Софийски Университет “Св. Климент Охридски”

Ф и з и ч е с к и ф а к у л т е т

Катедра „Теоретична физика“

КВАНТОВИ АЛГОРИТМИ

Христо Светленов Тончев

Автореферат на дисертация за придобиване на образователната и научна степен „доктор“

Професионално направление ***4.1.Физически науки***

Научна специалност 01.03.19 ***Физика на атомите и молекулите***

Научен ръководител:

проф. д.фз.н. Николай Витанов Витанов

София, 2017 г.

Дисертационният труд се състои от 144 страници, съдържащи 76 фигури и в библиографията са посочени 91 заглавия.

Публикациите по темата са три в международни списания, като две от тях са в списания с IF.

Дисертационният труд беше обсъден на катедрено заседание на катедра „Теоретична физика“, Физически факултет на СУ „Св. Климент Охридски“, състоял се на 14.06.2017 г.

Структура и обем на дисертацията

Тя е оформена в 6 глави, както следва:

Увод

Първият квантов алгоритъм е разработен през 1992 от Дейвид Дойтч и Ричард Джоса [1]. Целта на алгоритъма е да провери дали дадена функция е константна за всичките й възможни входни стойности. Понастоящем за квантовите компютри са разработени някои квантови алгоритми, които биха имали важно практическо приложение.

Алгоритъмът на Гровер за търсене в неподредена база данни [2] е един от квантовите алгоритми, за който се очаква да намери много широко приложение. Алгоритъмът намира търсения елемент за $N_G = (\pi/4)\sqrt{N}$ опита с квантов компютър, което е квадратично по-бързо от класическите алгоритми, които изискват $O(N)$ опита. Алгоритъмът на Гровер може да бъде адаптиран и към изчислително сложните проблеми със структура чрез влагане на квантовите търсения [3]. При граф с по-висока размерност от единица алгоритъмът за търсене с произволно преместване (обяснен по-долу) намира по-бързо търсения елемент, отколкото би го намерил алгоритъмът на Гровер. Експериментално търсенето на Гровер е демонстрирано по няколко начина: чрез ядрено-магнитен резонанс с два [4] и три [5] кюбита, отговарящи съответно на 4 и 8 елемента на регистъра; в линейната оптика, използвайки 4 елемента [6] и в йонен капан с четири елемента [7]. Алгоритъмът на Гровер е демонстриран и в индивидуални Ридбергови атоми с 8 различни нива, служещи като елементи на базата данни [8] и в класическа Фурие оптика с 32 елемента [9]. Последните два метода превъзхождат останалите по големината на базите данни, но имат горна граница за големината на регистрите, които могат да бъдат постигнати по този начин.

За алгоритъма за търсене на Гровер също е предложен вариант, използващ кюдити, където вместо Адамаровия гейт се използва дискретната Фурие трансформация (DFT) [10] или друга *d*-мерна трансформация [11] с цел да се построи оператор за отражение. Алгоритъм за търсене на Гровер с кютрити е разгледан в [12]. Някои от тези предложения изискват множество допълнителни физични взаимодействия.

Алгоритъмът на Гровер не може да се използва без да се знае предварително брой решения. Точният им брой може да се намери, като се използва алгоритъм на квантовия брояч [13, 14]. Алгоритъмът на квантовия брояч е експериментално демонстриран от Джонс и Моска [15] и Лий и колектив [16], чрез ядрено-магнитен резонанс.

Важна част от квантовия брояч е алгоритъмът за определяне на фазата [1, 17]. Целта на алгоритъма е да намери собствените стойности на унитарен оператор при известно собствено състояние. Абрамс и Лойд [18] са предложили вариант на алгоритъма за намиране на собствените стойности чрез използване на квантова Фурие трансформация. Траваглион и Милбърн [19] са предложили схема, по която алгоритъма може да се реализира с използване на кюбити, базирани на йони в уловка. Квантовият алгоритъм за определяне на фазата е демонстриран с помощта на ядрено-магнитен резонанс от Лий и съавтори [16].

Алгоритъмът за определяне на фазата също е важна част от алгоритъма на Шор за разделяне на число на прости множители [20].

Квантовият алгоритъм за произволно преместване е квантов аналог на класическите алгоритми за преместване. Съществуват два типа квантови алгоритми за произволно преместване - дискретни във времето DTRWA и непрекъснати CTRWA. По аналог с класическите алгоритми за търсене, базирани на произволното преместване, са направени и квантови. CTRWA е предложен за първи път от Е. Фари и С. Гутман [21]. Показано е, че в този алгоритъм преместването е експоненциално по-бързо през граф [22]. Той може да реши проблеми, в които се изисква минимален брой обръщения към черна кутия експоненциално по-бързо от всеки класически алгоритъм [23], като например проблема за търсенето. DTRWA е бил предложен за първи път от Ахаронов и колектив [24]. Примери за DTRWA алгоритми са квантовият алгоритъм за намиране на различаващи се елементи [25] и квантовият алгоритъм за търсене с произволно преместване [26]. Съществуват също така два типа квантови алгоритми за търсене с произволно преместване - непрекъснатият във времето (CTRWS) и дискретният във времето (DTRWS). А. Чилдс е показал, че непрекъснатият по време алгоритъм за търсене с произволно преместване (CTRWS) може да намери елемент в четиримерен граф по-бързо от алгоритъма на Гровер [27]. Оригиналният DTRWS алгоритъм предложен за пръв път от Н. Шенви и колектив [26] се означава като SKW. Алгоритъмът за търсене SKW при едно прилагане има вероятност по-малка от 0.4 да намери търсеното състояние. Б. Хейн е предложил по-бърз вариант на DTRWS, но за да бъде ефективен, началното състояние на алгоритъма трябва да бъде определено от елементите, които се търсят [28]. В. Потосек и колектив са показали, че ако пространството на търсене се раздели на две еднакви подпространства, то вероятността да се намери търсеният елемент може да се увеличи до повече от 0.8 [29] и също са показали, че модификации на оператора на смесване могат да увеличат вероятността да се намери търсеният елемент. А. Тулси показал, че DTRWS е по-бърз от алгоритъма на Гровер и CTRWS за пространствено търсене в двумерна база данни [30].

Дисертацията е посветена на усъвършенстване на възможностите на някои квантови алгоритми. Алгоритъмът на Гровер е приспособен да използва кютритен запис на информацията. Удвоена е вероятността да се намери търсеният елемент при квантовото търсене с произволно преместване върху хиперкуб чрез използване асиметрични монети. Направен е квантов алгоритъм за определяне на фазата, ползващ кютрити. Алгоритъмът е използван за да бъде направен квантов брояч за алгоритъма на Гровер с кюдити.

1. Основи

Изчислимостта на алгоритъма O(f(N)) [1] показва как се увеличават ресурсите, необходими при увеличаване на броя елементи, които трябва да обработва. Като ресурс тук ще бъде разглеждан броя стъпки на алгоритъма. Съвкупността от математически проблеми или алгоритми, които имат еднаква изчислимост, образуват класовете на изчислимост.

Класическият компютър може да използва детерминистични и вероятностни алгоритми. Чрез класическите детерминистични алгоритми компютърът би могъл да реши по-малък кръг от проблеми за полиномиално време от класически компютър, ползващ вероятностни алгоритми. Класът на изчислимост на детерминистичните алгоритми се означава с P. Когато класическият компютър ползва вероятностен алгоритъм и търсеният резултат се изисква с дадена точност, би могъл да се справи с по-голям кръг проблеми за полиномиално време. Класът му на изчислимост се нарича BPP. Квантовият компютър има клас на изчислимост BQP, който би могъл да се справи с още по голям кръг от задачи за полиномиално време. Отношението между различните класове на изчислимост е следното:

$$P \subseteq BPP \subseteq BQP$$

Класическият и квантовият компютър са базирани на различна физична основа и оттам идват и различните им възможности. Причината за разликата между класическия и квантовия компютър се дължи на концептуалните различия между класическата и квантовата информация. Кюбитът (квантовият аналог на бита) има безкрайно много възможни състояния. Той е произволна суперпозиция на чистите състояния, нормирана на единица.

Единичният кюбит е вектор $|\psi\rangle = a_1|0\rangle + a_2 e^{i\varphi}|1\rangle$, параметризиран от две комплексни числа, които удовлетворяват нормировката: $|\alpha_1|^2 + |a_2|^2 = 1$. Винаги е изпълнено $|e^{i\varphi}| = 1$.

Базисните вектори се изразяват като:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Операторите, които действат на кюбитите (кюбитните гейтове), трябва да запазват тази нормировка и се представят чрез $2^n \times 2^n$ унитарни матрици. Изискването матриците да са унитарни $UU^+ = 1$ се дължи на факта, че квантовите гейтове трябва да са обратими.

В случая на еднокюбитните гейтове n=1. Някои от по-важните еднокюбитни гейтове са:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{bmatrix}$$

Поради възможността състоянието да бъде не само някое от чистите базисни състояния, то квантовите гейтове могат условно да се разделят на три типа:

1) Гейтовете, които сменят състоянието с друго базово. Това са гейтовете I и X.

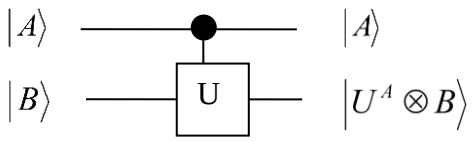
2) Гейтове, които дават фазова разлика след прилагането им, като Z и T.

3) Гейтове, които преобразуват началното състояние в суперпозиция от състояния, пример за това е H.

Докато 1) имат класически аналог, то 2) и 3) нямат.

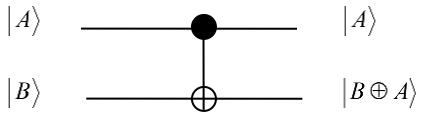
Съществуват безкрайно много еднокюбитни гейтове, понеже има безкрайно много 2×2 матрици.

Контролираните двукюбитни гейтове имат един контролен кюбит, който не се изменя при действието на гейта и един таргет кюбит, който се променя в зависимост от състоянието на контролния кюбит. Това е показано на Фиг. 1.1.



Фиг. 1.1. *Общ вид на двукюбитния контролен гейт. Кюбитът $|A\rangle$ е контролен. Той не се изменя при действието на гейта. Върху втория кюбит $|B\rangle$ се извършва унитарна трансформация U в зависимост от състоянието на кюбита $|A\rangle$.*

Пример за такъв гейт е CNOT. Той извършва следната трансформация: ако контролният кюбит е със стойност $|0\rangle$, то вторият кюбит остава непроменен, а ако контролният кюбит е със стойност $|1\rangle$, то вторият кюбит се обръща. Двукюбитният гейт е показан на Фиг. 1.2.



Фиг. 1.2. *CNOT гейт. Кюбитът $|A\rangle$ е контролен. На втория кюбит $|B\rangle$ се извършва събиране по модул 2. Следователно, ако контролният кюбит е със стойност $|0\rangle$, то вторият кюбит остава непроменен, а ако контролният кюбит е със стойност $|1\rangle$, то вторият кюбит се обръща.*

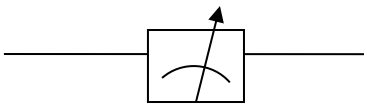
Двукюбитните гейтове могат да се представят с $2^2 \times 2^2$ матрици. Матричното представяне на CNOT гейта е:

$$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Състоянието на система от два кюбита се получава чрез тензорно произведение на състоянията на двата кюбита, участващи в системата.

Квантовите гейтове могат условно да се разделят на същите три типа, както в еднокюбитният случай. Произволен двукюбитен контролиран гейт CU може да бъде получен чрез CNOT гейт и еднокюбитни операции. Всеки унитарен оператор, независимо на колко кюбита трябва да се приложи, може да бъде разложен с произволно добра апроксимация на Адамаров, $\pi/8$, фазов и CNOT гейтове.

Измерващият уред се отбелязва по начина, показан на Фиг. 1.3



Фиг. 1.3. *Символът за уред за измерване в квантовите мрежи.*

Квантовите мрежи се състоят от няколко свързани помежду си квантови гейтове, като те се различават от обикновените мрежи по три неща:

1. Не са разрешени цикли и примки. При цикъла част от квантовата мрежа се обръща сама към себе си. При примка има обратна връзка от една част на квантовата мрежа към друга.
2. Всички операции трябва да са обратими.
3. Не е възможно копиране.

Кютритът е единица за квантова информация. Той е квантов аналог на класическия трит. Кютритът е система с три състояния, които се означават като $|0\rangle, |1\rangle$ и $|2\rangle$. Кютритът, също както и кюбитът, може да се намира в суперпозиция на трите базисни състояния. Стринг от n кютрити може да представя 3^n различни състояния едновременно. Базовите състояния на кютрита са ортогонални, затова кютритът може да се представи като линейна комбинация от тях.

Кютритите се представят по следния начин:

$$|\psi\rangle = a_0|0\rangle + a_1|1\rangle + a_2|2\rangle$$

$$|0\rangle = \begin{bmatrix}1\\0\\0\end{bmatrix} \qquad |1\rangle = \begin{bmatrix}0\\1\\0\end{bmatrix} \qquad |2\rangle = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

В кютритния случай нормировачното условие е:

$$|a_0|^2 + |a_1|^2 + |a_2|^2 = 1$$

Произволна еднокютритна операция може да бъде разложена по два начина. Единят е разложение като Ойлерови ротации и фазов гейт [32]. Вторият е разложение чрез

Хаусхолдерови отражения [33]. Мултикютритни операции могат да бъдат реализирани аналогично на мултикюбитните.

