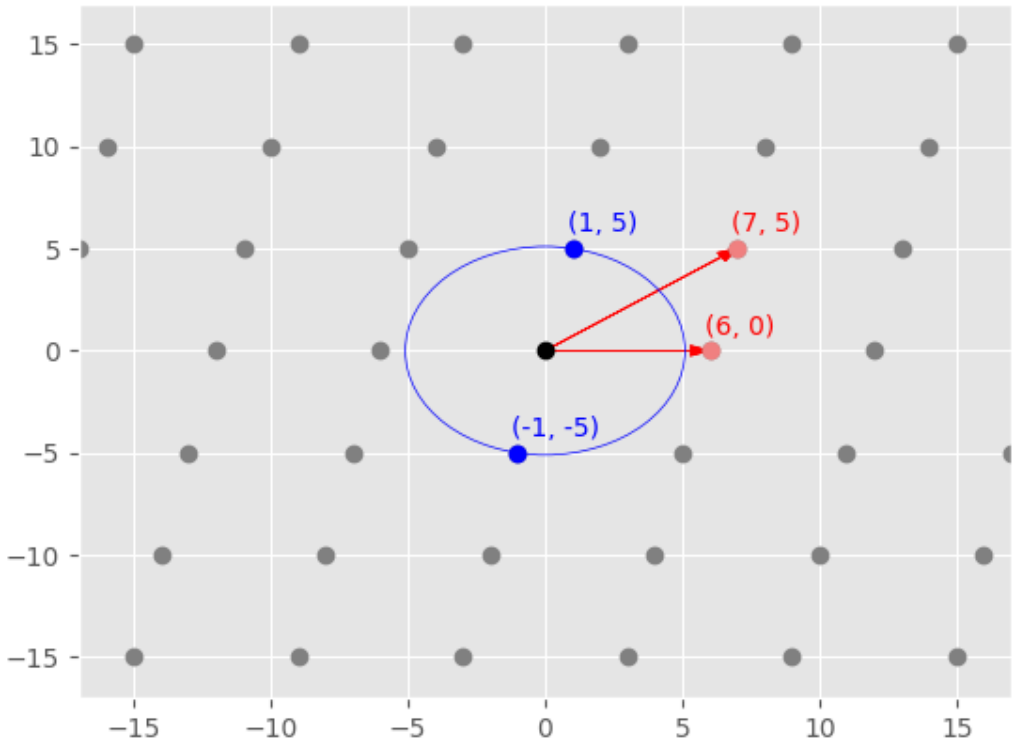
Задача 1.3 (SIVP)**.** Для довільного заданого базису $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ лінійного простору $\mathbb{R}^n$, знайти «найкоротший» набір n лінійно-незалежних векторів решітки $\mathcal{L}(B)$, а саме знайти базис:

$$S = \left\{ \vec{v}_1, \vec{v}_2, \ \ldots \ \vec{v}_n \right\} \subseteq \mathcal{L}(B)$$

такий що довжина найдовшого вектору базису є мінімальна, тобто:

$$\text{величина } \|S\| = \max_{\vec{v}_i \in S} \|\vec{v}_i\| \text{ – мінімальна.}$$

Для довільного заданого базису загального вигляду ці задачі вважаються обчислювально складними, навіть в квантовій моделі обчислень.



Базисом для цієї задачі є вектори $\vec{b}_1 = (7, 5)$ та $\vec{b}_2 = (6, 0)$, а розв'язком (найкоротшим ненульовим вектором решітки) – $\vec{v}_1 = (1, 5)$ або $\vec{v}_2 = -\vec{v}_1 = (-1, -5)$, норма якого дорівнює $\sqrt{26}$ (синє коло).

Рисунок 1.1 – Приклад задачі SVP

При цьому задача SVP є в певному сенсі базовою, оскільки до неї можна звести більшість інших задач на решітках. Таким чином вона задає верхню межу складності для них. В роботі [13, Схема 3] наведена схема 1.2, що вичерпно описує зведення між різними задачами на решітках.

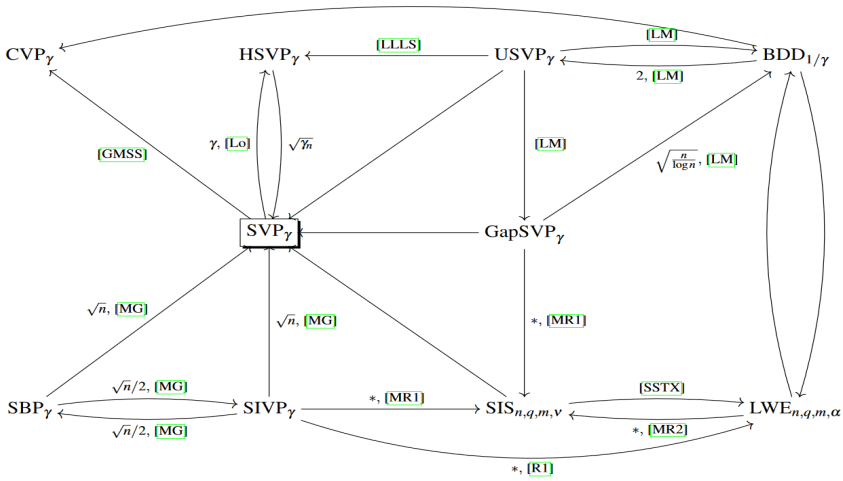


Рисунок 1.2 – Схема зведень задач на решітках

1.2 Важливі поняття квантової моделі обчислень. Основний стан гамільтоніану. Адіабатична теорема

Описані далі загальні теоретичні відомості, які стосуються квантової механіки або теорії квантових обчислень, переважно спираються на підручники «Квантова Механіка» [25] та «Quantum Computation and Quantum Information» [20].

Стан $|\psi\rangle$ довільної квантовомеханічної системи описується вектором у відповідному гільбертовому просторі станів $\mathcal{H}$.

В межах цієї роботи будемо розглядати лише n-кубітні квантові регістри, стан яких описується вектором в 2^n-вимірному гільбертовому просторі

$$\mathcal{H}_2^{\otimes n} = \underbrace{\mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \ldots \otimes \mathcal{H}_2}_{n},$$

де $\mathcal{H}_2$ – двовимірний гільбертів простір, який описує стан одного кубіту:

$$\mathcal{H}_2 = \{\alpha\,|0\rangle + \beta\,|1\rangle \;:\; \|\alpha\|^2 + \|\beta\|^2 = 1\}$$

Означення 1.3. Гамільтоніаном називають ермітовий оператор $\hat{H}$ в гільбертовому просторі $\mathcal{H}$, який описує повну енергію

квантово-механічної системи.

Означення 1.4. Основним станом гамільтоніану $\hat{H}$ називають вектор-стан в $\mathcal{H}$, що є власним вектором оператора $\hat{H}$ і який відповідає його найменшому власному значенню.

Основний стан є станом рівноваги квантово-механічної системи.

Еволюція в часі довільної квантово-механічної системи повністю визначається гамільтоніаном і описується диференційним **рівнянням Шредингґера**:

$$i\frac{\mathrm{d}}{\mathrm{d}t}\left|\psi(t)\right\rangle = \hat{H}(t)\left|\psi(t)\right\rangle \tag{1.1}$$

Зауваження. В загальному випадку гамільтоніан $\hat{H}(t)$ залежить від часу.

Важливим наслідком із цього факту є те, що будь-яке перетворення квантово-механічної системи описується **унітарним** (а отже і **оборотним**) оператором.

Розв'язком рівняння Шрьодінгера (1.1) в момент часу T з початковим станом $\left|\psi(0)\right\rangle$ є стан $U(T)\left|\psi(0)\right\rangle$, де унітарне перетворення $U(T)$ визначається як:

$$U(T) = e^{-i\int_0^T \hat{H}(\tau)d\tau}.$$

Точно реалізувати таке перетворення зазвичай неможливо, оскільки для цього необхідно обчислити інтеграл у степені. Тому частіше еволюцію квантово-механічної системи моделюють наближено:

Інтеграл можна наблизити дискретною сумою:

$$U(T) \approx e^{-i\sum_{k=0}^{r} \hat{H}(k\tau)\Delta\tau}.$$

В свою чергу, використавши формулу Лі-Троттера [20, Рівняння 4.103] для матричної експоненти від суми, можна отримати наближення:

$$U(T) \approx \prod_{k=0}^{r} e^{-i\hat{H}(k\tau)\Delta\tau}.$$

Для багатьох класів гамільтоніанів, що мають практичне значення, можливо *ефективно* [24] побудувати квантову схему, яка реалізовує унітарне перетворення $e^{-i\theta\hat{H}}$.

Детальніше наближена симуляція еволюції квантово-механічних систем за допомогою квантового комп'ютера описана в [20, Розділ 4.7]

Теорема 1.1 (Адіабатична теорема [8])**.** *Нехай поведінка квантово-механічної системи задається рівнянням Шрьодінгера:*

$$i\frac{\mathrm{d}}{\mathrm{d}t}\left|\psi(t)\right\rangle = \hat{H}(t)\left|\psi(t)\right\rangle$$

Для гладкого класу гамільтоніанів $\tilde{H}(s),\ s \in [0,1]$, *для гамільтоніану* $\tilde{H}(s)$ *позначимо* k*-е в порядку зростання власне значення* $E_k(s)$ *і відповідний власний вектор* $\left|E_k(s)\right\rangle$*:*

$$\tilde{H}(s)\left|E_k(s)\right\rangle = E_k(s)\left|E_k(s)\right\rangle$$

Тоді, якщо взяти $\hat{H}(t) = \tilde{H}(t/T)$*, з початковим станом* $\left|\psi(0)\right\rangle = \left|E_1(0)\right\rangle$*, то стан квантово-механічної системи* $\left|\psi(t)\right\rangle$ *в кінцевий момент часу* $t = T$ *наближається до власного стану* $\left|E_1(1)\right\rangle$ *гамільтоніану* $\tilde{H}(1)$*:*

$$\lim_{T\to\infty}\left|\left\langle\psi(T)|E_1(1)\right\rangle\right| = 1$$

При цьому,

$$\left|\left\langle\psi(T)|E_1(1)\right\rangle\right| = 1 - o(\varepsilon)$$

якщо зміна гамільтоніана $\hat{H}(t)$ *в часі буде відбуватись достатньо повільно, а саме:*

$$T = \omega\left(\frac{1}{\varepsilon\cdot g_{min}^2}\right)$$

де g_{min} *– різниця рівня енергії між основним станом та першим*

збудженим станом:

$$g_{min} = \min_{0 \leqslant s \leqslant 1} \{E_2(s) - E_1(s)\}$$

В 2000 році Е. Фархі та Дж. Голдстоун запропонували [8], як можна використовувати принцип адіабатичного переходу для розв'язання обчислювальних задач на квантовому комп'ютері. А в 2001 році опублікували роботу [7] із застосуванням такого підходу для розв'язання NP-повних задач.

