

***БАКАЛАВРСЬКА
РОБОТА***

БР.ІІ – 97.00.00.000 ІЗ

Група ІІ-22-2

Медик Микола

2026

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Медик Микола Васильович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

БАКАЛАВРСЬКА РОБОТА

Гібридні квантово-класичні алгоритми в комерційному програмному забезпеченні

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

**Робота містить результати власних досліджень, використання ідей, результатів і
текстів інших авторів мають посилання на відповідне джерело:**

Здобувач освітнього ступеня М.В. Медик
(підпис, ініціали та прізвище здобувача)

Науковий керівник Пасека Надія Мирославівна, к.т.н., доцент
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІПЗ

доцент.

В.В. Бандура

“ ” 2026 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Медик Миколі Васильовичу

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) "Гібридні квантово-класичні алгоритми в комерційному програмному забезпеченні"

керівник проекту (роботи) Пасєка Надія Мирославівна, к.т.н., доцент

затверджені наказом вищого навчального закладу від “ 21 ” квітня 2026 р. № 179/7

2. Строк подання студентом проекту (роботи) 10 червня 2026 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз вимог до програмного забезпечення

2. Аналіз вимог і постановка задачі

3. Проектування системи

4. Реалізація проекту

5. Тестування та впровадження

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Порівняння класичного та квантового TTS для задачі «N королів» (рис. 1.1, ст. 10)

2. Схема, що показує, як працює алгоритм зворотного відстеження (рис. 2.1, ст. 50)

3. Етапи алгоритму Метрополіс-Гастінгса (рис. 3.1, ст. 57)

4. Схема архітектури QMS. Система отримує файл у форматі JSON з описом можливих станів та відповідних їм значень (рис. 3.2, ст. 63)

5. Порівняння класичного та квантового TTS для задачі «N королів» (рис. 3.5, ст. 73)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

2. Дата видачі завдання _____ 2026 р.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення та обґрунтування теми роботи	17.01.2026	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	21.02.2026	виконано
3	Побудова моделі або алгоритму власного рішення	01.03.2026	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	21.04.2026	виконано
5	Оформлення пояснювальної записки бакалаврської роботи	02.05.2026	виконано

Студент _____ Медик М.В.
(підпис)

Керівник роботи _____ Пасєка Н.М.
(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 81 сторінку, 17 рисунків, список використаних джерел із 40 найменувань.

Метою роботи є дослідження принципів побудови гібридних квантово-класичних алгоритмів та аналіз їх застосування в комерційному програмному забезпеченні для підвищення ефективності обчислень і задач оптимізації.

Об'єкт дослідження: гібридні квантово-класичні обчислювальні системи.

Предмет дослідження: методи, алгоритми та інструменти реалізації гібридних квантово-класичних алгоритмів, зокрема квантові алгоритми пошуку, оптимізації та їх інтеграція з класичними підходами.

Результати дослідження: проведено аналіз гібридних квантово-класичних алгоритмів, досліджено основні принципи квантових обчислень.

У першому розділі розглянуто теоретичні основи квантових і класичних обчислень, принципи побудови гібридних алгоритмів, а також особливості методів пошуку та їх обмеження.

У другому розділі досліджено методи та інструменти реалізації гібридних алгоритмів, включаючи квантові алгоритми пошуку та оптимізації, квантові прогулянки та інші сучасні підходи.

У третьому розділі розглянуто практичну реалізацію гібридних алгоритмів, особливості їх застосування для задач оптимізації, а також проведено моделювання та оцінку ефективності отриманих результатів.

Висновок: використання гібридних квантово-класичних алгоритмів дозволяє підвищити продуктивність обчислювальних систем, покращити якість розв'язання задач оптимізації та відкрити нові можливості для впровадження квантових технологій у комерційне програмне забезпечення.

КЛЮЧОВІ СЛОВА: КВАНТОВІ ОБЧИСЛЕННЯ, ГІБРИДНІ АЛГОРИТМИ, КВАНТОВО-КЛАСИЧНІ СИСТЕМИ, ОПТИМІЗАЦІЯ, КВАНТОВІ ПРОГУЛЯНКИ, КОМЕРЦІЙНЕ ПЗ.

ANNOTATION

The bachelor's thesis contains of 81 pages, 17 figures, and a reference list with 40 sources.

The aim of the work is to study the principles of hybrid quantum-classical algorithms and to analyze their application in commercial software to improve computational efficiency and solve complex optimization problems.

Research object: hybrid quantum-classical computing systems.

Research subject: methods, algorithms, and tools for implementing hybrid quantum-classical algorithms, including quantum search and optimization techniques and their integration with classical approaches.

Research results: an analysis of hybrid quantum-classical algorithms was conducted, the main principles of quantum computing were studied, search and optimization methods were examined, and the effectiveness of their application in commercial software was evaluated.

The first section describes the theoretical foundations of quantum and classical computing and characteristics of search methods and their limitations.

The second section analyzes methods and tools for implementing hybrid algorithms, including quantum search and optimization algorithms, quantum walks, and other modern approaches.

The third section covers the implementation of hybrid algorithms, their application to optimization problems, a modification of the Monte Carlo method, as well as modeling and evaluation of the obtained results.

Conclusion: the use of hybrid quantum-classical algorithms improves computational performance, enhances the quality of solving optimization problems, and opens new opportunities for integrating quantum technologies into commercial software systems.

KEY WORDS: QUANTUM COMPUTING, HYBRID ALGORITHMS, QUANTUM-CLASSICAL SYSTEMS, OPTIMIZATION, QUANTUM WALKS, COMMERCIAL SOFTWARE.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ГІБРИДНИХ КВАНТОВО-КЛАСИЧНИХ АЛГОРИТМІВ	9
1.1 Сутність та принципи гібридних квантово-класичних обчислень	9
1.2 Методи пошуку та оптимізації в квантових і класичних системах	23
1.3 Класичні алгоритми пошуку та їх обмеження	30
1.4 Висновок по розділу	40
РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ РЕАЛІЗАЦІЇ ГІБРИДНИХ АЛГОРИТМІВ	41
2.1 Квантові алгоритми для задач пошуку та оптимізації	41
2.2 Квантові прогулянки як інструмент обчислень	46
2.3 Інші підходи до побудови гібридних квантово-класичних алгоритмів	49
2.4 Висновок по розділу	53
РОЗДІЛ 3. РОЗРОБКА ТА ОЦІНКА ЕФЕКТИВНОСТІ ГІБРИДНИХ РІШЕНЬ У КОМЕРЦІЙНОМУ ПЗ	54
3.1 Особливості реалізації задач оптимізації у гібридних системах	54
3.2 Гібридна квантово-класична модифікація методу	58
3.3 Результати моделювання та оцінка ефективності алгоритмів	71
3.4 Висновок по розділу	79
ВИСНОВКИ	80
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА	82

					БР.ІП – 97.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Гібридні квантово-класичні алгоритми в комерційному програмному забезпеченні	Літ.	Арк.	Акрушів
Розроб.		Медик М.В.						
Перевір.		Пасека Н.М.					7	
Реценз.		Гобир Л.М.				ІФНТУНГ ІІ-22-2		
Н. Контр.		Піх М.М.						
Затверд.		Бандура В. В.			Пояснювальна записка			

ВСТУП

У сучасному світі стрімкий розвиток інформаційних технологій, зростання обсягів даних та ускладнення обчислювальних задач зумовлюють необхідність пошуку нових підходів до підвищення продуктивності програмних систем. Традиційні класичні обчислення, незважаючи на значні досягнення, стикаються з обмеженнями при розв'язанні задач великої складності, зокрема задач оптимізації, пошуку та моделювання. У цьому контексті квантові обчислення розглядаються як перспективний напрям, здатний суттєво змінити підходи до обробки інформації. Проте сучасний рівень розвитку квантових технологій ще не дозволяє повністю замінити класичні системи, що зумовлює актуальність використання гібридних квантово-класичних алгоритмів.

Актуальність теми зумовлена необхідністю підвищення ефективності комерційного програмного забезпечення шляхом інтеграції квантових обчислень із класичними підходами. Гібридні квантово-класичні алгоритми дозволяють поєднати сильні сторони обох парадигм: високу стабільність і універсальність класичних систем із потенційною експоненційною швидкістю квантових алгоритмів. Існуючі комерційні рішення часто не здатні ефективно обробляти складні задачі оптимізації або пошуку у великих просторах станів, що відкриває можливості для застосування гібридних підходів у таких сферах, як фінанси, логістика, машинне навчання та аналіз даних.

Метою роботи є дослідження та розробка підходів до застосування гібридних квантово-класичних алгоритмів у комерційному програмному забезпеченні з метою підвищення ефективності обчислень і якості розв'язання складних задач.

Завданнями дослідження є аналіз теоретичних основ квантових і класичних обчислень, дослідження методів пошуку та оптимізації, вивчення сучасних квантових алгоритмів і підходів до їх інтеграції з класичними системами, розробка та модифікація гібридних алгоритмів (зокрема на основі методів Монте-Карло), реалізація моделей та проведення експериментального дослідження їх ефективності, а також оцінка можливостей практичного

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

застосування у комерційному програмному забезпеченні.

Об’єктом дослідження є процеси побудови та застосування гібридних квантово-класичних обчислювальних систем.

Предметом дослідження є методи, алгоритми та інструменти реалізації гібридних квантово-класичних алгоритмів, включаючи квантові алгоритми пошуку, квантові прогулянки, класичні методи оптимізації та їх інтеграцію в єдині програмні рішення.

Методи дослідження включають аналіз наукових джерел для вивчення сучасного стану квантових обчислень, порівняльний аналіз класичних і квантових алгоритмів, математичне моделювання гібридних підходів, експериментальне дослідження ефективності алгоритмів, а також статистичний аналіз отриманих результатів.

У рамках роботи передбачається розробка моделей гібридних алгоритмів для задач оптимізації та пошуку, їх програмна реалізація та оцінка ефективності шляхом моделювання. Особлива увага приділятиметься можливості інтеграції таких алгоритмів у сучасні комерційні програмні продукти, що дозволить підвищити їх продуктивність, скоротити час обчислень і покращити якість прийняття рішень.

Бакалаврська робота містить 81 сторінку, 17 рисунків, 3 розділи, список використаних джерел із 40 найменувань.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ГІБРИДНИХ КВАНТОВО-КЛАСИЧНИХ АЛГОРИТМІВ

1.1 Сутність та принципи гібридних квантово-класичних обчислень

Квантові обчислення передбачають застосування принципів квантової механіки до сфери обробки інформації. Квантова механіка, яка започаткувала революцію у фізиці 20-го століття, поставила під сумнів фундаментальні принципи, що існували століттями. Відомі фізики, такі як Макс Планк, який ввів концепцію квантів через дослідження випромінювання чорного тіла, Альберт Ейнштейн зі своєю роботою над фотоелектричним ефектом та значним внеском у квантову фізику, а також такі визначні постаті, як Луї де Бройль, Нільс Бор, Едвін Шредінгер, Макс Борн, Гейзенберг, Дірак та інші, значно просунули галузь квантової механіки [1]. Ця нелогічна та новаторська галузь знайшла практичне застосування у створенні лазерів, мікропроцесорів, надпровідників та хімічних моделей.

Однак моделювання складної квантової поведінки системи за допомогою класичного комп'ютера виявляється надзвичайно складним. Таке моделювання вимагає кодування складних станів та управління безліччю взаємодій між ними. Цей виклик спонукав Річарда Фейнмана запропонувати нетрадиційне рішення: використовувати самі принципи квантової фізики, використовуючи квантові стани для моделювання квантових систем. Фейнман, керований бажанням пришвидшити моделювання, ненавмисно заклав основу для того, що зараз визнається квантовим - очікуваної революції, яка має сформувати 21 століття. Пропозиція Фейнмана - це лише верхівка айсберга квантових обчислень, і він не був першим. У 1985 році Девід Дойч представив концепцію квантової машини Тюрінга [4]. Вона була еквівалентна машині Тюрінга, розробленій Аланом Тюрінгом [5], але довела, що можна виконувати той самий набір операцій за допомогою квантового обладнання. Слідуючи ідеї моделювання експериментів на квантових комп'ютерах, інші вчені запропонували обробляти інформацію швидше

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

та з меншими витратами енергії для розвитку таких галузей, як квантова хімія, зв'язок або штучний інтелект.

Якщо класичні обчислення розглядати як маніпулювання сигналами 0 та 1 (що вказують на наявність або відсутність електричного струму) та реалізацію логічних вентилів, які інтегрують ці сигнали (вентилі I, АБО, І-ТА, XOR тощо), то квантові обчислення мають спільні риси з точки зору логіки, але суттєво відрізняються за своїми властивостями. Квантові обчислення використовують квантові частинки, що існують у станах суперпозиції та заплутаності, що дозволяє їм не лише представляти стани, але й виконувати операції оптимізованим чином порівняно з класичними обчисленнями [3]. Основна подібність між ними полягає в тому, що в класичних і квантових обчисленнях одиницею інформації є біт (або кубіт) зі станами 0 та 1. Отже, обидва використовують двійкову систему кодування.

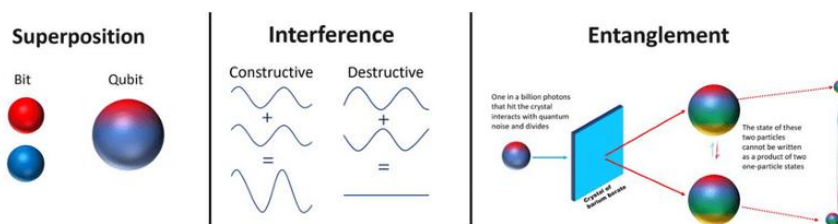


Рисунок 1.1 - Схема представляє явище суперпозиції в кубіті

Вона представляє явище суперпозиції в кубіті, представленому двома можливими станами: червоним і синім, і кубітом, який є комбінацією обох. Ефект інтерференції посередині може бути конструктивним, якщо дві хвилі інтерферують, створюючи візерунок, або деструктивним, якщо обидві компенсують одна одну [4]. Нарешті, заплутаність можна зрозуміти як дві частинки, які не можна записати як добуток.

Суперпозиція - це властивість, яка дозволяє квантовому стану перебувати у двох станах одночасно. Це означає, що кубіт може існувати у стані 0, 1 або в лінійній комбінації обох можливостей, що призводить до надзвичайно стислого представлення інформації. З іншого боку, заплутаність - це явище, завдяки якому

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

дві квантові частинки поведуться так, ніби вони є єдиним цілим. У заплутаній квантовій системі будь-яка зміна, що впливає на систему, однаково впливає на обидві частинки. Значення двох або більше заплутаних кубітів полягає в їхньому колективному впливі: операції (квантовий вентиль) над заплутаною системою можуть впливати на всі її кубіти [6]. На рисунку 1.1 ці три концепції візуально представлені.

Квантові обчислення пропонують численні переваги порівняно з класичними обчисленнями: надщільне кодування інформації, передача через захищені канали, зниження витрат енергії, операції в суперпозиції тощо. Ця робота буде зосереджена на здатності працювати з суперпозицією станів, що дозволяє зменшити обчислювальну складність алгоритмів [8].

Усі ці переваги мають вагоме теоретичне підґрунтя. Існують роботи з аналізу обчислювальної складності, які створили нові класи, такі як BQP [9]. В інших роботах сформульовано квантову машину Тюрінга [4] та виведено набір універсальних вентилів для квантових обчислень [10]. Також є дослідження, які демонструють переваги поліноміальних або суперполіноміальних алгоритмів над класичними алгоритмами [11]. Однак важливо підкреслити, що всі ці демонстрації є суто теоретичними та ще не були доведені в квантових алгоритмах, що виконуються на реальному обладнанні. Це означає, що на даний момент квантові алгоритми не мають переваги над класичними у вирішенні цікавих задач.

На цьому етапі варто запитати, чому квантові обчислення досі не вирішили реальних проблем. Відповідь проста: потрібно проаналізувати багато обмежувальних факторів.

— *Розмір квантового обладнання (кількість кубітів):* кількість кубітів часто вважається основною метрикою для оцінки масштабу квантового комп'ютера. Однак важливо підкреслити, що ця метрика, хоча й значуща, не є ні найважливішим, ні найбільш обмежувальним фактором. Хоча дехто може проводити паралелі між кількістю кубітів та ємністю пам'яті, розміром оперативної пам'яті або можливостями процесора класичного комп'ютера, такі порівняння не є точними.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Кількість кубітів безпосередньо пов'язана зі здатністю квантового комп'ютера представляти різні стани. Простіше кажучи, квантовий комп'ютер з більшою кількістю кубітів може представляти більшу кількість станів, що дозволяє йому звертатися до більш масштабні задачі. Однак важливо визнати, що здатність вирішувати більші задачі не обов'язково означає швидше вирішення задач. Натомість, справжній потенціал квантових обчислень полягає в їх здатності вирішувати задачі, які класичні комп'ютери можуть вважати неперевершеними. Також правда, що збільшення кількості апаратних шумових кубітів рухається до можливості отримання логічних кубітів без шуму, що пришвидшить квантові алгоритми, але це все ще майбутній сценарій.

— *Квантові помилки* : Виконання алгоритмів на апаратному пристрої ніколи не буває безпомилковим, ні класично, ні квантово. Класично ці помилки були ретельно проаналізовані та вивчені з використанням різних стратегій їх виправлення. З квантової точки зору, у пристроях існують притаманні помилки, які роблять класичні методи корекції помилок непридатними [5]. Ці помилки є фазовими помилками. Цей новий тип помилки, доданий до квантових обмежень неможливості копіювання станів без їх зміни, означає, що наразі квантові помилки обмежують виконання алгоритмів. Досягнення в квантовій корекції помилок означатимуть можливість виконання довших та складніших алгоритмів.

— *Часи когерентності* : Тривалість, протягом якої квантова система зберігає свої квантові атрибути, такі як суперпозиція та заплутаність, називається часом когерентності. Коли квантовий комп'ютер має обмежений час когерентності, це означає, що для виконання квантових вентилів доступний лише короткий період, перш ніж кубіти системи втратять свої квантові властивості. Отже, збільшення часів когерентності квантових чипів відкриває потенціал для виконання складніших і довших алгоритмів. Часи когерентності обмежують глибину схеми, що є справжнім обмежувальним фактором у сучасному квантовому обладнанні [12] .

— *Алгоритми* : Повне використання переваг квантових обчислень залишається предметом постійних досліджень, і, як наслідок, квантові алгоритми

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

ще не досягли своєї максимальної швидкості та ефективності. Квантові обчислення обробляються настільки відмінним від класичних обчислень чином, що поточні зусилля в основному полягають у спробах відтворити класичні алгоритми для визначення будь-яких потенційних переваг. Однак у довгостроковій перспективі ця стратегія повинна розвиватися, щоб охопити розробку повністю квантових алгоритмів, які не залежать від класичного натхнення.

— *Представлення* : Значне обмеження квантових алгоритмів полягає в тому, як представлена інформація. Багато з цих алгоритмів намагаються закодувати інформацію у формулу, роблячи квантову обробку схожою на квантове моделювання, а не на обробку інформації. Вкрай важливо визнати, що в терміні «квантові обчислення» акцент робиться радше на «обчисленнях», ніж на «квантових». Іншими словами, основна увага має бути зосереджена на обробці інформації.

— *Оператори* : Квантовий оператор складається з набору вентилів і є додатковим рівнем абстракції в процесі реалізації квантового алгоритму. Оператор у контексті квантових обчислень служить інструментом програміста для виконання певних операцій таким чином, що використовує можливості квантового комп'ютера [6]. Однак значна проблема виникає через відносно обмежену кількість доступних операторів. Фактично, багато алгоритмів, які теоретично демонструють квантову перевагу, походять від двох операторів або знаходяться під їхнім сильним впливом: оператора дифузії Гровера [13] та оператора оцінки амплітуди, відомого Шором у галузі факторизації простих чисел [14] .