Кюдитът е система с d нива, т.е. той има d чисти състояния $|0\rangle,\ldots, |d-1\rangle$. Когато се използва квантов регистър с кюдити вместо регистър с кюбити, той може да съдържа много повече информация при същия брой носители от регистър. Големината на регистъра при използване на кютрити вместо кюбити е $N = 3^n$, когато са използвани кюбити $N = 2^n$, т.е. базата данни е $(3/2)^n$ пъти по-голяма. Аналогично за кюдитите $N = d^n$ и базата данни е $(d/2)^n$ пъти по-голяма.

Дискретната квантова Фурие трансформация [31] взима като входно състояние квантово състояние в ортогонален базис $|0\rangle,\ldots,|N-1\rangle$:

$$|\psi\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$$

и го трансформира в състоянието:

$$\hat{F}\left(\sum_{j=0}^{N-1} x_j |j\rangle\right) = \sum_{k=0}^{N-1} y_k |k\rangle,$$

където амплитудите y_k са дискретните Фурие трансформации на амплитудите x_k. Трансформацията е дефинирана да бъде линеен оператор със следните действия върху базисните състояния.

$$\hat{F}|j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |k\rangle e^{2ijk\pi/N}$$

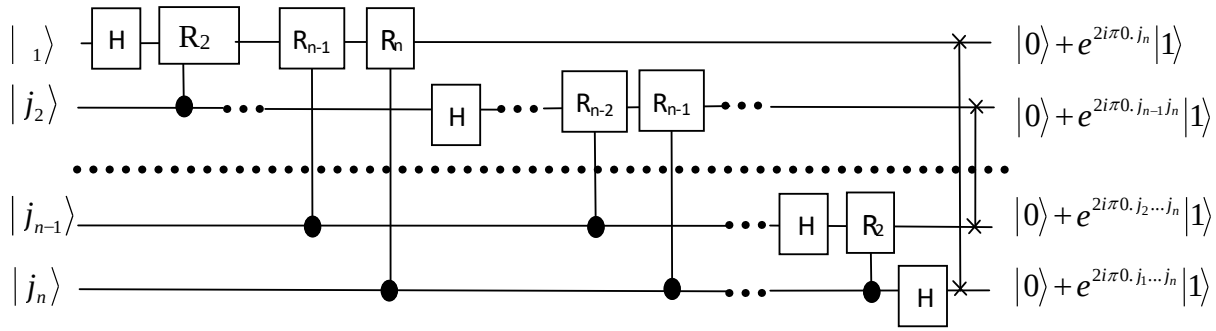
Тази трансформация е унитарна и може да бъде осъществена ефективно на квантов компютър.

Квантовата Фурие трансформация има следния матричен вид:

$$\hat{F} = \frac{1}{\sqrt{N}} \begin{pmatrix} w^{0*0} & w^{0*1} & w^{0*2} & \cdots & w^{0*(N-1)} \\ w^{1*0} & w^{1*1} & w^{1*2} & \cdots & w^{1*(N-1)} \\ w^{2*0} & w^{2*1} & w^{2*2} & \cdots & w^{2*(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w^{(N-1)*0} & w^{(N-1)*1} & w^{(N-1)*2} & \cdots & w^{(N-1)*(N-1)} \end{pmatrix},$$

където * е знак за умножение и:

$$w = e^{2\pi i/N}$$



Фиг. 1.4. *Квантова мрежа за осъществяване на квантова Фурие трансформация върху n кюбита. Фигурата е заимствана от [1]*

Броят необходими гейтове на алгоритъма за квантова Фурие трансформация (Фиг. 1.4) е $\Xi_1 = O(n^2)$. Алгоритъмът за класическата Фурие трансформация съдържа $\Xi_2 = O(n2^n)$ класически гейтове. Следователно квантовият алгоритъм е значително по-бърз от класическият.

Квантовата Фурие трансформация се бележи в квантовите мрежи по начина, показан на Фиг. 1.5:

Фиг. 1.5. *Символът за квантова Фурие трансформация в квантовите мрежи.*

2. Алгоритъм на Гровер с кюдити

Един квантов компютър би бил по-добър от класическия за решаването на редица задачи. При квантовата механика началното състояние на алгоритъма може да бъде равномерна суперпозиция от всички възможни класически начални състояния. Тогава той извършва паралелна обработка на данните и затова намира резултата далеч по-бързо от класическите алгоритми. Крайното състояние на алгоритъма е суперпозиция от всички чисти изходни състояния. При измерване на крайното състояние суперпозицията се разрушава и остава само едно от чистите изходни състояния. При правилно преобразуване на състоянието на регистъра, което се измерва, може да се постигне много голяма вероятност за намиране на търсеното състояние. Примери за квантови алгоритми за търсене са алгоритъма на Гровер [1], предназначен за търсене в неподредена база данни и алгоритъм за търсене с произволно преместване [26], предназначен за търсене върху неподредено дърво. И двата алгоритъма намират търсения елемент за $O(\sqrt{N})$ итерации.

Задачата, която решава алгоритъмът на Гровер, е намирането на M определени елемента в пространство на търсене от N елемента [1]. Елементите от множеството M се наричат още решения. Вместо да се търсят елементите директно, те се номерират и се търси по номерата на тези елементи в интервала от 0 до N-1. Броят елементи на системата е $N = 2^n$, тогава номерацията може да се съхранява в n бита. Броят решения M трябва да удовлетворява условието $1 \le M \le N$.

Алгоритъмът започва при начално състояние на системата $|0\rangle^{\otimes n} = |0\rangle \otimes |0\rangle \otimes ... \otimes |0\rangle = |0..0\rangle$. След прилагане на Адамаровата трансформация $H^{\otimes n}$ състоянието се променя в:

$$|\psi\rangle = H^{\otimes n}|0..0\rangle = H^{\otimes n}|0\rangle^{\otimes n} = H|0\rangle^{\otimes n} = \frac{1}{N^{1/2}}\sum_{x=0}^{N-1}|x\rangle$$

След това се прилага итерацията на Гровер $O(\sqrt{N})$ пъти, като итерацията се означава с G. Квантовата мрежа на алгоритъма на Гровер е показана на Фиг. 2.1. Квантовата мрежа на итерацията на Гровер е показана на Фиг. 2.2. Итерацията на Гровер може да се раздели на четири стъпки:

1) Прилага се Оракул.

Квантовият Оракул е черна кутия, която може да разпознае решенията на задачата за търсене. Той е унитарен оператор. Означава се с O и се дефинира по следния начин:

Кюбитът на Оракула е $|x\rangle$, който се обръща ако f(x)=1 и остава непроменен в противен случай. Когато елементът е решение, то функцията f(x) е единца, а когато не е решение е нула. Действието на Оракула се дефинира по следния начин:

$$\hat{O}|x\rangle = (-1)^{f(x)}|x\rangle$$

Целта на алгоритъма е да се намери решение на търсенето, като за това се използва възможно най-малък брой прилагания на Оракула.

2) Прилага се Адамаровата трансформация върху всички кюбити $H^{\otimes n}$.

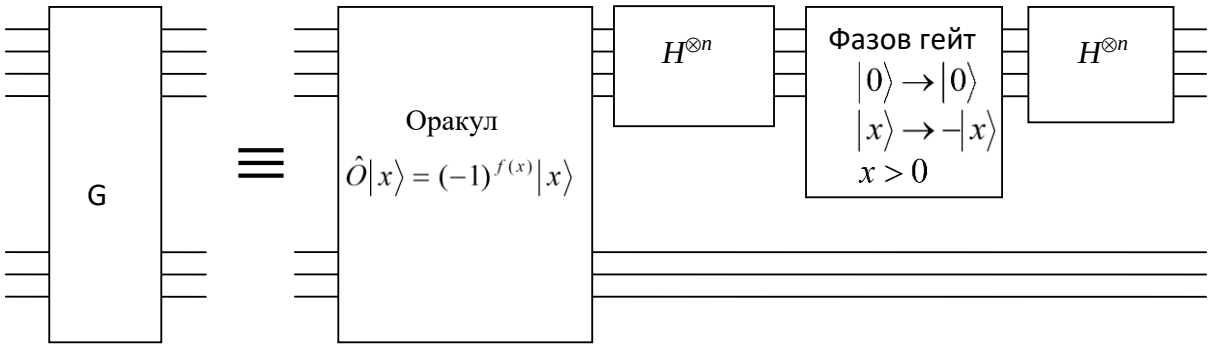
3) Всички състояния с изключение на състоянието $|0\rangle$ получават фаза -1. Операторът за това е:

$$\hat{Ph}|x\rangle = (-1)^{\delta_{x0}}|x\rangle.$$

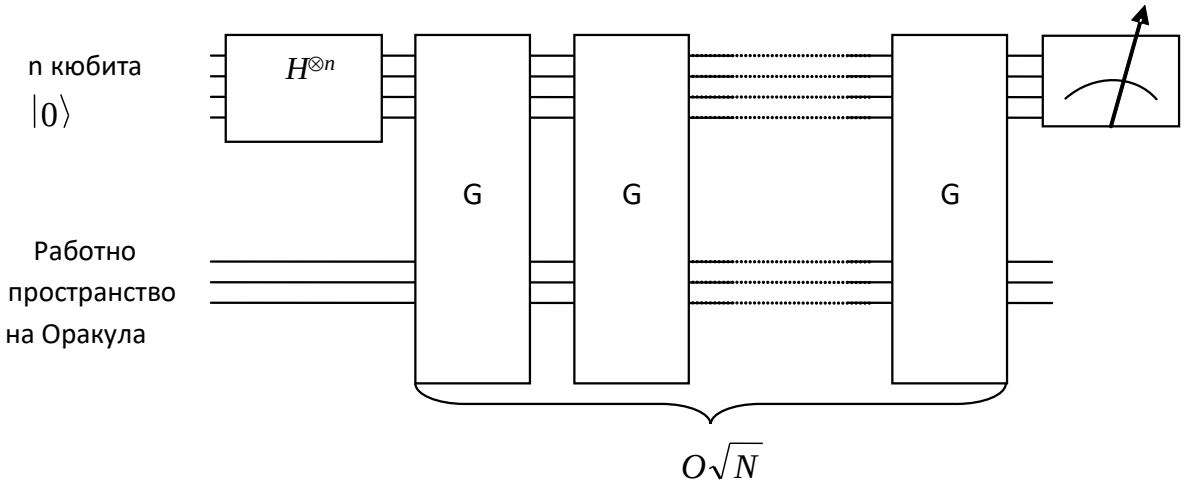
4) Прилага се отново Адамаровата трансформация върху всички кюбити $H^{\otimes n}$.

Всяка операция в итерацията на Гровер може да бъде ефективно изпълнена чрез квантов компютър. Всяка от стъпките 2) и 4) изисква $n = \log(N)$ операции. Операцията 3) изисква O(n) гейтове. Сумарно стъпките 2), 3) и 4) могат да се запишат по следния начин.

$$H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2(H^{\otimes n}|0\rangle)(\langle 0|H^{\otimes n}) - H^{\otimes n}H^{\otimes n} = 2|\psi\rangle\langle\psi| - I$$



Фиг. 2.1. *Квантова мрежа, която осъществява итерацията на Гровер и означението ѝ в квантовите мрежи. Фигурата е заимствана от [1]*



Фиг. 2.2. *Квантова мрежа, която осъществява алгоритъма на Гровер. Фигурата е заимствана от [1]*

Итерацията на Гровер $\hat{G} = (2|\psi\rangle\langle\psi| - I)\hat{O}$ може да бъде интерпретирана като ротация в двумерното пространство около началния вектор $|\psi\rangle$ и състоянието, съдържащо решенията на търсенето (Фиг. 2.3). Нека $|\alpha\rangle$ е векторът на всички елементи y, които не са решения, а $|\beta\rangle$ е векторът на всички останали елементи x, които са решения. Тогава:

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{\forall y} |x\rangle$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{\forall x} |x\rangle$$