Отже принцип адіабатичного квантового обчислення можна описати [8, Розділ 2] наступним чином:

1) звести досліджувану задачу до пошуку основного стану $|\varphi_P\rangle$ гамільтоніану $\hat{H}_P$;

2) розгянути певний інший гамільтоніан $\hat{H}_0$ для якого можна легко підготувати його основний стан $|\psi_0\rangle$;

3) побудувати систему в початковому стані $|\psi_0\rangle$, еволюція якої керується гамільтоніаном (для достатньо великого значення T):

$$\hat{H}(t) = (1 - \frac{t}{T})\hat{H}_0 + \frac{t}{T}\hat{H}_P.$$

Тоді система знаходиться в основному стані $|\psi_0\rangle$ гамільтоніану $\hat{H}_0$ в початковий момент часу $t = 0$, а наприкінці в момент часу $t = T$, згідно з твердженням теореми 1.1, в основному стані $|\varphi_P\rangle$ гамільтоніану $\hat{H}_P$, з якого можна отримати розв'язок досліджуваної задачі.

Можливо як фізично побудувати таку квантово-механічну систему для заданої задачі, так і симулювати еволюцію такої системи наближено за допомогою квантового комп'ютера. Найбільш яскравим представником першого підходу є компанія D-Wave [11], яка спеціалізується на побудові квантових комп'ютерів, що здатні знаходити основний стан для заданого гамільтоніану спеціального вигляду (так званий квантовий відпал [12]). Проте в межах цієї роботи будуть розглядатись лише дискретні квантові комп'ютери загального призначення.

1.3 Варіаційні квантові алгоритми. Алгоритм QAOA

Останніми роками варіаційні квантові алгоритми мають велику популярність серед дослідників, публікується багато робіт, які описують їхнє застосовування до різних задач, побудову нових модифікацій та інший пов'язаний аналіз. В роботах [4, 2] проведено актуальний та детальний огляд цього напрямку.

Варіаційний принцип до побудови квантових алгоритмів був вперше запропонований в 2013 році в роботі [21], разом із експрементальною реалізацією алгоритму «Variational Quantum Eigensolver», який обчислює основний стан для заданого гамільтоніана певного вигляду, використовуючи варіаційний метод [25, Параграф 53] квантової механіки.

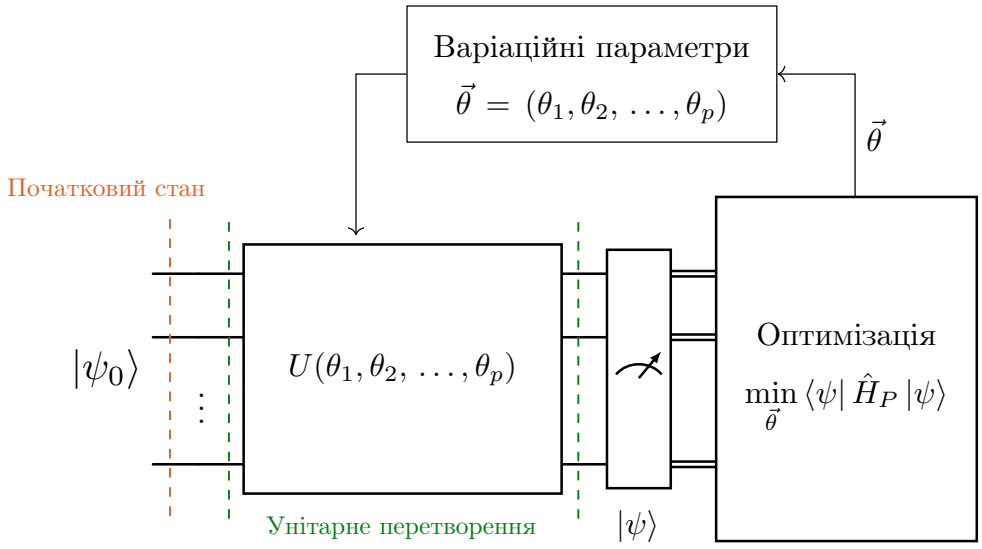


Рисунок 1.3 – Схема алгоритму VQE

В загальному випадку алгоритм VQE має вигляд:

1) Початкова задача зводиться до задачі пошуку основного стану $|\varphi\rangle$ гамільтоніану $\hat{H}_P$ в 2^n-вимірному гільбертовому просторі $\mathcal{H}_2^{\otimes n}$.

2) Класичний комп'ютер мінімізує значення функції ваги F_p, яка обчислюється як середнє значення багаторазового вимірювання спостережуваної величини $\hat{H}_P$ для стану $|\psi\rangle$, що отриманий в результаті

виконання параметризованої квантової схеми:

3) В загальному випадку вигляд квантової схеми може бути довільним – початковий стан $|\psi_0\rangle$ та параметричне унітарне перетворення $U(\vec{\theta})$. Важливо тільки те, щоб основний стан гамільтоніану $\hat{H}_P$ міг бути заданий як $|\varphi\rangle = U(\vec{\theta})\,|\psi_0\rangle$

Схема VQE наведена на рисунку 1.3. Цей алгоритм насправді описує загальну схему побудови довільного квантового варіаційного алгоритму.

Основною перевагою квантових варіаційних алгоритмів такого вигляду є те, що у випадку, коли гамільтоніан $\hat{H}_P$ можна ефективно описати, то кількість кубітів у квантовій схемі дорівнює кількості кубітів, необхідних для реалізації гамільтоніана $\hat{H}_P$. Також, використання квантового комп'ютера лише як допоміжної складової в рамках класичного алгоритму оптимізації дозволяє зменшити складність квантової схеми, що частково вирішує проблему помилок під час обчислення – одну із основних проблем сучасних квантових обчислень.

QAOA. Після того, як був запропонований вараційний підхід до побудови квантових алгоритмів, в 2014 році Е. Фархі та Дж. Голдстоун в роботі [6] запропонували алгоритм QAOA. Він поєднує принцип варіаційних квантових алгоритмів, де параметри квантової схеми оптимізуються для обчислення потрібного квантового стану, та принцим адіабатичних квантових обчислень, що достатньо повільна еволюція квантовомеханічної системи залишає її в стані рівноваги.

Алгоритм QAOA є алгоритмом, побудованим за варіаційною схемою, як і VQE, але унітарне перетворення U має додатковий параметр p, зі збільшенням якого, воно все більше буде відповідати адіабатичній еволюції системи.

Так само, як для VQE, початкова задача зводиться до задачі пошуку основного стану гамільтоніану $\hat{H}_P$ в 2^n-вимірному гільбертовому просторі $\mathcal{H}_2^{\otimes n}$. Далі класичний комп'ютер мінімізує значення функції ваги F_p, яка обчислюється як середнє значення багаторазового вимірювання стану $|\psi\rangle$,

що отриманий в результаті виконання параметризованої квантової схеми:

$$F_p(\vec{\theta}) = \langle\psi| \hat{H}_P |\psi\rangle .$$

Квантова схема, яка обчислює стан $|\psi\rangle$, залежить від $2p$ параметрів $(\vec{\gamma}, \vec{\beta}) = (\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)$ і має наступний вигляд:

1) Підготовка початкового стану $|\psi_0\rangle$, який є основним станом гамільтоніана $\hat{H}_M$, зазвичай:

$$\hat{H}_M = -\sum_{i=1}^{n} X_i,$$
$$|\psi_0\rangle = |+\rangle^{\otimes n} = H^{\otimes n} |0\rangle^{\otimes n},$$

де X_i – X-оператор Паулі, який діє на i-ий кубіт, H – оператор Адамара.

2) Послідовні p рівнів унітарних перетворень, кожен з яких має вигляд $U_M(\beta_i)U_P(\gamma_i)$:

$$U_M(\beta_p)U_P(\gamma_p) \ldots U_M(\beta_1)U_P(\gamma_1),$$

де унітарні перетворення U_M та U_P:

$$U_M(\beta) = e^{-i\beta\hat{H}_M},$$
$$U_P(\gamma) = e^{-i\gamma\hat{H}_P}.$$

Вектор-стан, який задається квантовою схемою в алгоритмі, позначають $\left|\psi(\vec{\gamma}, \vec{\beta})\right\rangle$ і обчислюють як:

$$\left|\psi(\vec{\gamma}, \vec{\beta})\right\rangle = U_M(\beta_p)U_P(\gamma_p) \ldots U_M(\beta_1)U_P(\gamma_1)H^{\otimes n} |0\rangle^{\otimes n} .$$

Стан $\left|\psi(\vec{\gamma}, \vec{\beta})\right\rangle$, заданий параметрами $(\vec{\gamma}, \vec{\beta})$, які відповідають мінімуму функції $F_p(\vec{\gamma}, \vec{\beta})$, є також і основним станом гамільтоніана $\hat{H}_P$.

Схема алгоритму QAOA наведена на рисунку 1.4.

Алгоритм QAOA наразі є одним із основних квантових алгоритмів,

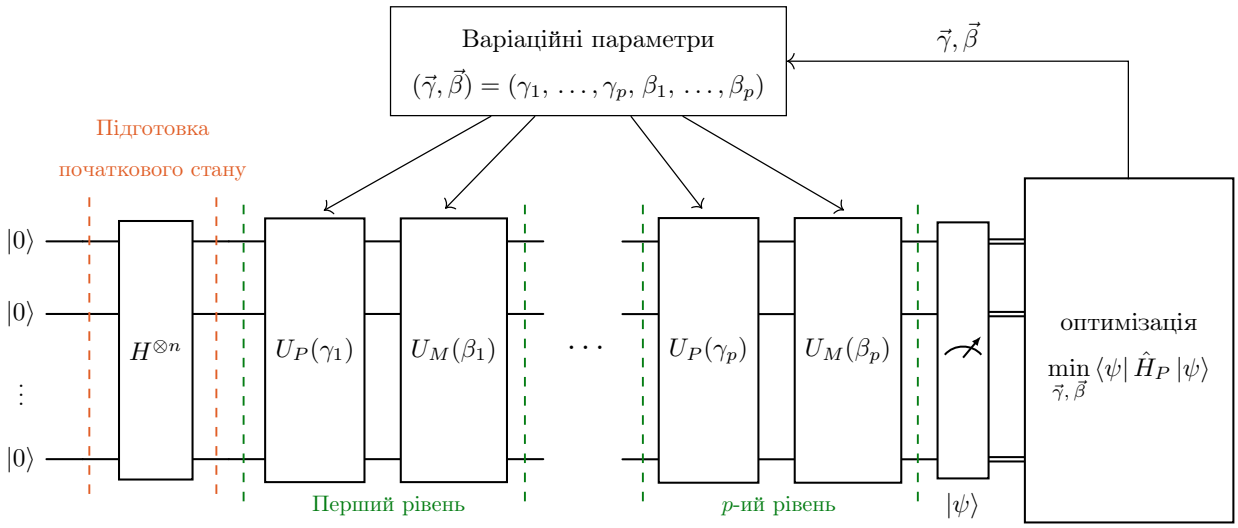


Рисунок 1.4 – Схема алгоритму QAOA

який досліджується в рамках використання сучасних NISQ-ком'ютерів. Існує чимала кількість робіт, які описують різні модифікації [3] цього алгоритму, а також експерементальні результати його застосування.