— *Теорема про відсутність клонування (зберігання інформації)* : Хоча квантові обчислення мають трансформаційні квантові властивості, вони також вводять складні особливості, які

ускладнюють його маніпулювання. Мабуть, найважливішою з цих проблем є той факт, що спостереження за квантовою системою за своєю суттю порушує її роботу. На практиці це означає, що квантові стани не можна

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

дублювати, не руйнуючи їх, оскільки сам акт дослідження стану неминуче його змінює [7]. Цей принцип викладено в теоремах про неклонування та впливає, а наразі забороняє, деякі важливі операції в інформатиці, такі як копіювання та зберігання інформації. Однак цієї проблеми можна уникнути, оскільки квантові стани можна зберігати в квантовій пам'яті та отримувати за допомогою квантової телепортації, що не порушує теореми про неклонування. З іншого боку, це обмеження стає перевагою в контексті квантової криптографії, де неможливість копіювання стану гарантує його секретність.

Усі ці фактори, перелічені вище, є лише переліком ключових моментів, що спрямовують розвиток цієї нової технології, і чим швидше вони розвиватимуться, тим швидше квантові обчислення стануть реальністю замість класичних обчислень або поряд з ними.

Наразі розвиток квантових обчислень – це глобальна та спільна робота - академічних кіл та компаній, у якій усі зосереджені на моделюванні, апаратному та програмному забезпеченні [8]. Ця співпраця майже завершена, і в результаті як компанії, так і університети публікують свої досягнення. Слід зазначити, що компанії готові до співпраці з кількох причин. По-перше, немає прямого застосування квантових обчислень у промисловості, тому вони ще не мають чіткого та визначеного бізнесу. По-друге, все ще існує багато проблем, які потрібно вирішити та над якими потрібно працювати, тому відносини між різними гравцями все ще є симбіотичними. Нарешті, обсяг бізнесу, що пропонується квантовими обчисленнями, настільки великий, що деякі компанії не в змозі його покрити, і їм вигідніше співпрацювати, ніж конкурувати. Це також призводить до великої нестачі працівників у сфері квантових обчислень.

Дивлячись на профіль вчених, які працюють сьогодні в галузі квантових обчислень, більшість із них – це фізики, що спеціалізуються на квантовій фізиці та зробили крок до квантової інформації як для квантових алгоритмів, так і для квантових багаточастинкових систем. Також багато фізиків-експериментаторів працюють над розробкою квантових комп'ютерів. Якщо шукати інші профілі, що працюють у сфері квантових обчислень, то більшість із них – інженери, які

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

працюють над суміжними схемами в квантових чіпах.

Однак у галузі алгоритмів та розробки програмного забезпечення кількість фахівців з інформатики мінімальна. Це зрозуміло, оскільки перші кроки квантових обчислень мали бути розроблені тими, хто розумів квантову теорію з великою точністю, а це вимагало додаткових зусиль від фахівців з інформатики. Сьогодні ця ситуація змінилася; вже можливо розробляти та впроваджувати алгоритми з певним рівнем складності, а орієнтований на моделювання підхід, який традиційно використовували фізики в розробці алгоритмів, виявляється неефективним. З цієї причини компанії та університети все частіше вимагають фізиків, здатних програмувати та розробляти алгоритми, або навіть фахівців з інформатики з досвідом у квантовій фізиці.

Ця робота, серед інших своїх внесків, намагається представити окремий погляд на розробку квантових алгоритмів. Грунтуючись на досвіді людини, яка звикла працювати з алгоритмами обробки даних та класичним штучним інтелектом, вона пропагує підхід, зосереджений на обробці інформації, а не на простому моделюванні фізичних явищ. Хоча це твердження може здатися дещо неоднозначним, у наступних розділах буде пояснено розробку алгоритмів, які наголошують на представленні інформації через змінні, причому різні значення призводять до появи безлічі станів. Згодом буде зроблено спробу поєднати ці змінні з конкретними правилами задачі для отримання рішень [9]. Цей підхід більше відповідає філософії обробки станів Гровера, ніж акценту Шора на оцінці фази або адіабатичних обчисленнях ґрунтується на гамільтонівській еволюції.

Дотримуючись цього підходу до створення квантових алгоритмів з сильнішим акцентом на вирішення промислових проблем з більш комп'ютерно-центричної точки зору, було розроблено цю докторську дослідницьку програму. Вона отримала відзнаку промислового доктора. Це визнання визнає здатність цієї роботи запропонувати рішення, що виникли в академічних колах, для промислової сфери, подолавши розрив, який часто існує між цими двома сферами. Ця відзнака особливо доречна, враховуючи, що останнє десятиліття стало свідком вибуху стартапів за моделлю спін-оффів. У цій моделі досвідчені вчені з різних

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

університетів розробили прототипи, які є як ринковими, так і цінними для промисловості.

У конкретному випадку цієї роботи іспанська компанія Quasar визначила квантові обчислення як перспективний напрямок бізнесу для своїх інтересів. Quasar – це аерокосмічна компанія, яка розробляє штучний інтелект та алгоритми оптимізації, що застосовуються до аналізу даних та просторових рішень [10]. Одним з основних клієнтів Quasar є Європейське космічне агентство, для якого вона розробила такі продукти, як автономний аналіз супутникових знімків для спостереження та моніторингу Землі або автономний роботизований телескоп, здатний самостійно керувати спостереженнями ночі в різних півкулях.

Quasar розуміла, що складність та обсяг космічних даних дедалі більше вимагають більшої обчислювальної потужності, особливо коли космічний сектор щойно вступив у так звану нову космічну еру, коли приватні компанії отримали ліцензії на запуск та експлуатацію супутників [15]. Це спонукало компанію шукати нові рішення для подолання вузьких місць у обробці даних та алгоритмах оптимізації. З цією метою у співпраці з групою GICC Мадридського університету Комплутенсе було запропоновано ступінь доктора промислових наук, який був нагороджений стипендією Мадридської спільноти на здобуття ступеня доктора промислових наук (IND2019/TIC17146).

Для того, щоб виконати цю роботу з промисловості та мати змогу створювати цікаві алгоритми для промисловості та співпраці з провідними компаніями, під час розробки цієї роботи було детально проаналізовано основні напрямки досліджень, що проводяться іншими компаніями та університетами.

Як обговорювалося вище, квантові обчислення пропонують можливості обробки станів, які набагато перевершують класичні [11]. Однак, було проаналізовано, що квантові обчислення все ще мають численні проблеми, які потребують вирішення. Деякі з цих проблем особливо добре вирішуються класичними обчисленнями. З цієї причини кілька років тому вчені почали обговорювати гібридні класично-квантові алгоритми. Ці алгоритми намагаються поєднати переваги обох світів: велику ємність обробки та зберігання даних

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

класичних комп'ютерів та суперпозиційну обробку квантових комп'ютерів.

Гібридні алгоритми складаються з архітектур, що включають модулі обох технологій. Як правило, перший модуль призначений для класичної підготовки та обробки даних, потім модуль квантового аналізу і, нарешті, класичний модуль для аналізу виходу квантового модуля, як показано на рисунку 1.2.

Дотримуючись цієї архітектури, гібридні алгоритми пропонують численні переваги у короткостроковій перспективі.

- Переваги обох світів : гібридні алгоритми поєднують швидкий аналіз та попередню обробку даних з класичних обчислень із можливістю виконувати обробку суперпозиції та квантові операції з квантових обчислень.
- Прискорення порівняно з класичними: Гібридні алгоритми, що містять квантовий модуль , виконують складні операції класичних алгоритмів зі значним прискоренням порівняно з чисто класичними алгоритмами.

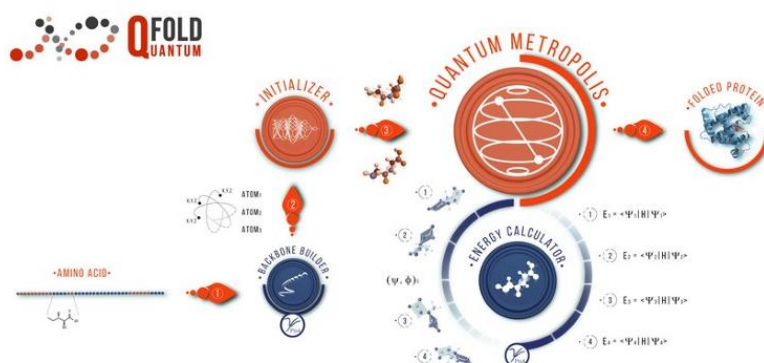


Рисунок 1.2 - Схема, що ілюструє гібридний алгоритм.

Червоні модулі позначені квантовими, а класичні – синіми. Квантові та класичні модулі співпрацюють для отримання рішення. Система починається з ланцюжка амінокислот, усі кути якого ініціалізовані рівними 0. Потім в ініціалізаторі виконується початкове припущення кутів, і ця ініціалізація уточнюється в модулі квантового мегаполісу. Калькулятор енергії, позначений синім кольором, – це класичний модуль для розрахунку енергій кожної можливої конфігурації. Зрештою, отримано складену структуру білка.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- Виконання на реальному обладнанні : Обмеження квантового обладнання, згадані раніше, можуть перешкоджати виконанню повного квантового алгоритму [12]. Однак сучасні квантові чіпи достатньо здатні виконувати певні компоненти або сегменти цих алгоритмів. Ця можливість дозволяє виконувати гібридні алгоритми на реальному квантовому обладнанні.

Ці гібридні алгоритми ідеально відповідають сучасному підходу до квантових обчислень. Розробка контролю якості (QC) вже обговорювалася як допомога у подоланні обмежень класичних обчислень. Однак класичні обчислення працюють на багатьох рівнях, від персональних комп'ютерів, офісних систем до обчислювальних центрів та високопродуктивних обчислень (HPC) [17].

Фундаментальна мета розробки квантових контролерів (QC) полягає в тому, щоб допомогти або замінити суперкомп'ютери, що використовуються в високопродуктивних обчисленнях (HPC). Це пов'язано з кількома причинами: по-перше, тому, що пошук квантових переваг за допомогою сучасних пристроїв вимагає вирішення проблем високої складності, які персональний комп'ютер ніколи не вирішить; по-друге, тому, що сучасні класичні обчислення є ефективними та результативними у вирішенні простих проблем; по-третє, квантові комп'ютери повинні контролюватися експертами, що ускладнює їх масове виробництво та обслуговування; і, нарешті, тому, що вартість квантового комп'ютера має покриватися великою корпорацією.

Поточний короткостроковий та середньостроковий підхід може бути змінений у довгостроковій перспективі [13]. Не дивно, що з часом квантові алгоритми знайдуть застосування для персональних комп'ютерів. Наприклад, вчені, які керували першими комп'ютерами розміром з кімнату, ніколи не думали про мобільні пристрої, віртуалізацію чи соціальні мережі, але з часом вони були розроблені. Однак очевидно, що якщо такий розвиток відбудеться, то це буде в довгостроковій перспективі.

Дослідження в галузі квантових обчислень спрямовані на створення квантових алгоритмів, здатних вирішувати проблеми, які є обчислювально

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

складними для класичних алгоритмів. Багато з цих алгоритмів, розроблених для вирішення таких складних проблем, зараз працюють на суперкомп'ютерах [18] . Отже, в найближчій перспективі розвиток квантових обчислень буде спрямований або на доповнення, або на потенційне витіснення суперкомп'ютерів у певних сферах.

Вона представляє можливу архітектуру в конвеєрі НРС, в якому квантові обчислення інтегровані як додатковий модуль для виконання певних алгоритмів. Згідно з запропонованою архітектурою, модуль контролю якості (QC) буде еквівалентний модулям графічного процесора (GPU) або центрального процесора (CPU) [14].

Тим не менш, фундаментальною характеристикою суперкомп'ютерів є їхня здатність обробляти великі обсяги даних, які часто називають великими даними. Квантові комп'ютери, з іншого боку, стикаються з обмеженнями при роботі з такими великими наборами даних. Тому в сучасному сценарії інтеграція обох технологій стає вкрай необхідною, а гібридні алгоритми постають як розумний підхід до подолання розриву між цими двома сферами.

Крім того, в різних роботах [19] гібридні алгоритми розглядаються як місток, що завершується чисто квантовими алгоритмами. Наразі недостатньо відомо, як квантувати деякі операції або як отримати переваги квантових обчислень у таких галузях, як квантовий штучний інтелект (QAI) або квантова хімія, і єдиним рішенням є застосування квантових обчислень до деяких модулів класичного конвеєра. Ці гібридні рішення будуть корисним проміжним кроком для вивчення квантової обробки інформації та розробки чисто квантових алгоритмів.

З іншого боку, є автори, які стверджують, що квантові обчислення будуть лише додатковим модулем обробки в рамках великої класичної архітектури [20] . Ця концепція полягала б у наявності квантового процесора (QPU) всередині класичного комп'ютера, еквівалентного сучасним графічним процесорам (GPU), як показано на рисунку 1.3.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

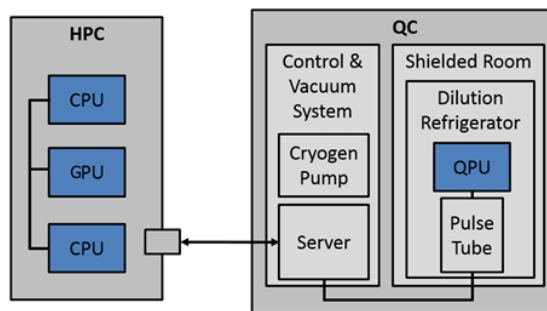


Рисунок 1.3 - Схема, що ілюструє гібридну архітектуру з інтеграцією QPU в систему HPC.

Який би погляд на гібридні алгоритми не переважав, це буде битва, яка триватиме в середньостроковій та довгостроковій перспективі. У короткостроковій перспективі гібридні алгоритми продовжуватимуть розвиватися, оскільки це найшвидший шлях до концепції квантової переваги, що розуміється як алгоритм, що використовує квантові обчислення та є швидшим за класичний [15]. Поступово будуть розроблені кращі інтерфейси між класичними та квантовими модулями, що оптимізує кодування інформації та її виконання.

Гібридні алгоритми також мають значення завдяки своїй здатності обробляти різні типи даних, що охоплюють як класичні, так і квантові дані. Як згадувалося раніше, квантові обчислення спочатку пропонувалися переважно для квантового моделювання, проте вони швидко стали цінним інструментом для ширшого спектру алгоритмів. Отже, виходячи з характеру інформації, що використовується, можна провести класифікацію, яка розрізняє класичні або квантові дані та класичні або квантові алгоритми, що призводить до чотирьох можливих категорій, як показано на рисунку 1.4.

Гібридні алгоритми здатні працювати в трьох із цих чотирьох категорій, таких як класичні алгоритми з квантовими даними [22] , квантові алгоритми з класичними даними [2] або квантові алгоритми з квантовими даними [23] .

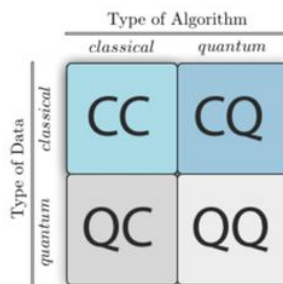


Рисунок 1.4 - На цьому рисунку можна спостерігати чотири різні парадигми для алгоритмів, що враховують тип алгоритму та тип даних.

Здатність запускати квантові алгоритми на реальному обладнанні є однією з важливих досягнень в галузі квантових обчислень за останні роки. Ця віха зумовлена появою шумних квантових комп'ютерів середнього масштабу (NISQ). Ці NISQ комп'ютери - це компактні пристрої, що містять від 10 до 1000 кубітів, що характеризуються значним рівнем шуму, але здатні виконувати алгоритми малого масштабу. Вони виконують подвійну функцію: як випробувальний майданчик для розробки алгоритмів, так і як проміжний крок до досягнення відмовостійких алгоритмів, де очікувана квантова перевага, як очікується, посправжньому проявиться [24] .

Розробка апаратного забезпечення NISQ створила промислові можливості для багатьох компаній. Великі компанії, такі як IBM, Google та Microsoft, розпочали дослідження цієї технології та досягли значних успіхів у квантових обчисленнях як в апаратному, так і в програмному забезпеченні, що стало значним завдяки дослідженням, проведеним в університетах. Цей момент став сприятливим ґрунтом для появи приблизно у 2015 році невеликих стартапів, очолюваних вченими з великих університетів та фінансово підтримуваних раундами фінансування та венчурного капіталу. Ці стартапи продовжили розвиток усіх напрямків досліджень контролю якості, дотримуючись радше академічної, ніж промислової моделі.

Така різноманітність учасників та їхньої співпраці головним чином зумовлена відсутністю конкуренції та ранньою стадією розвитку технології. Парадигматичним прикладом цієї ситуації є три основні пропозиції, які

продемонстрували квантову перевагу над штучною проблемою:

- Google: Використовуючи чіп Sycamore, вони вирішили задачу вибірки випадкових схем швидше, ніж класичний комп'ютер [25] .
- Використання чіпа, вони розв'язали задачу гаусової амплітудної вибірки швидше, ніж класичний комп'ютер [26] .
- Використовуючи чіп Borealis, вони вирішили задачу вибірки гауссових бозонів швидше, ніж класичний комп'ютер [27] .

Ці три експерименти були корисними для демонстрації того, що квантові обчислення можуть вирішувати певні проблеми швидше, ніж класичні алгоритми. Однак вони не мають реального застосування. Цікаві напрямки з реальним застосуванням, в яких очікується розробка квантових алгоритмів у майбутньому, такі:

- Оптимізація: Такі проблеми, як доставка посилок у місті, мінімізуючи кількість вантажівок або використане паливо, мають величезне значення та можуть бути цікавими для контролю якості, оскільки вони охоплюють багато можливих комбінацій для перевірки.
- QAI: Такі проблеми, як класифікація, розпізнавання мовлення, генерація зображень або прогнозування, наразі вирішуються класичними комп'ютерами, хоча й з певними властивими їм обмеженнями. Для вирішення цих проблем у великих масштабах необхідна така технологія, як квантові обчислення, яка використовує операції в суперпозиції.
- Квантова хімія: Класичні комп'ютери часто витрачають значні ресурси під час кодування хімічного моделювання. Натомість квантові комп'ютери за своєю суттю представляють квантові системи хімії, що дозволяє обробляти дані з набагато більшою ефективністю та точністю.
- Квантове моделювання: На відміну від квантової хімії, квантове моделювання матиме суттєві переваги від квантових обчислень. Це пояснюється тим, що виконання операторів, які є невід'ємною частиною квантової схеми, підвищує як точність, так і масштаб систем, які можна ефективно моделювати.
- Комунікації : Парадигма квантової механіки відкриває можливості

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

для вдосконалення класичних методів комунікації, забезпечуючи безпечні канали, надщільне кодування та швидкий зв'язок.

Однак, усі ці напрямки досліджень повинні супроводжуватися потужним розвитком квантового обладнання [16]. Як було показано раніше, обмеження шуму, когерентності та розміру квантових комп'ютерів впливають на алгоритми до такої міри, що роблять їх непотрібними. Наразі комп'ютери NISQ є адекватними як перевірка концепції, але якщо квантові обчислення мають використовуватися в реальних середовищах, необхідно перейти до більш надійної парадигми квантових чіпів. З'явилися відмовостійкі комп'ютери, здатні виправляти власні помилки та збільшувати час когерентності. Більше того, навіть ці надійні квантові комп'ютери, здається, використовують класичні комп'ютери, тобто гібридні алгоритми [28], що є додатковою мотивацією для цієї роботи. Принаймні, саме так розвиваються дослідження на даний момент.

У цій роботі досліджувалися гібридні алгоритми, що вирішують деякі з основних проблем, що цікавлять галузь. Інтерес полягав не лише в розробці алгоритмів для вирішення деяких проблем у цих областях, але й у кращому розумінні концепції обробки інформації в квантових обчисленнях та створенні алгоритмів, які були б не лише теоретичними, а й мали б кодову реалізацію та можливість виконання в симуляторах або квантовому обладнанні. Тому кожна з описаних робіт має пов'язаний код, який можна виконати. Теми, які розглядалися в цій роботі, - це оптимізація, квантовий штучний інтелект та квантова хімія, які є трьома основними з перелічених вище.

