

KVANTUMSZÁMÍTÁSTUDOMÁNY

Készítette: Kaczúr Flórián

Témavezető: Friedl Katalin

1. Bevezető

A kvantumalgoritmusok története a huszadik század második felében indult, az eredeti cél a kvantumfizikai folyamatok szimulálása volt. A terület absztrakt matematikai alapjait a 90-es években kezdték el lefedtetni, ez idő tájt alkottak meg egy - a legjobb klasszikus algoritmusnál gyorsabb - kereső, illetve prímfaktorizáló algoritmust. A félév elején az alapvető definíciókat, algoritmusokat ismertem meg, ezt követően pedig kvantum bolyongást használó módszerekkel foglalkoztam. Főként az [1]-es könyvet használtam, hivatkozásként is többnyire erre támaszkodok.

A kvantum számítási modellekben az alapegység a *qubit*, azaz a $\alpha|0\rangle + \beta|1\rangle$ alakban előálló \mathbb{C}^2 -beli elemek, ahol $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ és $|\alpha|^2 + |\beta|^2 = 1$. A 2-qubites rendszernek 4 bázisvektora van: $|0\rangle \otimes |0\rangle$, $|0\rangle \otimes |1\rangle$, $|1\rangle \otimes |0\rangle$, $|1\rangle \otimes |1\rangle$. Az n -qubites rendszert *regiszter*nek nevezzük, ennek értelemszerűen 2^n bázisvektora van, tehát $\alpha_0|0\rangle + \alpha_1|1\rangle + \dots + \alpha_{2^n-1}|2^n-1\rangle$ alakú, ahol $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$.

A regiszterekkel jellemzően két dolgot lehet csinálni: *megmérhetjük*, ami után a rendszer egy $|j\rangle$ állapotra esik össze $|\alpha_j|^2$ valószínűséggel.

Másrészt *transzformálhatjuk* egy másik qubitbe: matematikailag ez egy unitér (normatartó) mátrixszal való szorzást jelent. A leggyakoribb transzformációk az alábbiak: $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$. Ezt $\phi = \pi/4$ -re T -vel, $\phi = \pi$ -re Z -vel jelöljük. Egy másik gyakran használt unitér mátrix a *Hadamard*:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

1.1. Megjegyzés. $H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, $H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$.

A regiszterek jelölésénél a tenzorszorzást általában elhagyjuk: $|b_1\rangle \otimes |b_2\rangle \otimes \dots \otimes |b_k\rangle = |b_1 \dots b_k\rangle$, ahol $b_i \in \{0, 1\}$.

Ha egy regiszter különböző tagjaira egyszerre több transzformációt alkalmazunk, akkor ezeket összetenzorszorzunk. Például $H|0\rangle \otimes H|0\rangle = H^{\otimes 2}|00\rangle$.

1.2. Állítás. Ha $|i\rangle \in \{0, 1\}^n$ egy állapot, akkor

$$H^{\otimes n}|i\rangle = \frac{1}{\sqrt{2^n}} \sum_{j \in \{0, 1\}^n} (-1)^{i \odot j} |j\rangle, \quad (1)$$

ahol \odot a skalárszorást jelöli.

A leggyakoribb 2-qubites unitér mátrix a *controlled-not*: $CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

Ennek a 3-qubites verziója a CCNOT, vagy az ún. Toffoli: ez helybenhagyja a bázisvektorokat, kivéve ha az első két qubit 1-es, mert akkor negálja a harmadikat.

A kvantumszámítástudományban nem magától értetődő, hogy hogyan adjunk meg egy bemenetet. Ezt ún. *orákulumokkal* tesszük, amik maguk is unitér transzformációk. Ennek a legáltalánosabb alakja az alábbi: $O_{x,b} : |i, b\rangle \rightarrow |i, x_i \oplus b\rangle$, leggyakrabban ezt $b = 0$ -val használjuk: $O_x : |i, 0\rangle \rightarrow |i, x_i\rangle$. Egy másik lehetséges verziója az $O_{x,\pm} : |i\rangle \rightarrow (-1)^{x_i} |i\rangle$.