Висновки до розділу 1

В цьому розділі обгрунтовано актуальність дослідження складних задач на решітках, враховуючи важливість цієї теми для сучасної постквантової криптографії.

Описані основні необхідні поняття і властивості решіток та розглянуто основні складні задачі:

– (SVP) задача пошуку найкоротшого ненульового вектору;

– (CVP) задача пошуку вектору решітки, найближчого до заданого;

– (SIVP) задачу пошуку «найкоротшого» базису решітки.

Детальніше розглянуто задачу SVP, її зв'язок із іншими задачами на решітках, що показує її фундаментальну роль серед цих задач.

Також в цьому розділі наведенно основні відомості з квантової механіки та теорії квантових обчислень, які потрібні в рамках цього дослідження.

Наведено адіабатичну теорему та принцип адіабатичних квантових обчислень, ідея якого базується на цій теоремі.

Розглянуто історію розвитку та основні принципи побудови варіаційних квантових алгоритмів, описано їхню практичну цінність для сучасних NISQ-обчислень. Детальніше описано алгоритм QAOA, який є найбільш досконалим та універсальним представником такого класу алгоритмів.

Застосування алгоритму QAOA вимагає зведення досліджуваної задачі до задачі пошуку основного стану гамільтоніану, таким чином однією із основних задач цього дослідження є вивчення та аналіз наявного приниципу побудови такого зведення, а також побудова ефективнішого зведення, або покращення наявного.

2 АНАЛІЗ НАЯВНОГО ЗВЕДЕННЯ ЗАДАЧІ SVP ДО ПОШУКУ ОСНОВНОГО СТАНУ ГАМІЛЬТОНІАНУ

Другий розділ описує зведення задачі SVP до задачі пошуку основного стану гамільтоніану, яке необхідне для застосування варіаційних квантових алгоритмів для розв'язання цієї задачі, переважно спираючись на роботу [1]2023 року Мартіна Албрехта та інших.

Наведено необхідні теоретичні поняття, описано оцінки для довжини найкоротшого вектору решітки, а також побудовано і доведено оцінки кількості кубітів та складності квантової схеми алгоритму QAOA для зведення, запропонованого в роботі [1].

2.1 Зведення SVP до пошуку основного стану гамільтоніану

Для застосування довільного квантового алгоритму, який використовує варіаційний принцип, необхідно спочатку звести досліджувану задачу до задачі пошуку основного стану певного гамільтоніану. Стандартною практикою [14, 5] є зведення досліджуваної задачі спочатку до задачі QUBO, яку потім можна тривіально звести до задачі пошуку основного стану гамільтоніану.

Задача 2.1 (QUBO [5])**.** Для заданої квадратної n-вимірної матриці Q над дійсними числами знайти двійковий вектор $\vec{x} \in \{0,1\}^n$, який мінімізує значення

$$\vec{x}^T Q \vec{x} = \sum_{1 \leqslant i,j \leqslant n} Q_{ij} x_i x_j. \tag{2.1}$$

Існує відомий метод зведення задачі SVP до задачі QUBO, проте він не враховує обмеження на ненульову довжину вектора. Таке зведення описано в [1, розділ 3].

Неформальне зведення SVP $<$ QUBO. Нехай маємо екземпляр задачі SVP із заданим базисом $B = \left\{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\right\}$ решітки $\mathcal{L}(B)$. Розглянемо матрицю B, яка в стовпчиках складається із векторів базису B:

$$B = \begin{bmatrix} | & | & & | \\ \vec{b_1} & \vec{b_2} & \ldots & \vec{b_n} \\ | & | & & | \end{bmatrix}.$$

Надалі в позначеннях будемо ототожнювати базис B та матрицю B із векторами базису в стовпцях.

Тоді задача мінімізації довжини вектора (SVP):

$$\operatorname*{argmin}_{\vec{v}\in\mathcal{L}(B)} \|\vec{v}\| = \operatorname*{argmin}_{\vec{x}\in\mathbb{Z}^n} \|B\vec{x}\| \ ,$$

де $\vec{x} = [x_1, \ldots, x_n]$, $x_i \in \mathbb{Z}$, еквівалентна задачі мінімізації квадрату цієї довжини, який дорівнює $\|\vec{v}\|^2 = \vec{x}^T B^T B \vec{x}$:

$$\operatorname*{argmin}_{\vec{v}\in\mathcal{L}(B)} \|\vec{v}\| \ = \ \operatorname*{argmin}_{\vec{v}\in\mathcal{L}(B)} \|\vec{v}\|^2 \ = \ \operatorname*{argmin}_{\vec{x}\in\mathbb{Z}^n} \Big(\vec{x}^T \underbrace{B^T B}_{G} \vec{x}\Big), \tag{2.2}$$

де G – це матриця Грама для базису B.

Задача мінімізації (2.2) є задачею квадратичної необмеженої *цілочисельної* оптимізації, тобто мінімізації виразу вигляду:

$$\vec{x}^T G \vec{x} \ = \sum_{1\leqslant i,j\leqslant n} G_{ij} x_i x_j.$$

для цілочисельних змінних x_i.

Щоб перетворити її на задачу квадратичної *двійковоі* оптимізації (QUBO), необхідно представити цілочисельні змінні x_i в двійковому кодуванні:

$$x \ \longrightarrow \ (\hat{x}_k \ldots \hat{x}_1 \hat{x}_0)_2, \qquad \hat{x}_j \in \{0, 1\}$$

Для цього в свою чергу необхідно ввести обмеження на кількість бітів k_i

у двійковому представленні кожної змінної x_i, тобто потрібно знайти такі обмеження a_i, що $|x_i| < a_i$. Детальніше побудова таких обмежень описана в наступному підрозділі.

Тоді нові двійкові змінні $\hat{x}_{ij}$ для $i \in \{1, \ldots, n\}$ та $j \in \{0, \ldots, k_i\}$, де $k_i = \lceil \log_2(2a_i) \rceil$, задають наступним чином:

$$x_i = -\hat{x}_{ik_i} 2^{k_i} + \sum_{j=0}^{k_i - 1} \hat{x}_{ij} 2^j,$$

де $\hat{x}_{ij}$ для $j \in \{0, \ldots, k_i\}$ – це відповідні біти в двійковому представленні чисел x_i (в двоїстому двійковому кодуванні цілих чисел зі знаком).

Після такої заміни змінних отримують еквівалентну задачу оптимізації, яка є екземпляром QUBO для m двійкових змінних $\hat{x}_{ij}$, де

$$m = \sum_{i=1}^{n} k_i = \sum_{i=1}^{n} \lceil \log_2(2a_i) \rceil. \tag{2.3}$$

Проте таке «зведення» формально не є коректним, оскільки кінцевий екземпляр QUBO не враховує обмеження на ненульову довжину вектора. Тобто першим мінімумом для кінцевого екземпляру задачі QUBO буде набір змінних $\hat{x}_{ij} = 0$ для всіх $i \in \{1, \ldots, n\}$ та $j \in \{0, \ldots, k_i\}$, який відповідає нуль-вектору $\vec{0}$ в решітці $\mathcal{L}(B)$, а вектор $\vec{v} \in \mathcal{L}(B)$, який є розв'язком задачі SVP, буде відповідати другому найменшому значенню.

В своїй роботі ті ж автори також запропонували модифікацію [1, додаток A] кінцевого екземляру QUBO, яка додає «штрафний» доданок P до суми, якщо всі змінні $\hat{x}_{ij}$ дорівнюють нулю. Із цією модифікацією таке зведення вже є повністю коректним, проте вона вимагає введення додаткових $2n$ двійкових змінних до задачі QUBO.

Для задачі QUBO відоме [14, Розділ 1.1] стандартне взаємооднозначне співставлення із задачею пошуку основого стану гамільтоніану Ізінга. Для зручності, сформулюємо це як окреме твердження:

Твердження 2.1. *Існує зведення задачі QUBO з m змінними x_i, яка задана матрицею Q, до пошуку основного стану гамільтоніану Ізінга $\hat{H}$ в гільбертовому просторі $\mathcal{H}_2^{\otimes m}$. Гамільтоніан будується заміною змінних x_i у виразі (2.1) для задачі QUBO на однокубітні унітарні перетворення Паулі:*

$$x_i \longrightarrow \frac{1}{2}\left(Z_i + \mathbb{1}\right),$$

де Z_i – це Z-оператор Паулі, що діє на i-ий кубіт.

Тобто сам гамільтоніан має вигляд:

$$\begin{aligned}\hat{H} &= \sum_{1\leqslant i,j\leqslant n} Q_{ij}\frac{1}{2}\left(Z_i + \mathbb{1}\right)\frac{1}{2}\left(Z_j + \mathbb{1}\right) = \\ &= \frac{1}{4}\sum_{1\leqslant i,j\leqslant n} Q_{ij}\left(Z_iZ_j + Z_i + Z_j + \mathbb{1}\right) = \\ &= \frac{1}{4}\Bigg(\sum_{i\neq j} Q_{ij}Z_iZ_j + \sum_i L_iZ_i + C\cdot\mathbb{1}\Bigg)\end{aligned} \tag{2.4}$$

де $L_i = \sum_j (Q_{ij} + Q_{ji})$, а $C = \left(\sum_{i\neq j} Q_{ij} + 2\sum_i Q_{ii}\right)$.

Надалі будемо позначати гамільтоніан, отриманий в результаті описаного вище зведення, як $\hat{H}_{2P}$. Підсумовуючи, схематично наявне зведення, описане вище, виглядає як наведено на схемі 2.1.

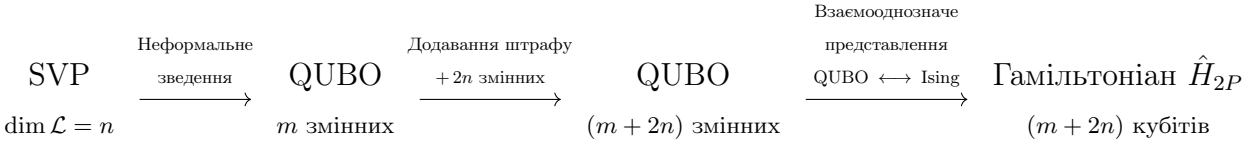


Рисунок 2.1 – Схема зведення SVP до пошуку основного стану гамільтоніану

2.2 Оцінка кількості кубітів для зведення

Для задачі SVP важливо розуміти, в яких межах знаходиться найкоротшний вектор $\vec{v}_s$, а саме потрібно знати верхню межу A, таку що $\|\vec{v}_s\| \leqslant A$.