1.2 Методи пошуку та оптимізації в квантових і класичних системах

Методи пошуку та вибірки охоплюють сімейство алгоритмів, які можна описати за допомогою різних визначень. Щоб представити ці концепції простими словами:

- Вибірка – це метод наближеного визначення невідомого розподілу, визначеного функцією. Це можна порівняти з роботою сантехніка, якому потрібно

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

зрозуміти форму та шлях труби, прихованої всередині стіни. Для досягнення цієї мети сантехнік робить невеликі отвори в стіні та досліджує трубу в різних точках, поступово отримуючи достатньо точне розуміння її форми для виконання необхідної роботи.

- Пошук – це метод, який передбачає знаходження певного елемента в розподілі, зазвичай максимуму або мінімуму, шляхом здійснення запитів до базової функції. Порівняно, це схоже на завдання сантехніка знайти отвір або проблему в трубі, відстежуючи потік води в трубі. Ефективний пошук виграє від попереднього відбору проб, оскільки це надає інформацію про розподіл, що шукається.

Ці аналогії ілюструють фундаментальні концепції методів пошуку та вибірки більш доступним способом. У сфері квантових обчислень алгоритми, засновані на пошуку та вибірці, розробляються для ефективного вирішення задач оптимізації, часто перевершуючи можливості класичних методів.

У цій роботі метод пошуку та вибірки визначається як ітераційний метод, що дозволяє виконувати пошук для знаходження скороченого підпростору повного простору пошуку та вибірково проводити вибірку цього підпростору для визначення його розподілу [17]. На наступному кроці, використовуючи інформацію з попередньої вибірки, здійснити новий пошук іншого цікавого підпростору. Використовуючи цю ітераційну методологію, можна буде дізнатися глобальний розподіл задачі для пошуку в деяких випадках мінімуму або максимуму (чистий пошук) або вибірково провести вибірку простору.

У цій роботі набір алгоритмів пошуку та вибірки (S&S) розглядається як окрема категорія в ширшій сфері методів вирішення задач оптимізації. Хоча цю перспективу можна вважати дещо інноваційною, важливо зазначити, що зосередження саме на алгоритмах пошуку та вибірки насправді не вносить нічого принципово нового. Натомість ця нова категорія спрощує алгоритми, дозволяючи абстрагувати певні функції алгоритмів оптимізації, які не є суттєвими для цих задач.

Причина, чому ці два типи алгоритмів можуть становити окрему категорію

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

в рамках методів задач оптимізації, полягає в тому, що вони мають багато подібностей та застосувань.

- Числові методи : методи пошуку та вибірки – це числові методи, які обчислюють розв'язок за допомогою повторюваних запитів до функції, яка повертає значення. Обидва методи уточнюють розв'язок, оскільки виконується більше ітерацій алгоритму. Крім того, важливо зазначити, що кожна ітерація алгоритму S&S дуже проста у виконанні, тому задачу можна ефективно вирішити, виконуючи багато ітерацій поспіль, на відміну від більш точних і складних методів. [29]

- Складні функції : методи пошуку та вибірки особливо корисні для складних розподілів, які важко аналізувати іншими методами. Наприклад, функції, виведення яких є складним. Метою пошуку та вибірки є зменшення кількості разів виконання функції, зокрема, якщо вичерпний пошук є єдиним відомим рішенням. [30]

- Наближені методи : оскільки вони є наближеними методами, у більшості випадків S&S повертають рішення, яке не є оптимальним, а лише субоптимальне наближення. Наприклад, вибірка функції вимагатиме вибірки такої ж кількості станів, яку має функція, щоб повернути точне рішення, що не є ефективним. У випадку пошуку, він може повернути максимум, який не є абсолютним мінімумом, але він достатньо близький.

- Масштабованість : Методи S&S повинні працювати з великим простором станів, щоб отримати перевагу над іншими складними методами. Класично, цей тип задачі з великим простором станів страждає від прокляття розмірності [31] , що призводить до проблеми для класичних алгоритмів через їх погану масштабованість. Кількість разів, коли алгоритм S&S повинен викликати функцію, має тенденцію до експоненціального масштабування, коли задача лінійно зростає за кількістю станів [32] .

- Тісно пов'язані : Алгоритми пошуку та вибірки тісно пов'язані, оскільки їх поєднання може бути дуже корисним у деяких задачах. Наприклад, виконання завдання пошуку набагато простіше, якщо до вирішення проблеми

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

було взято хороший зразок [33] .

Під час розробки цієї роботи було помічено, що алгоритми пошуку та вибірки особливо ефективні в задачах оптимізації, які відповідають певним характеристикам. У цій роботі під алгоритмами оптимізації розуміються ті, в яких необхідно знайти рішення, яке відповідає певним критеріям і має максимальну винагороду або мінімальні витрати, залежно від того, чи працюється над задачею максимізації чи мінімізації відповідно.

Основні характеристики, яким повинна відповідати задача оптимізації, щоб алгоритми S&S отримали будь-яку квантову перевагу, полягають у тому, що вона вимагає майже вичерпного пошуку в просторі станів для знаходження рішення, а також у тому, що простір станів є великим або зростає майже експоненціально. Хоча ці особливості можуть здаватися дуже детальними або специфічними, вони є поширеними в задачах оптимізації [34] .

Якщо розв'язання задачі оптимізації вимагає використання алгоритмів та комп'ютерів, це означає, що вона включає так багато змінних, що зв'язки між ними є значно великими, а функція оцінки задачі є складною для оцінки. Тому щоразу, коли задача змінюється або зростає, кількість ресурсів для її розв'язання також збільшуватиметься. Крім того, алгоритми оптимізації, оскільки вони вимагають пошуку найкращого рішення серед багатьох можливих рішень, необхідно спробувати майже всі комбінації між вузлами, доки не буде знайдено оптимальне або принаймні неоптимальне, близьке до рішення. Тому в багатьох алгоритмах оптимізації найкращим методом пошуку рішення є пошук можливих комбінацій. Більше того, цей пошук є майже вичерпним, оскільки дуже важко сказати, які комбінації вузлів не слід пробувати, оскільки вони не покращать якість рішення.

Складність алгоритму, який повинен виконувати вичерпний пошук задачі, зростає експоненціально. Це пояснюється тим, що кожного разу, коли додається нова змінна, кількість нових станів зростає шляхом об'єднання цієї нової змінної з усіма існуючими. Це означає, що масштабованість алгоритму погана, оскільки, незначне збільшення розміру задачі призводить до її нерозв'язності [35] . Ця погана масштабованість головним чином пов'язана з тим, що класичні алгоритми

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

оцінюють стани окремо. Квантові, з іншого боку, оцінюють суперпозицію. Таким чином, квантовий алгоритм логарифмічно збільшуватиме свою складність у представленні станів, але він все одно завжди діятиме на суперпозицію простору станів, і його масштабованість буде кращою, ніж у класичного алгоритму, завжди пам'ятаючи, що це стосується алгоритмів вичерпного пошуку [36]. Наприклад, у квантовому алгоритмі S&S, заснованому на Гровері, перевага буде до полінома.

Саме на цьому етапі низької масштабованості класичних алгоритмів з'являються квантові S&S-алгоритми з очікуваною перевагою в їхній масштабованості. Сила цих алгоритмів полягає в теоретичних перевагах, отриманих операторами Гровера та їх різними варіаціями [37]. Основною концепцією для розуміння того, чому квантові алгоритми можуть пришвидшити розв'язання задач оптимізації за допомогою QS&S-алгоритмів, є застосування операторів у суперпозиції.

Щоб зрозуміти цю концепцію, необхідно враховувати, що класично складність обробки полягає в еволюції станів до знаходження того з мінімальними витратами. У квантовому плані складність полягає у збільшенні ймовірності стану з мінімальними витратами. Для еволюції класичного стану необхідно застосовувати оператори окремо та послідовно. Однак модифікацію квантових ймовірностей можна виконувати по всій суперпозиції, при цьому один оператор одночасно впливає на всі можливі стани. Якщо кількість операцій для збільшення ймовірності квантового стану менша за кількість операцій для еволюції класичного стану, то в алгоритмі буде квантова перевага.

Знаходження задач оптимізації, в яких необхідно виконати майже вичерпний пошук і, отже, які, ймовірно, матимуть квантову перевагу, не є складним [18]. Класично, вчені-інформатики визначили деякі з цих задач як дуже репрезентативні та протестували на них різні алгоритми. Потім, коли вони хотіли ефективно вирішити інші задачі, вони адаптували їх, щоб вони були максимально схожими на еталонну задачу, і таким чином застосували той самий алгоритм. Навіть пішовши далі, вони визначили певні іграшкові задачі, які були спрощенням реальних задач, але точно відтворювали складність задачі, що значно

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

полегшувало перехід від теоретичної задачі до реальної. Деякі з теоретичних задач, які були запропоновані відповідно до категорії задач, які вони вирішували, були наступними:

- Теоретична задача маршрутизації: задача знаходження найкращого маршруту між двома містами. Запропонована тестова задача - це задача комівояжера (TSP), де мандрівник починає подорож з одного міста, відвідує кілька міст один раз і опиняється в тому ж місті, звідки він почав свою подорож [38, 39].

- Теоретична задача конфігурації: задача знаходження найкращого балансу навантаження для невеликого простору. Запропонованим прикладом задачі була задача рюкзака. У цій задачі метою є заповнення рюкзака, який підтримує задану вагу, конфігурацією об'єктів, яка максимізує певну винагороду.

- Теоретичний штучний інтелект: У машинному навчанні існує багато процесів, які потребують оптимізації, таких як пошук гіпотез або оптимізація гіперпараметрів. У цьому сенсі такі задачі, як N-ферзь, використовувалися для тестування алгоритмів, які потім застосовувалися в складніших методах. Задача N-ферзя полягає в розподілі n ферзів на шаховій дошці n на n так, щоб вони не атакували одна одну.

Теоретична вибірка: Задача характеристики функції шляхом взяття лише кількох її вибірок. Наприклад, щоб знайти значення числа π , можна взяти квадрат і коло таким чином, щоб діаметр кола дорівнював стороні квадрата. Беручи випадкові вибірки та дивлячись, які з них знаходяться всередині квадрата, а які зовні, можна оцінити досить точне значення числа π , як показано на рисунку 1.5.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

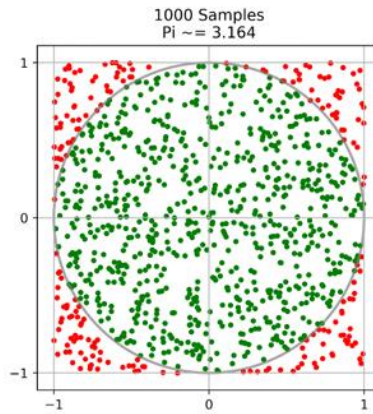


Рисунок 1.5 - Значення числа пі обчислюється на основі кількості зелених точок, що потрапляють у коло, та червоних точок, що випадають за його межі.

Це типовий приклад, що ілюструє метод Монте-Карло. Важливою рисою цих теоретичних проблем є їхня проста застосовність до практичних задач.

- **Задача маршрутизації:** Задача пошуку маршруту між двома або більше точками через серію проміжних точок – це задача, яку вирішує будь-яка компанія з перевезення вантажів, кур'єрська або транспортна компанія, щоб мінімізувати кількість транспортних засобів або пройденої відстані. Чим більша вирішена задача, чим більша кількість точок, транспортних засобів, людей або посилок, що перевозяться тощо, тим більший прибуток компанії. Існує безліч варіантів, таких як мінімізація відстані, часу, палива тощо.

- **Проблема конфігурації:** Проблема доставки певної кількості товарів у наборі контейнерів є поширеною для пакувальних компаній, міжнародних експедиторів або майже будь-якої компанії в галузі, відомої як доставка «останньої милі». Йдеться про розміщення більшої кількості товарів у меншому просторі. Більш ефективно вирішення цієї проблеми допомагає зменшити кількість або розмір контейнерів, наприклад.

- **Штучний інтелект:** Проблеми оптимізації у штучному інтелекті можуть варіюватися від вибору гіпотези, яка найкраще пояснює набір даних, більшого узагальнення та охоплення меншою кількістю пунктів, до вибору гіперпараметрів, які покращують продуктивність нейронної мережі для конкретної задачі. Наприклад, алгоритм навчання без учителя, k-середніх, вимагає в кожній ітерації

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

розподіляти центроїди та обчислювати відстані таким чином, щоб точки були розподілені якомога краще.

- Вибірка: Задачі вибірки широко використовуються на промисловому рівні, коли важко знати певний розподіл. Наприклад, у фінансовому секторі, щоб знати тенденцію активу або покращити аналіз ризиків. У цих випадках здатність вирішувати більші проблеми за короткий час дозволяє отримувати точніші короткострокові результати.

Досі інтерес до використання квантових алгоритмів S&S був загалом мотивованим, наступним кроком є більш детальне вивчення обох методів та пояснення актуальності цього типу алгоритмів, оскільки вони вписуються в задачі оптимізації.

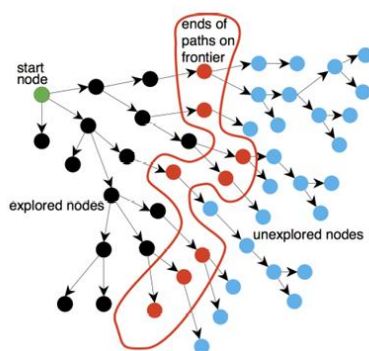


Рисунок 1.6 - Показано елементи пошуку, що починається з початкового вузла.

Під час пошуку вузли поділяються на різні категорії. Чорним кольором позначено вже обстежені вузли, які вже оцінено та для яких обчислено наступні вузли. Червоним кольором позначено вузли на межі, які оцінюються на цьому етапі. Нарешті, синім кольором позначено вузли, які ще не обстежено; рішення знаходиться серед цих вузлів.

1.3 Класичні алгоритми пошуку та їх обмеження

Алгоритми пошуку – це абстрактні методи для знаходження стану, що відповідає певним умовам, з множини. Залежно від характеристик задачі, може

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

бути використаний один або інші алгоритми. Основні типи пошуку показано на рисунку 1.7.

Оскільки це визначення може бути дещо абстрактним і важким для візуалізації, початкові алгоритми пошуку були проілюстровані у вигляді дерев. Ця метафора служить як дидактичним, так і практичним цілям, дозволяючи легко представити еволюцію алгоритму пошуку.

Таким чином, стан, у якому починається пошук, можна розглядати як вузол під назвою корінь. Цей кореневий вузол оцінюється, щоб визначити, чи є він вузлом, що представляє стан рішення. Якщо ні, він розширюється, щоб отримати своїх наступників. Коли вузол розширюється, він передається до списку відвіданих вузлів. Наступники переходять до списку розширених вузлів, а не відвіданих, з якого вибирається один для продовження пошуку. Таким чином, пошук по дереву - це процес відвідування вузла, розширення наступників та вибору наступника для повторення процесу.

Ці дерева пошуку можуть мати багато форм та характеристик. Можливо, один і той самий вузол можна досягти кількома шляхами, між вузлами можуть бути петлі, можуть бути вузли, які не мають дочірніх вузлів і відомі як листові вузли тощо. Важливо те, що під час пошуку алгоритм здатний уникнути двох фундаментальних проблем: застрягання в петлі та надлишкових шляхів. Для цього багато алгоритмів пошуку перевіряють, які вузли вже розгорнуті у списку розгорнутих вузлів та уникати їх внесення до списку вузлів для розгортання, списку вузлів-наступників.

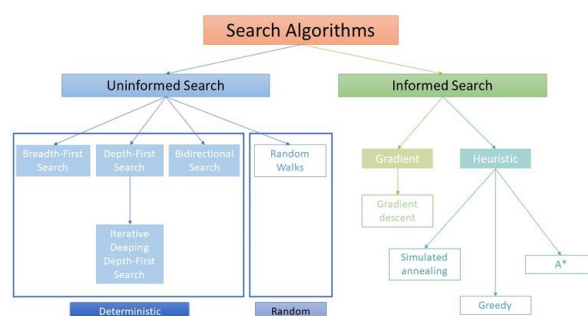


Рисунок 1.7 - Схема, на якій основні алгоритми пошуку розділені на категорії залежно від типу пошуку.

Залежно від того, чи використовує пошук якусь інформацію про проблему для керування, пошук може бути неінформованим або інформованим. У межах категорії неінформованого пошуку є детермінований пошук або випадкові алгоритми [19]. Серед алгоритмів інформованого пошуку інформація може бути надана за допомогою градієнта або евристики.

Ці алгоритми класифікуються за їхньою здатністю досягати концепцій оптимальності та повноти. Припускаючи, що кожне розширення вузла, покоління наступників, має вартість 1.

- **Оптимальність:** Оптимальність означає, що алгоритм гарантує знаходження рішення з мінімальною вартістю.
- **Повнота:** Повнота алгоритму гарантує, що він здатний знайти рішення незалежно від того, в якому вузлі простору станів він знаходиться.

Для огляду найважливіших алгоритмів пошуку, що існують та є актуальними для розробки квантових алгоритмів пошуку, буде проведено поділ на алгоритми пошуку з інформацією та без неї. У рамках цієї категорії будуть проаналізовані поняття повноти та оптимальності.

Неінформовані алгоритми пошуку характеризуються тим, що не мають інформації про те, наскільки близько чи далеко знаходиться цільовий стан. Цей тип алгоритму просто розширює вузли, доки один з них не буде знайдений як ціль. Повертаючись до схеми дерева, якість рішення буде визначатися відстанню від цільового вузла до кореня, ближче означає кращу якість. Вартість алгоритму буде визначатися кількістю вузлів, необхідних для розширення, доки рішення не буде знайдено [20].

Основні алгоритми неінформованого пошуку поділяються на послідовний або випадковий пошук, і кожна категорія має різні алгоритми.

На рисунку 1.8 вузли організовані за трьома рівнями глибини, кожен з яких представлений окремим відтінком синього, від світлішого до темнішого. Перед проходженням вузлів на третьому рівні глибини, всі вузли на другому рівні вичерпно відвідуються.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

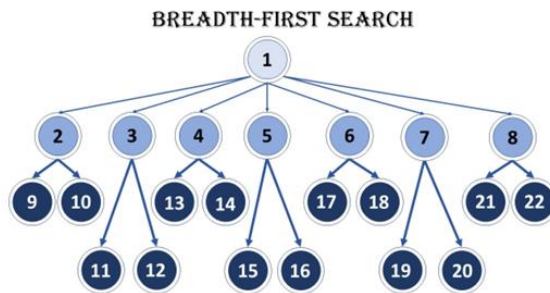


Рисунок 1.8 - Алгоритм пошуку в ширину систематично досліджує вузли на кожному рівні графа, перш ніж спуститися на глибші рівні.

- Пошук по ширині
- Пошук у глибину
- Ітеративне поглиблення. Пошук з обмеженою глибиною.
- Двонаправлений пошук
- Випадкові прогулянки

Пошук у ширину був першою спробою виконати пошук, за якого, навіть якщо не було інформації про стан цілі, можливо, не довелося б розширювати всі вузли, щоб її знайти. Цей тип пошуку починається з кореневого вузла, потім відвідує всі розширені вузли від кореня та зберігає наступників цих вузлів для наступного раунду. У наступному раунді він розширює всі вузли на відстані двох від кореня та зберігає їхніх наступників. Таким чином, він ітеративно проходить усе дерево пошуку в ширину, відвідуючи всі вузли на одному рівні, перш ніж перейти до наступного. Приклад можна побачити на рисунку 1.9.