1.3. Megjegyzés. A kvantumszámítógépek elkészítésénél felmerül a kérdés, hogy mely unitér transzformációkat valósítsuk meg fizikailag (ezeket kapunak nevezik). Ha az összes 1-qubites unitér mátrixot és a CNOT-ot használhatjuk, akkor bármely unitér transzformációt elő tudjuk állítani. Ez természetesen túl sok, ezért kevesebbet szoktak megengedni és ezzel approximálják az előállítandó transzformációt (a metrika mátrixnormában értendő). Belátható, hogy a $\{CNOT, H, T\}$ kapukkal tetszőlegesen jól közelíthető bármely U unitér mátrix. Ugyanez elmondható a $\{H, CCNOT\}$ -ről [3].

A klasszikus programozásban gyakori trükk, hogy egy változót lemásolunk, hogy aztán az értékét megvizsgálhassuk. A következő, ún. **no-cloning tétel** lényegében azt mondja, hogy a kvantumban ez nem működik.

1.4. Tétel. Nem létezik olyan 2-qubites A unitér mátrix, ami $|\phi\rangle |0\rangle$ -hez $|\phi\rangle |\phi\rangle$ -t rendel.

Bizonyítás. Indirekt tegyük fel, hogy van ilyen $A : |\phi\rangle |0\rangle \rightarrow |\phi\rangle |\phi\rangle$ transzformáció. Ekkor nyilván $A |0\rangle |0\rangle = |0\rangle |0\rangle$, illetve $A |1\rangle |0\rangle = |1\rangle |1\rangle$. Nézzük meg, hogy A mit csinál az $\frac{1}{\sqrt{2}}(|0, 0\rangle + |1, 0\rangle)$ -val.

Egyrészt a linearitás miatt ez $\frac{1}{\sqrt{2}}(|0, 0\rangle + |1, 1\rangle)$.

Másrészt $\frac{1}{\sqrt{2}}(|0, 0\rangle + |1, 0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$. Mivel a második qubit 0, ezért A (definíció szerint) belemásolja az első qubitet. Így $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}(|0, 0\rangle + |0, 1\rangle + |1, 0\rangle + |1, 1\rangle)$ -et ad eredményül, ami különbözik a fentitől. \square

2. Algoritmusok

2.1. Kvantum teleportálás

Tegyük fel, hogy Alice-nál van egy $|\phi\rangle = \alpha |0\rangle + \beta |1\rangle$ qubit, ezt akarja Bobhoz eljuttatni. Csak klasszikus csatornát használhatnak, vagyis csak biteket küldhetnek egymásnak. Az együtthatókat természetesen Alice sem ismeri, és ha megmérné $|\phi\rangle$ -t, akkor az összesen valamelyik bázisvektorra. Egy 3-qubites rendszerrel fognak dolgozni, A -nál lesz az első két qubit, B -nél pedig a harmadik. Értelmeszerűen mindketten

csak a saját qubitjeikre tudnak hatni. A cél, hogy végső soron a B -nél lévő qubit megegyezzen a kiindulási, A -nál lévő $|\phi\rangle$ qubittel.

Első lépésként a $|\phi\rangle$ -t tenzorszorozzák az ún. EPR-párral: $|\phi\rangle \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)$. Így egy 3-qubites rendszert kapnak: ebből az első két qubit A -nál, a harmadik pedig B -nél van.

Ezt követően A a saját qubitjeire $CNOT$ -ot alkalmaz: $\frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle)$, majd pedig $H \otimes I$ -t. Így az $\frac{1}{2}(|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)$ állapotot kapják, aminek a harmadik qubitje B -nél van.

Ezt követően A megméri a qubitjeit és az eredményt (vagyis a kapott bázisvektort) megmondja B -nek. Négy kimenet lehetséges, mindegyik esetben megadjuk, hogy B -nek mit kell csinálnia, hogy a $|\phi\rangle$ qubitet megkapja.

(i) Ha a mérés eredménye $|00\rangle$, akkor B -nél $\alpha|0\rangle + \beta|1\rangle$ van, vagyis éppen az a qubit, ami A -nál volt kezdetben.

(ii) Ha $|01\rangle$ -et kapunk, akkor B -nél $\alpha|1\rangle + \beta|0\rangle$ van, ezt $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ -val beszorozva a kezdeti $|\phi\rangle$ qubitet kapja meg.

(iii) $|10\rangle$ esetén B -nél $\alpha|0\rangle - \beta|1\rangle$ van, ezesetben ezt $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ -el kell beszoroznia.