Таке обмеження описується [19] теоремою Мінковського:

Теорема 2.1 (Мінковський)**.** *Для базису* $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ *довжина найменшого вектора* $\vec{v}_s$ *решітки* $\mathcal{L}(B)$ *обмежена:*

$$\|\vec{v}_s\| \leqslant \sqrt{n} \cdot |\det B|^{1/n}.$$

Також для оцінки складності зведення задачі SVP до задачі QUBO (і подальшої оцінки кількості кубітів в квантовій схемі), необхідно розуміти обмеження $|x_i| \leqslant a_i$ на коефіцієнти в розкладі найкоротшого вектора в заданому базисі. Наведемо важливу лему із [1, розділ 4], яка дозволяє отримати такі обмеження.

Означення 2.1. Дуальною решіткою решітки $\mathcal{L}(B)$ із базисом B називають решітку $\hat{\mathcal{L}}(\hat{B})$ із, так званим, **дуальним базисом** $\hat{B} = (B^T)^{-1}$.

Означення 2.2. Коефіцієнтом неортогональності базису $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ називають співвідношення:

$$\delta(B) = \frac{\prod_{i=1}^{n} \|\vec{b_i}\|}{|\det B|}.$$

Варто звернути увагу, що коефіцієнт неортогональності $\delta(B)$ завжди не менше 1, і досягає мінімального значення $\delta(B) = 1$, лише коли базис B є ортогональним.

Лема 2.1. *Для базису* $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ *решітки* $\mathcal{L}(B)$ *та для вектора* $\vec{v} = x_1\vec{b_1} + x_2\vec{b_2} + \ldots + x_n\vec{b_n}$ *виконується:*

$$\|\vec{v}\| \leqslant A \quad \Longrightarrow \quad |x_i| \leqslant A \cdot \|\hat{d}_i\| \;\; \textit{для } i = 1, \ldots, n\,,$$

де $\hat{d}_i$ – це вектор, що заданий i-им стовпчиком матриці $\hat{B}$ дуального базису для B.

Поєднавши теорему 2.1 Мінковського та вищенаведену лему 2.1, можна сформулювати необхідний нам наслідок:

Наслідок 2.1. *Для решітки $\mathcal{L}(B)$, що задана базисом $B = \left\{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\right\}$, коефіцієнти x_i в розкладі найменшого вектора $\vec{v}_s$ в базисі B можна обмежити:*

$$|x_i| \leqslant \sqrt{n} \cdot |\det B|^{1/n} \cdot \|\hat{d}_i\| \,,$$

де $\hat{d}_i$ – це вектор, що заданий i-им стовпчиком матриці $\hat{B}$ дуального базису для B.

За допомогою наслідку 2.1 можна отримати обмеження на зверху на кількість кубітів для гамільтоніану $\hat{H}_{2P}$:

Наслідок 2.2. *Нехай для задачі SVP на решітці $\mathcal{L}(B)$, що задана базисом $B = \left\{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\right\}$, гамільтоніан $\hat{H}_{2P}$ над гільбертовим простором $\mathcal{H}_2^{\otimes N}$ отримано в результаті зведення, описаного в розділі 2.1. Тоді кількість кубітів N, яка необхідна для опису простору станів $\mathcal{H}_2^{\otimes N}$ обмежена:*

$$N \leqslant 4n + \frac{1}{2} n \log_2 n + \log_2(\delta(\hat{B})) \,,$$

де $\delta(\hat{B})$ – коефіцієнт неортогональності для дуального до B базису $\hat{B}$.

Доведення. Кількість кубітів N, як видно із опису зведення, дорівнює кількості двійкових змінних m в задачі QUBO, яка отримується як проміжний крок зведення і ще $2n$ додаткових змінних із модифікації $\hat{H}_{2P}$. Тому оцінка зводиться до оцінки кількості змінних у відповідній задачі QUBO.

Із виразу 2.3 маємо, що кількість змінних дорівнює m:

$$m = \sum_{i=1}^{n} \lceil \log_2(2a_i) \rceil \,,$$

де a_i – це таке обмеження, що $|x_i| \leqslant a_i$.

Згідно з наслідком 2.1, можна розглядати:

$$a_i = \sqrt{n} \cdot |\det B|^{1/n} \cdot \|\hat{d}_i\| .$$

Тобто:

$$\begin{aligned} m &\leqslant \sum_{i=1}^{n} \lceil \log_2(2a_i) \rceil = \\ &= \sum_{i=1}^{n} \lceil \log_2(2\sqrt{n} \cdot |\det B|^{1/n} \cdot \|\hat{d}_i\|) \rceil \leqslant \\ &\leqslant 2n + \log_2 \left(\prod_{i=1}^{n} \sqrt{n} \cdot |\det B|^{1/n} \cdot \|\hat{d}_i\| \right) = \\ &= 2n + \log_2 \left(\sqrt{n}^n \cdot |\det B| \cdot \prod_{i=1}^{n} \|\hat{d}_i\| \right) = \\ &= 2n + \frac{1}{2} n \log_2 n + \log_2(\delta(\hat{B})) \end{aligned}$$

Звідси маємо, що $N = m + 2n = 4n + \frac{1}{2} n \log_2 n + \log_2(\delta(\hat{B}))$ □

Як видно із оцінки в наслідку 2.2, необхідна кількість кубітів залежить не тільки від розмірності решітки n, а й від того, наскільки ортогональним є дуальний базис $\hat{B}$.

Зауваження. Легко побачити, що величина $\log_2(\delta(\hat{B}))$ є необмеженою зверху. Таким чином, описані вище міркування не дають змоги побудувати обмежену зверху оцінку необхідної кількості кубітів для фіксованої розмірності решітки n.

В результаті попереднього аналізу не вдалось виявити якогось зв'язку між коефіцієнтами неортогональності базису B та дуального до нього $\hat{B}$, тому це питання залишається за межами цього дослідження.

2.3 Оцінка складності квантової схеми для такого зведення

Маючи загальний вигляд (2.4) гамільтоніану Ізінга $\hat{H}$, можна побудувати оцінку кількості вентилів, необхідних для реалізації квантової схеми алгоритму QAOA.

Під час оцінки складності квантової схеми доречно розділяти кількість 1-кубітних вентилів та кількость 2-кубітних (контрольованих) вентилів, оскільки на практиці складність та якість фізичної реалізації цих типів вентилів сильно відрізняється.

Сформулюємо і доведемо наступну теорему:

Теорема 2.2. *Квантова схема для алгоритму QAOA для гамільтоніану Ізінга $\hat{H}_P$ в гільбертовому просторі $\mathcal{H}^{\otimes m}$, який побудований відповідно до твердження 2.1, містить не більше ніж $p \cdot (m^2 - m)$ вентилів* `CNOT` *та $\frac{p}{2}m^2 + (\frac{7}{5}p + 1)m$ однокубітних вентилів.*

Доведення.

Згідно з побудовою алгоритму QAOA 1.4, квантова схема містить такі складові:

– **Підготовка початкового стану $H^{\otimes m}$:**

$$m \quad \text{однокубітних вентилів Адамара} \tag{1}$$

Далі йдуть p рівнів, кожен з яких складається із двох унітарних перетворень $U_M(\beta)$ та $U_P(\gamma)$ вигляду:

$$U_M(\beta) = e^{-i\beta\hat{H}_M},$$
$$U_P(\gamma) = e^{-i\gamma\hat{H}_P}.$$

– **Унітарне перетворення $U_M(\beta) = e^{-i\beta\hat{H}_M}$:**

Оскільки $\hat{H}_M$ має вигляд:

$$\hat{H}_M = -\sum_{i=1}^{m} X_i,$$

і, при цьому, оператори X_i та X_j комутують для $i \neq j$, то відповідне унітарне перетворення $U_M(\beta)$ можна описати як:

$$U_M(\beta) = e^{-i\beta\hat{H}_M} = e^{=i\beta\sum_{i=1}^{m} X_i} = \prod_{i=1}^{m} e^{i\beta X_i}.$$

Із вигляду квантової схеми для перетворення $e^{i\beta X_i}$, який наведено на рисунку 2.2(в), робимо висновок, що для перетворень $U_M(\beta_i)$ для усіх $i = 1, \ldots p$, необхідно буде:

$$p \cdot 3m \quad \text{однокубітних вентилів} \tag{2}$$

Аналогічні міркування можна застосувати і для $U_P(\gamma_i)$:

– **Унітарне перетворення $U_P(\gamma) = e^{-i\gamma\hat{H}_P}$:**

Із вигляду гамільтоніану Ізінга $\hat{H}_P$ 2.4 можна побачити, що відповідне унітарне перетворення $U_P(\gamma)$ має вигляд:

$$U_P(\gamma) = e^{-i\gamma\hat{H}_P} = e^{-i\gamma\left(\frac{1}{4}\sum_{i\neq j} Q_{ij}Z_iZ_j + \frac{1}{4}\sum_i L_iZ_i + \frac{C}{4}\mathbb{1}\right)} \tag{$*$}$$

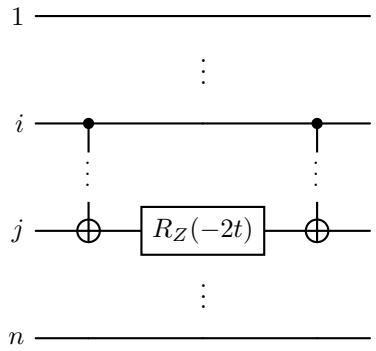
Враховуючи, що будь-які два оператора Z_iZ_j, Z_k та $\mathbb{1}$ в сумі в степені експоненти комутують між собою для довільних індексів i, j та k, можна розписати $(*)$ як:

$$\prod_{i\neq j} e^{-\frac{1}{4}itQ_{ij}Z_iZ_j} \cdot \prod_i e^{-\frac{1}{4}itL_iZ_i} \cdot e^{-\frac{C}{4}it\mathbb{1}}$$

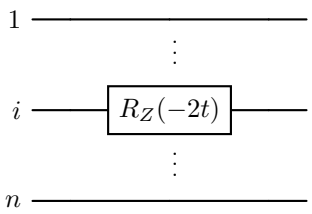
Тобто побудову та аналіз складності схеми для перетворення $U_P(\gamma)$ можна звести до перетворень вигляду $e^{-itZ_iZ_j}$ та e^{-itZ_k}, схеми для яких наведено на рисунку 2.2, (а) та (б) відповідно.