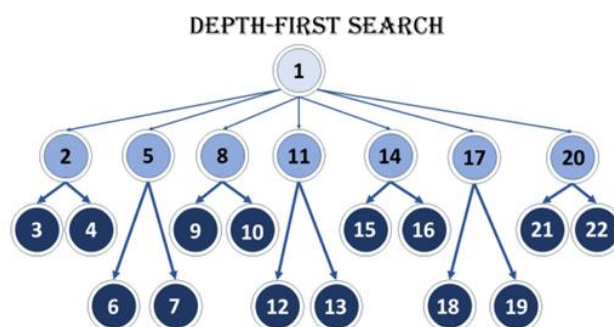


Рисунок 1.9 - Алгоритм пошуку в глибину досліджує вузли вздовж однієї гілки, доки не досягне кінцевого вузла, а потім переходить до наступної гілки.

Пошук у глибину спочатку пропонувався одночасно з пошуком у ширину, але пізніше про нього забули. Його головна проблема полягала в тому, що він не був ні повним, ні оптимальним алгоритмом. У випадку спроби знайти рішення в дуже глибокому дереві алгоритм втрачався.

Цей тип пошуку був спочатку запропонований, оскільки він задовольняє критерії повноти та оптимальності. Обходячи всі вузли на всіх рівнях дерева, можна бути впевненим, що в певний момент буде знайдено рішення, і воно буде оптимальним [21]. Оптимальність забезпечується тим, що спочатку відвідуються всі вузли мінімальної вартості, потім усі вузли наступного значення вартості тощо.

Хоча цей алгоритм є ідеальним з формальної точки зору, оскільки він відповідає двом дуже важливим критеріям: він завжди знаходить рішення, і це рішення має найвищу якість, на практиці він не є дуже корисним алгоритмом. Причиною такої відсутності корисності на практиці є кількість вузлів, необхідних для розширення. Це алгоритм вичерпного пошуку порівняння, тобто, поки рішення не буде знайдено, він розширить усі вузли.

Пошук у глибину дотримується протилежної парадигми до пошуку в ширину. Цей пошук завжди розширює найглибший вузол серед можливих наступників. Як тільки досягнуто вузла, який не має наступників, кінцевого вузла, він повертається, щоб продовжити розширення наступних найглибших вузлів. Приклад можна побачити на рисунку 1.10.

На відміну від пошуку в ширину, рівень глибини не має пріоритету. На відміну від підходу, зображеного на рисунку 10, цей алгоритм зосереджений на швидкому досягненні рішення, а не на вичерпному пошуку всіх вузлів.

Більше того, навіть якщо рішення було знайдено, неможливо було дізнатися, чи воно знаходиться на мінімальній глибині, і необхідно було продовжувати пошук майже так, ніби він виконувався вшир, доки інші рішення не були відкинуті.

Щоб вирішити проблему попереднього алгоритму та спробувати скористатися всіма його перевагами, була розроблена модифікація, відому як

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

ітеративний пошук з обмеженою глибиною поглиблення. Перевагою цієї нової версії алгоритму було те, що вона виконувала пошук в глибину до певної глибини [22]. Після досягнення цієї глибини алгоритм повертався до початку та починав знову з кореня або раніше позначеної глибини. Обмеження глибини ставало все більшим і більшим, все глибшим і глибшим, доки не було знайдено рішення. Приклад показано на рисунку 1.10 .

Тут вузли в одній гілці досліджуються до досягнення заданої межі глибини, позначеної пунктирною лінією (встановленої на 2 у цьому прикладі). Відмінною особливістю ітеративного пошуку в глибину, на відміну від попередніх алгоритмів, є його початкова поведінка, що нагадує пошук в ширину, як показано на рисунку 1.10. Однак, як тільки досягнуто межі, пошук стає вичерпним, відвідуючи всі вузли, глибина яких менша за межу.

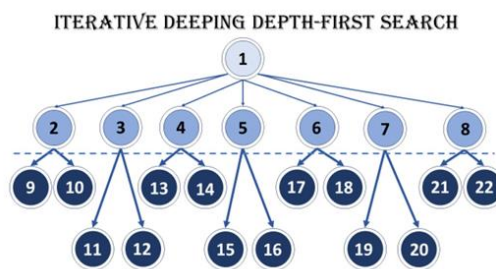


Рисунок 1.10 - Ця схема ілюструє порядок відвідувань вузлів в ітеративному пошуку в глибину.

Завдяки обмеженій глибині, на якій зупинявся алгоритм, уникалася проблема нескінченного пошуку. Це дозволило алгоритму бути повним, а також, у багатьох випадках, знаходити рішення швидше, ніж алгоритм пошуку в ширину. Це робить ітеративний алгоритм поглиблення в глибину кращим алгоритмом, коли потрібно було вибрати необізнаний алгоритм пошуку. Можна навіть розглядати цей алгоритм як пошук в ширину, якщо на кожній ітерації до граничної глибини пошуку додається одиниця [23]. У цьому випадку алгоритм виконуватиме пошук в ширину на кожному рівні. Навпаки, якщо максимально дозволена глибина більша, він знайде шуканий вузол раніше або, в гіршому

випадку, з тією ж складністю, що й пошук в ширину, завжди швидше або рівно.

Двонаправлений пошук виконує два пошуки одночасно. Один з початкового стану вперед, а інший з цільового стану назад. Таким чином, обидва пошуки зустрінуться в одній точці, і можна встановити шлях між початковим станом і цільовим станом.

Цей пошук є більш конкретним, оскільки він вимагає, щоб проблема мала певні характеристики. Цільовий стан має бути відомий, а проблема має полягати у знаходженні шляху до цього цільового стану. Крім того, цей алгоритм пошуку на кожному відвідуваному вузлі не повинен перевіряти, чи є це цільовий стан; він повинен перевіряти, чи цей вузол вже був розширений іншою частиною пошуку.

Цей алгоритм має повноту і, більше того, може бути оптимальним за умови, що певні перевірки виконуються при зустрічі як прямого, так і зворотного пошуку. Цей двонаправлений алгоритм пошуку експериментує з межами обох процесів пошуку, щоб спробувати мінімізувати кількість відвіданих станів і зменшити складність алгоритму.

На відміну від послідовного пошуку, існують алгоритми випадкового пошуку. Перевагою рандомізованого алгоритму є те, що час генерації наступника зводиться до мінімуму. Стохастичність дозволяє, замість обробки визначення найкращого вузла для відвідування, випадковим чином вибирати вузол для продовження пошуку. Недоліком цього пошуку є те, що він може повторювати вузли та повертати низькоякісні рішення.

Випадковий пошук є рішенням проблем, коли мало інформації про середовище або коли дуже складно та дорого оцінити, який тип пошуку працює найкраще. Крім того, поведінку випадкового пошуку дуже легко реалізувати.

Існують різні алгоритми випадкового пошуку. Найважливішим для цієї роботи є алгоритм випадкових блукань [24]. Випадкове блукання – це тип марковського процесу, в якому алгоритм досліджує простір станів повністю випадковим чином. Ідея цих алгоритмів полягає в тому, щоб здійснити пошук у просторі станів якомога швидше та без урахування повторення вузлів чи ефективності пошуку. Головною перевагою, окрім швидкості та стохастичності, є

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

те, що випадкове блукання ніколи не застрягає на локальних мінімумах чи максимумах і тому, з певною ймовірністю, створює рівномірне покриття простору станів. Основним недоліком є те, що немає інформації про якість рішення. Неможливо знати, чи є рішення оптимальним, чи наскільки воно далеке від оптимального рішення.

Випадкові шляхи мають різні застосування. Наприклад, вони дуже корисні для моделювання броунівського руху частинок або для виконання швидкого пошуку, щоб отримати перше рішення, яке пізніше можна уточнити, як у прогнозуванні зв'язків або напіваавторизованому навчанні. Хоча одне з їхніх основних застосувань, і причина, чому вони вивчаються в цій роботі, полягає в тому, що вони корисні як базовий метод для інших, більш складних алгоритмів пошуку на основі інформації.

Якщо під час пошуку розширення вузлів керується певними знаннями про середовище задачі, це вважається керованим пошуком. Головною перевагою цього типу пошуку є те, що він генерує вузли більш інтелектуально, в результаті чого менша кількість вузлів розширюється, доки не буде досягнуто рішення. Крім того, це дозволяє знати якість рішення, знати, чи є воно оптимальним, або відстань до оптимального рішення [25]. У цій роботі буде проаналізовано градієнтний пошук та евристичний пошук.

Інформований пошук наразі є найбільш широко використовуваним методом для вирішення задач пошуку, оптимізації та штучного інтелекту. Цей тип пошуку виявився корисним, оскільки, вибираючи вузли за певними критеріями, він значно зменшує досліджуваний простір пошуку. Залежно від інформації, яку алгоритм використовує для керування пошуком, розрізняють пошук з інформацією, незалежною від предметної області, та з інформацією, залежною від предметної області.

Пошуки з використанням інформації, незалежної від предметної області, не потребують знання конкретної інформації про проблему, лише про сам процес пошуку [26]. Наприклад, евристичний пошук на основі орієнтирів. Ці пошуки, незалежні від предметної області, корисні, якщо відомо мало інформації про

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

предметну область, але, як правило, є повільними та досліджують більший набір простору станів. На противагу цьому, пошуки, залежні від предметної області, хоча й вимагають досить точних знань про проблему, що розв'язується, дозволяють виконувати дуже точний пошук, який не досліджує стільки вузлів простору станів і, отже, є швидшими.

Градiєнтний пошук за спуском аналізує різниці значень функції значень між різними вузлами простору пошуку, щоб завжди просуватися до точки мінімальної енергії. У цьому випадку значення можна розуміти як функцію оцінки або функцію вартості, побудовану з урахуванням конкретних знань проблеми. Наприклад, у випадку нейронних мереж градієнт виводиться з різниці між прогнозами та фактичними даними, розрахованими за допомогою такої функції, як помилка найменших квадратів.

Градiєнтний спуск дотримується схем мінімізації з опуклим розподілом та одним абсолютним мінімумом, а в деяких випадках і кількома локальними мінімумами. Мета пошуку - якомога швидше знайти вузол, який представляє абсолютний мінімум функції, не застрягаючи в локальних мінімумах. Для цього виконується аналіз нахилу функції в кожному аналізованому вузлі, що дозволяє визначити напрямок, у якому потрібно продовжувати пошук.

Замість використання градієнта, для керування пошуком можна використовувати функцію, яка повертає оцінку. Ця схема відома як евристичний пошук. Евристика надає алгоритму інформацію про те, як він повинен керувати пошуком на основі відстані до цілі або часу пошуку. Існують різні типи евристичного пошуку та численні алгоритми. Цей пошук, розширюючи вузол, обчислює значення евристики для кожного з його наступників [27]. Потім, враховуючи евристичне значення кожного вузла, він вибирає той, що представляє найбільший інтерес, дотримуючись певного критерію.

Евристика - це функція $h(x)$, яка повертає значення, де x - вузол, який потрібно оцінити. Для побудови такої функції h обмеження задачі послаблюються та спрощуються [28]. Головне - зробити h максимально інформативною, але також швидкою для обчислення. Наприклад, якщо задача пошуку полягає в тому, щоб

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

знайти найшвидший шлях до центру лабіринту, однією з можливих евристик буде евклідова відстань, яка є спрощенням, оскільки вона ігнорує стіни лабіринту.

Жадібний пошук – це найпростіший тип евристичного пошуку. У цьому випадку алгоритм завжди вибиратиме вузол з найкращим евристичним значенням. Таким чином, алгоритм сліпо покладатиметься на здатність евристики наблизитися до рішення. Ця стратегія має дуже чітке обмеження: оскільки евристика є спрощенням задачі, будуть випадки, коли жадібний пошук потраплятиме в локальні мінімуми і не зможе вийти з них.

Алгоритм імітації відпалу використовує евристику, щоб вказати алгоритму, коли виконувати дослідницький пошук, а коли виконувати сфокусований пошук у досліджуваному просторі. Цей алгоритм, заснований на металургійних процесах, використовує алгоритм випадкового блукання. Спочатку він встановлює високу температуру, що робить вибір наступного вузла для відвідування повністю випадковим. Однак, зі збільшенням кількості ітерацій алгоритму температура знижується, а вибір вузлів менше визначається випадковістю, а більше зменшенням енергії вузлів. Цей алгоритм дозволяє поєднувати широке початкове дослідження з точним пошуком після заданої кількості ітерацій.

Найважливішим евристичним алгоритмом пошуку є A^* . Цей алгоритм подібний до жадібного пошуку, але замість вибору вузла з найвищим евристичним значенням він враховує вартість попередніх вузлів, розширених на цей момент. Таким чином, для вибору вузла для розширення використовується не $h(x)$, а $f(x)$. $f(x)$ дорівнює сумі $h(x)$, евристичного значення вузла, та $g(x)$, фактичної неевристичної вартості досягнення цього вузла. A^* вирішує проблему застрягання в локальних мінімумах і, крім того, забезпечує оптимальність пошуку.

У цьому розділі обговорюються два різні способи проведення обґрунтованого пошуку: один заснований на градієнті, а інший - на евристиці. Цей аналіз, який класично має певну актуальність, квантово набуває життєво важливого значення. Як буде видно далі, існують гібридні класично-квантові

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

алгоритми, які використовують класичний градієнтно-керований пошук для оптимізації квантових схем [29]. Було помічено, що цей пошук страждає від проблеми, яка називається безплідними плато (BP), що ускладнює знаходження абсолютного мінімуму. Можливим рішенням може бути використання евристично-керованого та неградієнтно-керованого пошуку, щоб уникнути феномену BP.

Як було пояснено, класичний пошук, як обґрунтований, так і необґрунтований, є широкою темою з численними варіантами, невелику частину яких було проаналізовано.

Хоча це правда, що ці алгоритми пошуку мають численні застосування, також необхідно розуміти, що вони мають чіткі обмеження. Ці обмеження настільки сильні, що призводять до дуже обмеженої масштабованості алгоритмів. Як результат, алгоритми пошуку повільно прогресують у вирішенні дедалі складніших проблем і дуже обмежені у вирішенні задач оптимізації в промисловому середовищі.

1.4 Висновок по розділу

У першому розділі було розглянуто теоретичні основи гібридних квантово-класичних алгоритмів у контексті їх застосування в комерційному програмному забезпеченні. Проаналізовано сутність та ключові принципи гібридних обчислень, що поєднують можливості квантових і класичних систем. Досліджено методи пошуку та оптимізації, які використовуються в обох підходах, а також розглянуто класичні алгоритми та їх обмеження в задачах високої складності. Отримані результати формують теоретичну базу для подальшого вивчення гібридних обчислювальних моделей.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ РЕАЛІЗАЦІЇ ГІБРИДНИХ АЛГОРИТМІВ

2.1 Квантові алгоритми для задач пошуку та оптимізації

Квантові алгоритми пошуку та вибірки (QS&S) представляють собою зміну парадигми у підході до вирішення задач, що включають певну форму квантових алгоритмів для пошуку або вибірки. Вони досить інтуїтивно зрозумілі та легко застосовуються до задач, де необхідно проаналізувати весь простір пошуку та, отже, використовувати вичерпні алгоритми. Однак, алгоритми QS&S також корисні, якщо для знаходження рішення необхідно дослідити важливу частину простору станів, і це просторово дуже велика частина.

Як обговорювалося вище, прокляття розмірності впливає на класичні алгоритми, оскільки вони повинні виконувати експоненціально більшу кількість операцій, навіть якщо розмірність даних збільшується меншою мірою. Зіткнувшись із цією проблемою, алгоритми QS&S цікаві тим, що їхня складність не так сильно залежить від розмірності даних, як буде видно з деяких алгоритмів, пояснених нижче [30]. Однак це лише інтуїція, якої дотримується спільнота і яка ще не доведена. Насправді, деякі автори закликають до обережності з цим типом алгоритмів і проблем, кажучи, що необхідно знайти кращі переваги, щоб квантові обчислення були справді корисними.

У квантових алгоритмах виграшною ставкою, здається, є зменшення кількості операцій, необхідних для знаходження рішення. Ця менша кількість операцій означає, що потрібно застосовувати менше квантових вентилів, ніж у класичних. Якщо це твердження виявиться правильним, це зрештою призведе до меншої кількості операцій, що, завдяки фізичним основам квантових обчислень, також призведе до зниження витрат енергії. Така економія енергії, що витрачається на виконання алгоритму для вирішення задачі оптимізації, наприклад, не є тривіальною, оскільки класичні комп'ютери, зокрема центри високопродуктивних обчислень (HPC), часто мають споживання енергії, яке є

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

непідйомним для обчислювальних центрів.

Проблеми, що розв'язуються алгоритмами QS&S, викликали великий інтерес з моменту зародження концепції квантових алгоритмів. Вважалося, що в задачах пошуку, вибірки та оптимізації QC матиме великий вплив. Тому частина академічних зусиль була присвячена цій темі, що призвело до появи численних методів та алгоритмів. У цьому підрозділі будуть проаналізовані основні з них. По-перше, буде проаналізовано оператор Гровера, оскільки це метод, який зазвичай використовується іншими алгоритмами [31]. На основі Гровера можна вивчати квантові блукання та їх реалізацію в квантовому алгоритмі Метрополіса-Гастінгса та квантових алгоритмах зворотного відстеження. Також будуть проаналізовані алгоритми, засновані на квантовому відпалі, через їх широке використання та легку реалізацію в адіабатичному обладнанні. Нарешті, буде проведено аналіз того, як ці алгоритми можуть бути адаптовані до гібридної класично-квантової парадигми, з особливою увагою до квантового алгоритму Метрополіса-Гастінгса (QM-H), який матиме особливе значення в цій роботі.

Слід зазначити, що ці алгоритми QS&S мають широкий спектр застосування як для теоретичних, так і для реальних задач. Хоча задачі, що наразі тестуються, обмежені. Враховуючи обмеження квантового апаратного забезпечення, яке стосується скоріше академічного, ніж промислового середовища, очікується, що в короткостроковій та середньостроковій перспективі застосування QS&S почнуть помічатися на промисловому рівні. У цій роботі практичне застосування алгоритмів QS&S буде показано як варіанти використання з доказами концепції з використанням реальних даних проблеми, яку вони вирішують, щоб трохи наблизити демонстрацію практичної корисності цих алгоритмів у цих проблемах.

Оператор Гровера не виник як такий, а був запропонований як алгоритм. Після глибокого вивчення науковою спільнотою [37] він був визнаний базовим методом для інших алгоритмів, що виходять за рамки початкових застосувань, задуманих його творцем.

Важливість алгоритму/оператора Гровера полягає в тому, що

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

фундаментальною операцією класичних комп'ютерів є зберігання та отримання інформації, тобто пошук і сортування інформації у великих обсягах даних. Операція пошуку була ізольована та вивчена з часів перших комп'ютерів [29]. Наприклад, одним із природних рішень для швидшого пошуку інформації є індекси. Класичні комп'ютери використовують усілякі індекси, які дозволяють здійснювати пошук у логарифмічній складності. Однак неможливо створити індекси для всіх наборів даних, оскільки це зробить пошук за індексами ще повільнішим, ніж сам пошук. Таким чином, єдиним рішенням є створення скороченого набору індексів для певних наборів даних та виконання більшості пошуків вичерпно.

Цей вичерпний пошук, який зазвичай називають «пошуком методом грубої сили», полягає у відвідуванні кожного можливого стану в просторі станів, що робить його дуже дорогою операцією [32]. Вартість алгоритму визначається як кількість запитів або кількість разів, коли елемент оцінюється. Якщо елементів N , вартість становитиме N . Гровер зрозумів, що можна виконати подібну операцію, але з вартістю $\propto \sqrt{N}$, використовуючи квантовий алгоритм.

Щоб побачити це на практичному та дуже спрощеному прикладі для пояснення концепції, якщо в бібліотеці є 100 мільйонів книг без будь-якого порядку, і потрібна конкретна книга, то в гіршому випадку необхідно прочитати назви 100 мільйонів книг. Це завдання практично нездійсненне для однієї людини. Використовуючи алгоритм Гровера, потрібно було б докласти зусиль, еквівалентних прочитанню лише 10 000 назв, щоб знайти потрібну, що, хоча й нудно, цілком здійсненно для однієї людини. Кажуть, що квантовий алгоритм докладає зусиль, еквівалентних прочитанню 10 000 назв, оскільки немає точної еквівалентності між класичним читанням назви та її перебуванням у квантовій суперпозиції, але аналогію складності можна встановити.