(iv) $|11\rangle$ esetén az $\alpha|1\rangle - \beta|0\rangle$ -ához pedig $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ lesz a megfelelő választás.

2.2. Deutsch-Józsa probléma

Az alábbi két feladatot úgy kell érteni, hogy az x indexei kettes számrendszerben vannak írva.

Probléma: Adott egy $x \in \{0,1\}^N$ valamilyen $N = 2^n$ -re. Feltesszük, hogy x vagy kiegyensúlyozott $\left(\sum_{i=1}^N x_i = N/2\right)$ vagy konstans. El kell dönteni, hogy a két eset közül melyik áll fenn.

Algoritmus: A $|0^n\rangle$ -ből indulunk, erre $H^{\otimes n} O_{x,\pm} H^{\otimes n}$ -ot alkalmazunk. Ekkor a rendszer az

$\frac{1}{2^n} \sum_{i \in \{0,1\}^n} (-1)^{x_i} \left(\sum_{j \in \{0,1\}^n} (-1)^{i \odot j} |j\rangle \right)$ állapotba kerül (az $i \odot j$ skalárszorzásként értendő).

Vegyük észre, hogy a $|0^n\rangle$ együtthatója $\frac{1}{2^n} \sum_{i \in \{0,1\}^n} (-1)^{x_i}$. Ennek az értéke

- 1, ha x azonosan 1,
- -1, ha x azonosan 0,
- 0, ha x kiegyensúlyozott.

Ezért, ha x konstans, akkor a fenti állapotot megmérve 1 valószínűséggel a $|0^n\rangle$ -t kapjuk. Ha x kiegyensúlyozott, akkor 1 valószínűséggel egy ettől különböző bázisvektort kapunk. Tehát a mérés eredménye

pontosan megmondja, hogy a kettő közül melyik eset áll fenn.

2.1. Megjegyzés. Ha olyan klasszikus megoldást keresünk, ami a kvantumhoz hasonlóan csak az egyes biteket tudja lekérdezni, akkor $N/2 + 1$ kérdést kell feltennünk. Ugyanakkor, a fenti kvantumalgoritmus $O(n)$ futási idejű, vagyis exponenciálisan gyorsabb ennél.

2.3. Simon-féle probléma

Probléma: Legyen $N = 2^n$, $x = (x_0, \dots, x_{N-1})$, ahol $x_i \in \{0, 1\}^n$, azaz x -ben összesen $n2^n$ bit van. Egy olyan $s \in \{0, 1\}^n$ -et keresünk, amire $x_i = x_j \iff i = j$ vagy $i = j \oplus s$, ahol \oplus a bitenkénti $\text{mod } 2$ összeadást jelöli.

Algoritmus: A $|0^n 0^n\rangle$ -ből indulunk, először az első n qubitre Hadamardot alkalmazunk, majd a második n -esre az O_x orákulumot. Ekkor a rendszer állapota az alábbi: $\frac{1}{2^n} \sum_{i \in \{0, 1\}^n} |i\rangle |x_i\rangle$. Ezután megmérjük a második n -est, így egy ilyen alakot kapunk: $\frac{1}{\sqrt{2}} (|i\rangle + |i \oplus s\rangle) |x_i\rangle$. Ezt követően az első n -esre Hadamardot alkalmazva kapjuk az alábbi: $\frac{1}{\sqrt{2^{n+1}}} \left(\sum_{j \in \{0, 1\}^n} (-1)^{i \odot j} \cdot (1 + (-1)^{s \odot j}) |j\rangle \right)$, majd itt ismét mérünk. Csak olyan $|j\rangle$ -t kaphatunk, aminek nemnulla az együtthatója, azaz $s \odot j = 0 \text{ mod } 2$.

Az egész algoritmust addig ismételjük meg, amíg egy $n - 1$ egyenletből álló olyan egyenletrendszert nem kapunk, ahol a sorok lineárisan függetlenek. Belátható, hogy elég $O(n)$ ismétlést végezni, ezután pedig csak a lineáris egyenletrendszert kell megoldani.

3. Keresés, bolyongás

3.1. Grover-féle keresőalgoritmus

A számítástudományban nagyon gyakran kell keresést végrehajtanunk, vagyis egy lineárisnál gyorsabb algoritmussal számos program futási idejét felgyorsíthatjuk. A lent bemutatott módszer a **Grover-algoritmus**, ami négyzetes javítást jelent a klasszikus kereséshez képest.