Отже для унітарних перетворень $U_P(\gamma_i)$ для усіх $i = 1, \ldots p$ в квантовій схемі QAOA необхідно:

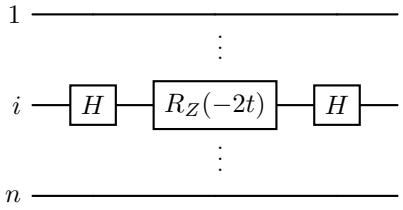
$$\begin{aligned} &p \cdot \frac{1}{2}(m^2 + m) \quad \text{однокубітних вентилів} \\ &p \cdot (m^2 - m) \quad \text{двокубітних вентилів } \texttt{CNOT} \end{aligned} \tag{3}$$



(а) Перетворення $e^{-itZ_i \otimes Z_j}$



(б) Перетворення e^{-itZ_i}



(в) Перетворення e^{-itX_i}

Рисунок 2.2 – Квантові схеми для унітарних перетворень вигляду e^{-itP}, де P – оператор Паулі певного вигляду.

Додавши разом оцінки 1, 2 та 3, отримуємо:

$$\begin{aligned} & p \cdot \left(\frac{1}{2}m^2 + \frac{7}{5}m\right) + m && \text{однокубітних вентилів} \\ & p \cdot (m^2 - m) && \text{двокубітних вентилів } \texttt{CNOT} \end{aligned}$$

для всієї квантової схеми QAOA.

Побудова квантових схем для відповідних унітарних перетворень на рисунку 2.2 наведена в роботі [24]. □

Із твердження цієї теореми видно, що кількість вентилів в квантовій схемі алгоритму QAOA безпосередньо залежить від кількісті кубітів гамільтоніану $\hat{H}_P$ та, очевидно, лінійно зростає зі зростанням параметра p, який визначає кількість рівнів в алгоритмі.

Для гамільтоніану $\hat{H}_{2P}$, побудованого в результаті зведення, описаного вище, підставивши оцінку для кількості кубітів 2.2 в твердження теореми 2.2, отримуємо таку оцінку:

Наслідок 2.3. *Нехай задача SVP задана базисом* $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ *на* n*-вимірній решітці* $\mathcal{L}(B)$*, а відповідний гамільтоніан* $\hat{H}_{2P}$ *побудовано в результаті зведення задачі SVP, тоді квантова схеми алгоритму QAOA для цього гамільтоніану містить не більше ніж:*

$$4n + \frac{1}{2}n\log_2 n + \Delta \qquad \textit{кубітів,}$$

$$p \cdot \left[\frac{1}{8}n^2(\log_2 n + 8)^2 + \frac{1}{2}\Delta n(\log_2 n + 8) + \frac{1}{10}(5\Delta^2 + 14\Delta)\right] \qquad \textit{однокубітних,}$$

$$p \cdot \left[\frac{1}{4}n^2(\log_2 n + 8)^2 + n\Delta(\log_2 n + 8) + \Delta^2\right] \qquad \textit{вентилів } \texttt{CNOT},$$

де $\Delta = \log_2 \delta(\hat{B})$.

Із цього наслідку видно, що збільшення кількості кубітів, окрім самого собою збільшення розміру квантової схеми, також ще і збільшує кількість вентилів в схемі.

Варто відзначити, що для NISQ-комп'ютерів будь-яке додаткове збільшення складності квантової схеми, є набагато більш суттєвим, ніж для класичних комп'ютерів, навіть якщо асимптотично складність залишається в межах того ж класу.

Висновки до розділу 2

В другому розділі детально описано наявний принцип побудови зведення задачі SVP до задачі пошуку основного стану гамільтоніану, також наведено необхідні для цього задачі та твердження. Оскільки такий принцип побудови зведення не враховує обмеження на ненульову довжину вектора, то як формально коректне зведення розглядається модифікація, що описана в роботі [1].

Для наявного зведення побудовано точну оцінку кількості кубітів для опису відповідного гамільтоніану. Наведено теореми, твердження і поняття із теорії решіток, які необхідні для побудови цієї оцінки.

Детальніше розглянуто вплив рівня ортогональності заданого базису в задачі SVP на ефективність такого зведення.

Також, на основі оцінки для кількості кубітів, побудовано точну оцінку кількості вентилів квантової схеми алгоритму QAOA для гамільтоніану, що відповідає описаному зведенню. Ця оцінка окремо описує кількості однокубітних вентилів та двокубітних вентилів `CNOT`.

3 НОВЕ ЗВЕДЕННЯ ЗАДАЧІ SVP ДО ПОШУКУ ОСНОВНОГО СТАНУ ГАМІЛЬТОНІАНУ. ПОРІВНЯЛЬНИЙ АНАЛІЗ ЗАПРОПОНОВАНОГО ЗВЕДЕННЯ ІЗ НАЯВНИМ

В цьому розділі запропоновано нове модифіковане зведення задачі SVP до задачі пошуку основного стану гамільтоніану, яке грунтується на вже наявному принципі; доведена його коректність.

Для запропонованого зведення побудовані аналогічні до наявного зведення оцінки для кількісті кубітів та складності квантової схеми алгоритму QAOA.

Також проведено порівняльний аналіз ефективності запропонованого та наявного зведень на основі практично значимих екземплярів задачі SVP.

3.1 Нове зведення задачі SVP до пошуку основного стану гамільтоніану та доведення його коректності

Замість того, щоб модифікувати екземпляр задачі QUBO після зведення SVP до QUBO, як це запропоновано в роботі [1], щоб отримати гамільтоніан $\hat{H}_{2P}$, можна натомість модифікувати вже кінцевий гамільтоніан Ізінга. Опишемо такий метод.

Нехай $\hat{H}_P$ – гамільтоніан Ізінга, що побудований в результаті таких дій: для початкової задачі SVP на решітці $\mathcal{L}(B)$ екземпляр задачі QUBO будується, не враховуючи обмеження на ненульову довжину вектора, в результаті «неформального» зведення, як описано в розділі 2; потім із отриманої задачі QUBO будується гамільтоніан Ізінга згідно із твердженням 2.1.

Позначимо λ_i – i-те найменше значення серед власних чисел

гамільтоніану $\hat{H}_P$. Також звернемо увагу, що λ_i дорівнює квадрату i-го послідовного мінімуму $\lambda_i^2(\mathcal{L})$ решітки $\mathcal{L}(B)$ із якої було отримано цей гамільтоніан.

Тоді розглянемо новий оператор $\hat{H}_{3P}$, який визначимо таким чином:

$$\hat{H}_{3P} = \hat{H}_P + \alpha \left|\mathbf{0}\right\rangle\left\langle\mathbf{0}\right|, \tag{3.1}$$

де:

– $\left|\mathbf{0}\right\rangle = \left|00\ldots0\right\rangle = \left|0\right\rangle^{\otimes m}$ – це вектор-стан, який також відповідає нуль-вектору $\vec{0}$ решітки $\mathcal{L}(B)$,

– $\left|\mathbf{0}\right\rangle\left\langle\mathbf{0}\right|$ – оператор проектування на $\left|\mathbf{0}\right\rangle$,

– α – це дійсний коефіцієнт штрафування, причому $\alpha > |\lambda_0 - \lambda_1|$.

Покажемо, що оператор $\hat{H}_{3P}$ є гамільтоніаном та його основний стан відповідає розв'язку відповідної задачі SVP.

Теорема 3.1. *Нехай $\left|\phi\right\rangle$ – власний вектор гамільтоніану $\hat{H}_P$, який відповідає найкоротшому ненульовому вектору решітки $\mathcal{L}(B)$. Тоді оператор $\hat{H}_{3P}$, визначений як 3.1, також є ермітовим оператором, а вектор $\left|\phi\right\rangle$ є його основним станом.*

Доведення.

– Оператор проектування $\alpha \left|\mathbf{0}\right\rangle\left\langle\mathbf{0}\right|$ є ермітовим, тому $\hat{H}_{3P}$ – також ермітовий, як сума двох ермітових операторів, тобто $\hat{H}_{3P}$ є гамільтоніаном.

– Оскільки всі власні вектори оператора $\hat{H}_P$ є станами обчислювального базису, то власне значення $\hat{H}_{3P}$, що відповідає вектору $\left|\mathbf{0}\right\rangle$ дорівнює $(\lambda_0 + \alpha)$ і є більшим за λ_1:

$$\hat{H}_{3P}\left|\mathbf{0}\right\rangle = (\hat{H}_P + \alpha \left|\mathbf{0}\right\rangle\left\langle\mathbf{0}\right|)\left|\mathbf{0}\right\rangle = \lambda_0 \left|\mathbf{0}\right\rangle + \alpha \left|\mathbf{0}\right\rangle.$$

А довільне інше власне значення, що відповідає власному вектору $\left|\varphi\right\rangle$ залишиться без змін, оскільки:

$$\hat{H}_{3P}\left|\varphi\right\rangle = \lambda_\varphi \left|\varphi\right\rangle + \alpha \underbrace{\left\langle\mathbf{0}|\varphi\right\rangle}_{=0} \left|\varphi\right\rangle = \lambda_\varphi \left|\phi\right\rangle,$$

тобто λ_1 буде найменшим власним значенням гамільтоніану $\hat{H}_{3P}$, а отже і $|\phi\rangle$ – його основним станом.

□

Зауваження. На практиці, оскільки $|\lambda_0 - \lambda_1| = |0 - \lambda_1| = |\lambda_1|$, то коефіцієнт α можна оцінити за теоремою Мінковського 2.1:

$$\alpha := \sqrt{n} \cdot |\det B|^{1/n}$$

Також сформулюємо лему, яка допоможе зрозуміти, як буде виглядати квантова схема в алгоритмі QAOA для гамільтоніану $\hat{H}_{3P}$, що допоможе і в подальшому аналізі складності такої схеми.

Лема 3.1. *Для гамільтоніану $\hat{H}_{3P}$, побудованого згідно з виразом 3.1, відповідне унітарне перетворення $e^{-it\hat{H}_{3P}}$ дорівнює добутку двох перетворень:*

$$e^{-it\hat{H}_{3P}} = e^{-it\hat{H}_P} \cdot e^{-it\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}$$

Доведення. Оператор-проектор $|\mathbf{0}\rangle\langle\mathbf{0}|$ комутує з довільним іншим оператором P, тому виконується властивість матричної експоненти:

$$e^{P+|\mathbf{0}\rangle\langle\mathbf{0}|} = e^P \cdot e^{|\mathbf{0}\rangle\langle\mathbf{0}|}.$$

А отже оператор $e^{-it\hat{H}_{3P}}$ можна розписати як:

$$e^{-it\hat{H}_{3P}} = e^{-it(\hat{H}_P+\alpha|\mathbf{0}\rangle\langle\mathbf{0}|)} = e^{-it\hat{H}_P} \cdot e^{-it\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}.$$

□

Це означає, що в квантовій схемі алгоритму QAOA для гамільтоніану $\hat{H}_{3P}$ перетоврення $U_P(\gamma)$ можна реалізовувати як два окремі послідовні перетворення: $e^{-i\gamma\hat{H}_P}$, побудова якого була описана в розділі 2, та $e^{-i\gamma\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}$, побудову якого буде описано далі.