Для побудови свого алгоритму Гровер використовує оракул $f(x)$ за заданим $x \in \{0, 1, \dots, N - 1\}$. Цей оракул оцінює кожне значення x , кожен можливий стан системи (назва книги в попередньому прикладі). Потім оракул може приймати два значення: $f(x) = 1$, якщо значення x є бажаним значенням, $f(x) = 0$ в будь-якому

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

іншому випадку. Якщо цей оракул застосувати до суперпозиції всіх можливих значень x , визначених у рівнянні 2.1, суперпозиція створюється шляхом застосування вентиля Адамара до кожного кубіта, що представляє стан, який спочатку знаходиться в стані $|-\rangle$. Оскільки $N = 2^n$, оракул поверне 1 лише тоді, коли він збігається з правильним значенням, і завдання завершиться.

$$H^{\otimes n} | \cdot \rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \quad (2.1)$$

Здається, що алгоритм завершено, і він поверне 1 у правильному стані. Тим не менш, квантові обчислення не такі прості. Як тільки ймовірність правильного стану зросте, різниця між ймовірністю правильного стану та рештою станів буде настільки малою, що під час вимірювання схеми не буде розрізняти, який стан має найвищу ймовірність [33]. Це поширене явище в квантових обчисленнях у багатьох алгоритмах, і тому необхідно повторювати певний набір вентилів кілька разів, доки ймовірність правильного стану не стане достатньо великою відносно інших.

У випадку алгоритму Гровера це збільшення ймовірності позначеного стану досягається шляхом повторення двох операторів обертання. Ці два обертання взаємодіють з фазою станів, здійснюючи інверсію відносно середньої ймовірності. Перше обертання позначає стан, в якому $f(x)$ дорівнює 1, пов'язуючи з ним фазу, протилежну фазі інших станів. Друге обертання, відоме як оператор дифузії, виконує обертання початкового стану, перетворюючи різницю фаз між станами на амплітуди ймовірностей. Таким чином, позначений стан матиме вищу ймовірність бути виміряним.

Перше обертання в рівнянні 2.2, яке ототожнюється з оракулом, відповідає за позначку шуканого стану.

$$R|x\rangle = (-1)^{f(x)}|x\rangle \quad (2.2)$$

Результатом застосування оператора R , $oracle$, до суперпозиції станів у

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

рівнянні 2 є одиниця у всіх станах, крім шуканого стану, який дорівнює -1. Потім необхідно застосувати друге обертання, оператор дифузії. Оператор обертання має вигляд:

$$U_D := H^{\otimes n} (1 - 2|0\rangle\langle 0|) H^{\otimes n} \quad (2.3)$$

Цей оператор дифузії U_D бере всі можливі стани в суперпозиції, застосовуючи вентиль Адамара $H^{\otimes n}$ до відбиття від середнього значення, яке раніше було позначено як -1 ($(1 - 2|0\rangle\langle 0|) < 0$). Використовуючи спрощений приклад [65], дано стан ϕ з 4 можливими елементами, кожен з яких має ймовірність 4 та коефіцієнт, амплітуду ймовірності, 1.

$$|\psi\rangle = [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}] \quad (2.4)$$

Якщо позначений елемент знаходиться на другій позиції, то після застосування оракула в рівнянні результуючий стан буде наступним:

$$|\psi\rangle = [\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{1}{2}] \quad (2.5)$$

Середнє значення цього стану (ϕ) дорівнює 1. Застосовуючи рівняння 3 та 4 кілька разів до ϕ , керуючись рівнянням 4, отримуємо стан:

$$|\psi\rangle = [0, 1, 0, 0] \quad (2.6)$$

Одиниця знаходиться на другій позиції, що відповідає шуканому елементу. Це пояснювальний приклад, дуже спрощений, але він допомагає зрозуміти механізм, який використовує Гровер у своєму алгоритмі. Ці оператори відбиття та дифузії вийшли далеко за рамки алгоритму Гровера та стали справді важливими у спільноті алгоритмів квантового пошуку. Окрім їхнього застосування в задачах пошуку неіндексованих ключів, вони були застосовані до всіх тих алгоритмів, які

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

потребували посилення амплітуди певного стану .

2.2 Квантові прогулянки як інструмент обчислень

Класичні алгоритми пошуку, описані в розділі 2.1, мали спільну проблему: необхідність пошуку майже по всіх, якщо не по всіх, вузлах простору станів, щоб знайти стан розв'язку. Один із описаних алгоритмів, випадкові блукання, пропонував дуже швидке, але низькоякісне дослідження, щоб вони з певною ймовірністю досліджували область простору пошуку, достатньо велику, щоб знайти рішення з якістю, що наближається до оптимального.

Основна проблема випадкових блукань полягала в тому, що, хоча вони й швидко виконувалися, дослідження значної частини простору пошуку було дуже дорогим [34]. Більше того, таке неповторне дослідження не гарантувалося через стохастичну природу алгоритму. З цієї причини їм було необхідно виконати велику кількість повторень, щоб забезпечити правильне дослідження простору станів.

У квантовому випадку простір станів можна представити як суперпозицію станів, застосовуючи кожен операцію до всього простору пошуку одночасно. Це вирішує проблему обмежень дослідження випадкових блукань шляхом переходу до їхніх квантових аналогів, які називаються квантовими блуканнями. У випадку квантових блукань складність полягає не в дослідженні всіх можливих станів, а в тому, щоб розрізнити, який стан слід вибрати, оскільки операції застосовуються до всієї суперпозиції, а не до окремих станів.

Будь-який тип прогулянки легко уявити, якщо уявити граф G з вершинами (станами, S) та ребрами (переходами між станами, T). Цей граф перетинає агент, який прокладає шлях, випадково переходячи через S станів. Зі стану s в інший стан s' він переходить з ймовірністю, визначеною в ребрі t . Агент може виконати серію випадкових W кроків, доки не досягне стану, який вважатиметься кінцевим.

Початкова пропозиція Амбайніса щодо квантових блукань [36] , що використовує оператор Гровера , продемонструвала квадратичну перевагу над

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

класичними випадковими блуканнями [35]. Інша модель квантових блукань, базується на графах і представляє простір станів як двочастинковий граф. Сегеді визначає двочастинковий граф як два гільбертові простори , де кожен H представляє простір станів X . Це дві ідентичні копії оригінального простору. Потім визначаються два оператори для переміщення по графу:

$$U |j\rangle |\cdot\rangle := |j\rangle \sum_{i \in X} \sqrt{W_{ij}} |i\rangle = |j\rangle |p_j\rangle \quad (2.7)$$

$$V |\cdot\rangle |i\rangle := \sum_{j \in X} \sqrt{W_{ij}} |j\rangle |i\rangle = |p_i\rangle |i\rangle \quad (2.8)$$

Матриця W_{ij} можна розглядати як ваги ребер T на графіку, що представляє ймовірнісний перехід від $|i\rangle$ до $|j\rangle$. U та V пов'язані так:

$$SU = VS \quad (2.9)$$

Використовуючи обертання, подібні до запропонованих Гровером, оператор квантового блукання, визначений Сегеді, визначається як:

$$W = R_B R_A = U^{\dagger} S U R_A U^{\dagger} S U R_A \quad (2.10)$$

Квантова блука, про яку щойно йшлося, є квантовою блукаючою з дискретністю в часі. Це дозволяє виконувати кожен крок квантової блукання незалежно та налаштовувати його відповідно до моделі міркувань марковських агентів.

Як видно вище, класичні випадкові блукання були повільними через їх випадкове дослідження, яке могло повторювати кілька станів, чого не відбувається з їхнім квантовим аналогом. Квантові блукання особливо цікаві в тих задачах, де потрібно виконати практично вичерпний пошук простору станів. Так само, як Гровер говорив про вичерпний пошук у телефонній книзі або в прикладі

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

з бібліотекою, якщо квантове блукання потрібно виконати для задачі, яка потребує майже всіх вузлів простору станів, воно матиме перевагу в суперпозиції простору станів.

Однак, обмеження практичної реалізації перешкоджають реалізації теоретичних пропозицій щодо квантових блукань, зокрема ідеї Сегеді про представлення всього простору станів у двочастковому графі. Отже, деякі оператори в W необхідно модифікувати для досягнення ефективнішого представлення простору станів у квантових блуканнях.

Так само, як класичний пошук служить основою для інших алгоритмів, квантовий пошук на основі квантових блукань використовується в багатьох випадках, як для прискорення класичних алгоритмів у гібридних схемах, так і для побудови більших квантових алгоритмів [36]. У пояснюється, як шукати позначені елементи в ланцюгу Маркова за допомогою квантових блукань. У квантовий пошук на основі квантових блукань використовується для прискорення процесу використання алгоритму навчання з підкріпленням.

Як видно вище, різниця між алгоритмами пошуку та вибірки в квантових обчисленнях не така очевидна, як це виглядає класично. Те саме стосується і квантових блукань. Якщо в попередньому абзаці пояснювалися квантові блукання, що застосовуються до пошуку, їх також можна застосовувати до процесів вибірки. Найкращим прикладом у цьому напрямку є робота, які модифікували оператор Сегеді, замінивши двочастинковий граф монетою, та перетворили оператор W квантового блукання на оператор для побудови квантового алгоритму Метрополіса-Гастингса, який буде детально пояснено в наступному розділі [37].

2.3 Інші підходи до побудови гібридних квантово-класичних алгоритмів

Окрім алгоритмів, заснованих на квантових блуканнях, існують інші пропозиції щодо квантових обчислень для алгоритмів S&S. Розуміння обмежень

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

деяких із цих пропозицій посилює можливості алгоритмів на основі квантових блукань.

Одна з таких пропозицій - квантовий зворотний пошук. Візьмемо, наприклад, агента, який рухається лабіринтом, або проблему завантаження посилок у вантажівку, і корисним буде інший тип пошуку, який називається зворотним пошуком. Ключовим моментом алгоритмів зворотного пошуку є те, що вони виконують вичерпний пошук, але інтелектуально, уникаючи дослідження варіантів, які, як відомо, не призводять до дійсного рішення. Це допомагає підвищити ефективність порівняно з підходами грубої сили.

У цьому випадку запропонував застосувати методи квантових блукань для пришвидшення алгоритмів зворотного відстеження. Основна функція алгоритму зворотного відстеження полягає в тому, що він представляє простір станів так, ніби це дерево. Цей алгоритм розширює та досліджує гілки з різними станами, доки не буде знайдено стан розв'язку. Різниця з іншими алгоритмами полягає в здатності оцінювати вузли таким чином, щоб знати, чи всі наступники, всі гілки, згенеровані цим вузлом, наближаються до розв'язку чи ні. У випадку, якщо знайдено неперспективну гілку для пошуку розв'язку, він повертається, повертаючись, до кореня цієї гілки та бере іншу, щоб продовжити пошук, як показано на рисунку 2.1.

Основний недолік алгоритмів квантового зворотного відстеження є невід'ємною частиною їхньої переваги. Ці алгоритми успадковують обмеження представлення простору станів, описані вище для квантових блукань Сегеді. Вони навіть додають додаткові оператори, які, хоча й збільшують перевагу, також додають високу складність, щоб привести їх до реальної реалізації, або в симуляторі, або на апаратному забезпеченні, що може вирішити практичну задачу. Обмеження квантового зворотного відстеження залишається таким самим, як і у багатьох квантових алгоритмів; досі важко довести його перевагу, оскільки дуже малі квантові апаратні ресурси роблять його реалізацію неможливою ще кілька років.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