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle$$

Следователно първоначалното състояние на системата е в равнината, определена от $|\alpha\rangle$ и $|\beta\rangle$. Състоянието на началния вектор с коефициентите пред базисните състояния може да се напише, като се използва следното полагане:

$$\cos\left(\frac{\theta}{2}\right)=\sqrt{\frac{N-M}{N}}$$

Тогава:

$$\sin\left(\frac{\theta}{2}\right)=\sqrt{\frac{M}{N}}$$

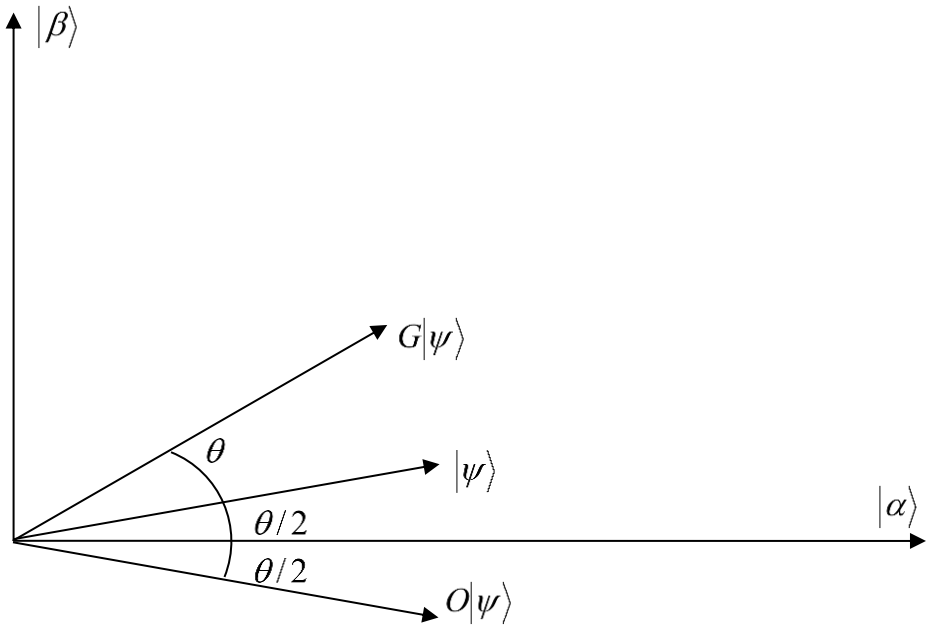
При това полагане състоянието на началния вектор може да се запише като:

$$|\psi\rangle=\cos\left(\frac{\theta}{2}\right)|\alpha\rangle+\sqrt{\frac{M}{N}}|\beta\rangle$$

От полагането следва, че действието на оператора на Гровер е ротация в равнината, определена от базисния вектор на всички решения и базисния вектор на всички нерешения:

$$\hat{G}|\psi\rangle=\cos\left(\frac{3\theta}{2}\right)|\alpha\rangle+\sin\left(\frac{3\theta}{2}\right)|\beta\rangle$$

$$\hat{G}^k|\psi\rangle=\cos\left(\frac{(2k+1)\theta}{2}\right)|\alpha\rangle+\sin\left(\frac{(2k+1)\theta}{2}\right)|\beta\rangle$$



Фиг. 2.3. *Оракулът отразява вектора $|\psi\rangle$ в равнината, определена от векторите $|\alpha\rangle$ и $|\beta\rangle$ на ъгъл $\theta/2$ спрямо оста $|\alpha\rangle$. Понеже $2|\psi\rangle\langle\psi|-I$ отразява $O|\psi\rangle$ спрямо вектора $|\psi\rangle$ в равнината, определена от векторите $|\alpha\rangle$ и $|\beta\rangle$ на ъгъл θ, то като цяло итерацията на Гровер води до завъртане на вектора $|\psi\rangle$ на ъгъл θ. Фигурата е заимствана от [1]*

В базис $|\alpha\rangle$, $|\beta\rangle$ итерацията на Гровер може да се запише като:

$$G = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Векторът $|\psi\rangle$ трябва да се завърти на $\arccos\left(\sqrt{M/N}\right)$ радиана за да съвпадне с вектора $|\beta\rangle$. Броят повторения R на итерацията на Гровер се изчислява по следния начин:

Ако $M << N$ то:

$$R = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rfloor$$

При $M = 1$:

$$N = 2^N$$

$$R = \left\lfloor \frac{\pi}{4}\sqrt{2^N} \right\rfloor$$

С увеличаване броя на решенията се увеличава броят итерации, необходим за намиране на едно от решенията. Най-малък брой итерации е необходим, когато има само едно решение. Когато $M \geq N/2$, алгоритъмът на Гровер не може да работи без модифициране.

Две реални матрици за кюдитния случай, които могат да бъдат използвани за аналог на Адамаровата матрица [34], се образуват по следния начин:

$$H_1 = Re(F) + Im(F)$$

$$H_2 = Re(F^{-1}) + Im(F^{-1})$$

За двумерния случай H_1 и H_2 съвпадат с Адамаровата матрица.

За случая на d=3:

$$H_1 = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2}-\frac{1}{2}\sqrt{3} & -\frac{1}{2}+\frac{1}{2}\sqrt{3} \\ 1 & -\frac{1}{2}+\frac{1}{2}\sqrt{3} & -\frac{1}{2}-\frac{1}{2}\sqrt{3} \end{bmatrix}$$

$$H_2 = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2}+\frac{1}{2}\sqrt{3} & -\frac{1}{2}-\frac{1}{2}\sqrt{3} \\ 1 & -\frac{1}{2}-\frac{1}{2}\sqrt{3} & -\frac{1}{2}+\frac{1}{2}\sqrt{3} \end{bmatrix}$$

Матриците също удовлетворяват описаните по-горе четири свойства.

Адамаровата матрица представлява двумерния случай на дискретната Фурие трансформация. Матриците на нормираната дискретна Фурие трансформация при размерност 3, както и обратната й матрица също могат да бъдат използвани за аналог на Адамаровата матрица за кюдитния случай.

Нормираната матрица на дискретната Фурие трансформация при размерност 3, както и обратната ѝ матрица могат да се запишат по следния начин:

$$F = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \exp\left(\frac{2i\pi}{3}\right) & \exp\left(-\frac{2i\pi}{3}\right) \\ 1 & \exp\left(-\frac{2i\pi}{3}\right) & \exp\left(\frac{2i\pi}{3}\right) \end{bmatrix}$$

$$F^{-1} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \exp\left(-\frac{2i\pi}{3}\right) & \exp\left(\frac{2i\pi}{3}\right) \\ 1 & \exp\left(\frac{2i\pi}{3}\right) & \exp\left(-\frac{2i\pi}{3}\right) \end{bmatrix}$$

В случай, че H_1 или H_2 се използват като аналог на Адамаровата матрица, алгоритъмът съдържа следните стъпки:

Прилага се аналога на Адамаровата трансформация ${H_{12}}^{\otimes n}$ върху началното състояние, което го променя в:

$$\left|\psi\right\rangle = {H_{12}}^{\otimes n}\left|0..0\right\rangle = {H_{12}}^{\otimes n}\left|0\right\rangle^{\otimes n} = \left(H_{12}\left|0\right\rangle\right)^{\otimes n} = \frac{1}{N^{1/2}}\sum_{x=0}^{N-1}\left|x\right\rangle,$$

където H_{12} може да е H_1 или H_2

След това се прилага $O\left(\sqrt{N}\right)$ пъти итерацията на Гровер за кютритния случай. Тя съдържа същите стъпки, както и в кюбитния случай [34]. Единствено се променят матриците в стъпки 2) и 4). Тези стъпки се променят както следва:

2) Прилага се аналогът на Адамаровата трансформация върху всички кютрити ${H_{12}}^{\otimes n}$.

4) Прилага се същият аналог на Адамаровата трансформация върху всички кютрити ${H_{12}}^{\otimes n}$.

В случай, че F или F^{-1} се използват за аналог на Адамарова матрица, алгоритъмът съдържа следните стъпки:

Началното състояние се трансформира в равномерна суперпозиция, както в предишният случай:

$$\left|\psi\right\rangle = {F_F}^{\otimes n}\left|0..0\right\rangle = {F_F}^{\otimes n}\left|0\right\rangle^{\otimes n} = \frac{1}{N^{1/2}}\sum_{x=0}^{N-1}\left|x\right\rangle,$$

където F_F може да е F или F^{-1}.

Когато за аналог на Адамарова матрица се използва F_F се променят отново стъпките 2) и 4) в итерацията на Гровер. Стъпките 1) и 3) остават както са описани по-горе [34]. Стъпките 2) и 4) се променят по следния начин:

2) Прилага се единят аналог на Адамаровата трансформация $(F_F)^{\otimes n}$.

4) Прилага се другият аналог на Адамаровата трансформация $\left((F_F)^{-1}\right)^{\otimes n}$.

Всяка операция в итерацията на Гровер може да бъде ефективно изпълнена чрез квантов компютър. За всяка от стъпките 2) и 4) се изискват $n = \log_3(N)$ операции. 3) изисква O(n) на брой гейтове.

Геометричната интерпретация е идентична с тази при случая на кюбити.

Броят итерации на Гровер, необходим за намиране на някое от решенията е:

$$R = \left[\frac{\pi}{4}\sqrt{\frac{N}{M}}\right] = \left[\frac{\pi}{4}\sqrt{\frac{3^n}{M}}\right]$$

Когато има само едно решение (M=1):

$$R = \left[\frac{\pi}{4}\sqrt{3^n}\right]$$

Като аналог на Адамаровата матрица в итерацията на Гровер може да се ползва произволна унитарна матрица S, която има равни модули на елементите в първата колона [34].

Началното състояние на системата е $|0_k\rangle = |0 \dots 0\rangle$, т.е. всеки един от кюдитите е поставен в състояние $|0\rangle$.

Началното състояние на системата $|0 \dots 0\rangle$ под действието на унитарната матрица с произволни амплитуди S се привежда до състояние $|\alpha\rangle$. Състоянието $|\alpha\rangle$ е суперпозиция от състояния с равни амплитуди, но с произволни относителни фази. Операторът S трябва да се приложи поотделно върху всеки един от кюдитите:

$$S|0_k\rangle = \sum_{q=0}^{d-1} \xi_k |q_k\rangle,$$

където $|\xi_к| = 1/\sqrt{d}$.

Трансформацията на всеки един от кюдитите (k=1,…,n) може да се запише по следния начин:

$$|\alpha\rangle = S^{\otimes n}|0\rangle = \sum_{x=1}^{N} \alpha_x |x\rangle,$$

където $|\alpha_x| = 1/\sqrt{N}$ и $N = 3^m$ е големината на базата данни.

Относителните фази в суперпозицията може да са произволни. Трябва да се използва една и съща матрица S на всяка стъпка.

Матрицата трябва да е унитарна, за да може итерациите да са кохерентни.

$$SS^{-1} = I$$

$$S^* = (\bar{S})^T = S^{-1}$$

Следващите стъпки на алгоритъма са същите като при алгоритъма на Гровер с кюдити, когато е използвана F или F^{-1}. Стъпките 1) и 3) са като в описаните по-горе кютритни случаи. Стъпките 2) и 4) се променят както следва:

2) Прилага се аналогът на Адамаровата трансформация върху всички кюдити S.

4) Прилага се S^{-1} върху всички кюдити.

Получената вероятност за намиране на търсеното състояние е една и съща, независимо коя от петте матрици (S, F , F^{-1}, H_1 или H_2) се използва:

3. Значение на квантовия алгоритъм за определяне на фазата

В квантовата информация се изисква изчисленията да бъдат обратими (с изключение на еднопосочните квантови компютри) и квантовата система да бъде изолирана от външни въздействия. Това въвежда условието за унитарност на операторите. Много задачи могат да бъдат сведени до намирането на собствените стойности на унитарни оператори. Примери за такива задачи са факторизацията, намирането на броя решения, удовлетворяващи условията на търсене, алгоритъм за намиране на реда, алгоритъм за намиране на дискретния логаритъм и други [1]. Задачи от такъв тип са с голяма трудност за класическия компютър, понеже необходимият брой операции нараства експоненциално O(n) с увеличаване големината на регистрите. Алгоритъмът на квантовия брояч намира броя решения експоненциално по-бързо O(log(n)) от класическите му аналози, а алгоритъмът на Шор разделя число на прости множители квадратично по-бързо O($\sqrt{n}$) от всеки класически алгоритъм, като използват в себе си като съставна част алгоритъма за пресмятането на фазите.

3.1. Алгоритъм за пресмятане на фазата за кюбити

Квантовият алгоритъм за определяне на фазата намира неизвестната фаза [1] φ в следното уравнение:

$$U|u\rangle = \exp(2\pi i\varphi)|u\rangle,$$

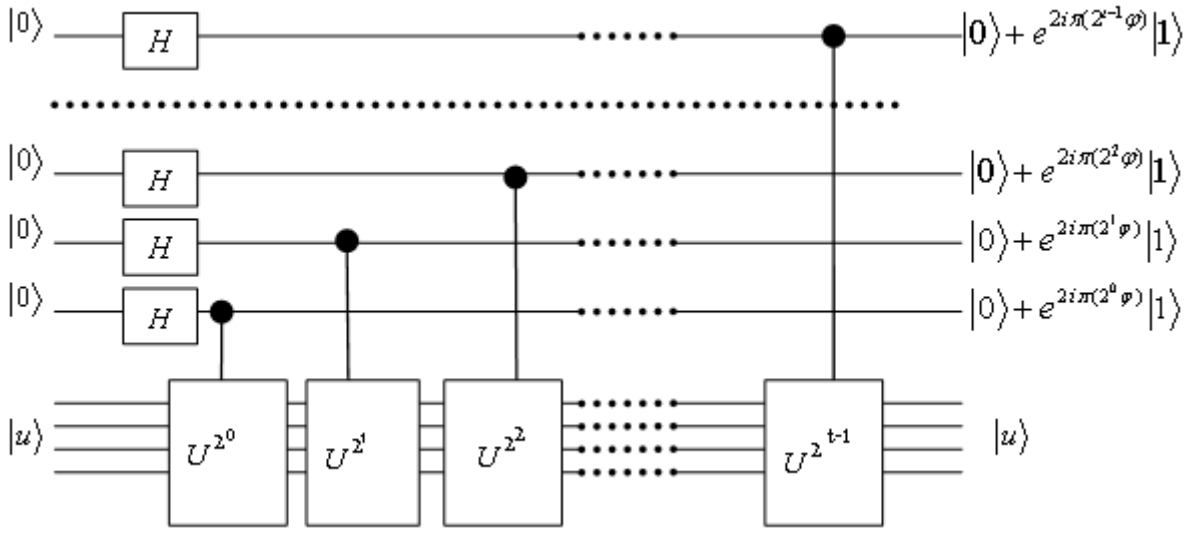
където U е унитарен оператор със собствен вектор $|u\rangle$ и собствена стойност $\exp(2i\pi\varphi)$.