Probléma: Adott egy $x \in \{0, 1\}^N$ vektor, ahol $N = 2^n$. Egy olyan i indexet keresünk, amire $x_i = 1$. Tudjuk, hogy x -nek pontosan t db nemnulla eleme van, itt csak azzal az esettel foglalkozunk, ahol t ismert. Ha x nem 2-hatvány elemű, akkor megtehetjük, hogy kibővítjük megfelelő számú 0-val.

Megvalósítás: A lényeg, hogy az N -dimenziós térben egyfajta semleges állapotból indulunk, ami minden bázisvektorral egyforma szöveget zár be. Ezt kezdjük el forgatni, amíg meg nem közelítjük az i . bázisvektort: ekkor megmérve az állapotot nagy valószínűséggel megkapjuk a keresett indexek egyikét.

Jelölje $|G\rangle = \frac{1}{\sqrt{t}} \sum_{j: x_j=1} |j\rangle$ a "jó állapotokat", $|B\rangle = \frac{1}{\sqrt{N-t}} \sum_{j: x_j=0} |j\rangle$ pedig a "rossz állapotokat". Ezeket

természetesen nem ismerjük, azt azonban tudjuk, hogy ha az $|U\rangle := \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$ és $|B\rangle$ által bezárt szög θ , akkor $|U\rangle = \sin \theta |G\rangle + \cos \theta |B\rangle$ alakban előáll. Itt $\sin \theta = \sqrt{\frac{t}{N}}$.

Formálisabban: A $|0^n\rangle$ a kiinduló állapot, amire $H^{\otimes n}$ -et alkalmazunk: így $|U\rangle$ -t kapjuk, ami az összes bázisvektorral ugyanakkora szöveget zár be. Erre $k = O\left(\sqrt{\frac{N}{t}}\right)$ -szer az alábbi transzformációt végezzük el: először $O_{x,\pm}$ (ez a $|B\rangle$ -re való tükrözés), majd $H^{\otimes n} R H^{\otimes n}$ -t, ahol R a $|0\rangle^\perp$ -re tükrözés. Ez utóbbi transzformáció az $|U\rangle$ -ra való tükrözés.

Figyeljük meg, hogy k lépés után $\sin((2k+1)\theta) |G\rangle + \cos((2k+1)\theta) |B\rangle$ -ben vagyunk. Itt egy mérést hajtunk végre: annak a valószínűsége, hogy $|G\rangle$ -t kapjuk $\sin^2((2k+1)\theta)$. A k megválasztásánál azt szeretnénk, hogy ez közel legyen 1-hez. Ha k éppen $\frac{\pi}{4\theta} - \frac{1}{2}$ lenne, akkor ez a valószínűség pontosan 1. Ez azonban nem feltétlenül egész: ilyenkor a legközelebbi egész értéket kell választanunk. Mindkét esetben $k \in O(\sqrt{N})$.

Amennyiben t -t nem ismerjük, akkor nem tudjuk, hogy milyen k -t válasszunk. Vegyük észre, hogy k függvényében a hiba valószínűsége egy periodikus függvény, vagyis okosan kell az ismétlésszámot megválasztani. A fenti algoritmusnak van olyan módosítása, ami ezesetben is megtalál egy keresett indexet $O\left(\sqrt{\frac{N}{t}}\right)$ ismétlésszámban, ezzel azonban nem foglalkoztam.

3.1. Megjegyzés. *Konkrét keresési feladat esetén a Grover-féle keresés megvalósítása azon múlik, hogy az orákulumot, vagyis a $|B\rangle$ -re való tükrözést meg tudjuk-e konstruálni. Az alábbi egy olyan példa, ahol $O_{x,\pm}$ könnyen előállítható.*

Satisfiability: Legyen Φ egy n -változós konjunktív normálforma, olyan $i \in \{0, 1\}^n$ -et keresünk, amire $\Phi(i) = 1$. Összesen 2^n -féle lehetőségünk van, ezek között keresünk: a fenti algoritmust használva ez $O(\sqrt{2^n})$ időbe telik.