Загальний схематичний вигляд запропонованого зведення зображено на рисунку 3.1.

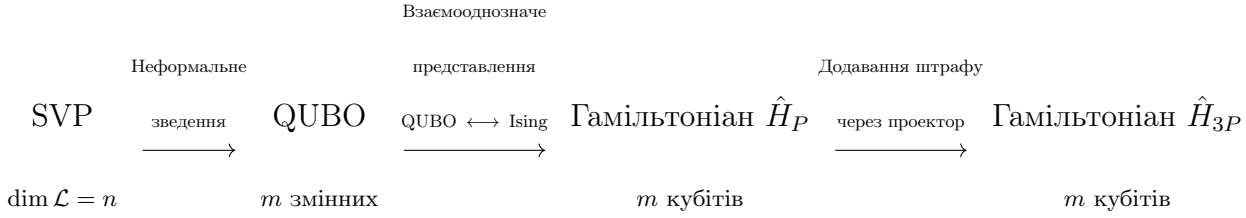


Рисунок 3.1 – Схема зведення SVP до пошуку основного стану гамільтоніану для нового зведення

3.2 Аналіз складності запропонованого зведення

Для наведеного гамільтоніану $\hat{H}_{3P}$ побудуємо оцінки для кількості кубітів та складності квантової схеми алгоритму QAOA, аналогічно до оцінок, побудованих для $\hat{H}_{2P}$ в розділі 2.

Лема 3.2. *Нехай для задачі SVP на решітці $\mathcal{L}(B)$, що задана базисом $B = \left\{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\right\}$, гамільтоніан $\hat{H}_{3P}$ над гільбертовим простором $\mathcal{H}_2^{\otimes N}$ побудовано як у виразі 3.1. Тоді кількість кубітів N, яка необхідна для опису простору станів $\mathcal{H}_2^{\otimes N}$ обмежена:*

$$N \leqslant 2n + \frac{1}{2} n \log_2 n + \log_2(\delta(\hat{B}))\,,$$

де $\delta(\hat{B})$ – коефіцієнт неортогональності для дуального до B базису $\hat{B}$.

Доведення. Кількість кубітів гамільтоніану $\hat{H}_{3P}$ дорівнює кількості кубітів для гамільоніану $\hat{H}_P$, оскільки додавання оператора-проектора $\alpha\,|\mathbf{0}\rangle\langle\mathbf{0}|$ не впливає на це.

Далі доведення повністю ідентичне до доведення наслідку 2.2, тільки без додавання $2n$ кубітів, оскільки гамільтоніан $\hat{H}_P$ будується із екземпляру QUBO з m змінними. □

Використовуючи оцінку кількості кубітів із леми 3.2, теорему 2.2 та наявні результати щодо побудови схеми перетворення $e^{-it\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}$, сформулюємо та доведемо остаточну оцінку складності у вигляді теореми:

Теорема 3.2. *Нехай задача SVP задана базисом* $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ *на* n*-вимірній решітці* $\mathcal{L}(B)$*, а відповідний гамільтоніан* $\hat{H}_{2P}$ *побудовано в результаті зведення задачі SVP, тоді квантова схеми алгоритму QAOA для цього гамільтоніану містить:*

$$2n + \frac{1}{2}n\log_2 n + \Delta \qquad \textit{кубітів,}$$

$$p \cdot \left[\frac{1}{8}n^2(\log_2 n + 8)^2 + \frac{1}{2}\Delta n(\log_2 n + 8) + \frac{1}{10}(5\Delta^2 + 94\Delta)\right] \qquad \textit{однокубітних,}$$

$$p \cdot \left[\frac{1}{4}n^2(\log_2 n + 8)^2 + n\Delta(\log_2 n + 8) + \Delta^2\right] \qquad \textit{вентилів } \texttt{CNOT},$$

де $\Delta = \log_2 \delta(\hat{B})$.

Доведення.

Розглянемо унітарне перетворення $U_P(\gamma)$ в квантовій схемі QAOA для гамільтоніану $\hat{H}_{3P}$:

$$U_P(\gamma) = e^{-i\gamma\hat{H}_{3P}} = e^{-i\gamma\hat{H}_P} \cdot e^{-i\gamma\alpha|\mathbf{0}\rangle\langle\mathbf{0}|} \text{ – згідно з лемою 3.1.}$$

Із цього видно, що квантова схема алгоритму для гамільтоніана $\hat{H}_{3P}$ буде відрізнятись від схеми для гамільтоніану $\hat{H}_P$ лише додатковим перетворенням $e^{-i\gamma\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}$ на кожному рівні. Тобто кількість вентилів N буде дорівнювати:

$$N = N_1 + pN_2\,,$$

де N_1 – кількість вентилів в квантовій схемі алгоритму QAOA для $\hat{H}_P$;

N_2 – кількість вентилів, необхідна для реалізації $e^{-i\gamma\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}$.

– N_1 можна отримати, підставивши оцінку для кількості кубітів із леми 3.2 в твердження теореми2.2.

– N_2 оцінимо наступним чином:

Перетворення $e^{-i\gamma\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}$ задає, так званий, виколотий

мультиконтрольований фазовий зсув для одного кубіту:

$$e^{-i\gamma\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}\,|\psi\rangle = \begin{cases} e^{-i\gamma\alpha}\,|\psi\rangle\,, & \text{якщо } |\psi\rangle = |\mathbf{0}\rangle \\ |\psi\rangle\,, & \text{інакше} \end{cases}.$$

Схему для такого перетворення можна отримати із $(m-1)$-контрольованого Z-обертання та певних однокубітних унітарних перетворень. Схема і побудова наведені на рисунку 3.2. При цьому, згідно з оцінками, наведеними у роботах [23, 10] про ефективну декомпозицію багатокубітних вентилів Тоффолі та інших мультиконтрольованих унітарних вентилів, перетворення $(m-1)$-контрольованого Z-обертання можна реалізувати, використовуючи $8m$ вентилів `CNOT` та $8m$ однокубітних унітарних вентилів.

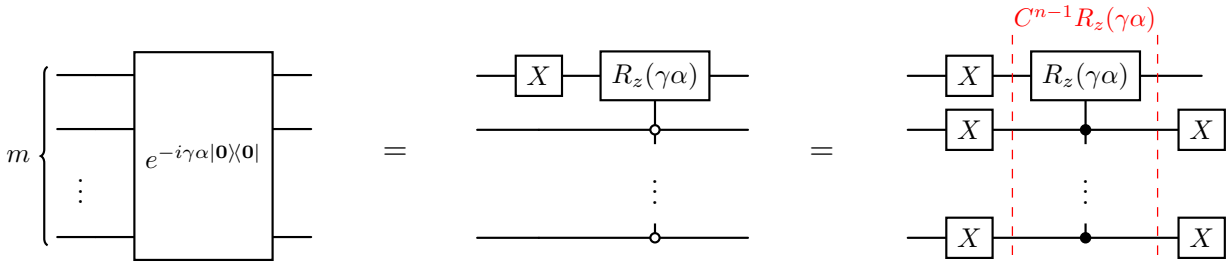


Рисунок 3.2 – Квантова схема для перетворення $e^{-i\gamma\alpha|\mathbf{0}\rangle\langle\mathbf{0}|}$

Таким чином отримуємо оцінки:

$$p \cdot \left[\frac{1}{8}n^2(\log_2 n + 8)^2 + \frac{1}{2}\Delta n(\log_2 n + 8) + \frac{1}{10}(5\Delta^2 + 94\Delta)\right] \quad \text{однокубітних,}$$

$$p \cdot \left[\frac{1}{4}n^2(\log_2 n + 8)^2 + n\Delta(\log_2 n + 8) + \Delta^2\right] \quad \text{вентилів } \texttt{CNOT},$$

□

В результаті зведення згідно із запропонованим методом, квантова схема QAOA для гамільтоніану $\hat{H}_{3P}$ має на $2n$ менше кубітів, де n – це розмірність початкової решітки.

Згідно з побудованими оцінками, кількість вентилів T квадратично залежить від кількості кубітів m, тобто $T = \mathcal{O}(m)$. Відповідно, порівняно з гамільтоніаном $\hat{H}_{2P}$, очікується, що зменшення кількості кубітів для гамільтоніану $\hat{H}_{3P}$ має мати суттєво більший позитивний ефект, ніж додавання лінійної кількості вентилів – негативний.

3.3 Порівняння складності запропонованого зведення із наявним для практично значимих решіток

Важливо також за допомогою побудованих оцінок дослідити, які обчислювальні ресурси квантових комп'ютерів необхідні для застосування алгоритму QAOA до практичних екземплярів SVP.

Інтернет ресурс «Lattice Challange» має публічний список [18] обчислювально складних екземплярів задачі SVP, для якого дослідники змагаються в тому, хто зможе обчислити кращий наближний розв'язок для більших розмірностей решіток.

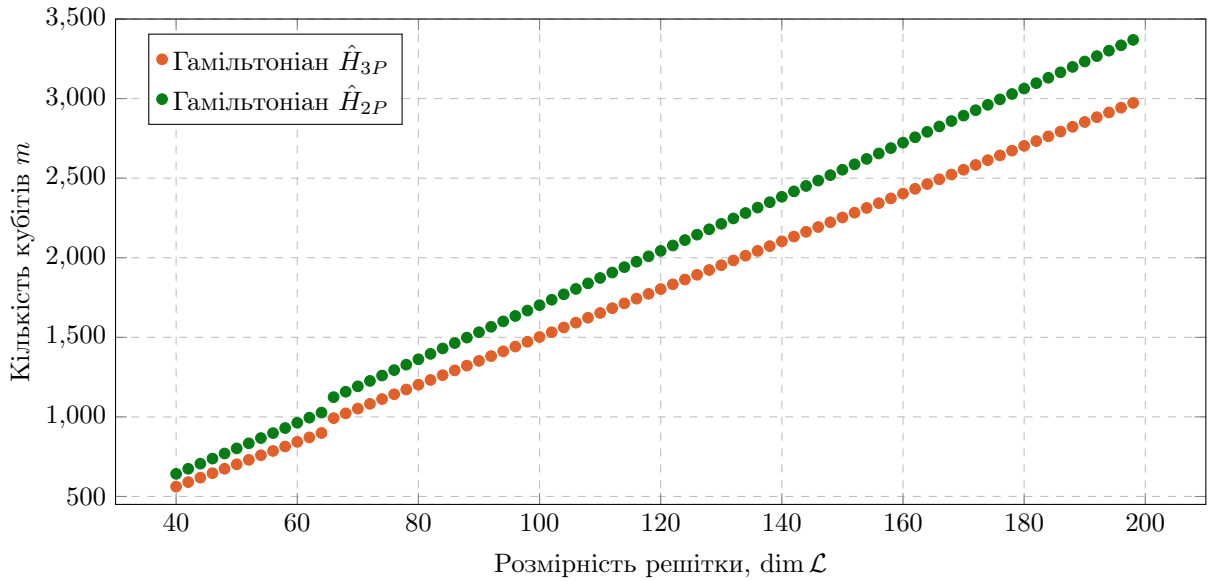


Рисунок 3.3 – Кількість кубітів в квантовій схемі QAOA

Базиси $B = \{\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n}\}$ решіток $\mathcal{L}(B)$ для задач SVP із цього списку є дуже неортогональними, тобто:

$$\forall \vec{b_i}, \vec{b_j} \ : \ \langle \vec{b_i}, \vec{b_j} \rangle \longrightarrow \|\vec{b_i}\| \cdot \|\vec{b_j}\| \, .$$

А сама довжина $\|\vec{b_i}\|$ базисних векторів для всіх $i = 1, \ldots n$ є дуже великою.