Пресмятането на фазата φ използва последователност от контролирани гейтове, съдържащи оператори U на различни степени и обратна квантова Фурие трансформация. Квантовото пресмятане на фазата използва два регистъра [1]. Първият регистър, в началния момент преди да започне алгоритъма, съдържа t кюбита в състояние $|0\rangle$. Избора на стойност за t зависи от броя цифри след десетичната запетая, с които трябва да бъде определено φ и с каква вероятност е необходимо да бъде успешно пресмятането на фазата. Началното състояние на втория регистър е

собствения вектор $|u\rangle$ и се състои от толкова кюбити, колкото са необходими, за да може да се запише $|u\rangle$.

Пресмятането на фазата има два етапа.

1) Първият етап се състои от две части [1]. Първо се прилага Адамаровият гейт върху всеки кюбит от първия регистър, за да се получи равномерна суперпозиция от всички възможни състояния на регистъра. След това се прилагат контролирани гейтове с различните степени на оператора U. Квантовата мрежа на първия етап е представена на Фиг. 3.1:



Фиг. 3.1. *Квантова мрежа, подготвяща първия етап от пресмятането на фазата. Фигурата е заимствана от [1]*

Състоянието на втория регистър остава непроменено по време на първата част от алгоритъма. Крайното състояние на първия регистър след прилагането на квантовата мрежа става $|\psi\rangle$:

$$|\psi\rangle = \frac{1}{2^{t/2}}\left(|0\rangle + e^{2i\pi 0.\varphi_t}|1\rangle\right)\left(|0\rangle + e^{2i\pi 0.\varphi_{t-1}\varphi_t}|1\rangle\right)..\left(|0\rangle + e^{2i\pi 0.\varphi_1,\varphi_2...\varphi_t}|1\rangle\right).$$

Неизвестната фаза φ може да бъде записана точно в t бита чрез разложение по степени на двойката:

$$\varphi = \frac{\varphi_1}{2} + \frac{\varphi_2}{4} + ... + \frac{\varphi_\infty}{2^{\infty-1}} = \sum_{i=1}^{\infty}\frac{\varphi_i}{2^i} = 0.\varphi_1\varphi_2..\varphi_\infty$$

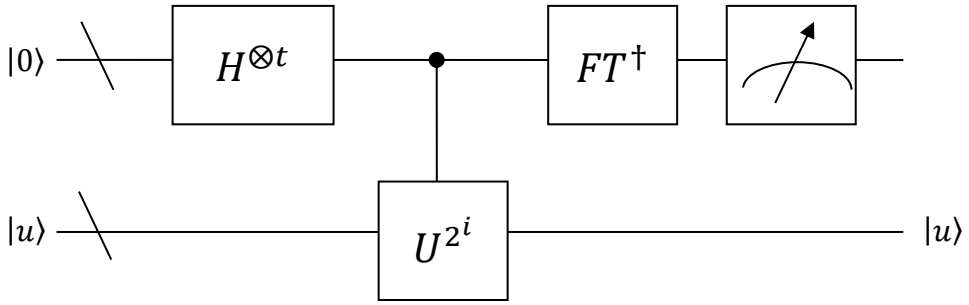
2) Вторият етап от пресмятането на фазата е прилагането на обратна Фурие-трансформация върху първия регистър. Обратната квантова Фурие трансформация изисква $O(t^2)$ стъпки.

3) Последната стъпка е прочитане състоянието на първия регистър чрез измерване в изчислителния базис. След обратната Фурие трансформация измерването в

изчислителния базис ще намери φ в контролният регистър с точност, която позволява броя кюбити. Ако φ може да се запише в t бита, то резултатът е φ:

$$\hat{F}^{-1}\left|\psi\right\rangle = \frac{1}{2^{t/2}}\sum_{k,l=0}^{2^{t-1}} e^{\frac{-2i\pi kl}{2^t}} e^{2i\pi\varphi k}\left|l\right\rangle = \frac{1}{2^{t/2}}\sum_{k,l=0}^{2^{t-1}} e^{\frac{2^t\varphi}{l}}\left|l\right\rangle = \left|\varphi\right\rangle$$

Квантовата мрежа за осъществяването на квантовото определяне на фазата е дадена на Фиг. 3.2.



Фиг. 3.2: *Квантова мрежа за осъществяване на алгоритъма за определяне на фазата в кюбитния случай. Фигурата е заимствана от [1]*

Когато φ не може да се запише точно, в регистъра се записва най-добрата t-значна апроксимация на φ, която е по-малка от φ [1].

За да бъде получено успешно φ с точност от n бита с вероятност за успех поне $1-\varepsilon$, то трябва:

$$t = n + \left\lceil \log\left(2 + \frac{1}{2\varepsilon}\right)\right\rceil.$$

3.2. Алгоритъм за квантов брояч

Броят пъти, който трябва да бъде приложена итерацията на Гровер в алгоритъмът на Гровер, за да се намери някой от търсените елементи, може да се пресметне по формулата:

$$R \le \left\lfloor \frac{\pi}{2\theta} \right\rfloor \le \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil.$$

За да се пресметне R трябва да се знае точният брой решения M на търсените елементи. За да е изпълнено горното неравенство трябва $M < N/2$.

Броят решения може да се намери с едно приложение на алгоритъма за определяне на фазата, наречено квантов брояч [35]. Квантовият брояч е много по-ефективен от класическия алгоритъм за търсене и може да намери много по-бързо от всеки класически алгоритъм дали съществуват решения и колко са те. Действието на квантовия брояч може да се обясни по начина, показан по-долу.

Нека $|\alpha\rangle$ са всички елементи, които не са решения на проблема за търсене, а $|\beta\rangle$ са всички елементи, които са решения.

Операторът за усилване на амплитудата съвпада с оператора на итерацията на Гровер и се отбелязва по същия начин с G. Векторите $|\alpha\rangle$ и $|\beta\rangle$, както бе описано в главата за алгоритъма на Гровер, могат да се запишат по следния начин:

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}}\sum_{x\in\alpha}|x\rangle \qquad\qquad |\beta\rangle = \frac{1}{\sqrt{M}}\sum_{x\in\beta}|x\rangle$$

Прилага се Адамаров гейт върху началното състояние на системата и се получава равномерна суперпозиция от всички начални състояния.

$$|\psi\rangle = H|00\ldots0\rangle = \frac{1}{\sqrt{M}}\sum_{x=0}^{N-1}|x\rangle$$

Векторът $|\psi\rangle$ може да се изрази в базиса $|\alpha\rangle$ и $|\beta\rangle$ по следният начин:

$$|\psi\rangle = \frac{\sqrt{N-M}}{\sqrt{N}}|\alpha\rangle + \frac{\sqrt{M}}{\sqrt{N}}|\beta\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{\theta}{2}\right)|\beta\rangle.$$

Ако се дефинират $|+\rangle$ и $|-\rangle$ по следните състояния:

$$|+\rangle = \frac{1}{\sqrt{2}}(|\alpha\rangle + i * |\beta\rangle) \qquad\qquad |-\rangle = \frac{1}{\sqrt{2}}(|\alpha\rangle - i * |\beta\rangle)$$

Тогава векторът $|\psi\rangle$ може да се изрази в базиса определен от $|+\rangle$ и $|-\rangle$ по следния начин:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(e^{-i\theta/2}|+\rangle + e^{i\theta/2}|-\rangle\right)$$

$$G|+\rangle = \left(e^{-i\theta}\right)|+\rangle \qquad\qquad G|-\rangle = \left(e^{i\theta}\right)|-\rangle$$

Това означава, че $|+\rangle$ и $|-\rangle$ са собствени вектори на G със собствени стойности $\left(e^{-i\theta}\right)$ и $\left(e^{i\theta}\right)$.

$$G|\psi\rangle = \frac{1}{\sqrt{2}}\left(e^{-i\theta/2}G|+\rangle + e^{i\theta/2}G|-\rangle\right) = \frac{1}{\sqrt{2}}\left[e^{-i\theta/2}\left(e^{-i\theta}\right)|+\rangle + e^{i\theta/2}\left(e^{i\theta}\right)|-\rangle\right]$$

По този начин задачата за намиране броя решения се свежда до задача за намиране на собствени стойности на оператор.

θ е и ъгълът на ротация на итерацията на Гровер:

$$G = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Квантовата мрежа, която отговаря на квантов брояч, е тази показана на Фиг. 3.3, където за унитарен оператор се използва операторът на Гровер, а началното състояние на втория регистър е равномерна суперпозиция.

Мрежата на квантовия брояч определя θ с m бита точност и с вероятност на успех най-малко $1-\varepsilon$. Първият регистър съдържа t кюбита. Стойността на t се

определя в зависимост от това, каква точност и каква вероятност на успех са необходими:

$$t = m + \lceil \log(2 + (2\varepsilon)^{-1}) \rceil.$$

След като ъгълът θ вече е определен чрез квантовата мрежа, броят решения и ъгълът θ участват в следното равенство:

$$M = 2N \sin^2\left(\frac{\theta}{2}\right).$$

На класическия алгоритъм му трябват $O(N)$ обръщения към Оракула, за да определи броя решения M. Квантовият алгоритъм за търсене и алгоритъмът на квантовия брояч изискват $O(\sqrt{N})$ обръщения към Оракула. Това прави комбинацията от квантовия алгоритъм за търсене и квантов брояч толкова перспективна.

3.3. Пресмятане на фазата за кюдити

Ефективен начин да бъде построен квантовият брояч с кюдити е чрез използване на кюбити в контролния регистър A и кюдити в таргет регистър B. Стъпките на квантовия брояч с кюдити са същите, като тези на алгоритъма с кюбити. Единствената разлика е, че контролният регистър (A) и таргет регистър (B) имат различна размерност.

3.4. Квантов брояч с кюдити

Операторът, който се използва в алгоритъма на Гровер с кюдити, може да се използва за построяване и на квантов брояч с кюдити [36]. Квантовият брояч с кюдити съдържа два регистъра A и B. Регистъра A съдържа k кюбита, а регистъра B съдържа n кюдита. Алгоритъмът започва с привеждане на регистъра в начално състояние $|0\rangle_A|0\rangle_B = |00 \dots 0\rangle$.

След това състоянието се преобразува в равномерна суперпозиция на всички базисни състояния. Върху началното състояние $|0\rangle_A|0\rangle_B$ се прилагат: Адамаровият гейт H върху всеки един от кюбитите в регистъра A и операторът на дискретната Фурие трансформация F върху всеки кюдит от регистъра B. Това трансформира всеки кюдит в равномерна суперпозиция от състоянията му с размерност d.

$$H^{\otimes k}|0\rangle_A S^{\otimes n}|0\rangle_B = \frac{1}{\sqrt{2^k d^n}} \sum_{j=0}^{2^k-1} |j\rangle_A \sum_{l=0}^{d^n-1} |l\rangle_B$$

$$G = RO = S^{\otimes n} Ph (S^{\dagger})^{\otimes n},$$

където Ph е фазовият гейт (оператора, който сменя знака на състоянието $|00 \dots 0\rangle$), O е Оракула (оператор, който сменя знака на търсеното състояние), а S е аналог на Адамаровия гейт.

След това се прилага квантовият алгоритъм за определяне на фазите, който намира собствените стойности на G. Измерването на състоянието на регистъра A дава стойността на ъгъла θ и $2\pi - \theta$. След това може да се пресметне броят решения по формулата:

$$M = Nsin^2\left(\frac{\theta}{2}\right)$$

3.5. Примери за числени симулации на квантов брояч с кюдити

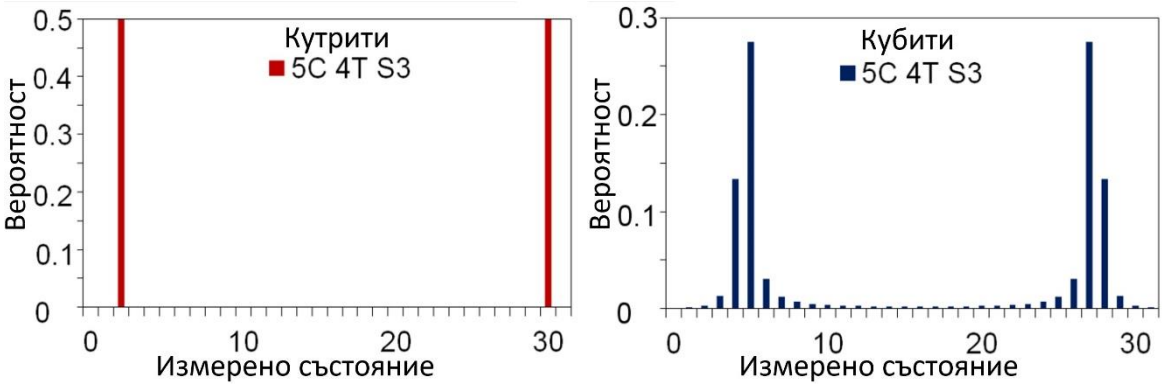
Резултатите от числените симулации на квантовия брояч с кюбити и кюдити [36] е показани на Таблица 3.1.