Ennél a problémánál az $O_{x,\pm}$ orákulumot könnyen elő tudjuk állítani: legyen $U_\Phi : |i, 0, 0\rangle \rightarrow |i, \Phi(i), w_i\rangle$, ahol w_i egyfajta plusz munkaterület. Az $|i, 0, 0\rangle$ -ra U_Φ -t alkalmazzuk, a második qubitre Z -t ($Z|0\rangle$ -hoz $|0\rangle$ -t, $|1\rangle$ -hez $-|1\rangle$ -et rendel), majd pedig U_Φ^{-1} -et. Így a $|i, 0, 0\rangle \rightarrow (-1)^{x_i} |i, 0, 0\rangle$ leképezést konstruáltuk meg.

3.2. Kvantum bolyongás

Ebben a részben főként a [2]-es cikk eredményeire, illetve az [1]-es könyv 8.fejezetére támaszkodok. A klasszikus véletlen bolyongásból ismert jelöléseket használom: P az átmenetmátrixot, P^* a megfordított Markov-lánc átmenetmátrixát, π a stacionárius eloszlást jelöli. Ha λ_1 az átmenetmátrix legnagyobb sajátértékét, λ_2 a második legnagyobb abszolútértékű sajátértékét jelöli, akkor legyen $\delta = \lambda_1 - |\lambda_2|$ az ún. spektrális hézag. Néhány eredményre is támaszkodok: egyrészt tudjuk, hogy tetszőleges kezdeti eloszlásból indítva a bolyongást az eloszlásvektor a stacionáriushoz tart. Másrészt reguláris gráfban a stacionárius és az egyenletes eloszlás egybeesik.

Legyen X az elemek halmaza, ebből néhány meg van jelölve: jelölje ezeket M . Ennek egy elemét szeretnénk megtalálni. Feltesszük, hogy a jelölt eleme arányára ismert egy ε alsó korlát. Egy kvantum bolyongásos módszert, az ún. **MNRS-algoritmust** [5] ismertetem. Mivel a P átmenetmátrix nem unitér, ezért elkészítjük ennek a megfelelőjét a $W(P)$ unitér mátrixot. A klasszikus véletlen bolyongással ellentétben itt nem a csúcsokon, hanem az éleken lépkedünk, ezért a $\mathcal{H} = \mathbb{C}^{X \times X}$ teret fogjuk vizsgálni. Legyen $\mathcal{M} = \mathbb{C}^{M \times X}$ a jelölt elemek altere, egy olyan $|x, y\rangle$ élt keresünk, amire $x \in M$. Vezessük be ennek két alterét:

$$A = \text{Span}\{|x\rangle |p_x\rangle : x \in X\}, B = \text{Span}\{|p_y^*\rangle |y\rangle : y \in X\}, \text{ ahol } |p_x\rangle = \sum_{y \in X} \sqrt{p_{xy}} |y\rangle, |p_y^*\rangle = \sqrt{p_{yx}^*} |x\rangle.$$

A fenti alterekre való tükrözést jelölje $ref(A)$, illetve $ref(B)$.

A $W(P) = ref(B) \cdot ref(A)$ unitér operátor felel meg a **kvantum bolyongás** egy lépésének.

Bolyongások esetén futási idő helyett általában költségeket nézünk, ehhez vezessük be a következő fogalmakat:

– **Setup cost** S: a $\sum_{x \in X} \sqrt{\pi_x} |x\rangle |0\rangle$ kezdőállapot felállításának költsége, ahol π a stacionárius eloszlást jelöli.

– **Update cost** U: az alábbi lépések költsége:

$$\text{a) } |x\rangle |0\rangle \rightarrow |x\rangle \sum_{y \in X} \sqrt{p_{xy}} |y\rangle$$

$$\text{b) } |0\rangle |y\rangle \rightarrow \sum_{x \in X} \sqrt{p_{yx}^*} |x\rangle |y\rangle$$

A $W(P)$ kvantum bolyongás egy lépése $4U + 2$ költségből valósítható meg: a $ref(A)$ és a $ref(B)$ is $2U + 1$ költségű.

– **Checking cost** C: $|x\rangle |y\rangle \rightarrow \begin{cases} |x\rangle |y\rangle, & x \notin M \\ -|x\rangle |y\rangle, & x \in M \end{cases}$ megvalósításának költsége.