procedure EXPLORE(node n)
  if REJECT(n) then return
  if COMPLETE(n) then
    OUTPUT(n)
  for  $n_i$  : CHILDREN(n) do EXPLORE( $n_i$ )

```

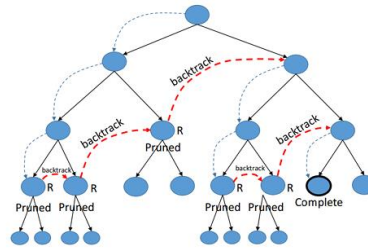


Рисунок 2.1 - Схема, що показує, як працює алгоритм зворотного відстеження.

Цей алгоритм відвідує вузли після пошуку в глибину. Якщо він виявляє, що рішення не знаходиться в досліджуваній гілці, він обрізає гілку та повертається до іншої гілки.

Відмінною парадигмою реалізації алгоритмів QS&S, відмінною від квантових блукань, є квантовий відпал. Ця пропозиція, яка майже одночасно є алгоритмом і пристроєм, оскільки апаратне забезпечення, на якому він виконується, призначене лише для виконання цього типу алгоритмів, натхненна класичною технікою імітованого відпалу. Ці алгоритми починаються з легкодоступної квантової конфігурації, а потім, поступово охолоджуючи систему, прагнуть «відпалити» оптимальне рішення. Основна ідея полягає в тому, щоб налаштувати систему в квантовому стані відповідно до початкових умов задачі, а потім дозволити системі еволюціонувати, доки вона не досягне мінімальної енергетичної конфігурації, яка відповідає розв'язку задачі.

Для кодування задач квантового відпалу можна використовувати різні методи. Одним з найпоширеніших підходів є формулювання їх як задач безумовної квадратичної бінарної оптимізації (QUBO). Такі представлення дозволяють двійкове кодування змінних та обмежень задачі. Таким чином, алгоритм ефективно розв'язує систему рівнянь, де потрібно максимізувати або мінімізувати певні значення, дотримуючись заданих обмежень.

Бінарне кодування QUBO може бути досить жорстким, коли йдеться про представлення складних структур, які можуть бути необхідними в задачі оптимізації. Тому його було зведено до методу вирішення дуже специфічних

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

проблем. Тим не менш, воно залишається альтернативою для розгляду певних задач QS&S. Масштабованість є ще одним обмеженням, тісно пов'язаним з відсутністю формальної демонстрації його переваги над класичними алгоритмами у великомасштабних задачах [38]. Хоча квантовий відпал вирішує задачі великої складності для класичних алгоритмів, забезпечуючи перевагу, масштабованість адіабатичних квантових апаратних пристроїв виявилася недостатньою. Ця відсутність масштабованості усуває перевагу для більших задач, що призводить до визнання важливості квантового відпалу в алгоритмах QS&S, але виключає його як дослідницький метод для побудови алгоритмів та вирішення проблем у цій роботі.

Інша перспектива для QS&S виникла, коли дослідники, які працювали над алгоритмами квантової оптимізації, усвідомили, що виконання чисто квантового алгоритму для вирішення задачі оптимізації швидше, ніж класичного, було надзвичайно складним через повільну еволюцію апаратних пристроїв. Тому, шукаючи золоту середину між складними чисто квантовими алгоритмами зі значними, але сумнівними перевагами, було задумано нове сімейство алгоритмів. Ці алгоритми, що поєднують класичну та квантову технології, відомі як квантово-натхненні алгоритми.

У категорії квантово-натхнених алгоритмів існує багато типів алгоритмів: квантові алгоритми, що виконуються на класичному обладнанні, класичні алгоритми, прискорені невеликими квантовими апаратними модулями, алгоритми, що використовують квантові методи, але реалізовані повністю класично, або алгоритми, що поєднують квантові та класичні модулі, як апаратні, так і програмні. Ця робота спеціально зосереджена на гібридних алгоритмах.

Гібридні алгоритми, що є центральним елементом цієї роботи, наразі створюють великі очікування стати парадигмою, завдяки якій квантові обчислення входять у промисловий сектор. Під егідою цієї концепції існують різні алгоритмічні схеми, але спільною рисою є їхня модульна природа, яка поєднує як класичні, так і квантові модулі.

Гібридні алгоритми мають різні підходи, але основні з них двоякі. З одного

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

боку, гібридні алгоритми можна вважати самоціллю. Вони побудовані відповідно до цієї парадигми, оскільки будуть операції, де квантові обчислення ніколи не будуть швидшими за класичні, і тому завжди краще поєднувати обидва [39]. У цьому підході квантові процесори (QPU) розробляються як процесорні модулі, що додаються до класичних суперкомп'ютерів, подібно до процесорів оперативної пам'яті або графічних процесорів. З іншого боку, гібридні алгоритми можна розглядати як проміжний крок, який необхідно зробити, оскільки квантові пристрої ще недостатньо зрілі, щоб підтримувати весь квантовий алгоритм. З розвитком технологій класичні модулі цих гібридних алгоритмів можуть бути поступово виведені з експлуатації.

Ця робота більше відповідає філософії другого підходу, який полягає в розробці гібридних алгоритмів, оскільки квантові пристрої наразі не мають можливості аналізувати розмір проблеми, яку потрібно вирішити. Однак алгоритми розроблені з достатньою масштабованістю, щоб після появи квантового обладнання можна було виконати весь алгоритм квантовим способом.

Здійснення цього проміжного кроку через класично-квантові гібридні обчислення – це спосіб спробувати отримати квантові переваги в задачах оптимізації набагато раніше. Очікується, що комп'ютери NISQ, спеціально розроблені для таких алгоритмів, будуть доступні раніше, ніж інші парадигми квантових комп'ютерів, такі як ранні відмовостійкі комп'ютери. Це передбачає можливість отримання переваги в певних задачах у середньостроковій перспективі. Однак, великомасштабні квантові комп'ютери, здатні виконувати суто квантові алгоритми, не очікуються в довгостроковій перспективі, без уточнення дати.

У цьому першому розділі, присвяченому квантовому пошуку та вибірці, буде запропоновано гібридний алгоритм, який поєднує технологію квантових блукань для створення квантового алгоритму Метрополіса-Гастингса, здатного вирішувати задачі оптимізації, тісно пов'язані з промисловими потребами. У наступних розділах будуть розглянуті різні випадки використання, де було застосовано пояснюваний гібридний алгоритм.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4 Висновок по розділу

У другому розділі було досліджено методи та інструменти реалізації гібридних квантово-класичних алгоритмів. Розглянуто квантові алгоритми, орієнтовані на задачі пошуку та оптимізації, які дозволяють підвищити ефективність обчислень. Проаналізовано квантові прогулянки як один із перспективних інструментів квантових обчислень. Також досліджено інші підходи до побудови гібридних алгоритмів, що забезпечують поєднання квантових і класичних обчислювальних моделей. У результаті визначено основні методи реалізації гібридних систем для подальшого використання у прикладному програмному забезпеченні.

РОЗДІЛ 3. РОЗРОБКА ТА ОЦІНКА ЕФЕКТИВНОСТІ ГІБРИДНИХ РІШЕНЬ У КОМЕРЦІЙНОМУ ПЗ

3.1 Особливості реалізації задач оптимізації у гібридних системах

Існує додаткова складність у вирішенні задач оптимізації – представлення. Деякі з них, такі як зв'язки між містами в задачі комівояжера (ЗКМ), можна легко

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

представити на комп'ютері [40]. Однак інші, такі як моделювання повітря навколо крила літака, створюють більш суттєву проблему, що вимагає використання спрощених моделей. У деяких випадках може знадобитися навіть надмірне спрощення через нерозв'язність початкової задачі. Так було історично з ґратчастими моделями згортання білків.

Існує кілька різних варіантів представлення для перетворення задачі на випадок, що розв'язується алгоритмом. У цій роботі обрано чотириелементне представлення. Елементи такі:

- Стани: Можливі значення, які система може прийняти для заданої задачі.
- Переходи: Можливі стани, що генеруються з кожного стану.
- Функція оцінювання: Функція для обчислення винагороди/цінності кожного стану.
- Мета: Мета задачі, мінімізація або максимізація функції оцінювання.

Щоб представити квантовий алгоритм, який ми вибрали в цій роботі, нам спочатку потрібно пояснити деякі класичні алгоритми. Класичні випадкові блукання не тільки дуже потужні, але й формують основу для дуже широко використовуваних алгоритмів Монте-Карло, які регулярно використовуються для задач оптимізації. Метод Монте-Карло полягає у випадковій вибірці простору станів для апроксимації функції. Він краще працює з більшою популяцією, оскільки похибка класично зменшується. Найважливішим аспектом методу Монте-Карло є те, що він служить основою для алгоритмів оптимізації.

Споріднений метод, також заснований на випадкових блуканнях та натхненний статистичною фізикою, - це алгоритм імітації відпалу. Основна концепція полягає в алгоритмі пошуку, який завжди приймає переходи, що знижують енергію, але з певною ймовірністю він також приймає переходи до вищих енергій. Ймовірність переходу до стану з вищою енергією на початку виконання висока, оскільки імітована температура починається з теплішої. Однак, у міру виконання кроків алгоритм охолоджується, і ймовірність зменшується. Цей процес допомагає алгоритму досліджувати багато станів на початку, уникаючи

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

локальних мінімумів. Через це алгоритм повільно сходиться до стану мінімальної енергії.

Поєднання випадкової вибірки методом Монте-Карло з розподілу ймовірностей та керованого стохастичного пошуку випадкових блукань та імітованого відпалу призводить до алгоритму під назвою Метрополіс-Гастінгс. Він використовується для апроксимації розподілу ймовірностей p_x шляхом його змішування з випадковим блуканням W , *доки* не буде досягнуто рівноваги, $Wn_x = p_x$.

Алгоритм Метрополіса-Гастінгса вимагає трьох методів: (i) процедури вибірки станів ініціалізації, (ii) процедури пропозиції переходів між станами та (iii) функції оцінки, яка оцінює, наскільки добрим є заданий стан. Останню часто називають «енергією» E через її зв'язок зі статистичною фізикою. Вона визначатиме ймовірність прийняття переходу, запропонованого в точці (ii), $\min(1, \exp(-\beta \Delta E))$, де β – це параметр, який називається оберненою температурою.

Існує квантова версія випадкових блукань. Квантові блукання також можна розуміти як узагальнення алгоритму Гровера. Перша пропозиція, запропонована Амбайнісом, була обмежена графами Джонсона. Невдовзі Сегеді представив двочасткові квантові блукання, узагальнювані на будь-яку ергодичну ланцюгову задачу.

Можна показати, що обидва пропонують квадратичне прискорення, подібне до прискорення Гровера, щодо часу влучання, іншими словами, щодо часу, необхідного для знаходження позначеного елемента. Останній квантовий шлях широко використовується, наприклад, у контексті алгоритмів Quantum Metropolis. Також пропозиція Сегеді знайшла різноманітне застосування.

Більшість попередніх квантових алгоритмів Metropolis передбачали повільно змінний параметр β і часто оцінювання фази для еволюції стану від рівномірної суперпозиції до цільового стаціонарного розподілу. Однак це відрізняється від того, як працюють класичні алгоритми, *де* змінюється набагато швидше, і жодних додаткових методів, окрім випадкових блукань, не використовується. З цієї причини Лем'є та ін. запропонували квантову версію

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

алгоритму Метрополіса-Гастінгса, яка евристично використовує квантові блукання, подібно до того, як випадкові блукання використовуються класично.

У цій роботі ми обговорюємо та аналізуємо поведінку квантового алгоритму Метрополіса-Гастінгса (МН) в задачі оптимізації, що виникає в галузі штучного інтелекту (ШІ). Щоб протестувати алгоритм МН та полегшити використання квантового МН іншими користувачами, ми реалізували програмний інструмент під назвою Quantum Metropolis Solver (QMS). Наш інструмент використовує як вхідні дані опис задачі оптимізації та генерує рішення з мінімальною вартістю. Крім того, він має додаткові функції, такі як побудова графіків, класичне порівняння рішень та глибокий аналіз рішень квантового алгоритму МН. Інструмент орієнтований на філософію з відкритим вихідним кодом та був написаний на Python з використанням модулів Qiskit.

Застосування QMS має три аспекти. По-перше, його можна використовувати як метричний інструмент для перевірки продуктивності квантового алгоритму МН у конкретній пошуковій задачі. Крім того, QMS можна інтегрувати в гібридні класично-квантові алгоритми, оскільки вхідні дані QMS є класичним описом задачі, але вони можуть генерувати класичний або квантовий вихід. Нарешті, ми порівнюємо класичні та квантові варіанти для обчислювальної оцінки потенційних квантових переваг.

На продуктивність класичного алгоритму МН сильно впливають два фактори: час змішування (МТ) ланцюга Маркова, визначеного в рівнянні 1, та похибка Монте-Карло. Ці два аспекти визначають кількість необхідних кроків для отримання прийняттого рішення, і вони є ключовими у створенні квантової версії, яка покращує обидва. Час змішування ланцюга Маркова – це час, необхідний для досягнення ланцюгом Маркова стаціонарного розподілу. У роботі пояснюється важливість цієї оптимізації значень. На противагу цьому, помилка Монте-Карло – це відстань між розрахованим розподілом на кроці N та стаціонарним розподілом. У роботі наведено приклади зменшення помилки методів Монте-Карло.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Час змішування $MT(c)$ можна інтерпретувати як мінімальну кількість кроків t ланцюга Маркова, які слід застосувати до будь-якого початкового розподілу (n_0) , щоб результат був ϵ -близьким до стаціонарного розподілу (n) на відстані D .

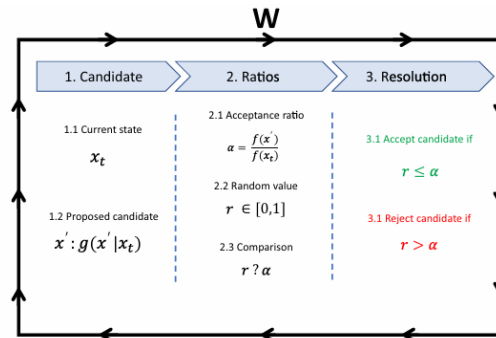


Рисунок 3.1 - Етапи алгоритму Метрополіс-Гастінгса

Першим кроком є вибір нового кандидата x , пов'язаного з x_t за допомогою функції розподілу $g(x_t)$. Після генерації кандидата необхідно обчислити числове значення α , щоб визначити, наскільки він кращий або гірший за поточний стан. Спочатку генерується випадкове число r у діапазоні від 0 до 1 включно. Потім порівнюються обидва значення - α та r . Якщо α більше або дорівнює r , зміна приймається і x стає x_t ; в іншому випадку x відкидається, і нове x буде обчислено знову в тому ж x_t . Цей процес повторюється, поки не буде досягнуто фіксованої кількості w або еволюціонований розподіл, з часом, s_0 не стане меншим за значення, відмінне від цільового розподілу π .

Генерація наступників у МН визначається функцією розподілу ймовірностей, її також можна використовувати для вибірки невідомого стаціонарного розподілу ланцюга Маркова. Алгоритм може генерувати вибірки з розподілу ймовірностей, які можна використовувати для апроксимації функції щільності ймовірності. Зокрема, його можна застосовувати до задач, які забороняють повне перерахування всіх шляхів. Знову ж таки, ця вибірка МН може бути квантово версійована природним шляхом, оскільки квантова схема МН може виконуватися багаторазово, доки не буде отримано розподіл результату кожного пострілу виконання.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Як пояснювалося вище, обмежувальним фактором для збільшення швидкості за допомогою алгоритму МН є час змішування та зменшення помилок. Йдучи ще далі, можна визначити три моменти, які ми можемо оптимізувати у виконанні МН: зменшити кількість оцінюваних станів, уникнути застрягання на локальних мінімумах та швидше оцінювати стани. Квантові обчислення можуть допомогти в цих моментах, оскільки розрив власних значень квантового блукання квадратично менший за класичний розрив власних значень з формулою $\Delta = \delta^{1/2}$, де δ – розрив власних значень класичного блукання, а Δ – фазовий розрив квантового блукання. Хоча існують класичні підходи до вирішення цієї проблеми, алгоритм Гровера та квантові блукання додають амплітуди замість ймовірностей, і їхня різниця зрештою проявляється як квадратичне прискорення.

3.2 Гібридна квантово-класична модифікація методу

Квантова версія алгоритму Метрополіса-Гастінгса використовує знижену складність завдяки меншому розриву власних значень порівняно з його класичними аналогами. Цей факт допомагає швидше визначити стан мінімальної енергії. Суть цієї переваги полягає в застосуванні оператора квантового блукання до початкової рівномірної суперпозиції всіх можливих станів, що зменшує кількість кроків для змішування ланцюга.

За допомогою двох відбиттів, будує квантову блукання на двочастковому графі. Однак у цій роботі ми замінюємо двочастковий граф монетою $|c\rangle$ через ізоморфізм, що створює запутаність зі станами $|s\rangle$. Це призводить до квантової блукання $|\Phi\rangle = |s, c\rangle$, де стани представлені як суперпозиція $|s\rangle$ можливих станів.

$$|s\rangle = \{ \sum_{\{x \in N\}} \alpha_x |x\rangle \} \in \{N\}_s \quad (3.1)$$

Квантове блукання Сегеді використовується як основа для побудови схеми квантового алгоритму Метрополіса-Гастінгса. У цій роботі ми показуємо складність реалізації W у квантовій схемі з використанням унітарних операторів.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Основною проблемою є застосування оператора та його оберненого (унітарного оператора) в кожному W , оскільки оберненість оператора залежить від того, чи зміна прийнято чи ні. В ідеальному випадку необхідно було б застосовувати умовний обернений оператор на кожному кроці. В алгоритмі МН після запропонованої зміни є два варіанти: прийняти або відхилити запропоновану зміну, і ця подвійність є ключовою проблемою у створенні унітарного оператора. Рішення, являє собою інший унітарний оператор для W , який ізоморфний вихідному оператору блукання Сегеді U_W та представлений як

$$\tilde{U} = R V^{\dagger} B^{\dagger} F B V \quad (3.2)$$

де R – оператор відбиття, V – оператор підготовки ходу, B – оператор монети, а F – оператор спін-фліпу. Оператори F , B та V можна розглядати як пряме квантування класичних операцій в алгоритмі МН, що полягає в пропонуванні кандидата та його прийнятті/відхиленні залежно від того, мінімізує він чи максимізує функцію значення. З іншого боку, R – єдиний чисто квантовий оператор для квантового алгоритму МН, який використовується для генерації обертань, подібних до обертань Гровера, з потенціалом демонструвати поліноміальну перевагу. Всі ці оператори спрощують реалізацію до схеми, якщо порівнювати з реалізацією W Сегеді.

Ми представляємо програмний фреймворк, ядром якого є схема та створюємо бібліотеку навколо квантового алгоритму Метрополіса-Гастингса, що дозволяє будь-якому користувачеві вирішувати задачі оптимізації за допомогою квантового алгоритму М-Н. Ми називаємо цю програмну архітектуру Квантовим Розв'язувачем Метрополіса (QMS). Ми реалізуємо її в квантовому симуляторі, що працює на класичному комп'ютері. QMS претендує на те, щоб бути на крок далі еволюції квантових блукань та квантового алгоритму МН.

Існує безліч статей про квантові блукання. У нашому випадку ми хочемо продовжити шлях розвитку, який, на нашу думку, має два основні попередні пункти (і багато проміжних): пропозицію квантових блукань Сегеді та

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

квантування квантового алгоритму МН Лем'є та ін. Починаючи із запропонованих U - операторів, ми модифікували цей алгоритм таким чином, щоб він міг вирішувати задачі оптимізації. Для цього ми реалізували всі схеми в бібліотеці qiskit та модифікували її для можливості зчитування даних з оракула. Ми також модифікували способи ініціалізації схеми, оптимізували певні операції та включили опцію роботи з різними бета-версіями, що дозволяє тестувати алгоритм з квантовим блуканням поза рівновагою. Ми створили квантову модель МН, здатну вирішувати будь-яку задачу оптимізації, визначену як набір станів та значень, пов'язаних з цими станами. Наш основний внесок полягає в наступному:

- Програмний інструмент: Ми надаємо громаді просте у використанні програмне забезпечення для вирішення проблем, у яких Метрополіс-Гастингс має доведену перевагу.

- Дослідження квантового Метрополіса-Гастингса для задачі оптимізації, пов'язаної зі штучним інтелектом: Ми

надають докази того, що квантова перевага квантових блукань та алгоритму QM-N може бути застосована до штучного інтелекту для оптимізації процесу пошуку, властивого будь-якій техніці ШІ. Тематичним дослідженням для підтвердження цієї ідеї є задача N-Queen, яка неодноразово використовувалася і досі використовується як орієнтир для нових класичних алгоритмів ШІ.

- Закон масштабування: Ми аналізуємо ефективність QMS, знаходячи закон масштабування для задачі N-Queen. Таким чином, ми додаємо ще один приклад застосування квантового алгоритму МН до раніше проаналізованого тематичного дослідження задачі згортання білків.

- Порівняння різних реалізацій квантових блукання: Ми порівняли три різні реалізації квантових блукань та квантового методу Метрополіса-Гастингса для задачі N-Квін. Пропозиція використовує двочастинковий граф для пошуку в просторі станів. Цей алгоритм пізніше був модифікований у роботі з дискретними операторами та заміною n -вимірного пошуку на бінарний пошук, що виконується за допомогою монети.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Нарешті, ми також порівняли результати з порядком операторів, відмінним від методу кубітизації.

- Квантовий вихід з рівноваги: У QMS ми включили опцію мати різні бета-коефіцієнти. графіки. Це ϵ з'являється в рівнянні і пов'язане з ймовірністю прийняття запропонованої зміни. Мета зміни значення бета під час виконання квантової блукання полягає в тому, щоб квантова блукання вийшла з рівноваги та швидше перемішала ланцюг Маркова, ніж звичайна квантова блукання.

- Oracle: Ми розробили підпрограму для завантаження всіх даних про енергію/значення в квантовий блукальний процес. Це дозволяє нам виконувати більші задачі з попередньо розрахованими значеннями.