Кубити				
Брой кубити в A	Брой кубити в B	Състояния в B	Обща вероятност	Оценка на M
5	4	5 и 27	0.549	3.555
6	3	13 и 51	0.533	2.839
6	4	9 и 55	0.953	2.925
6	5	6 и 58	0.676	2.696
7	3	27 и 101	0.931	3.028
7	4	18 и 110	0.819	2.925
8	4	36 и 220	0.413	2.925

Кутрити				
Брой кубити в A	Брой кутрити в B	Състояния в B	Обща вероятност	Оценка на M
5	4	2 и 30	0.998	3.083
6	3	7 и 57	0.981	3.064
6	4	4 и 60	0.990	3.083
6	5	2 и 62	0.793	2.335
7	3	14 и 114	0.925	3.064
7	4	8 и 120	0.961	3.083
8	4	16 и 240	0.852	3.083

Таблица 3.1: *Резултати от числената симулация на квантовия брояч за състоянията на регистър B, отговарящи за ъглите θ и $2\pi - \theta$, заедно с вероятността да бъдат измерени (четвъртата колона) и оценката за броя решения на измереното състояние (петата колона). Резултатите са за три елемента на търсене.*

Може да се види в Таблица 3.1, че когато се използват кютрити вместо кюбити, то вероятността да се намери броят на решенията нараства значително.



Фиг. 3.3: *Числени симулации на квантовия брояч с кюбити и кютрити, показващи вероятността да се измери всяко състояние на контролния регистър A. Сините графики показват случая, когато контролният регистър A и таргет регистърът B са съставени от кюбити. В червените графики контролният регистър A отново е съставен от кюбити, а таргет регистърът B е съставен от кютрити. Броят търсени елементи е равен на три.*

Процедурата за намиране решение на задачата за търсене е следната: след края на квантовата част на алгоритъма с кюбити или кютрити се стига до вероятностно разпределение между всички състояния (2^k) на контролния регистър. Пресмятането при произволно състояние става чрез формулата:

$$\theta[rad] = \frac{2\pi}{2^k}\vartheta,$$

където ϑ е измерената стойност от регистъра.

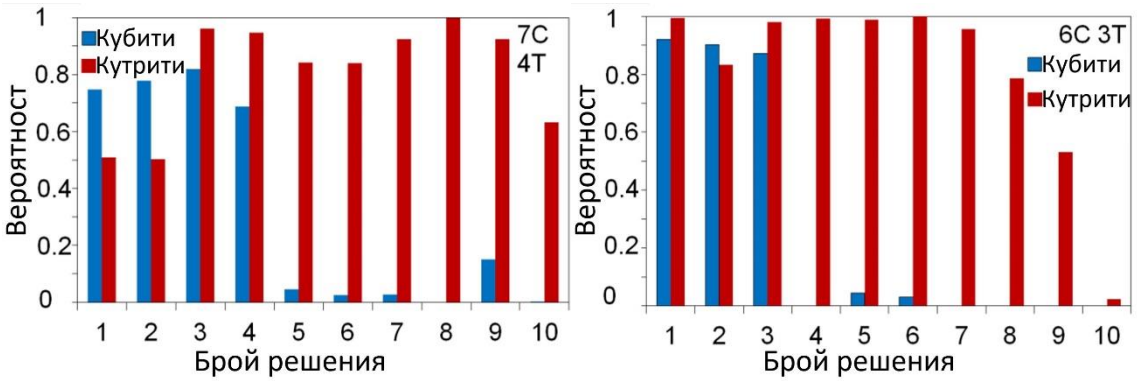
$$\mathrm{M} = \mathrm{N}\sin^2\left(\frac{\theta}{2}\right) = \mathrm{N}\sin^2\left(\frac{2\pi - \theta}{2}\right)$$

Следователно трябва да се вземе сборът на вероятностите и на двата ъгъла θ и $2\pi - \theta$ (съответно P_θ и $P_{-\theta}$), за да се получи вероятността P_M за измерване на M:

$$P_M = P_\theta + P_{-\theta}$$

На Фиг. 3.3 (A=5, B=4 и S=3) се вижда, че заселеността на състоянията на кюбитите е разпределена между няколко състояния в контролния регистър A. При кютритите заселеността е концентрирана главно в двете състояния, които отговарят на θ и $2\pi - \theta$. Затова вероятността да се намери броят решения с алгоритъма на квантовия брояч с кютрити е много по-висока.

При малък брой кюбити в контролния регистър A алгоритъмът на квантовия брояч се държи хаотично, поради голямата стъпка на ъгъла θ (и следователно и броя решения M). Когато нараства броят на контролните кюбити, стъпката на θ намалява и алгоритъмът на квантовия брояч става по-точен за намирането на броя решения. Вариантът на алгоритъма с кютрити дава по-добри резултати от резултатите на варианта с кюбити (Фиг. 3.4). Кюбитите работят ефективно само за малък брой решения M, докато вариантът с кютритите работи добре дори и при голям брой решения M.



Фиг. 3.4: *Числена симулация на ефективността на квантовия брояч при различен брой кюбити в контролния регистър A и различен брой кюдити в таргет регистъра B, като функция на броя решения M. Вероятността при кютритите е отбелязана с червено, а за кюбитите са отбелязана със синьо.*

Алгоритъмът на квантовия брояч работи добре, ако броят състояния в регистър A (тоест 2^k) е много по-голям от броя състояния в регистъра B (тоест d^n). Това води до малка ($\Delta M \ll 1$) стъпка на стойността на M. Същевременно размерът на регистрите A и B трябва да е достатъчно голям, за да може да се намери фазата θ с достатъчно голяма точност. Затова условията за алгоритъма на квантовия брояч с кюдити са:

$$2^k \gg d^n, \qquad k \gg 1, \qquad n \gg 1.$$

Квантовият брояч с кютрити работи значително по-добре от кюбитния вариант на алгоритъма даже при подобна големина на регистъра

4. Квантови алгоритми, базирани на произволно преместване

В някои практически случаи се изисква обхождане на линейна неподредена база данни, например при записаните резултати от сблъсъците на частици [37]. При решаване на задачата с квантов компютър квантовият алгоритъм на Гровер е достатъчен, за да се обходи ефективно базата данни, по-бързо от произволен класически детерминистичен или вероятностен алгоритъм. При много случаи се изисква обхождане на дадена структура, различна от линейната база данни, например обхождане на дърво [38]. При по-сложна структура на базата данни алгоритъмът на Гровер не може да се приложи без предварителна трансформация на базата данни, което би направило алгоритъма крайно неефективен. При обхождане на такава структура трябва да се използват алгоритмите за търсене, базирани на квантовото произволно преместване. Те могат да се използват за ефективно търсене на върховете на произволен граф [39, 40]. Квантовите алгоритми за търсене, базирани на произволно преместване са далеч по-ефективни от класическите.

Квантовият алгоритъм за произволно преместване е квантов аналог на класическото произволно преместване. Те ползват терминология от теорията на вероятностите, която е разработвана при теория на игрите. Затова се използват термини като киралност и монета.

4.1. Непрекъснат и дискретен квантов алгоритъм за произволно преместване

Съществуват два вида квантови алгоритми за произволно преместване –дискретен [41] и непрекъснат [27].

Непрекъснат алгоритъм за произволно преместване върху граф [27] е марковски процес, с фиксирани вероятности за всяка единица време да премине от върха, в който се намира в момента, в съседен връх.

Дискретният квантов алгоритъм за произволно преместване [41] е квантов аналог на класическия алгоритъм за произволно преместване. Също както и при класическия алгоритъм за произволно преместване, всяка стъпка на алгоритъма, която се описва с унитарен оператор U, има две части. Първата е прилагането на монетата. Състоянието на монетата се определя от нейния унитарен оператор C_0, която действа в пространството на монетата H_C. Операторът на монетата действа върху Хилбертовите пространства на монетата и върховете $H_C \otimes H_S$ и се записва като $C = C_0 \otimes I$. Резултатът от действието на оператора на монетата върху пространството на монетата е „кирално" състояние. То е аналог на класическата вероятност. Понеже киралността е квантов оператор, то може да има квантова суперпозиция от „посоките". Състоянието на системата се променя в зависимост от вида на структурата, която се обхожда и резултата от „хвърлянето" на монетата. Квантовият оператор, отговарящ за структурата, е матрица на смесването S и извършва контролирана смяна в зависимост от състоянието на пространството на монетата. В зависимост от матрицата на смесване киралните посоки са различни. Операторът S действа на пространството $H_C \otimes H_S$. Сумарно всяка стъпка на квантовото произволно движение по линия може да се запише като:

$$U = SC$$

Квантовото произволно преместване за N стъпки може да се запише като:

$$|\mathrm{d_f}, \mathrm{x_f}\rangle = (SC)^{\mathrm{N}}|\mathrm{d_i}, \mathrm{x_i}\rangle,$$

където $\mathrm{x_i}$ и x_f са началното и крайното състояние на вълновата функция на частицата, извършваща произволното движение върху линията. От своя страна $\mathrm{d_i}$ и d_f са началното и крайното състояние на пространството на монетата.

4.2. Квантов алгоритъм на произволно преместване върху линия

Квантовото произволно преместване върху линия може да се дефинира по следния начин: частица се движи по линия съгласно параметъра киралност. Той има две стойности – "ляво" и "дясно". Описва се с двукомпонентна вълнова функция [42].

$$\Psi(\mathrm{n}, \mathrm{t}) = \begin{pmatrix} \psi_{\mathrm{L}}(\mathrm{n}, \mathrm{t}) \\ \psi_{\mathrm{R}}(\mathrm{n}, \mathrm{t}) \end{pmatrix},$$

където n е позицията на частицата, а t е номерът на стъпката.

Матрицата на смесване за произволно движение върху линия (S_L) е:

$$S_L = \sum_{d=0}^{1} \sum_x |d, x-(-1)^d\rangle\langle d, x|,$$

където x е позицията на частицата върху линията. Стойностите на d (0 и 1) се променят под действие на монетата и отговарят на двете киралните състояния (“ляво” и “дясно”) в позицията x.

Преходът между две стъпки (състояния на системата) става чрез унитарни трансформации, които променят киралността на състоянието.

В общия случай за монета може да се използва произволен двумерен унитарен оператор. Резултатът от квантовото произволно движение по линия зависи не само от началното състояние, а и от монетата, която е избрана. Ако за монета се вземе матрицата на Адамар:

$$C_{LH} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Двете киралните състояния ляво $|L\rangle$ и дясно $|R\rangle$ във всяка позиция върху линията са произведение от състоянието на позицията на частицата $|x\rangle$ и състояние (0 или 1), което се променя под действие на монетата. Те имат следния запис:

$$|R\rangle = |x, 1\rangle \qquad\qquad |L\rangle = |x, 0\rangle$$

Действието на монетата върху състояние x може да се запише като:

$$C_{LH}|R\rangle = \frac{1}{\sqrt{2}}(|x, 0\rangle - |x, 1\rangle) \qquad\qquad C_{LH}|L\rangle = \frac{1}{\sqrt{2}}(|x, 0\rangle + |x, 1\rangle)$$

Преместването на следващата позиция става чрез оператора на смесване:

$$S_L|x, 0\rangle = |x-1, 0\rangle \qquad\qquad S_L|x, 1\rangle = |x+1, 0\rangle$$

Всяка стъпка на дискретния квантов алгоритъм за произволно преместване е:

$$U_L = S_L C_{\mathbb{L}}$$

$$C_{\mathbb{L}} = C_L \otimes I,$$

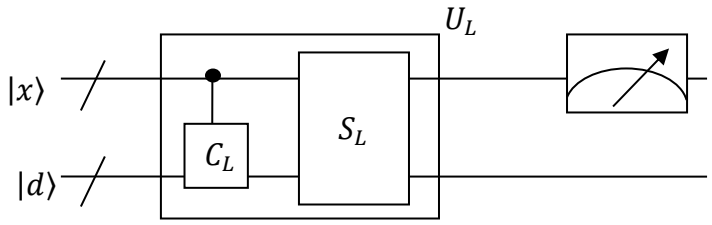
където I е единичен оператор с размерност на пространството на възлите.

Разпределението в резултат на произволното преместване по линията не е симетрично, когато началното състояние е |0> или |1> и за монета се използва матрицата на Адамар. Тази асиметрия се дължи на факта, че монетата действа по различен начин на $|0\rangle$ и $|1\rangle$, по-точно умножава по -1 само при началното състояние на монетата $|1\rangle$. Това води до деструктивна интерференция в случаите, когато частицата отива надясно и конструктивна интерференция, когато частицата отива наляво.