Вважається, що такі властивості базису в задачі SVP роблять її обчислювально складною. Тому такі екземпляри задачі SVP є важливими і цікавими з точки зору постквантової криптографії.

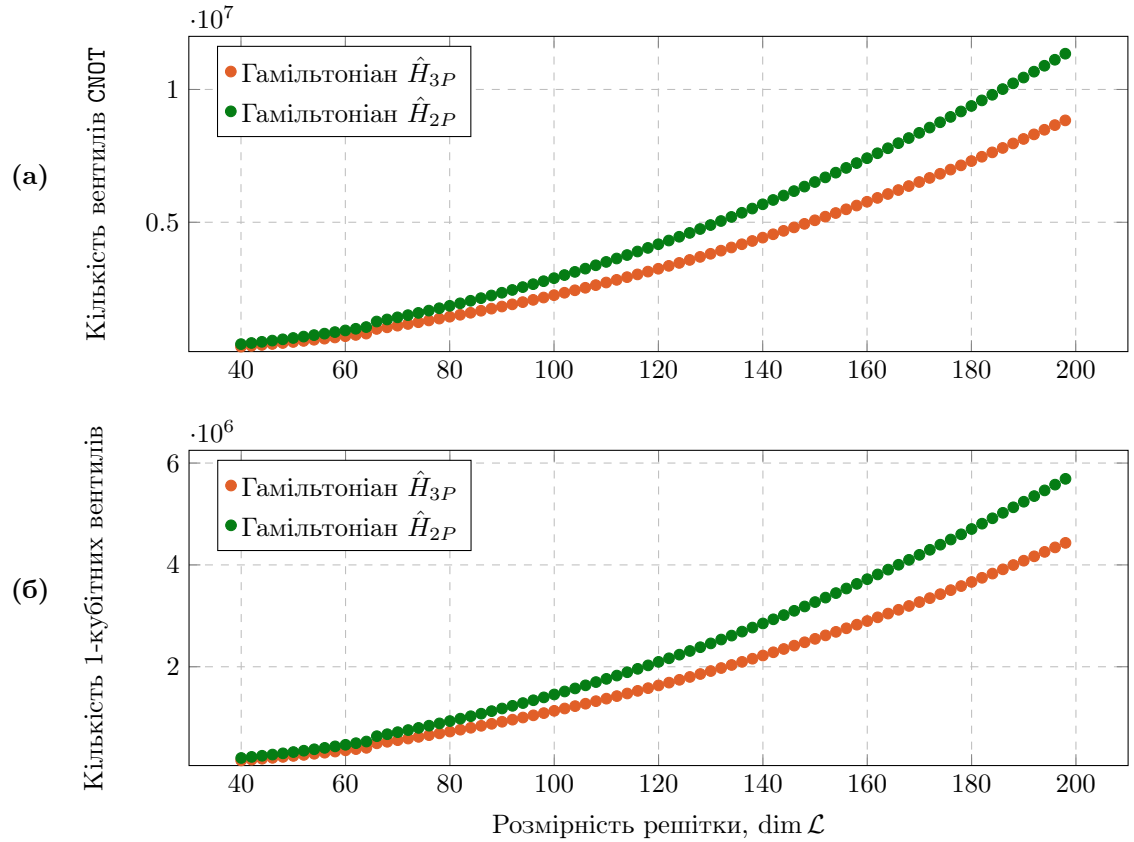


Рисунок 3.4 – Кількість вентилів в квантовій схемі QAOA

Було обчислено оцінки складності квантової схеми алгоритму QAOA для екземплярів задачі SVP зі списку [18] для решіток розмірностей $\dim \mathcal{L} = 40, \ldots, 200$. Залежність кількості кубітів в квантовій схемі від розмірності решітки продемонстрована на рисунку 3.3. Залежність кількість вентилів (окремо для одно- та

двокубітних) в схемі від розмірності решітки – на рисунку 3.4 відповідно.

Як видно із результатів на графіках, лінійне (від кількості кубітів) збільшення кількості вентилів є не таким суттєвим, як квадратичне зменшення кількості вентилів від зменшення кількості кубітів. Що підтверджує очікуваний результат.

Висновки до розділу 3

В третьому розділі запропоновано нове зведення, яке базується на відомому принципі, проте опис кінцевого гамільтоніану вимагає меншої кількості кубітів, доведено коректність такого зведення.

Аналогічно до оцінок, побудованих в розділі 2 для наявного зведення, побудовано і обгрунтовано оцінки для запропонованого зведення. В процесі побудови таких оцінок також описано особливості побудови квантової схеми алгоритму QAOA для гамільтоніану, що отримано в результаті такого зведення.

Також проведено порівняння складностей квантових схем для двох описаних методів зведення на основі практично складних екземплярів задачі SVP для решіток великих розмірностей. В результаті порівняння для реальних задач видно, що запропонований метод зведення призводить до суттєвого покращення складності квантової схеми QAOA, хоч побудовані оцінки для цих методів і є асимптотично еквівалентні.

ВИСНОВКИ

В роботі проведено дослідження застосування сучасного варіаційного квантового алгоритму QAOA до задачі SVP, – однієї із основних задач на решітках для постквантової криптографії.

1) Проведено огляд наявних результатів по темі застосування квантових варіаційних алгоритмів до задачі SVP, детально було досліджено принцип побудови зведення задачі SVP до задачі пошуку основного стану гамільтоніану, яке необхідне для застосування таких алгоритмів. Основним джерелом, в якому наведені необхідні наявні результати та інші джерела, стала робота [1].

2) В роботі також наведено необхідні теоретичні відомості:

– Основні властивості решіток, досліджувані складні задачі на решітках та пов'язані з ними поняття, детальніше розглянуто задачу SVP.

– Необхідні принципи квантової механіки та квантових обчислень, зокрема адіабатичну теорему та пов'язаний із нею принцип квантових обчислень.

– Історію розвитку варіаційних квантових алгоритмів та ключові алгоритми, зокрема детально описано досліджуваний алгоритм квантової наближеної комбінаторної оптимізації – QAOA.

3) Описано відомий принцип побудови зведення задачі SVP до задачі пошуку основного стану гамільтоніану через проміжне зведення до задачі QUBO. Також розглянуто конкретну побудову такого зведення, запропоновану в роботі [1], яке вимагає додаткового збільшення кількості кубітів для кінцевого гамільтоніану.

4) Для наявного зведення побудовано та доведено оцінку кількості кубутів, необхідних для опису кінцевого гамільтоніану. Із цієї оцінки, в свою чергу, побудовано точні оцінки складності (кількості кубітів, однокубітних вентилів та вентилів `CNOT`) квантової схеми алгоритму QAOA при застосуванні такого зведення.

5) В результаті дослідження запропоноване нове зведення задачі SVP до задачі пошуку основного стану гамільтоніану, яке побудовано на основі того ж відомого принципу, проте не вимагає збільшення кількості кубітів, на відміну від зведення, запропонованого в роботі [1].

6) Для запропонованого зведення побудовано аналогічні оцінки складності квантової схеми QAOA: кількості кубітів, однокубітних вентилів та вентилів `CNOT`.

7) Проведено порівняння запропонованого та відомого зведень на основі екземплярів задачі SVP на решітках практичних розмірів, із базисами зі спеціальними властивостями, що робить їх обчислювально складними і відповідно практично значимими з точки зору постквантової криптографії. В результаті такого порівняння продемонстровано, що нове запропоноване зведення дозволяє помітно покращити складність квантової схеми в алгоритмі QAOA.

У подальшому планується продовжувати дослідження на цю тему в декількох напрямках:

– побудова більш змістовних оцінок складності квантової схеми. Зокрема оцінка, так званої, глибини квантової схеми, що є більш показовою метрикою складності для квантових обчислень;

– перевірка побудованих оцінок складності на практиці при реальному застосуванні алгоритму QAOA до задач SVP невеликих розмірів, використовуючи класичні симулятори квантових обчислень, або справжні квантові комп'ютері із віддаленим доступом;

– застосування модифікацій алгоритму QAOA до задачі SVP та його аналіз;

– застосування алгоритму QAOA до інших, важливих в постквантовій криптографії, складних задач на решітках. Зокрема побудова зведення таких задач до задачі пошуку основного стану гамільтоніану.

ПЕРЕЛІК ПОСИЛАНЬ

[1] Martin Albrecht та ін. «Variational quantum solutions to the Shortest Vector Problem». В: *Quantum* 7 (2023), с. 933. DOI: 10.22331/q-2023-03-02-933.

[2] Kishor Bharti та ін. «Noisy intermediate-scale quantum algorithms». В: *Rev. Mod. Phys.* 94 (1 2022), с. 015004. DOI: 10.1103/RevModPhys.94.015004. URL: https://link.aps.org/doi/10.1103/RevModPhys.94.015004.

[3] Kostas Blekos та ін. «A review on Quantum Approximate Optimization Algorithm and its variants». В: *Physics Reports* 1068 (черв. 2024), 1–66. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2024.03.002. URL: http://dx.doi.org/10.1016/j.physrep.2024.03.002.

[4] M. Cerezo та ін. «Variational quantum algorithms». В: *Nature Reviews Physics* 3.9 (серп. 2021), 625–644. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00348-9. URL: http://dx.doi.org/10.1038/s42254-021-00348-9.

[5] Hristo N. Djidjev та ін. *Efficient Combinatorial Optimization Using Quantum Annealing*. 2018. URL: https://arxiv.org/abs/1801.08653.

[6] Edward Farhi, Jeffrey Goldstone та Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411 . 4028 [quant-ph].

[7] Edward Farhi та ін. «A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem». В: *Science* 292.5516 (квіт. 2001), 472–475. ISSN: 1095-9203. DOI: 10.1126/science.1057726. URL: http://dx.doi.org/10.1126/science.1057726.