A kezdőállapot a $|\pi\rangle = \sum_{x \in X} \sqrt{\pi_x} |x\rangle |p_x\rangle$, végső soron pedig egy $|\mu\rangle$ -höz közeli állapotot szeretnénk kapni, ahol is $|\mu\rangle$ az \mathcal{M} -re való lenormált vetítést jelenti, vagyis $|\mu\rangle = \frac{1}{\sqrt{\varepsilon}} \sum_{x \in M} \sqrt{\pi_x} |x\rangle |p_x\rangle$, $\varepsilon = \sum_{x \in M} \pi_x$. Az algoritmusban a $|\pi\rangle$ -re való tükrözés is megjelenik, nem magától értetődő, hogy tudunk ilyet konstruálni.

3.2. Tétel. *Létezik olyan $R(P)$ transzformáció, amire $R(P)|\pi\rangle = |\pi\rangle$ és $R(P)$ a $|\pi\rangle$ -re merőleges elemeket (kis hibával) $|\pi\rangle$ -re tükrözi.*

Ezt felhasználva az algoritmus a kezdőállapot felállítása után $\frac{1}{\sqrt{\varepsilon}}$ -szor az alábbi transzformációkat végzi el:

1. negálja az olyan $|x\rangle |y\rangle$ bázisvektorokat, ahol $x \in M$,

2. alkalmazza $R(P)$ -t.

Végül pedig megméri az első regisztert, és kiadja outputként, ha \mathcal{M} -beli. Belátható, hogy nagyon kis valószínűséggel kapunk rossz eredményt. Az MNRS-algoritmus összköltsége $S + \frac{1}{\sqrt{\varepsilon}} \left(C + \frac{1}{\sqrt{\delta}} \log\left(\frac{1}{\sqrt{\varepsilon}}\right) U \right)$.

Egy további javítással a logaritmikus faktor is eltüntethető, ezt mondja ki az alábbi tétel.

3.3. Tétel. [5] Legyen P egy ergodikus és megfordítható Markov-lánc átmenetmátrixa, ε pedig egy alsó korlát a $\frac{\sum_{x \in M} \pi_x}{\sum_{x \in X} \pi_x}$ arányra. Ekkor létezik olyan kvantumalgoritmus, ami nagy valószínűséggel megtalál egy jelölt elemet $S + \frac{1}{\sqrt{\varepsilon}} \left(C + \frac{1}{\sqrt{\delta}} U \right)$ összköltségen.

3.2.1. Alkalmazások:

1. Grover-keresés: Legyen egy N csúcsú teljes gráfunk, mindegyik csúcs x egy koordinátájának felel meg. Legyen az i . csúcs jelölt, ha $x_i = 1$: egy ilyet szeretnénk megtalálni.

A teljes gráf P átmenetmátrixának a legnagyobb sajátértéke 1, a második legnagyobb pedig $-\frac{1}{N}$, vagyis $\delta = 1 - \frac{1}{N}$. Ezen esetben $S = U = C = O(1)$, valamint $\varepsilon = \frac{t}{N} \geq \frac{1}{N}$, így a fenti formulába behelyettesítve kapjuk, hogy az összköltség $O(\sqrt{N})$.

A következő két példához szükség lesz egy gráfelméleti fogalomra: a $J(n, r)$ Johnson-gráf csúcsai az $[n]$ r -elemű részhalmazai. Két csúcs között pontosan akkor megy él, ha egy cserével megkaphatók egymásból, azaz pontosan két elemben különböznek. Ismert, hogy a $J(n, r)$ Johnson-gráfban a δ spektrális hézag $\Theta\left(\frac{1}{r}\right)$ -ben van.

2. Mátrix-szorzás: Adott A, B, C három $n \times n$ -es mátrix, ha $AB \neq C$, akkor mutassunk egy (i, j) -t, amire $(AB)_{i,j} \neq C_{i,j}$.

Legyen $R \subseteq [n]$ egy indexhalmaz, $r := |R|$, M_R pedig jelölje az M mátrix $r \times n$ méretű részmátrixát, ahol az R -nek megfelelő sorokat vesszük. Analóg módon definiáljuk M^R és M_R^R -et is.