Началните състояния се разпространяват независимо и не интерферират, когато се използва симетричната монета $C_{LS} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$ и начално състояние $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. Това дава оптимално обхождане на линията като няма преференция за положителната или отрицателната посоки.

Зависимостта между статистическата вариация и броя стъпки t между класическото и квантовото произволно движение по линия силно се различава. Дискретното квантово произволно движение по линия се разпространява като $O(t)$, за разлика от класическото движение по линия, което се разпространява като $O(\sqrt{t})$.

Мрежа на квантовия алгоритъм за произволно движение по линия е показана на Фиг. 4.1. На нея U_L е една стъпка на произволно преместване по линия, S_L и C_L са съответно операторите на смесването и монетата. Векторите на състоянията $|x\rangle$ и $|d\rangle$ отговарят за позицията и монетата.



Фиг. 4.1: *Мрежа, описваща квантовото произволно преместване върху линия. Тази квантова мрежа важи за произволен граф след замяната на* $S_L \to S$ *и* $C_L \to C$. *Фигурата е заимствана от [42]*

4.3. Алгоритъм за произволно преместване върху хиперкуб

Всеки от върховете на n-мерен хиперкуб е свързан с n други върха. Върховете на един хуперкуб могат да се означат с двоичен числов низ от n символа [26]

Хаминговата тежест [1] е броя символи в даден стринг, които се различават от стринг със същия брой символи, но запълнен само с нули. Хамингово разстояние [1] между два стринга е броя символи в двоичен запис, по който се различават.

Два нода на един хиперкуб са свързани, ако разликата в Хаминговата им тежест е 1, тоест двоичният запис на номерацията им се различава само по една цифра (например 11011 е съседен с 10011) [26].

Класическият и квантовият алгоритми за произволно движение върху хипиркуб може да се сведат до класически или съответно квантов алгоритъм за произволно движение върху линия [26]. Започва се от върха с Хамингова тежест 0 и се отива в някой от върховете с Хамингова тежест 1. Въпрос на означение е кой връх ще се означи като 00…0. Останалите се означават в зависимост от това, кой е избран като нулев. При избран връх с Хамингова тежест 0, всички върхове с Хамингова тежест 1 могат да се разменят произволно, без да се налага да се модифицира произволното движение. Аналогично всички върхове, които са на равно Хамингово разстояние от връх с Хамингова тежест 0 (тоест са с равна Хамингова тежест) могат да се разменят, без да се промени произволното движение. Това позволява всички върхове с еднаква Хамингова тежест да се проектират в една точка. Така симетричното произволно движение по хиперкуб се свежда в произволно несиметрично движение по линия. Хилбертовото пространство от върховете на хиперкуба и броя страни, с които те са свързани е:

$$\mathrm{H} = \mathrm{H_C} \otimes \mathrm{H_P} = \mathrm{H^n} \otimes \mathrm{H^{2^n}}.$$

При всеки връх има равнопоставеност на различните посоки на преместване в произволното преместване върху хиперкуб.

Следователно монетата на Гровер е най-добрата монета за максимално бързо обхождане на състоянията. Тя се използва и в алгоритъма на Гровер за смесване на състоянията:

Монетата на Гровър може да се запише и по следния начин [28]:

$$\mathrm{C_{H1}} = G = I - 2|sv\rangle\langle sv|,$$

където $|sv\rangle = |s\rangle \otimes |v\rangle$ е равномерна суперпозиция на всички начални състояния.

$$|\mathrm{sv}\rangle = \frac{1}{\sqrt{\mathrm{n}}} \sum_{\mathrm{i}=0}^{\mathrm{n}} |\mathrm{i}\rangle$$

Операторът за смесване при Хиперкуб е:

$$\mathrm{S_H} = \sum_{\mathrm{d}=0}^{\mathrm{n}-1} \sum_{\vec{\mathrm{x}}}^{\mathrm{r}} |\mathrm{d}, \vec{\mathrm{x}} \oplus \vec{\mathrm{e}}_{\mathrm{d}}\rangle\langle \mathrm{d}, \vec{\mathrm{x}}|,$$

където $\vec{\mathrm{e}}_{\mathrm{d}}$ е d-тия базисен вектор на хиперкуба, $\vec{\mathrm{x}}$ е битов стринг, отговарящ за позицията върху хиперкуба.

Квантовата мрежа, която може да осъществи квантовото произволно преместване върху хиперкуб, е същата като показаната на Фиг. 4.1, само операторите на смесване и монетата трябва да се сменят с такива за хиперкуб.

4.4. Алгоритъм на произволно преместване и търсене върху хиперкуб и симулация на алгоритъма

Квантовият алгоритъм за търсене при произволно преместване (QRWS) върху хиперкуб използва две монети. Едната осигурява най-добро обхождане на елементите, а другата маркира решенията. Алгоритъмът съдържа Оракул, аналогично на алгоритъма на Гровер, който трябва да разпознае решението, когато му бъде подадено.

$$|\mathrm{x}, \mathrm{d}\rangle \xrightarrow{O} \begin{cases} \mathrm{C}_{H1}|\mathrm{x} \neq \mathrm{x_{target}}, \mathrm{d}\rangle \\ \mathrm{C}_{H2}|\mathrm{x} = \mathrm{x_{target}}, \mathrm{d}\rangle \end{cases}$$

Следователно всяка стъпка на произволното движение върху хиперкуба може да се запише така:

$$U_L|\mathrm{x}, \mathrm{d}\rangle = \begin{cases} S_H \mathrm{C_{H1}}|\mathrm{x} \neq \mathrm{x_{target}}, \mathrm{d}\rangle \\ S_H \mathrm{C}_{H2}|\mathrm{x} = \mathrm{x_{target}}, \mathrm{d}\rangle \end{cases},$$

където C_{H1} и $\mathrm{C_{H2}}$ са две различни монети, $|\mathrm{x}, \mathrm{d}\rangle$ е комбинираното състояние на позицията $|\mathrm{x}\rangle$ и състоянието на монетата $|\mathrm{d}\rangle$. Трябва да се отбележи, че Оракула може да маркира и повече от едно състояние. Най-добрата монета за максимално бързо

обхождане на състоянията е монетата на Гровер. Затова нека тя да е монетата, която определя преместването между състоянията:

$$\mathrm{C_{H1}} = G$$

На маркираните върхове на хиперкуба се прилага единичната монета със знак минус, вместо монетата ползвана за обхождане на елементите:

$$\mathrm{C}_{H2} = -\mathbb{I}$$

Действието на придвижващата монета върху комбинираното състояние на върховете и монета може да се запише по следният начин:

$$C_H = C_{H1} \otimes \mathbb{I}_{2^{\mathrm{n}}\mathrm{n}}.$$

Маркиращата монета, която действа на пълното Хилбертово пространство, може да се запише по следния начин:

$$C_H{}' = C'_{H1} \otimes \mathbb{I}_{2^{\mathrm{n}}\mathrm{n}}$$

$$C_H{}' = C_{H1} \otimes \mathbb{I}_{2^{\mathrm{n}}\mathrm{n}} - (C_{H1} - C_{H2}) \otimes \left|\mathrm{x_{target}}\right\rangle\!\left\langle\mathrm{x_{target}}\right|$$

Оператора U' може да се изрази чрез C_{H1} и C_{H2}:

$$\mathrm{U}' = S_H C_H{}' = S_H C_H - S_H \left((C_{H1} - C_{H2}) \otimes \left|\mathrm{x_{target}}\right\rangle\!\left\langle\mathrm{x_{target}}\right|\right) = U(\mathbb{I}_{2^{\mathrm{n}}\mathrm{n}} - 2|\mathrm{sv}\rangle\langle\mathrm{sv}|)$$

$|\mathrm{sv}\rangle$ е равномерната суперпозиция от всички начални състояния на монетата и на върховете и те са съответно $|\mathrm{s}\rangle$ и $|\mathrm{v}\rangle$.

$$|\mathrm{sv}\rangle = |\mathrm{s}\rangle \otimes |\mathrm{v}\rangle = |\mathrm{s},\mathrm{v}\rangle$$

$$C_{H1}|\mathrm{s}\rangle = |\mathrm{s}\rangle$$

Стъпки на алгоритъма:

1. Началното състояние на регистъра на монетата и регистъра на върховете на хиперкуба е равномерна суперпозиция от всички състояния на регистъра на върховете и регистъра на всички връзки с други върхове които има.

2. Оператора W=S.C’ се прилага h пъти:

$$h = \left\lceil \frac{\pi}{2}\sqrt{2^n} \right\rceil = \lceil x \rceil,$$

където $\lceil x \rceil$ е числото x закръглено нагоре:

а) върху немаркираните върхове се използва монетата C_0, а върху маркираните се прилага монетата C_1 ;

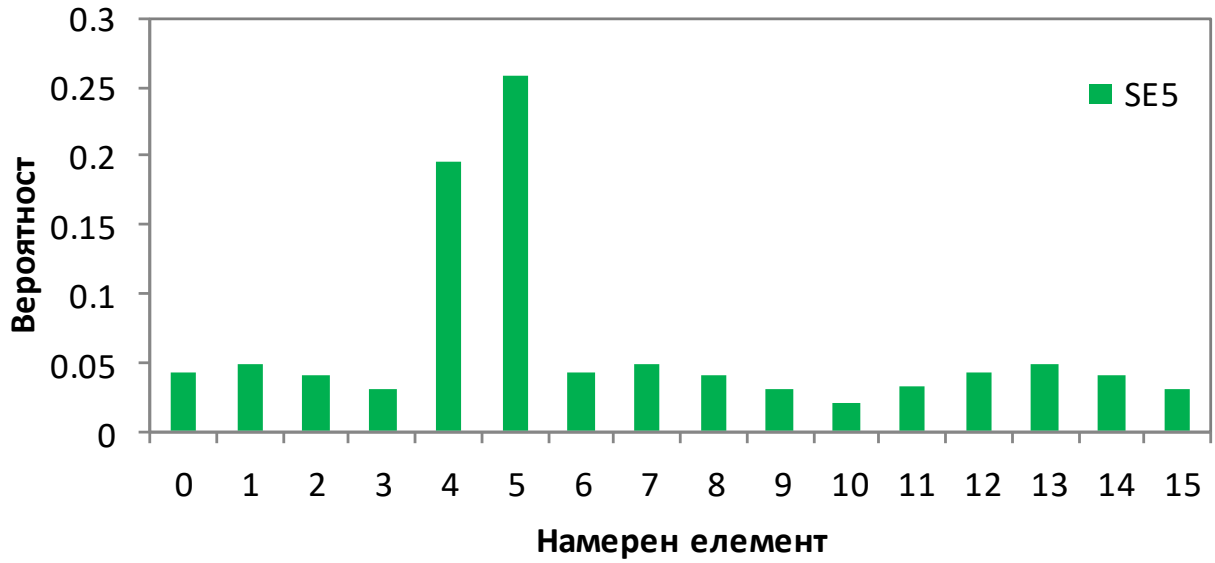
б) прилага се оператора на смесване на състоянията

$$|\vec{x}, \mathrm{i}\rangle \rightarrow \left|\vec{x}^{i}, \mathrm{i}\right\rangle,$$

където x^{i} е състоянието x със сменено състоянието на i-тия бит.

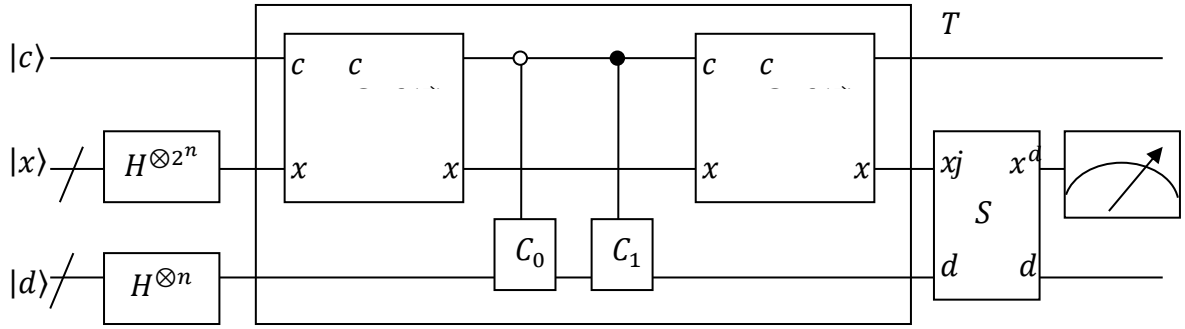
3. Измерва се състоянието на регистъра на върховете на хиперкуба. С известна вероятност измереното състояние е търсеното. Проверява се дали измереното състояние е търсеното. Ако не е, алгоритъмът се повтаря.

Този вариант на квантовия алгоритъм за търсене при произволно преместване при топология на хиперкуб са нарича SKW [26].



Фиг. 4.2: *Пример за квантово търсете с произволно преместване при големина на пространството на търсене 4 кюбита. Броят итерации е 25, а броят на кюбитите в регистъра на монетата е 2. Елементите, които се търсят са |5> .*

Пример за симулация на SKW е показан на Фиг. 4.2. Квантовата мрежа за осъществяването на QRWS върху хиперкуб е показана на Фиг. 4.3:



Фиг. 4.3: *Квантова мрежа на алгоритъма за търсене с произволно преместване върху хирперкуб. Фигурата е заимствана от* [43].