- Дельти: Ми попередньо обробляємо всі значення, пов'язані зі станами, щоб зменшити складність схеми та кубітів, необхідних для їх представлення. Ми обчислюємо різницю між значеннями станів і кодуємо її в допоміжні регістри оракула.

- Ініціалізації: Ми розробили наш програмний інструмент для створення квантових схем для різних ініціалізацій задачі. Наприклад, ми можемо почати з рівноймовірного розподілу станів, навіть якщо кількість станів не є степенем 2, або з функції, визначеної користувачем.

Ми можемо працювати із задачами з граничними умовами або без них тощо.

Основна мотивація для впровадження QMS полягає в тому, щоб надати науковій та промисловій спільноті випробувальний майданчик для вирішення задач оптимізації за допомогою квантових алгоритмів, включаючи порівняння з їх класичним аналогом. Цей програмний інструмент абстрагує притаманну складність реалізації квантових алгоритмів. Користувачі можуть визначити проблему для вирішення за допомогою лише вхідного файлу. У цьому файлі проблема визначається як набір станів та пов'язаних з ними витрат. Потім QMS поверне стан з найменшими витратами як рішення. Передбачається, що стани пов'язані між собою в графі, щоб можна було застосувати алгоритм ланцюга Маркова.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

QMS був розроблений не лише для того, щоб запропонувати кінцевий результат, але й для відображення статистичних даних, які допомагають зрозуміти продуктивність алгоритму. Існує три можливі виходи QMS: стан мінімальних витрат, результат TTS або розподіл ймовірностей. Акцентуючи увагу на тестовому стенді, QMS дозволяє порівняти TTS між квантовим алгоритмом та його класичною версією. У такому режимі та сама задача виконується в класичному алгоритмі Метрополіса-Гастингса та квантовому, і показано обидва мінімальні TTS, досягнуті кожним з них. Крім того, у (9) генерується графік з кривою TTS як функція t для квантового та класичного алгоритмів МН.

Класичний модуль МГ для порівняння з квантовим МГ також був реалізований нами. Він є точним аналогом квантового Метрополіса-Гастингса. Ми реалізували класичне випадкове блукання за алгоритмом МГ, щоб вирішити, чи прийняти зміну чи ні. Він приймає ті ж вхідні дані, що й квантове блукання, і виконує пошук з тією ж кількістю кроків (W). Потім класичне МГ виконується n разів , щоб також отримати ймовірності наявності того чи іншого стану в результаті. Тому ми виконуємо класичне МГ w n разів.

Наш програмний інструмент, детально описаний на рис. 3.2, був задуманий як зручна бібліотека, що спрощує його використання. Користувач просто визначає проблему у вхідному JSON, файлі, який зберігає прості структури даних та об'єкти у стандартному форматі. Потім можна налаштувати виконання QMS за допомогою файлу конфігурації, де визначено деякі параметри. Наприклад, кількість кроків, яка еквівалентна кількості разів застосування оператора W , можна налаштувати, просто змінивши значення параметрів *initial_step* та *final_step*. Потім буде проаналізовано будь-яку кількість кроків між ними.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

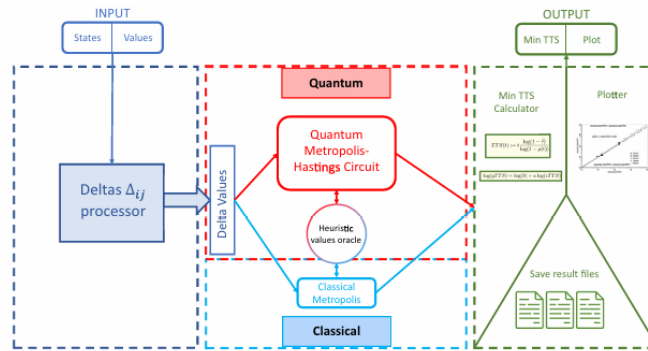


Рисунок 3.2 - Схема архітектури QMS. Система отримує файл у форматі JSON з описом можливих станів та відповідних їм значень.

Основою QMS є реалізація схеми, запропонованої в Lemieux et al. (2020) з деякими оптимізаціями та модифікаціями. З рівняння 5 оператори ті самі:

- V – це *підготовка_до_переміщення*: цей оператор створює суперпозицію всіх можливих переходів станів.
- B – це *підготовка_монети*: цей оператор створює суперпозицію - монети та обертає монету в тих станах, де приймається решта.
- F - умовний_переміщення: цей оператор виконує зміну станів, у яких оберталася монета. Це крок прийняття/відхилення МН.
- R – відбиття: Аналогічно, відбиття Гровера використовується для посилення амплітуди позначених станів.

Однак, представлення станів (також званих регістрами), виконане в попередній роботі, було оптимізовано. У попередній роботі використовувалося представлення з N кубітів для представлення M можливих регістрів для переміщення, оскільки $N = M$, тобто один кубіт для кожного регістра. З таким представленням усі регістри мали значення 0, крім регістра для переміщення, який мав значення 1. Ми замінюємо це унарне представлення в регістрі переміщення на двійкове представлення, використовуючи лише $\log^2 (M)$ кубітів , де M - кількість регістрів. Ця зміна дозволяє нам зменшити кількість кубітів у всій системі. Крім того, евристика, яка керує алгоритмом, базується на оберненому температурному параметрі Ψ .

Вхідний файл QMS – це набір кортежів (стан, вартість). Стани

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

представлені у двійковій системі числення, починаючи з 0, а вартість – це дійсне число. Прикладом кортежу може бути [(101): 65.53], де 101 – це стани у двійковій системі числення, а 65.53 – вартість.

Для внутрішнього представлення QMS вимагає кубітів $\lceil \log_2(n) \rceil$ для представлення n вхідних станів. Вартість, пов'язана з кожним станом, не представлена безпосередньо; навпаки, у QMS зберігається різниця вартості між пов'язаними станами, представлена як A . A_{ij} встановлюється на 1, якщо вартість стану-кандидата j нижча за вартість стану i . В іншому випадку A_{ij} зберігає кодифікацію різниці вартості між станами, ця кодифікація відповідає ймовірності зміни з урахуванням в кроку. Потім усі ймовірності зберігаються в QRAM як оракул, до якого можна отримати доступ у кожному W .

Ми визначаємо чотири регістри для зберігання всієї інформації в СУЯ:

- Стани $|-\rangle_S$: Містить кожен можливий стан, на який впливає підготовка переміщення оператора (V).
- Переміщення $|-\rangle_M$: Містить стан кандидата на переміщення, на який впливає підготовка оператора переміщення (V).
- Монета $|-\rangle_c$: Монета, на яку впливає підготовка монети оператором (B).
- Oracle $|-\rangle_o$: Містить ймовірність прийняття між усіма зв'язаними станами, на які впливає умовний переміщення операторів (F) та відбиття (R).

Регістр переміщення $|-\rangle_M$ можна розділити на два регістри: регістр ідентифікатора переміщення $|-\rangle_{Mj}$, який вказує, який стан буде змінено, та регістр значення переміщення $|-\rangle_{Mv}$, який вказує, як регістр має бути змінений (+1, -1, поміна місцями ліворуч, поміна місцями праворуч тощо).

У будь-якій задачі оптимізації критично важливо вибрати розподіл початкових станів. Розрив між початковим станом і цільовим станом визначає час виконання або якість рішення. QMS отримує вхідний набір станів, з яких вона повинна знайти найменш енергійний. QMS приймає початкову точку та застосовує оператор W кількість разів, визначену користувачем. Стан, досягнутий після застосування операторів W , є рішенням, що повертається QMS. З цієї

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

причини наша бібліотека дозволяє використовувати різні початкові стани, щоб надати користувачеві свободу вибору, з якої точки слід розпочати пошук. Можливо навіть вибрати не просто точку, а розподіл ймовірностей, який визначатиме початкову точку для пошуку. Підготовка такого розподілу ймовірностей здійснюється за допомогою процесу, який називається ініціалізацією.

- **Виправлено:** QMS починає пошук у стані, вибраному користувачем. Ця опція корисна для задач уточнення, в яких існує перше наближене рішення, а оптимізація полягає в пошуку більш якісного рішення, близького до початкового наближення.

- **Випадковий:** СМЯ починається з рівномірної суперпозиції станів, що еквівалентно випадковому стану, оскільки ніхто не має більшої ймовірності бути обраним, ніж інші.

Ініціалізацію СМК також можна класифікувати за способом генерації станів, а саме:

- **Послідовний:** Цей режим генерує кандидатів послідовно, просто додаючи або віднімаючи одну одиницю від координати. Наприклад, якщо задача - пішак, що переміщується на шахову дошку, стани можна представити як рядок і стовпець, зайняті пішаком (стовпець, рядок). Отже, якщо пішак знаходиться в (4, 5), одним із можливих кандидатів є додавання 1 позиції в стовпці, що призведе до стану (5, 5).

- **Циркуляр:** Цей варіант дозволяє з'єднувати прикордонні стани з періодичними граничними умовами. Наприклад, якщо попередній пішак знаходиться в позиції (5, 7), тобто у верхньому ряду, можна додати один ряд, щоб розмістити пішака в нижньому ряду, стан (5, 0).

- **Некруговий:** Цей параметр не дозволяє з'єднувати граничні стани, тому він налаштований з неперіодичними граничними умовами. Наприклад, якщо пішак знаходиться на межі рядка (5, 7), не дозволяється підсумовувати 1 у рядках.

- **Обмін:** Цей режим генерує кандидатів, які обмінюються координатами, і не допускає колізій. Наприклад, якщо на дошці розміщено 3 ферзі, і кожна ферзь

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

знаходиться в одному стовпці, то в одному стовпці немає двох ферзів. Стан представлятиме ряд ферзів. Один з можливих станів - (1, 0, 2), перша ферзь у ряду 1, друга ферзь у ряду 0 тощо. Кандидати генеруються шляхом обміну позиціями ферзів; наприклад, кандидат, який поміняє місцями першу та другу ферзі, призведе до (0, 1, 2), що означає першу ферзь у ряду 0, другу ферзь у ряду 1 тощо.

СУЯ – це програмний інструмент з дизайном «білої скриньки» та модульною архітектурою. Це означає, що він складається з агрегації модулів, які отримують вхідні дані, обробляють їх та передають результат іншим модулям. Ці модулі можна аналізувати, змінювати або навіть замінювати іншими модулями з тим самим інтерфейсом даних, дотримуючись аналогічних правил формату введення/виведення. Ця відкрита та гнучка архітектура є перевагою, яка може покращити використання інструменту спільнотою, оскільки його легко зрозуміти, включаючи нові функції, які можуть бути додані в майбутньому, або виправлення помилок. Архітектура СУЯ детально описана на рис. 3.2 .

Вся архітектура була реалізована за допомогою Python3, оскільки це мова з відкритим вихідним кодом, яка дуже використовується спільнотою розробників. Квантовий модуль викликає Qiskit та Qiskit-Aer для квантового моделювання на класичному комп'ютері. Qiskit також є модулем Python з відкритим вихідним кодом, який спрощує створення та моделювання схем, і він довів хорошу продуктивність з великими схемами з більш ніж 25 кубітами та багатопотоковим виконанням.

Будь-який алгоритм машинного навчання (ML) змінює свій внутрішній стан та представлення світу після обговорення. Цей процес базується на міркуваннях про вхідні дані для отримання моделі знань проблеми. Під час міркування система генерує різні гіпотези для пояснення середовища та виконання правильної дії. Наприклад, коригування ваг зв'язків у штучній нейронній мережі або зміна значення дії в агенті навчання з підкріпленням.

Етапи генерації та вибору гіпотез вимагають швидкого пошуку в просторі станів (просторі гіпотез), щоб оцінити, яка з усіх доступних гіпотез краще відповідає проблемі, яку потрібно розв'язати. Цей процес пошуку є одним із

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

вузьких місць будь-якого алгоритму машинного навчання, і квантові обчислення можуть його пришвидшити. З цієї причини ми розглядаємо QMS як допомогу в області штучного інтелекту, покращуючи продуктивність процесу пошуку. Ми вирішили перевірити наш інструмент на проблемі, яка широко використовується як еталон для ШІ, - проблемі N-Queen. Оскільки ця проблема, по суті, є проблемою пошуку з численними подібностями до пошуку алгоритму ШІ, будь-який алгоритм, який отримує хорошу продуктивність в N-Queen, може бути легко адаптований до інших проблем ШІ на основі пошуку, наприклад, пошуку гіпотез в алгоритмі машинного навчання.

Задача N-ферзів є похідним від класичної шахової задачі. Цільовий стан N-ферзів – це шахова дошка з n рядків та n стовпців з n ферзями, такими що жоден ферзь не атакує інших. Отже, немає двох ферзів в одному рядку, стовпці або діагоналі, як показано на рис. 3.3. Для задачі N-ферзів було запропоновано багато рішень.

Проблему N-ферзів досліджували багато авторів як NP-повну задачу. Однак ця категоризація ще не була чітко продемонстрована. Дослідження, яке містить найдетальніший опис складності задачі N-ферзів, ставить її за межі P -класу. Однак показали, що N-ферзеве доповнення, варіація N-ферзів, в якій деякі ферзі розміщуються заздалегідь, а завдання полягає в тому, щоб розмістити інших, є NP-повною задачею. Ця робота була використана Torggler для пропозиції квантового рішення для іншого варіанта N-ферзів, з виключеними діагоналями, яке також є NP-повним.

Такий вид міркувань, пошук гіпотези, яка пояснює та узагальнює вхідні дані, можна розглядати як абстракцію загальних знань з конкретних прикладів, і він відомий як індуктивні міркування. Як пояснюється в роботі, метою індуктивних міркувань є пошук гіпотези з найкращим балансом між класифікацією існуючих прикладів та узагальненням нових прикладів. Пошук цієї гіпотези з найкращим узагальненням/класифікацією винагороди еквівалентний пошуку N-королеви, і з цієї причини N-королева є типовою задачею, яка використовується як орієнтир у класичних роботах про штучний інтелект.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

Наприклад, ця задача була використана для введення пошуку зі зворотним відстеженням.

Під час взаємодії з середовищем будь-який раціональний агент генерує - набір гіпотез або асоціацій з вхідних даних, що подібно до навчання людини з середовища або опису проблеми. Потім він шукає, яка гіпотеза найкраще пояснює світ. Цей процес відомий як пошук у дереві рішень. Іноді агент вибирає одну гіпотезу та адаптує її до нових вхідних даних, але це подібно до пошуку варіацій навколо фіксованої точки, що виконується системою управління якістю (QMS). Як символічне навчання (наприклад, з використанням логіки першого порядку), так і несимволічне (наприклад, з використанням ваг у нейронній мережі) вимагають пошуку в просторі гіпотез, щоб знайти найкраще наступне рішення.

Цей процес подібний до більшості алгоритмів машинного навчання. Як описує, процес навчання можна розуміти як завдання пошуку у великому просторі гіпотез та пошук гіпотези, яка найкраще відповідає навчанню та майбутнім прикладам. Це основна причина, чому, якщо можливо пришвидшити процес пошуку за допомогою квантових обчислень, можна отримати переваги в алгоритмах штучного інтелекту. Причина полягає в тому, що те, що зазвичай називають «процесом навчання», - це просто процес пошуку найкращого пояснення (гіпотези) для отриманих даних, намагаючись бути максимально готовим до майбутніх даних, що надходять до системи.

Оскільки кінцевий стан задачі, розв'язаної за допомогою QMS, невідомий через невизначеність у виході (завдання полягає в тому, щоб знайти невідому конфігурацію з мінімальними витратами), алгоритм QMS виконує пошук у просторі станів, керуючись метою мінімізації функції витрат. Як описує Кнут, існує тип пошуку, заснований на порівнянні ключів (значень) станів. У нашому випадку ми виконуємо обґрунтований пошук .

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

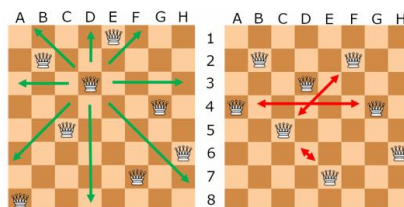


Рисунок 3.3 - Пояснення задачі «N королев» із 8 королевами на шахівниці розміром 8×8 .

Зліва наведено правильне рішення задачі, оскільки жодна з ферзей не атакує іншу, як пояснено для ферзі на полі D3. Справа наведено приклад тієї самої шахівниці, але з 4 ферзями, що атакують одна одну (A4, C5, E7, F6 та G4)

Звичайно, послідовна концепція є лише формалізмом для опису пошуку, оскільки квантово він виконується в суперпозиції. Ось чому ми розглядаємо QMS, застосовану до N-ферзів, як алгоритм пошуку.

Агент III виконує подібний пошук, щоб оптимізувати свою взаємодію з середовищем та пришвидшити досягнення цілей. Через подібність між пошуком у просторі гіпотез та пошуком QMS навколо початкового стану (початкової гіпотези), QMS можна розуміти як метод, який реалізує один із найважливіших кроків у будь-якому агенті штучного інтелекту.

Ще однією особливістю задачі N-ферзів, розв'язаної за допомогою QMS, є ймовірності переходу. Оскільки ми розв'язуємо задачу за допомогою методики, заснованої на Метрополісі-Гастингсі, ймовірність переходу в один чи інший стан з поточного стану має особливе значення. Ймовірність завжди буде дорівнювати одиниці, що евристичне значення зменшується (кількість атак між ферзями менша), та значенню між 0 та 1, в іншому випадку. Ця ймовірність між 0 та 1 дозволяє QMS переходити в новий стан, навіть якщо евристичне значення збільшується, уникаючи локальних мінімумів. Значення ймовірності розраховується пропорційно різниці евристичного значення між поточним станом та наступним станом. Якщо різниця евристичного значення мала, ймовірність буде близькою до одиниці; в іншому випадку вона буде близькою до нуля. Крім того, цю ймовірність можна регулювати за допомогою коефіцієнта ϵ .

QMS вимагає вхідного файлу, який є набором кортежів (стан, вартість). У задачі N-ферзів стани – це всі можливі комбінації дощок, які можуть з'явитися для заданого n. Щоб зменшити розмір простору станів, ми представляємо кожен стан як список з n позицій (по одній на ферзя), і кожна позиція представляє ряд ферзя. Припускаючи, що неможливо мати двох ферзів в одному стовпці, рух – це просто перестановка в позиції (рядку) двох ферзів. Схема, показана на рис. 3.4, буде представлена списком (0, 1, 2, 3), що вказує, що перший ферзь знаходиться в рядку 0, другий – у рядку 1 тощо.

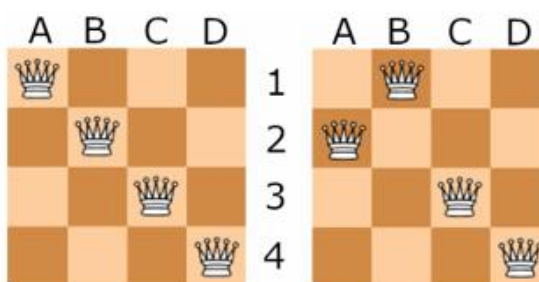


Рисунок 3.4 - Цей стан для $n = 4$ зображено зліва як (0, 1, 2, 3), оскільки перша ферзь знаходиться в рядку 0 (A1), друга ферзь - у рядку 1 (B2) тощо.

Таке представлення є ефективною кодифікацією проблеми, яка зменшує кількість станів. Можливі рухи – це перестановки між позиціями. Наприклад, на рис. 4 перший і другий ферзі міняються місцями між ними. Після пояснення кодифікації дошки необхідно пояснити евристичне значення, пов'язане з кожною позицією на дошці. Розроблена нами евристика підраховує кількість атакуючих ферзів і штрафує їх додатковими витратами, якщо один ферзь атакує більше одного іншого ферзя. Евристика (H) визначається рівнянням 8, а метою є мінімізація евристики-значення тика. Якщо евристика дорівнює 0, то ця конфігурація плати є рішенням.

Праворуч було виконано обмін місцями між першою та другою королевою, і отриманий стан - (1, 0, 2, 3), оскільки перша королева знаходиться в рядку 1 (A2), друга королева - у рядку 0 (B1), де перша сума підраховує евристичне значення для всіх ферзів, а друга сума підраховує евристичне значення для кожного ферзя. 8 рядок W_i , рядок j дорівнює 1, якщо *ферзь* та *ферзь j* знаходяться в одному рядку

(або однакові для діагоналі), інакше воно дорівнює 0. Y – це акумулятивне значення, яке підраховує кількість ферзів, яких *queen*i вже атакує. Це мультиплікативний коефіцієнт, який збільшується на 1 для кожного додаткового атакованого ферзя. $Y = 1$ для першого атакованого ферзя *queen*i, $Y = 2$ для другого атакованого ферзя *queen*i тощо. Ця евристика повертає високий штраф для дощок, на яких ферзь розташована дуже погано, тобто атакує кількох інших ферзів, чого необхідно уникати.

У задачі N-ферзи кількість рішень для кожного розміру n визначає її складність. Більша кількість рішень означає більше цільових станів та більш керований пошук, оскільки градієнт між станами більший, а перехід до цільового стану швидший. Призводить до швидшої генерації рішень. Можна побачити, що $n = 6$ має значно менше рішень, ніж попередній випадок, $n = 5$, та наступний випадок, $n = 7$. Це впливає на складність задачі. Як видно з графіка моделювання на рис. 3.5, TTS, отриманий для $n = 6$ та $n = 7$, є подібним, незважаючи на те, що для $n = 7$ простір станів набагато більший, ніж для $n = 6$.

3.3 Результати моделювання та оцінка ефективності алгоритмів

Для перевірки інструменту QMS із задачею N-Queen ми виконали різні симуляції як з класичними, так і з квантовими алгоритмами Метрополіса-Гастингса. Ці симуляції були виконано за допомогою фреймворку *Qiskit* та включеного симулятора вільного шуму *QASM*. Задача N-Queen вимагає багатьох кубітів для представлення своїх станів, а фактичний симулятор має обмежені можливості для представлення великих обсягів даних. З цієї причини ми розраховували кількість кубітів, необхідних для нашої кодифікації, та споживання пам'яті в симуляторі *QASM* для цієї кількості кубітів. Детально описано кількість кубітів, необхідних для кожного регістра в програмному засобі QMS.

Список необхідних реєстрів такий:

- Регістр координат представляє кожен стан. Необхідно кодувати кожен стан у двійковій формі, тому кількість кубітів дорівнює кількості регістрів,

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

помноженій на кількість кубітів, необхідних для двійкового представлення.

- Регістр ідентифікатора переміщення представляє координату для переміщення, тому це індекс, який вимагає двійкового представлення кількості регістрів.

- Регістр значення переміщення вказує, чи відбувається рух вгору чи вниз (вліво чи вправо у випадку обміну). Для цього потрібен лише один кубіт.

- Реєстр монет відображає бінарне рішення про прийняття або відхилення запропонованого кандидата.

- Регістр Ancilla використовується для зберігання ймовірності зміни запропонованого кандидата. Він може бути представлений 3 або більше кубітами залежно від вибраної точності ймовірності.

У таблиці 3.1 представлено кількість кубітів для кожного розміру n у задачі N-Queen. Ця таблиця корисна для розуміння складності схеми та необхідних ресурсів для її виконання на класичному комп'ютері. У нашому випадку доступно 128 ГБ оперативної пам'яті, але ми маємо результати лише для $n = 7$ через час виконання, який становить близько 2 тижнів на екземпляр задачі з $n = 7$.