Először elkészítjük a $J(n, r)$ -t: az R részhalmazok lesznek a csúcsok, mindegyiknél egyúttal eltároljuk A^R -t, B_R -t és C_R^R -t. A $J(n, r)$ gráfban akkor lesz egy R csúcs megjelölve, ha $\exists i, j \in R : (AB)_{i,j} \neq C_{i,j}$. Rövid számolás mutatja, hogy $\varepsilon \in \Omega\left(\frac{r^2}{n^2}\right)$.

A kvantum bolyongás költségei ekkor a következők: $S = O(rn)$, $U = O(n)$ és $C = 0$. Az $r = n^{2/3}$ választással az összköltség $O(n^{5/3})$ lesz.

3. Háromszög keresése egy gráfban: Legyen H egy n csúcsú gráf, ebben szeretnénk egy háromszöget megtalálni. Az előzőhöz hasonlóan, itt is elkészítjük $J(n, r)$ gráfot, a bolyongást ezen végezzük. Egy $R \subseteq \{0, 1, \dots, n-1\}$ csúcsot jelöltnek tekintünk, ha tartalmazza a háromszög egyik élét. A Johnson-gráf minden csúcsára eltároljuk, hogy H mely csúcsai alkotják.

A setup cost $S = \binom{r}{2}$, az update cost $U = 2(r-1)$, hiszen úgy lépünk át egy szomszédba, hogy R -ben egy H -beli csúcsot kidobunk (ilyenkor a többi $r-1$ csúccsal való viszonyát töröljük ki) és a helyére beveszünk egy másikat (hasonlóan, megnézzük, hogy közte és a többi csúcs között megy-e él).

A C checking cost kiszámolása kicsit komplikáltabb. Egy R csúcsban vagyunk, azt akarjuk eldönteni, hogy ez tartalmaz-e olyan élet, ami az eredeti H gráfban egy háromszög-él. Ha R -t és egy $u \in V(H)$ csúcsot rögzítünk, akkor az alábbi módon el tudjuk dönteni, hogy van-e olyan $v, w \in R$, hogy u, v, w háromszöget alkot H -ban.

Először is vesszük a $J(r, r^{2/3})$ gráfot, aminek minden csúcsa egy $R' \subseteq R$ részgráfnak felel meg $r' := r^{2/3}$ választással. Ezen esetben δ' nagyságrendileg $r^{-2/3}$. Egy R' -t jelöltnek tekintünk, ha vannak olyan v és w csúcsai, amik u -val háromszöget alkotnak. Ekkor a jelölt csúcsok aránya $\varepsilon' \approx r^{-2/3}$. Szubrutinként az R' csúcsokon való bolyongást hívjuk meg, azaz a $J(r, r^{2/3})$ gráfon bolyongunk. Ennek jelen esetben a következők lesznek a költségei: $S' = O(r^{2/3})$, $U' = O(1)$ (mert itt csak adott $u \in V(H)$ -ra nézzük meg, hogy háromszöget alkotnak-e az élek), valamint $C' = 0$. Tehát rögzített u -ra és R -re $O(r^{2/3})$ költséggel el tudjuk dönteni, hogy R valamely két csúcsa és u háromszöget alkot-e. Az összes $u \in V(H)$ csúcson Grover-kereséssel végig tudunk menni, amiből kapjuk, hogy $C = O(\sqrt{nr^{2/3}})$.

Így az összköltség $O(r^2 + \frac{n}{r}(\sqrt{nr^{2/3}} + r^{3/2}))$. Ez $O(n^{13/10})$, ha r -et $n^{3/5}$ -nek választjuk.

4. Kvantum bonyolultságelmélet

A fenti algoritmusok után felmerül a kérdés, hogy vajon minden probléma felgyorsítható-e. Sajnos nem ez a helyzet. Megmutatom, hogy a legtöbb n -változós logikai függvény kiszámításához exponenciálisan sok kaput kell használni.

Feltesszük, hogy a kapuknak csak egy k elemű halmazát használhatjuk és összesen C darabot. Továbbá azt is feltesszük, hogy mindegyik legfeljebb 3 bemenetű és 3 kimenetű. Hány fajta hálózatot tudunk ebből készíteni?

Minden kapu esetén eldönthetjük, hogy milyen fajta legyen: ez k^C lehetőség. Egy kapunak 3 kimenete lehet, ezeket más kapuk kapják meg bemenetként: erre $(3C)^3$ lehetőségünk van. Mivel összesen C db kapu lehet, így azt kapjuk, hogy $k^C(3C)^{3C} = C^{O(C)}$ db különböző hálózat készíthető.