Вероятността да се намери търсеният елемент след h стъпки е $P = 1/2 - O(1/N)$. Вероятността да се намери търсения елемент е периодична в зависимост от броя итерации.

4.5. Алгоритъм за търсене с произволно преместване оптимизиран за хиперкуб

Ако се раздели квантовото произволно преместване върху хиперкуб на две независими квантови произволни придвижвания върху хиперкуб чрез модифициране на оператора на смесване, би могло да се удвои вероятността да се намери търсения елемент [29]. Един метод за това е да се конструира $n' = n + 1$ мерен хиперкуб, като в този нов куб елементите от стария куб да са на четните позиции. Нечетните елементи да са копие на четните и оператора на смесване реализира две независими смесвания едното само на четните, другото само на нечетните елементи.

4.6. Квантов алгоритъм за търсене с произволно преместване върху хиперкуб с асиметрични монети

Ако регистъра се раздели на две подпространства - за четните и нечетните елементи чрез оператора на смесване, тези подпространства могат да еволюират поотделно. Такова разделяне увеличава вероятността да се намери търсеният елемент два пъти. Същият резултат може да се получи и ако вместо модификация на оператора на смесване се използват подходящи монети [44].

4.6.A. Обща форма на монетите

Стандартната монета на Гровер в алгоритъма за търсене с произволно преместване в оригиналния SKW алгоритъм за търсене се представя по следния начин:

$$C_0 = I - 2|s^c\rangle\langle s^c|$$

$$|s^c\rangle = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle,$$

където $|j\rangle$ е j-тия базисен вектор ($|s^c\rangle = |0,\dots,0,1,0,\dots,0\rangle$) и $|s^c\rangle$ е вектора на равномерната суперпозиция. За маркираща монета C_1 е взето произволно Хаусхолдерово отражение с фаза π:

$$C_1 = I - 2|\chi_1\rangle\langle\chi_1|$$

$$|\chi_1\rangle = \sum_{j=0}^{N-1}\frac{a_j}{a}|j\rangle$$

$$a = \sqrt{\sum_{j=0}^{N-1}{a_j}^2}$$

Тук a_j е реално и $a \neq 0$.

Монетата и стъпката на произволно преместване са непроменени, както е при стандартният SKW алгоритъм за търсене:

$$C_0 = C \otimes I_{2^n}$$

$$U = SC$$

Обобщената форма на пертурбищата монета за произволно преместване е:

$$C' = C_0 \otimes I_{2^n} - (C_1 - C_0) \otimes |v\rangle\langle v| = C - 2(|\chi_1\rangle\langle\chi_1| - |\vec{s}^c\rangle\langle\vec{s}^c|) \otimes |\vec{x}_t\rangle\langle\vec{x}_t|,$$

$$U' = SC' = U - 2S(|\chi_1\rangle\langle\chi_1| - |\vec{s}^c\rangle\langle\vec{s}^c|) \otimes |\vec{x}_t\rangle\langle\vec{x}_t|$$

4.6.B. Алгоритъм

Стъпките на този вариант на алгоритъма са същите като тези на SKW алгоритъма за търсене. Разликите са, че е различна маркиращата монета и е добавен кюбит, който не е необходимо да се измерва в края на алгоритъма. Квантовата мрежа на този алгоритъм е почти същата като тази на SKW и мрежите при един вариант на монети са показани на Фиг. 4.4.

4.6.C. Примери за подходящи монети

Дадените тук примери на някои монети, подходящи за търсене с произволно преместване, вероятно не са единствените, които са ефективни, но тези могат да се реализират относително лесно с N-под система.

Нека размерността на хиперкуба да е 2K, вместо N:

$$N = 2^n = 2K = 2^{2k}$$

С y_i ще се означава произволно число и то може да съвпада или не с y_j когато $i \neq j$. Числото x е произволно с модул, по-голям от модула на y_i при всяко i. Броят на всички числа y_i е n-1.

Един възможен случай на монети е, когато $a_r = x,\ a_{i \neq r} = y_i$, където $|y_i| < |x\ |$, i е цяло число и $i \in [0, n-2]$ и r може да бъде 0 или n-1. Тези монети ще се наричат асиметрични и са предназначени за произволно преместване върху хиперкуб. Те имат такава асиметрична форма, която ефективно води до разделяне на пространството на търсене на две части.

В първото подпространство на търсене монетата маркира елемента, посочен от Оракула. Във второто подпространство на търсене маркиращата монета маркира едно състояние, което е съседно в пълното пространство на маркираното от Оракула. Матрицата на маркиращата монета определя кое от състоянията във второто подпространство да се маркира. Разделянето на пространството за търсене на две изисква един допълнителен кюбит (броят на състоянията на регистъра преди разделянето е 2K) за да може да се направи търсене и да има вероятност по-голяма от 80% да се намери решение.

Два примера на такива монети в зависимост от това как се удвоява броя на състоянията чрез добавяне на кюбит са:

1. Първият вид такава монета е, когато $a_r = x$ и $a_{i\neq(n-1)} = y_i$, където $|y_i| < |x\ \ |$, i е цяло число и $i \in [0, n-2]$, и r може да бъде 0 или n-1:

$$a = \sqrt{x^2 + \sum_{i=0}^{K-2} y_i^2}$$

Действието на монетата може да се обясни лесно в случаите, когато $a_n = 1$, $a_{i\neq n} = 0$ и $\alpha = 1$. Монетите C_1, C_0, C_{1SKW} се различават една от друга само по знака на компонентите на матриците им, затова те маркират едни и същи върхове с еднакви амплитуди. Монетата C_0 маркира всички върхове, свързани с този връх със знак плюс. Монетата C_{1SKW} маркира всички върхове, свързани със съответния връх със знак минус. Монетата C_1 маркира всички върхове без последния със знак плюс, а последния със знак минус. Първият хиперкуб с големина K за квантовия алгоритъм на търсене с произволно преместване е получен от маркираното състояние по такъв начин, че да не се включи състоянието маркирано със знак минус от монетата. От друга страна върхът, който е маркиран със знак минус, участва в хиперкуб, в който не се включва върхът, маркиран от Оракула. Това е вторият хиперкуб в пространството на търсене, разделено на две.

Броят стъпки, който е необходим, зависи от точните стойности на x и y_i, Квантовата мрежа, необходима за този вид монети, е показана на Фиг. 4.4.

Симулацията е направена с двукюбитни монети, поради недостатъчна изчислителна мощност, за да може да бъде реализиран алгоритъмът с монети, съдържащи повече кюбити.

В примера с конкретни стойности на параметрите: $a_n = 1$, $a_{i\neq n} = 0$ и $\alpha = 1$. Когато се използва двукюбитна монета и тези стойности на параметрите, вероятността да се намери решение при шестата итерация е 0.678, а девет итерации са необходими за да се достигне максималната вероятност да се намери елемента 0.859.

При двукюбитните монети $a_n = 5/8$, $a_{i\neq n} = 1/8$ алгоритъмът се нуждае от шест стъпки с произволно преместване. Числените симулации показват, че тези монети могат също да се използват и когато има повече от едно маркирано състояние.

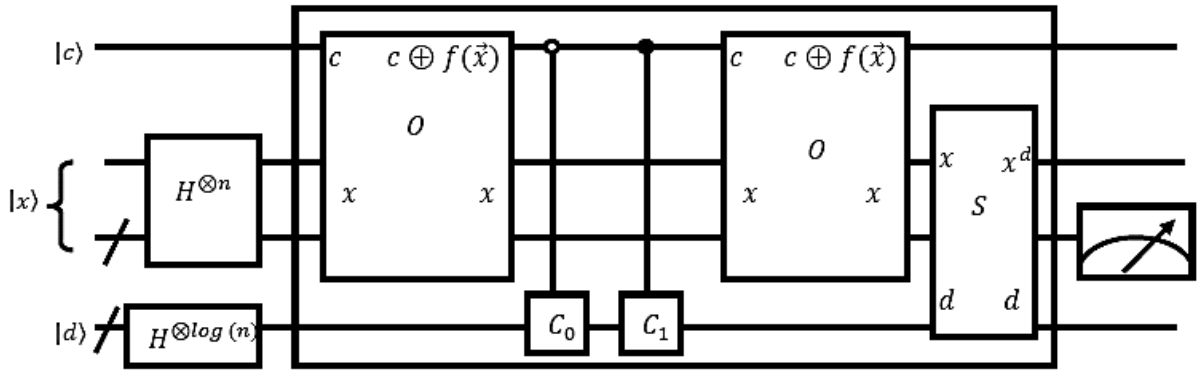
Изчисленията показват, че вероятността да се намери търсения елемент е по-голяма, когато се използват монетите с $a_n = 5/8$, $a_{i\neq n} = 1/8$, отколкото при $a_n = 1$, $a_{i\neq n} = 0$ и $\alpha = 1$.

Друг тип такива монети са $a_0 = x$ $a_{i\neq 0} = y_i$, i е цяло число и $i \in [0, n-2]$, където $|y_i| < |x\ \ |$. Монета от този тип е, когато $a_0 = 1$ и $a_{i\neq 0} = 0$. Квантовата мрежа за този тип монети е показана на Фиг. 4.5. Алгоритъмът се нуждае от шест стъпки на произволно преместване, когато регистъра се състои от два кюбита на монетата, $a_0 =$

$5/8$ и $a_{i\neq 0} = 1/8$. Симулациите демонстрират, че тези монети могат също да се използват, когато има повече от едно маркирано състояние.

Когато $a_0 = 1$ и $a_{i\neq 0} = 0$, алгоритъмът се нуждае от девет итерационни стъпки за да се получи максимална вероятност, равна на 0,859. По-голяма вероятност да се получи търсеният елемент е, когато $a_0 = 5/8$ и $a_{i\neq 1} = 1/8$ по аналогия със случаите, когато $a_n = 5/8$ и $a_{i\neq n} = 1/8$.

Всички останали случаи, имащи структура на монетата, от вида $a_r = x$, $a_{i\neq r} = y_i$, i е цяло число и $i \in [0, r) \vee (r, n-1]$, където $|y_i| < |x\ \ |$, r може да бъде всяко число в интервала $[0, n-1]$. В тези случаи квантовата мрежа за получаване на резултата би била по-сложна или ще се изисква допълнително класическо обработване на информацията.

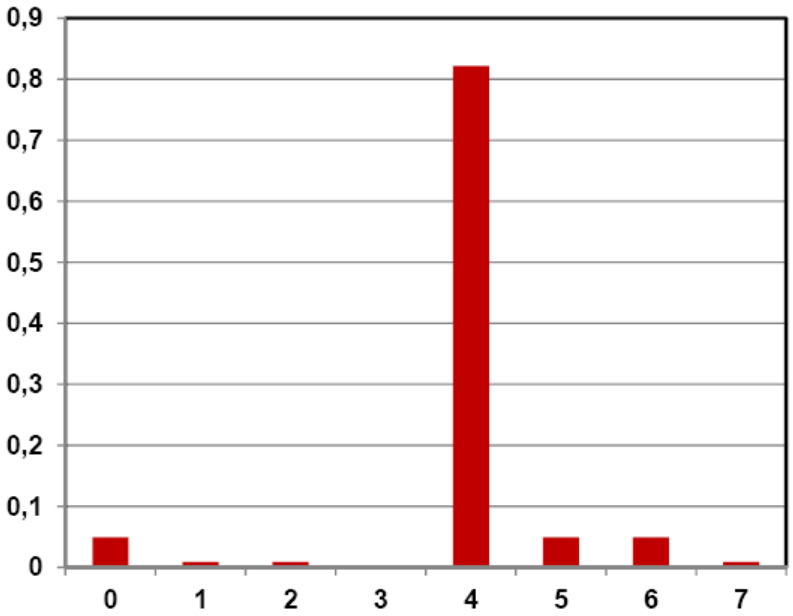


Фиг. 4.4: *Квантова мрежа за търсене с произволно преместване при монетата от вида* $a_K = x,\ \ a_{i\neq K} = y_i$, *където* $|y_i| < |x|$, i *е цяло число и* $i \in [0, n-2]$. *Кутията означена с T е итерацията на търсене с произволно преместване и трябва да се повтаря определен брой пъти. Точният брой зависи от конкретната монета.*

Числените симулации показват, че когато големината на регистъра на върховете е N=16, тогава има и други монети с друга форма на вектора $|\chi\rangle$, които могат да се използват ефективно.

Примери за такива монети са когато:

$\alpha_i = 1/(n+1-i)^w$ $\qquad$ $\alpha_i = (n+1-i)^w$ $\qquad$ $\alpha_i = 1/i^w$ $\qquad$ $\alpha_i = i^w$



Фиг. 4.5: *Резултат от квантово търсене с произволно преместване, когато се използва монета* $a_K = 5/8$ *и* $a_{i\neq K} = 1/8$. *Елементът на търсене е 4 и големината на регистъра на нодовете е 3 кюбита. Постигането на максимална вероятност да се намери търсеният елемент изисква 6 итерации.*

Пример за такава монета е $\alpha_i = i^3$. Използваната квантова мрежа е дадена на Фиг. 4.4 и вероятността да се намери решение е 0.77.