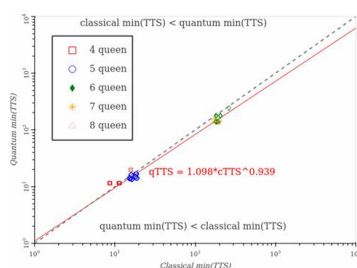


Рисунок 3.5 - Порівняння класичного та квантового TTS для задачі «N королев» з $n = 4, 5, 6$ та 7 на основі 74 зразків задачі.

Штрихована сіра лінія розділяє області класичної переваги (верхній трикутник) та квантової переваги (нижній трикутник). Ключовим аспектом, на який слід звернути увагу на цьому рисунку, є те, що для $n = 4$ більшість точок знаходиться в області класичної переваги, але коли складність задачі зростає,

більшість точок переходить в область квантової переваги. Експонента масштабування становить 0,939.

Ми виконуємо симуляції, використовуючи метрику TTS, пояснену в рівнянні 9, як показник якості. Ми тестуємо QMS для $n = 4, 5, 6$ та 7. Рішення зупинитися на $n = 7$ безпосередньо пов'язане з часом, який витрачається на кожне виконання. Щоб отримати більше випадків задачі з різними початковими конфігураціями, ми дещо змінюємо правила N-ферзів. У кожному випадку ми фіксували одну ферзь в одній позиції, враховуючи, що ця ферзь застрягла в цій позиції з будь-якої причини. Таке обмеження є поширеним явищем у задачі машинного навчання, наприклад, обов'язкова точка для відвідування на маршруті, згенерованому за допомогою глибокого навчання. Це нове правило дає нам додаткові бали для оцінки нашої задачі.

Без цієї модифікації N-Queen у нас було б лише 4 різні зразки задачі, по одному на кожне значення n . У симуляції ми маємо 74 різних екземпляри задачі N-Queen: 9 екземплярів для $n = 4$, 42 екземплярів для $n = 5$, 21 екземплярів для $n = 6$ та 2 екземплярів для $n = 7$. Зменшена кількість екземплярів для $n = 7$ призводить до 4 тижнів виконання.

Метрика, запропонована та використана Лем'є та ін. (2020), називається Час до знаходження рішень і позначається як TTS. Це показник якості, який вимірює очікувану кількість кроків, необхідних для знаходження рішення. Корисно порівнювати процедури, які необхідно повторити у разі невдачі, як-от цей алгоритм вибірки. TTS забезпечує баланс між збільшенням ймовірності та кількістю кроків у кожному виконанні, що означає, що нижчий TTS означає менший очікуваний час виконання.

За допомогою цієї метрики та аналізу показника закону масштабування, Лем'є та ін. отримали поліноміальне прискорення 0,75, наприклад, класичний $TTS = O(n^2)$ (квантовий $TTS = O(n^{0.75})$), стверджуючи, що їхня пропозиція масштабується краще, ніж класичний алгоритм Метрополіса-Гастінгса, і тому може бути вигідною у більших випадках задачі. Показник показує, як масштабується зв'язок між класичним та квантовим алгоритмами. Значення нижче 1 означає, що

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

квантова складність масштабується сприятливіше, ніж класична. Ми можемо оцінити цей показник за допомогою лінійного методу найменших квадратів у логарифмічній шкалі як для класичного, так і для квантового мінімального TTS. Оскільки ми хочемо побачити показник закону масштабування квантового TTS відносно класичного TTS, ми дотримуємося рівняння $y = bx^a$, x та y , класичного ($cTTS$) та квантового ($qTTS$) TTS відповідно. У логарифмічному масштабі:

$$\log(qTTS) = \log(b) + a \log(cTTS) \quad (3.3)$$

a є показником степеня, що визначає масштабування між $qTTS$ та $cTTS$.

Результати показано на рис. 3.5. Цей графік показує зв'язок між класичною та квантовою ТТП. Він розділений на два трикутники сірою пунктирною лінією. Верхній трикутник показує область класичної переваги, де знаходиться більшість точок з $n = 4$, а нижній трикутник з квантовою перевагою, де знаходяться всі точки з $n = 6$ та $n = 7$. Графік показує, що отримані точки мають тенденцію рухатися до області квантової переваги зі збільшенням розміру задачі n , і ми можемо зробити висновок, що існує можлива квантова перевага квантової МГ над класичною. Ми можемо кількісно визначити її за допомогою лінійного методу найменших квадратів з показником a (визначеним у рівнянні 11) 0,939.

Ми також тестуємо ядро QMS, квантові блукання, щоб знайти найкращий алгоритм. Завдяки дискретизації, проведеній Лем'є та ін. у квантовому МН унітарному операторі U , ми перевіряємо, чи може сортування операторів вплинути на результати. Ми також включаємо два інші альтернативні варіанти сортування у порівняльне дослідження. Ми визначаємо підготовку за допомогою операторів VB , вибір за допомогою оператора F , обернену підготовку за допомогою $B \wedge V \wedge$ та відображення за допомогою R .

• Лем'є та ін.: Підготовка-Вибір-Зворотна підготовка-Відображення. А саме, це відповідає сортуванню:

$$U = RV^\dagger B^\dagger FBV' \quad (3.4)$$

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

Зворотна підготовка-рефлексія-підготовка-вибір. А саме, це відповідає сортуванню:

$$U = FVBRB^{\dagger}V^{\dagger} \quad (3.5)$$

Підготовка. А саме, це відповідає сортуванню:

$$U = VBRB^{\dagger}V^{\dagger}\Phi' \quad (3.6)$$

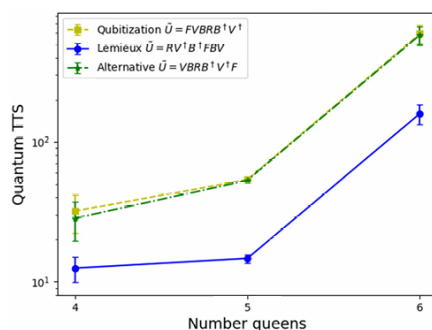


Рисунок 3.6 - На цьому рисунку показано три різні варіанти сортування, які ми перевіряємо для оператора квантового ходу W .

Синім кольором позначено сортування, яке дає мінімальне значення TTS, нижче за інші два варіанти сортування для $n = 4$, $n = 5$ та $n = 6$. Можна спостерігати схожість між еволюцією W у рівнянні 3.4 та іншими двома варіантами сортування у рівняннях 3.5 і 3.6.

Щоб порівняти ці три різні варіанти сортування, ми виконаємо задачу N-ферз з $n = 4, 5$ та 6 , включаючи кілька випадків фіксованих ферзів. Для кожного сортування та кожного n ми отримуємо середнє значення та стандартне відхилення. Використовуючи ці параметри, легко порівняти, чи мають TTS суттєві відмінності між собою. Результати показано на рис. 6. Вибір оператора сортування, обраний Лем'є та ін., досягає нижчого значення TTS для всіх протестованих розмірів задачі. Крім того, можна спостерігати подібну тенденцію між різними варіантами сортування, але розділеними проміжком. Таким чином, ці симуляції показують, що сортування Лем'є та ін. працює краще.

Ми вивчили алгоритми квантової оптимізації, щоб перевірити, як вони можуть кинути виклик існуючим класичним алгоритмам для промислових задач. Класичні алгоритми оптимізації сприяють пошуку рішень складних задач, які інакше неможливо вирішити за допомогою комп'ютерів методом перебору. Однак класичні алгоритми оптимізації мають обмеження в масштабованості та можуть бути оптимізовані за допомогою квантових обчислень. Зокрема, ми зосередилися на алгоритмі Метрополіса-Гастингса, який має багато застосувань для промислових задач, та його квантовому аналогу, квантовому Метрополіса-Гастингса.

Квантову версію МН модифікували для отримання реалізованої версії. Ми використовуємо обидві роботи для створення програмного інструменту, в основі якого лежить алгоритм МН, і який може вирішити будь-яку задачу оптимізації, задану у простому форматі як список кортежів вартості станів. Цей інструмент є простим у використанні модулем Python, який отримує опис задачі та повертає позицію мінімальної вартості. Крім того, QMS оцінює пошук рішення, повертаючи значення часу до вирішення (TTS). Ця метрика TTS є важливим показником при оцінці продуктивності інструменту та порівнянні його з класичним алгоритмом.

Як ми показали, можна використовувати QMS з будь-якою комбінаторною оптимізаційною задачею, яка має невизначеність у виході. Таким чином, метою є пошук невідомого стану з деякими властивостями. Ця вимога виконується в більшості оптимізаційних задач на промисловому рівні (задача рюкзака, TSP, маршрутизація тощо). Процес пошуку стану з мінімальними витратами в цих задачах перетворює їх на задачу NP-складності. Саме в цій родині задач поліноміальні переваги квантових обчислень можуть бути найбільш корисними, тому ми їх і обрали. Роботи Сегеді та Лем'є та ін. показують поліноміальну перевагу з використанням квантових блукань, яку ми екстраполюємо на універсальний інструмент для будь-якої оптимізаційної задачі.

Робота зосереджена на вивченні алгоритмів, що дотримуються гібридної класично-квантової парадигми. Як було пояснено, ця парадигма являє собою

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

проміжний шлях, який намагається скористатися перевагами широко вивчених класичних алгоритмів та нових можливостей, що пропонуються квантовими алгоритмами. Хоча це може не бути остаточним рішенням, логічнішим видається рух до сценарію, в якому алгоритми є або класичними, або квантовими, це проміжний крок, який полегшує виконання квантових алгоритмів, і можливі сценарії, в яких існуючі алгоритми прискорюються у дуже складних задачах.

Гібридні алгоритми охоплюють гетерогенну групу методів, починаючи від класичних алгоритмів з квантовим модулем, відповідальним за виконання складного, але обмеженого завдання, алгоритмів, що включають лише класичний модуль попередньої обробки для зменшення обсягу даних, що кодуються в класичній схемі, і закінчуючи архітектурами, в яких квантова схема оптимізується класичним алгоритмом. Існують навіть інші формулювання, в яких класичний алгоритм аналізує квантовий алгоритм для оптимізації його вартості та аналізу його властивостей.

Першими проблемами, на які звернули увагу квантові обчислення, були проблеми моделювання. Пізніше стало зрозуміло, що найкращим сценарієм для квантового моделювання з практичним застосуванням є квантова хімія. У наступному поколінні розвитку квантових обчислень та з появою більшої кількості запропонованих алгоритмів, проблеми оптимізації також почали розглядатися як хороші кандидати для прискорення квантовими алгоритмами. Тому в цій роботі для застосування розроблених алгоритмів було обрано ці дві родини проблем – оптимізацію та квантову хімію. Мотивацією цього вибору було створення алгоритмів, які можуть мати значний вплив на промисловий сектор.

Щодо задач оптимізації, дослідження зосередилося на певній підкатегорії, а саме на алгоритмах пошуку та вибірки. Ця підкатегорія визначає алгоритми, здатні виконувати пошук у просторі станів для знаходження розподілу даних, мінімуму чи максимуму. Для цього було проаналізовано основні класичні алгоритми, що належать до цієї категорії, та визначено їхні обмеження.

Було проаналізовано різні квантові пропозиції та їхні обмеження. Потім було обрано ту, яка досі викликала менший інтерес у світі квантової оптимізації,

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

– квантові блукання. Використовуючи квантові блукання як центральний алгоритм, інші автори побудували на їх основі квантову версію алгоритму Метрополіса-Гастингса. У цій роботі ці версії квантового алгоритму МН були взяті як еталон та модифіковані для його реалізації в квантовій схемі. Ця реалізація призвела до створення програмного інструменту під назвою квантовий розв'язувач Метрополіса, QMS, який був застосований до різних випадків використання.

Система управління якістю (QMS) була застосована до випадку використання штучного інтелекту, оскільки фундаментальним процесом багатьох алгоритмів ШІ, особливо алгоритмів машинного навчання, є пошук. Алгоритми машинного навчання виконують процес абстракції та логічного висновку, в якому їм потрібен процес пошуку, щоб знайти найкращу гіпотезу для пояснення вхідних прикладів. Оскільки виконання тестів цих алгоритмів є занадто складним через обмежені можливості квантового обладнання, була використана редукована задача, яка класично використовується як еталон для цих задач, задача N-Queen. Результати показують, що гібридний алгоритм має поліноміальну перевагу над чисто квантовим алгоритмом.

3.4 Висновок по розділу

У третьому розділі було розглянуто розробку та оцінку ефективності гібридних квантово-класичних рішень у комерційному програмному забезпеченні. Проаналізовано особливості реалізації задач оптимізації в гібридних системах, що дозволяє підвищити продуктивність обчислень. Розглянуто гібридну модифікацію класичних методів, яка поєднує переваги квантових і класичних підходів. Також проведено моделювання та оцінку ефективності запропонованих алгоритмів, що підтвердило їх потенціал для використання у складних комерційних задачах. Отримані результати свідчать про перспективність гібридних квантово-класичних алгоритмів у сучасному програмному забезпеченні.

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході виконання роботи було досліджено та реалізовано підходи до використання гібридних квантово-класичних алгоритмів у комерційному програмному забезпеченні, що є комплексним процесом, спрямованим на підвищення ефективності обчислень і розв’язання складних задач оптимізації. У роботі проаналізовано сучасні підходи до побудови квантових і класичних алгоритмів, визначено їхні переваги та обмеження, а також досліджено можливості їх інтеграції в рамках єдиних програмних рішень. Процес дослідження охопив аналіз теоретичних основ, моделювання алгоритмів, їх програмну реалізацію та оцінку ефективності, що дозволило сформулювати цілісне уявлення про перспективи використання гібридних обчислень.

На початковому етапі було проведено аналіз предметної області, визначено ключові поняття квантових обчислень, принципи побудови гібридних алгоритмів та особливості задач пошуку і оптимізації. Розглянуто класичні методи пошуку, їх обмеження при роботі з великими обсягами даних, а також потенційні переваги квантових алгоритмів, зокрема можливість прискорення обчислень за рахунок використання квантових властивостей, таких як суперпозиція та

					БР.ІІ – 97.00.00.000 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

заплутаність. Це дозволило обґрунтувати доцільність застосування гібридного підходу.

У другому розділі було досліджено методи та інструменти реалізації гібридних квантово-класичних алгоритмів. Особливу увагу приділено квантовим алгоритмам пошуку та оптимізації, квантовим прогулянкам, а також іншим сучасним підходам до побудови квантових алгоритмів. Було визначено, що поєднання квантових та класичних методів дозволяє досягти більш ефективних результатів порівняно з використанням лише одного підходу, особливо в задачах із великою розмірністю та складною структурою.

Практична частина роботи була спрямована на реалізацію гібридних алгоритмів та дослідження їх ефективності. Зокрема, було розглянуто особливості застосування гібридних підходів для задач оптимізації, реалізовано модифікацію методу Монте-Карло з використанням квантових елементів, а також проведено моделювання отриманих рішень. Результати експериментів показали, що використання гібридних алгоритмів дозволяє підвищити швидкість пошуку оптимальних рішень та покращити якість результатів у порівнянні з класичними методами.

Проведене дослідження підтвердило, що гібридні квантово-класичні алгоритми мають значний потенціал для застосування у комерційному програмному забезпеченні. Вони можуть бути ефективно використані в задачах оптимізації, аналізу великих даних, фінансового моделювання, логістики та машинного навчання. Основними перевагами такого підходу є підвищення продуктивності, можливість обробки складних задач та гнучкість інтеграції з існуючими системами.

Разом з тим, існують певні обмеження, пов'язані з недостатнім рівнем розвитку квантових апаратних засобів, складністю реалізації квантових алгоритмів та необхідністю спеціалізованих знань для їх використання. Крім того, інтеграція квантових і класичних компонентів потребує додаткових досліджень у сфері архітектури програмних систем і оптимізації обчислювальних процесів.

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

У перспективі подальші дослідження можуть бути спрямовані на вдосконалення гібридних алгоритмів, розширення сфер їх застосування, розробку більш ефективних методів інтеграції квантових обчислень у комерційні програмні продукти, а також використання новітніх квантових платформ. Це дозволить значно розширити можливості сучасного програмного забезпечення та підвищити його конкурентоспроможність.

Загалом, поставлена мета роботи досягнута, а отримані результати підтверджують доцільність і перспективність використання гібридних квантово-класичних алгоритмів у сучасних програмних системах.

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Nielsen, M., & Chuang, I. (2010). *Quantum Computation and Quantum Information*. cambridge.org. <https://doi.org/10.1017/CBO9780511976667>
2. Preskill, J. (2018). Quantum Computing in the NISQ era and beyond. *Quantum*, 2, 79. <https://doi.org/10.22331/q-2018-08-06-79>
3. Farhi, E., Goldstone, J., & Gutmann, S. (2014). A Quantum Approximate Optimization Algorithm. arxiv.org. <https://arxiv.org/abs/1411.4028>
4. Peruzzo, A., et al. (2014). A variational eigenvalue solver on a quantum processor. *Nature Communications*, 5, 4213. <https://doi.org/10.1038/ncomms5213>
5. Kandala, A., et al. (2017). Hardware-efficient variational quantum eigensolver. *Nature*, 549, 242–246. <https://doi.org/10.1038/nature23879>
6. IBM Quantum. (2025). Documentation. quantum.ibm.com. <https://quantum.ibm.com/docs>
7. Qiskit Documentation. (2025). qiskit.org. <https://qiskit.org/documentation/>
8. Microsoft Quantum Development Kit. (2025). learn.microsoft.com. <https://learn.microsoft.com/en-us/azure/quantum/>
9. Google Quantum AI. (2025). quantumai.google. <https://quantumai.google/>

					БР.ІІІ – 97.00.00.000 ПЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

10. Cirq Documentation. (2025). [cirq.readthedocs.io.
https://cirq.readthedocs.io/](https://cirq.readthedocs.io/)
11. Rigetti Computing. (2025). Forest SDK Documentation. [rigetti.com.
https://docs.rigetti.com/](https://docs.rigetti.com/)
12. D-Wave Systems. (2025). Ocean Documentation. [dwavesys.com.
https://docs.ocean.dwavesys.com/](https://docs.ocean.dwavesys.com/)
13. Schuld, M., & Petruccione, F. (2018). *Supervised Learning with Quantum Computers*. springer.com. <https://doi.org/10.1007/978-3-319-96424-9>
14. Biamonte, J., et al. (2017). Quantum machine learning. *Nature*, 549, 195–202. <https://doi.org/10.1038/nature23474>
15. Montanaro, A. (2016). Quantum algorithms: an overview. *npj Quantum Information*, 2, 15023. <https://doi.org/10.1038/npjqi.2015.23>
16. Grover, L. (1996). A fast quantum mechanical algorithm for database search. arxiv.org. <https://arxiv.org/abs/quant-ph/9605043>
17. Shor, P. (1994). Algorithms for quantum computation: discrete logarithms and factoring. IEEE. <https://doi.org/10.1109/SFCS.1994.365700>
18. Cerezo, M., et al. (2021). Variational quantum algorithms. *Nature Reviews Physics*, 3, 625–644. <https://doi.org/10.1038/s42254-021-00348-9>
19. Arute, F., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574, 505–510. <https://doi.org/10.1038/s41586-019-1666-5>
20. Harrow, A., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103, 150502. <https://doi.org/10.1103/PhysRevLett.103.150502>
21. Bernstein, E., & Vazirani, U. (1997). Quantum complexity theory. *SIAM Journal on Computing*, 26(5), 1411–1473.
22. Dong, D., & Petersen, I. (2010). Quantum control theory and applications. *IET Control Theory & Applications*, 4(12), 2651–2671.
23. McGeoch, C. (2014). *Adiabatic Quantum Computation and Quantum Annealing*. synthesis lectures.

					БР.ІІІ – 97.00.00.000 ІІЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

24. Johnson, M., et al. (2011). Quantum annealing with manufactured spins. *Nature*, 473, 194–198.
25. Häner, T., et al. (2020). Software for quantum computing: a review. *ACM Computing Surveys*, 53(2).
26. Lloyd, S. (1996). Universal quantum simulators. *Science*, 273(5278), 1073–1078.
27. Wiebe, N., et al. (2014). Quantum deep learning. [arxiv.org. https://arxiv.org/abs/1412.3489](https://arxiv.org/abs/1412.3489)
28. Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine. *Physical Review Letters*, 113, 130503.
29. Oracle. (2025). Cloud Infrastructure for Quantum Computing. [oracle.com. https://www.oracle.com/cloud/quantum/](https://www.oracle.com/cloud/quantum/)
30. Amazon Braket Documentation. (2025). [docs.aws.amazon.com. https://docs.aws.amazon.com/braket/](https://docs.aws.amazon.com/braket/)
31. Azure Quantum Overview. (2025). [learn.microsoft.com. https://learn.microsoft.com/en-us/azure/quantum/overview](https://learn.microsoft.com/en-us/azure/quantum/overview)
32. Gartner. (2025). Top Strategic Technology Trends: Quantum Computing. [gartner.com](https://www.gartner.com).
33. Deloitte. (2024). Quantum computing for business. [deloitte.com. https://www2.deloitte.com/](https://www2.deloitte.com/)
34. McKinsey & Company. (2024). Quantum computing use cases. [mckinsey.com. https://www.mckinsey.com/](https://www.mckinsey.com/)
35. Accenture. (2024). Commercial applications of quantum computing. [accenture.com. https://www.accenture.com/](https://www.accenture.com/)
36. Chen, S., et al. (2023). Hybrid quantum-classical algorithms for optimization problems. *IEEE Access*, 11, 12345–12360.
37. Zhou, L., et al. (2020). Quantum approximate optimization algorithm: Performance and applications. *Physical Review X*, 10, 021067.
38. Bharti, K., et al. (2022). Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94, 015004.

					БР.ІІІ – 97.00.00.000 ІІЗ	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

39. Hadfield, S. (2019). From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2), 34.

40. LaRose, R., & Coyle, B. (2020). Robust data encodings for quantum classifiers. *Physical Review A*, 102, 032420.

					БР.ІІІ – 97.00.00.000 ІІЗ	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

БІБЛІОГРАФІЧНА ДОВІДКА

Тема бакалаврської роботи: "Гібридні квантово-класичні алгоритми в комерційному програмному забезпеченні "

Обсяг пояснювальної записки: 81 аркуш

Дата закінчення **бакалаврської** роботи 10 червня 2026р.

Підпис студента _____