[8] Edward Farhi та ін. «Quantum Computation by Adiabatic Evolution». В: *arXiv: Quantum Physics* (2000). URL: https://api.semanticscholar.org/CorpusID:16467419.

[9] Lov K. Grover. «A fast quantum mechanical algorithm for database search». В: STOC '96 (1996), 212–219. DOI: 10.1145/237814.237866. URL: https://doi.org/10.1145/237814.237866.

[10] Yong He та ін. «Decompositions of n-qubit Toffoli Gates with Linear Circuit Complexity». В: *International Journal of Theoretical Physics* 56 (2017). DOI: 10.1007/s10773-017-3389-4.

[11] D-Wave Quantum Systems Inc. URL: https://www.dwavesys.com/ (дата зверн. 06.06.2024).

[12] D-Wave Quantum Systems Inc. *What is Quantum Annealing?* URL: https://docs.dwavesys.com/docs/latest/c_gs_2.html (дата зверн. 06.06.2024).

[13] Thijs Laarhoven, Joop van de Pol та Benne de Weger. *Solving Hard Lattice Problems and the Security of Lattice-Based Cryptosystems.* Cryptology ePrint Archive, Paper 2012/533. https://eprint.iacr.org/2012/533. 2012. URL: https://eprint.iacr.org/2012/533.

[14] Andrew Lucas. «Ising formulations of many NP problems». В: *Frontiers in Physics* 2 (2014). ISSN: 2296-424X. DOI: 10.3389/fphy.2014.00005. URL: http://dx.doi.org/10.3389/fphy.2014.00005.

[15] Daniele Micciancio та Oded Regev. «Lattice-based Cryptography». В: *Post-Quantum Cryptography.* За ред. Daniel J. Bernstein, Johannes Buchmann та Erik Dahmen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, с. 147—191. ISBN: 978-3-540-88702-7. DOI: 10.1007/978-3-540-88702-7_5. URL: https://doi.org/10.1007/978-3-540-88702-7_5.

[16] «Module-Lattice-Based Digital Signature Standard». В: (2023). DOI: 10.6028/nist.fips.204.ipd. URL: http://dx.doi.org/10.6028/NIST.FIPS.204.ipd.

[17] «Module-Lattice-Based Key-Encapsulation Mechanism Standard». В: (2023). DOI: 10 . 6028 / nist . fips . 203 . ipd. URL: http://dx.doi.org/10.6028/NIST.FIPS.203.ipd.

[18] N. Gama M.Schneider. *SVP CHALLENGE*. URL: https://latticechallenge.org/svp-challenge/ (дата зверн. 18.06.2024).

[19] Phong Q. Nguyen. «Hermite's Constant and Lattice Algorithms». В: *The LLL Algorithm: Survey and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, с. 19—69. ISBN: 978-3-642-02295-1. DOI: 10 . 1007 / 978 - 3 - 642 - 02295 - 1 _ 2. URL: https://doi.org/10.1007/978-3-642-02295-1_2.

[20] Michael A. Nielsen та Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011. ISBN: 9781107002173. URL: https://www.amazon.com/Quantum-Computation-Information-10th-Anniversary/dp/1107002176.

[21] Alberto Peruzzo та ін. «A variational eigenvalue solver on a photonic quantum processor». В: *Nature Communications* 5.1 (лип. 2014). ISSN: 2041-1723. DOI: 10 . 1038 / ncomms5213. URL: http://dx.doi.org/10.1038/ncomms5213.

[22] P.W. Shor. «Algorithms for quantum computation: discrete logarithms and factoring». В: (1994), с. 124—134. DOI: 10.1109/SFCS.1994.365700.

[23] Rafaella Vale та ін. *Decomposition of Multi-controlled Special Unitary Single-Qubit Gates*. 2023. arXiv: 2302.06377 [quant-ph].

[24] James D. Whitfield, Jacob Biamonte та Alán Aspuru-Guzik. «Simulation of electronic structure Hamiltonians using quantum computers». В: *Molecular Physics* 109.5 (2011), 735–750. DOI: 10.1080/00268976.2011.552441.

[25] Вакарчук І. О. *Квантова Механіка*. ЛНУ імені Івана Франка, 2012. ISBN: 978-966-613-921-7. URL: http://www.ktf.franko.lviv.ua/books/QM4/QM4.pdf.

ДОДАТОК А ПРОГРАМНИЙ КОД КЛАСУ SVP

Цей описує клас задачі SVP, який містить необхідні функції та параметри для опису екземпляру задачі SVP, обчислення властивостей базису/решітки та побудови описаних в роботі оцінок.

А.1 Програмний код

```
from numpy import linalg as LA
import numpy as np
import itertools
import numpy.random as rnd
import math

import matplotlib.pyplot as plt
plt.style.use('ggplot')

from sympy import Matrix, integer_nthroot, integer_log


class SVP:
    def __init__(self, basis):
        self._n = len(basis)
        self._B = np.array(basis)
        self._sol = None

    @property
    def dim(self):
        return self._n

    @property
    def basis(self):
        return self._B


    @property
    def solution(self):
        if self._sol is None:
            self._sol = self.bruteforce_solution()

```

```
33          return [self._sol, -self._sol], np.matmul(self._sol, self.basis), LA.norm(np.matmul(
    self._sol, self.basis))
34
35      def bruteforce_solution(self):
36          zeroos = np.array(np.zeros(self.dim))
37          best_sol = zeroos
38          best_len = float('inf')
39
40          if self.dim == 2:
41              X = []
42              Y = []
43
44          max_a = max(self.get_var_bounds())
45
46          lists = [range(-max_a, max_a+1) for a in self.get_var_bounds()]
47          print(lists)
48          for coeffs in itertools.product(*lists):
49              sol = np.array(coeffs)
50              if np.array_equal(sol, zeroos):
51                  continue
52
53              if self.dim == 2:
54                  xy = np.matmul(sol, self.basis)
55                  X.append(xy[0])
56                  Y.append(xy[1])
57
58              if LA.norm(np.matmul(sol, self.basis)) < best_len:
59                  best_sol = sol
60                  best_len = LA.norm(np.matmul(sol, self.basis))
61
62          if self.dim == 2:
63              plt.plot(X, Y, 'o', color='gray')
64              plt.plot([0], [0], 'o', color='black')
65              plt.plot(np.transpose(self.basis)[0], np.transpose(self.basis)[1], 'o', color='
    lightcoral')
66
67              plt.gca().add_patch(plt.arrow(0, 0, self.basis[0][0], self.basis[0][1], color='
    red', length_includes_head=True, head_width=0.5, head_length=0.7))
68              plt.gca().add_patch(plt.arrow(0, 0, self.basis[1][0], self.basis[1][1], color='
    red', length_includes_head=True, head_width=0.5, head_length=0.7))
69              plt.text(self.basis[0][0]-0.2, self.basis[0][1]+0.8, str(tuple(self.basis[0])),
    color='red')
70              plt.text(self.basis[1][0]-0.2, self.basis[1][1]+0.8, str(tuple(self.basis[1])),
    color='red')
71              plt.gca().add_patch(plt.Circle((0, 0), best_len, color='b', fill=False))
```

```
72
73              best_vec = np.matmul(best_sol, self.basis)
74
75              plt.plot(best_vec[0], best_vec[1], 'o', color='blue')
76              plt.plot(-best_vec[0], -best_vec[1], 'o', color='blue')
77              plt.text(best_vec[0]-0.2, best_vec[1]+0.8, str(tuple(best_vec)), color='blue')
78              plt.text(-best_vec[0]-0.2, -best_vec[1]+0.8, str(tuple(-best_vec)), color='blue'
    )
79
80          print(f'MIN_LEN_2: {best_len ** 2}')
81          return best_sol
82
83
84      def get_var_bounds(self):
85          n = self.dim
86          A = math.sqrt(n) * iroot(SVP.vol(self.basis), n)
87          print(f'A={A}\n')
88          a = [math.inf] * n
89          BT = SVP.dual_basis(self.basis)
90
91          for i in range(n):
92              a[i] = A * math.sqrt(BT[i, :].dot(BT[i, :]))
93
94          print(f'Variable bounds = \n {a}\n')
95          return a
96
97
98      def get_dual(self):
99          return SVP(SVP.dual_basis(self.basis))
100
101     @staticmethod
102     def dual_basis(basis):
103         BT = np.transpose(basis).tolist()
104         M = Matrix(BT).inv()
105
106         return M
107
108
109     @staticmethod
110     def vol(basis):
111         return int(Matrix(basis.tolist()).det())
112
113
114     @staticmethod
115     def red_req_qubits(basis):
```

```
116         L = SVP(basis)
117
118         sum = 0
119         for a in L.get_var_bounds():
120             sum += math.ceil(math.log2(2 * a))
121
122         return sum
123
124     @staticmethod
125     def from_seed(seed, dim, max_len = 10):
126         rnd.seed(seed)
127
128         basis = max_len * rnd.rand(dim, dim)
129         return SVP(basis)
130
131     @staticmethod
132     def from_txt(filename):
133         with open(filename, 'r') as file:
134             data = file.read().replace(']', '').replace('[', '')
135             data = data[0:-2]
136             basis = []
137
138             for line in data.split('\n'):
139                 b_row = [int(x) for x in line.split(' ')]
140                 basis.append(b_row)
141
142         return SVP(np.array(basis))
143
144
145     @staticmethod
146     def dual_log_orthogonality_defect(basis):
147         BT = Matrix(np.transpose(basis).tolist()).inv()
148         vol = int(Matrix(basis).det())
149         sum = 0
150         for i in range(len(basis)):
151             sum += math.log2(math.sqrt(BT[i, :].dot(BT[i, :])))
152
153         sum += integer_log(vol, 2)[0]
154         return sum
155
156     @staticmethod
157     def nqubits_svp_challange(filename):
158         with open(filename, 'r') as file:
159             data = file.read().replace(']', '').replace('[', '')
160             data = data[0:-2]
```

```
            basis = []

            for line in data.split('\n'):
                b_row = [int(x) for x in line.split(' ')]
                basis.append(b_row)

        n = len(basis)

        vol = int(Matrix(basis).det())

        A = math.sqrt(n) * integer_nthroot(vol, n)[0]

        a = [math.inf] * n

        BT = Matrix(np.transpose(basis).tolist()).inv()

        dual_orth_defect = SVP.dual_log_orthogonality_defect(basis)

        for i in range(n):
            a[i] = A * math.sqrt(BT[i, :].dot(BT[i, :]))

        nqubits = 0
        for ai in a:
            nqubits += math.ceil(math.log2(ai) + 1)

        return n, nqubits, dual_orth_defect


    @staticmethod
    def ngates_svp_challange(m, p):
        unitary1 = p * (0.5*pow(m, 2) + 3.5*m) + m
        cnot2 = p * (pow(m, 2) - m)

        return unitary1, cnot2
```

Лістинг 1: клас SVP