Друг пример е $\alpha_i = (n+1-i)^3$. Вероятността да се намери решение е също 0.77.

Броят стъпки за постигане на максимална вероятност в тези примери е установен числено и все още не е намерено аналитично обяснение.

Предимството на този метод е, че запазва топологията на хиперкуба и не го разделя чрез промяна на оператора на смесване. Монетите могат тривиално да бъдат получени чрез използване на Хаусхолдерови отражения, които е лесно да се реализират експериментално чрез N-pod система.

Източници

[1] Michael A. Nielsen, Isaac L. Chuang, Cambridge University Press (2010), Quantum Computation and Quantum Information

[2] L. K. Grover, Phys. Rev. Lett. 79, 325 (1997), Quantum Mechanics Helps in Searching for a Needle in a Haystack

[3] N. J. Cerf, L. K. Grover, C. P. Williams, Phys. Rev. A 61, 032303 (2000), Nested quantum search and structured problems

[4] I. L. Chuang, N. Gershenfeld, M. Kubinec, Phys. Rev. Lett.80, 3408 (1998), Experimental Implementation of fast quantum search

[5] L. M. K. Vandersypen, M. Steffen, M. Sherwood, C. S. Yannoni, G. Breyta, I. Chuang, Appl. Phys. Lett. 76, 646 (2000), Implementation of three quantum bit search algorithm

[6] P. Walther, K. J. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger, Nature (London) 434, 169 (2005), Experimental one-way quantum computing

[7] Implementation of Grover's quantum Search algorithm in a scalable system (K. A. Brickman, P. C. Haljan, P. J. Lee, M. Acton, L. Deslauriers, C. Monroe) Phys. Rev. A 72, 050306(R) (2005).

[8] J. Ahn, T. C.Weinacht, P. H. Bucksbaum, Science 287, 463 (2000), Information storage and retrieval through quantum phase

[9] N. Bhattacharya, H. B. van Linden van den Heuvell, R. J. C. Spreeuw, Phys. Rev. Lett. 88, 137901 (2002), Implementation of quantum search algorithm using classical fourier optics

[10] Y. Fan, arXiv:0809.0932v4 [quant-ph] (2010), Applications of Multi-Valued Quantum Algorithms

[11] H. Y. Li, C. W. Wu, W. T. Liu, P. X. Chen, C. Z. Li, Phys. Lett. A 375, 4249 (2011), Fast quantum search algorithm for databases of arbitrary size and its implementation in cavity QED systems

[12] Y. Wang and M. Perkowski, Proceedings of the 41st International Symposium on Multiple Valued Logic (2011), Improved complexity of quantum oracles for ternary Grover Algorithm for Graph coloring

[13] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp, Fortsch. Phys. 46, 493 (1998), Tight Bounds on Quantum Searching

[14] Z.Diao, C.Huang, and K.Wang, Springer, Berlin, (2012), Quantum Counting: Algorithm and Error Distribution

[15] J. A. Jones and M. Mosca, Phys. Rev. Lett. 83, 1050 (1999), Approximate Quantum Counting on an NMR Ensemble Quantum Computer

[16] J.-S. Lee, J. Kim, Y. Cheong, and S. Lee, Phys. Rev. A 66, 042316 (2002), Implementation of phase estimation and quantum counting algorithms on an NMR quantum-information processor

[17] A. Y. Kitaev, arXiv:quant- ph/9511026 (1995), Quantum measurements and the Abelian Stabilizer Problem

[18] D. S. Abrams and S. Lloyd, Phys. Rev. Lett. 83, 5162 (1999), Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors

[19] B. C. Travaglione and G. J. Milburn, Phys. Rev. A 63, 032301 (2001), Generation of eigenstates using the phase-estimation algorithm

[20] P. W. Shor, SIAM J. Comput. 26, 1484 (1997), Polynomial time Algorithm for Prime Factorization and Discrete Logarithms on a Quantum Computer

[21] E. Farhi, S. Gutmann, Phys. Rev. A, 58, 915 (1998), Quantum Computation and Decision Trees

[22] A. Childs, E. Farhi, and S. Gutmann) Quantum Information Processing, 1, 35-43 (2001), An Example of the Difference between Quantum and Classical Random Walks

[23] A. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, arXiv:quant-ph/0209131v2 (2002), Exponential Algorithmic Speedup by Quantum Walk

[24] Y. Aharonov, L. Davidovich, N. Zagury, Phys. Rev. A, 48, 1687 (1993), Quantum Random Walks.

[25] A. Ambainis, arXiv:quant-ph/0311001 (2003), Quantum Walk Algorithm for Element Distinctness

[26] N. Shenvi, J. Kempe, K. Whaley, Phys. Rev. A, 67, 052307 (2003) Quantum Random-Walk Search Algorithm

[27] A. Childs, J. Goldstone, Phys. Rev. A, 70, 022314 . (2004), Spatial Search by Quantum Walk

[28] B. Hein, G. Tanner, Journal of Physics A: Mathematical and Theoretical, 42, 085303 (2009), Quantum Search Algorithms on the Hypercube

[29] V. Potocek, A. Gabris, T. Kiss, I. Jex, Phys. Rev. A, 79, 012325 (2009), Optimized Quantum Random-Walk Search Algorithms on the Hypercube

[30] A. Tulsi, Phys. Rev. A, 78, 012310 (2008) Faster Quantum Walk Algorithm for the Two Dimensional Spatial Search

[31] Richard J. Lipton Kenneth W. Regan, The MIT Press (2014), Quantum algorithms via linear algebra - A Primer

[32] Nikolay V. Vitanov, Phys. Rev. A 85, 032331 (2012), Synthesis of Arbitrary SU(3) transformation of atomic qutrits

[33] P. A. Ivanov, E. S. Kyoseva, Nikolay V. Vitanov, Phys. Rev. A 74, 022323 (2006), Engineering of Arbitrary U(N) transformations by quantum Hauseholder reflections

[34] Time-efficient implementation of quantum search with qudits (S. S. Ivanov, H. S. Tonchev, and N. V. Vitanov) Phys. Rev. A 85, 062321 (2012)

[35] Michele Mosca, Theor. Comput. Sci. 264, 139 (2001), Counting by quantum eigenvalue estimation

[36] Hristo Tonchev, Nikolay Vitanov, Phys. A 94, 042307 (2016), Quantum phase estimation and quantum counting with qudits

[37] J. Beringer, J. Arguin, R. Barnett et al., Phys Rev. D 86, 010001 (2012), Review of Particle physics Particle Data Group

[38] Jean-baptiste Hoock, Chang-shing Lee, Arpad Rimmel, Fabien Teytaud, Mei-hui Wang, Olivier Teytaud, IEEE Computional Intelligence Magazine 5, 4 (2010), Intelligent Agents for the Game of Go

[39] Andris Ambainis, Int. J. Quantum Inform. 01, 507 (2003), Quantum walks and their algorithmic applications

[40] Julia Kempe, arXiv:quant-ph/0303081 (2003) , Quantum random walk – an introductory overview

[41] Neil Lovett, Daniel Mosby, Daniel Stockton, and Viv Kendon, Nat Comput 11, 23 (2012), Spatial search using discrete quantum walks

[42] Ashwin Nayak, Ashvin Vishwanath, arXiv:quant-ph/0010117 (2000), Quantum Walk on the Line

[43] Stephan Hoyer, Phd Thesis (2008), Quantum random walk search on satiability problems

[44] H. Tonchev, Journal of Quantum Information Science 5 (2015), Alternative Coins for Quantum Random Walk Search Optimized for Hypercube

5. Приноси по тематиката на дисертацията

Тезата е посветена на разширяване на следните квантови алгоритми:

1. Алгоритъмът на Гровер, когато се използва кютритен запис на информацията.

2. Квантовото търсене с произволно преместване върху хиперкуб, като се използват асиметрични монети.

3. Квантовият алгоритъм за определяне на фазата, ползващ кютрити. Алгоритъмът е използван за да бъде направен квантов брояч за алгоритъма на Гровер с кюдити.

Основните приноси на докторанта, отразени в настоящата дисертация, могат да се обобщят както следва:

Глава 2: Алгоритъм на Гровер с кюдити

Показан е нов начин за адаптиране на алгоритъма на Гровер за кютрити, като са въведени оригинални матрици H_1 и H_2. Новите матрици са аналог на Адамаровата матрица в кюбитния случай, като те са реални, унитарни и симетрични. При използването на някоя от тях за направата на алгоритъма, тя е достатъчна и не се налага да се намира обратната матрица, както е при използването на Фурие трансформацията. За всеки един от горните случаи е направена компютърна симулация на действието на алгоритъма на Гровер с кютрити при различен брой кютрити и различен елемент на търсене. Разгледано е как се променя вероятността за намиране на търсения елемент при пространство на търсене от 3 кютрита в зависимост от броя итерации на алгоритъма на Гровер.

Глава 3: Квантов алгоритъм за определяне на фазата и квантов брояч с кюдити

Направен е алгоритъм за определяне на фазата с кюдити. Използването на кюбити в контролния регистър и кютрити в таргет регистъра дава възможност да се направи контролиран NOT гейт, необходим за алгоритъма за определянето на фазата. Алгоритъмът за определяне на фазата е използван за направата на квантов брояч за алгоритъма на Гровер с кюдити. Направени са числени симулации на алгоритъма на квантовия брояч с кюбити и кютрити при различни големини на регистрите и различен брой елементи на търсене. Симулирани са случаите на аналозите на Адамаровия гейт (F^{-1}, F, H_1 и H_2). Показано е, че вероятността за намиране на броя решения се увеличава при кютритния случай.

Глава 4. Квантов алгоритъм за търсене с произволно преместване, ползващ асиметрични монети, оптимизиран за хирперкуб

Предложен е нов метод за разделяне на хиперкуб на две чрез асиметрични монети, получени от Хаусхолдеров оператор. Разделянето води до удвояване на

вероятността за намиране на търсения елемент. Методът е алтернативен на публикувания наскоро в литературата метод за разделяне на хиперкуба чрез модификация на оператора на смесване. Намерени са четири варианта на различен вид монети, които дават най-добри резултати. Модифицирана е квантовата мрежа на квантовия алгоритъм за търсене с произволно преместване върху хипиркуб за случаите на новите предложени монети. Направени са числени симулации на търсене при новите монети и са намерени числено броя стъпки, необходими при някои конкретни монети.

5. 1. Публикации по тематиката на дисертацията

Резултатите от работата по тематиката на дисертацията са отразени в следните научни публикации:

1. Time-efficient implementation of quantum search with qudits (S. S. Ivanov, H. S. Tonchev, and N. V. Vitanov)) Phys. Rev. A 85, 062321 (2012)
2. Quantum phase estimation and quantum counting with qudits (Hristo Tonchev, Nikolay Vitanov) Phys. Rev. A 94, 042307 (2016)
3) Alternative Coins for Quantum Random Walk Search Optimized for Hypercube (H. Tonchev) Journal of Quantum Information Science 5 (2015)

5.2. Участия в конференции по тематиката на дисертацията

Резултати от работата по тематиката на дисертацията са представени на следните конференции:

1. “Grover’s algorithm with qudits”(Hristo S. Tonchev, S. S. Ivanov, N. V. Vitanov) CAMEL VII - Control of Quantum Dynamics of Atoms Molecules and Ensembles by Light), Nessebar, Bulgaria, 04-08.07.2011, постер
2. “Алгоритъм за търсене с произволно преместване върху хиперкуб, ползващ асиметрични монети”(Христо Тончев) на XVIII Зимен Семинар ИНТЕРДИСЦИПЛИНАРНА ФИЗИКА на младите учени и докторанти от институтите на Научен комплекс 2 на БАН / ИФТТ, ИЯИЯЕ, ИЕ, ЦЛ СЕНЕИ, НИМХ и ИА/, устен доклад
3. “Алгоритъм на Гровер и квантов брояч с кюдити” (Христо Тончев) на XIX Зимен Семинар ИНТЕРДИСЦИПЛИНАРНА ФИЗИКА на младите учени и докторанти от институтите на Научен комплекс 2 на БАН / ИФТТ, ИЯИЯЕ, ИЕ, ЦЛ СЕНЕИ, НИМХ и ИА/, устен доклад

5.3. Забелязани цитати на статиите отразяващи тематиката на дисертацията

В научната литература са забелязани следните цитати на публикациите по тематиката на дисертацията:

1. Time-efficient implementation of quantum search with qudits

Цитати: 6:

1A) Simplified construction and physical realization of n-qubit controlled phase gates (Shi-Biao Zheng) Phys. Rev. A 86, 012326 (2012)

1B) Simplified construction of n-qubit controlled phase gates and physical realization (Shi-Biao Zheng) arXiv:1207.2882 [quant-ph]

2) Shaped Energy-Time Entangled Two-Photon States for Quantum Information (Bänz Bessire) Phd Thesis (2013)

3) Two-state behavior in N-state quantum systems: The Morris–Shore transformation reviewed (Bruce W. Shore) Journal of Modern Optics, 61, 10 (2013)

4) Amplified Quantum Transforms (David Cornwell) Phd Thesis (2014)

5) Phase context decomposition of diagonal unitaries for higher-dimensional systems (Kerstin Beer and Friederike Anna Dziemba) Phys. Rev. A 93, 052333 (2016)