Egy $f : \{0, 1\}^n \rightarrow \{0, 1\}$ függvényből 2^{2^n} van, egy kvantumhálózat pedig legfeljebb egy ilyen függvényt tud kiszámolni.

Az összes n -változós Boole-függvényt ki szeretnénk számolni, tehát $C^{O(C)} \geq 2^{2^n}$, azaz $C \geq \Omega\left(2^n \frac{1}{n}\right)$. Ez azt jelenti, hogy exponenciálisan sok kapura van szükség.

A klasszikus esethez hasonlóan a kvantumszámításelméletben is vannak bonyolultsági osztályok. A **BPP**-nek a kvantum megfelelője a **BQP**, amibe azok a nyelvek tartoznak, amelyek polinom időben kvantumhálózattal legalább $2/3$ valószínűséggel eldönthetők.

4.1. Megjegyzés. $BPP \subseteq BQP$.

4.2. Tétel. $BQP \subseteq PSPACE$.

Bizonyítás. Legyen L egy **BQP**-beli nyelv, $x \in \Sigma_0^*$. Az L -et felismerő hálózat $T = poly(n)$ időben dönti el, hogy $x \in L$, S qubiten működik, továbbá feltehetjük, hogy kizárólag Hadamard és Toffoli kaput használ. Az általánosság megszorítása nélkül az is feltehető, hogy $S \leq 3T$, hiszen $S > 3T$ esetén lenne olyan qubit, amire egyáltalán nem hatunk (ha az összes kapu Toffoli lenne, az is legfeljebb $3T$ qubitet befolyásolna).

A j -edik lépésben alkalmazott unitér transzformációt jelölje U_j , az $|i_0\rangle = |x\rangle |0^{S-n}\rangle$ pedig legyen a kezdőállapot. A végső $U_T U_{T-1} \dots U_1 |i_0\rangle$ állapotot jelölje ψ_x . Az $|i_T\rangle$ bázisvektor amplitúdója a végső

állapotban $\langle i_T | \psi_x \rangle = \langle i_T | U_T \dots U_1 | i_0 \rangle$. Vegyük észre, hogy az U_j -k közé identitásokat beszúrva az eredmény nem változik.

$$\langle i_T | \psi_x \rangle = \langle i_T | U_T \left(\sum_{i_{T-1} \in \{0,1\}^S} |i_{T-1}\rangle \langle i_{T-1}| \right) U_{T-1} \left(\sum_{i_{T-2} \in \{0,1\}^S} |i_{T-2}\rangle \langle i_{T-2}| \right) U_{T-2} \dots U_2 \left(\sum_{i_1} |i_1\rangle \langle i_1| \right) U_1 |x, 0\rangle = \sum_{i_{T-1}, \dots, i_1} \prod_{j=1}^T \langle i_j | U_j | i_{j-1} \rangle.$$

Az $\langle i_j | U_j | i_{j-1} \rangle$ -ket könnyen ki tudjuk számolni, ezek összeszorozását, illetve összeadását is el tudjuk végezni polinomiális tárban. Feltehető, hogy a kvantum-hálózat válasza (0 vagy 1) az első qubit megméréséből adódik. Ekkor az elfogadási valószínűség $\sum_{i_T: (i_T)_1=1} |\langle i_T | \psi_x \rangle|^2$, ami szintén kiszámolható polinomiális tárban.

□

4.3. Következmény. $P \subseteq BPP \subseteq BQP \subseteq PSPACE$.

Nem ismert, hogy bármely kettő között fennáll-e az egyenlőség.

Hivatkozások

- [1] R. de Wolf, *Quantum computing: Lecture Notes* (<https://arxiv.org/pdf/1907.09415.pdf>)
- [2] M. Santha, *Quantum walk based search algorithms* In Proceedings of 5th TAMC, pages 31–46, 2008. arXiv/0808.0059.
- [3] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [4] A. Ambainis. *Quantum Walk Algorithm for Element Distinctness*. SIAM Journal on Computing, 37:210–239, 2007.
- [5] F. Magniez, A. Nayak, J. Roland, and M. Santha. *Search via quantum walk*. In Proc. of the 39th ACM Symposium on Theory of Computing, pages 575–584, 2007.