

Etsintäongelman kvanttialgoritmi

Jari Tuominiemi

Helsinki 22.11.2004
Vaihtoehtoiset laskentaparadigmat -seminaari
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

1	Johdanto	1
2	Kvanttilaskennan perusteita	1
3	Groverin algoritmi	3
	3.1 Algoritmin vaiheet	3
	3.2 Esimerkki algoritmin käytöstä	6
	3.3 Algoritmin toimivuus käytännössä	7
	3.4 Algoritmin sovelluskohteet	10
4	Yhteenveto	10
	Lähteet	11

1 Johdanto

Etsintäongelma, jossa halutaan tutkia, löytyykö järjestämättömästä joukosta tietyn ehdon täyttävä alkio, on perinteisillä laskentamenetelmillä työläs. Ainoa keino on käydä joukkoa läpi alkio kerrallaan ja testata ehtoa. Ongelman vaativuus on siis luokkaa $O(N)$. Kvanttilaskenta voi kuitenkin tuoda helpotusta tilanteeseen. Lov Grover kehitti vuonna 1996 algoritmin, joka ratkaisee ongelman kvanttietokoneella $O(\sqrt{N})$ askeleella. Tämä työ esittelee Groverin algoritmin perusteita, ominaisuuksia, käyttöä ja rajoituksia. Aluksi esitellään muutamia perusasioita kvanttilaskennasta.

2 Kvanttilaskennan perusteita

Tässä luvussa käsitellään kvanttilaskentaa sen verran, että lukija pystyy ymmärtämään Groverin algoritmin toimintaa. Perusteellinen esitys aiheesta löytyy esimerkiksi Riefelin ja Polakin julkaisusta [RiP00].

Kvanttilaskennassa käytetään kvanttibittejä. Kvanttibitin tilaa voi kuvata yksikkövektoreilla $a|0\rangle + b|1\rangle$, jossa kantavektori $|0\rangle = (1\ 0)^T$ ja $|1\rangle = (0\ 1)^T$ muodostavat etukäteen kiinnitetyn ortonormaalin kannan. Tällaista kantavektorien lineaarikombinaatiota nimitetään yleensä superpositioksi. Kantavektorien kertoimia a ja b kutsutaan amplitudeiksi. Niille pätee yhtälö $|a|^2 + |b|^2 = 1$. Erilaisia kvanttitiloja voi olla ääretön määrä, mutta mitattaessa tila romahtaa joko kantavektoriksi $|0\rangle$ tai $|1\rangle$ vastaavilla todennäköisyyksillä $|a|^2$ ja $|b|^2$. Mittaamalla voidaan siis saada vain kaksi mahdollista tilaa, jotka voidaan samaistaa klassisiksi biteiksi 0 ja 1. Kvanttilaskennan varsinainen voima tulee useiden kvanttibittien yhdistelmästä. Jos n bittiä yhdistetään, saadaan 2^n erilaista kantavektoria. Kvanttibitit yhdistetään tensoritulolla. Esimerkiksi kahden bitin tensoritulo voidaan kirjoittaa $(a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$. Kun lyhennetään $|x\rangle \otimes |y\rangle = |xy\rangle$ (tai $|x, y\rangle$), tulo

voidaan kirjoittaa $ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$. Vektorien tensoritulo saadaan seuraavasti:

$$\text{esimerkiksi } |01\rangle = \begin{pmatrix} 1^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad (1)$$

Nyt on neljä kantavektoria, ja mitattaessa superposition tila romahtaa johonkin niistä todennäköisyydellä, jota kuvaa amplitudin itseisarvon neliö. Esimerkiksi tilaan $|01\rangle$ päädytään todennäköisyydellä $|ad|^2$.

Kvanttitiloja voidaan operoida unitaarilla matriiseilla. Matriisi on unitaarinen, jos sen ja sen konjugaattitranspoosin tulo on yksikkömatriisi. Tärkeä esimerkki on Hadamardin muunnos: Kvanttitila kerrotaan matriisilla

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2)$$

Kun muunnos H tehdään n kvanttibitille erikseen, saadaan Walsh-Hadamardin muunnos $W = H \otimes H \dots \otimes H$, joka on $2^n * 2^n$ -kokoinen matriisi. Matriisien tensoritulo lasketaan samalla tavalla kuin vektorienkin: esimerkiksi matriisien A ja B tensoritulossa $A:n$ jokaisen alkion a tilalle tulee matriisi B kerrottuna a :lla.

3. Groverin algorimi

Lov Grover on kehittänyt kvanttilaskentaan perustuvan algoritmin [Gro96], joka tarjoaa ratkaisun hyvin yksinkertaiseen ongelmaan: On annettu joukko, jossa on N alkioita. Tehtävänä on etsiä joukosta ehdon P täyttävä alkio. Merkitään $P(x)=1$, jos ja vain jos x on etsitty alkio, ja $P(x) = 0$ muuten. Perusidea on nyt, että alkioit koodataan n kvantti-bitin ($2^n \geq N$) kantatiloiksi $|000 \dots 00\rangle$, $|000 \dots 01\rangle \dots |111 \dots 11\rangle$. Superpositiotila romahtaa mitattaessa tietyllä todennäköisyydellä johonkin kantatilaan. Muodostetaan aluksi tila, jossa kaikki amplitudit (ja siis todennäköisyydet) ovat yhtä suuria. Algoritmin seuraavat askeleet vahvistavat etsittyjä alkioita vastaavien kantatilojen amplitudeja. Näitä vaiheita toistetaan tietty lukumäärä, ja lopuksi tila mitataan. Nyt saadaan haluttu alkio (jos sellainen on olemassa) suurella todennäköisyydellä. Katsotaan seuraavaksi yksityiskohtaisesti eri vaiheita

3.1 Algoritmin vaiheet

VAIHE 1

Muodostetaan aluksi n -bittinen kantatila $|0000 \dots 00\rangle$. Sitten tilaa operoidaan Walsh-Hadamardin muunnoksella. Näin saadaan tila, jossa kaikki amplitudit ovat yhtä suuria.

$$W(|000 \dots 00\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{x=2^n-1} |x\rangle \quad (3)$$

Liitetään ylläolevaan superpositioon lisäksi tilassa $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ oleva tulosbitti b .

Tällä valinnalla on tärkeä merkitys vaiheen 2 kannalta.

VAIHE 2

Tässä vaiheessa suoritetaan kyselyoperaatio Q , joka tarkistaa ehdon P toteutuvuuden. Mikäli $P(x)=1$, lisätään tulosbittiin 1 (modulo 2). Katsotaan seuraavaksi, miten Q :lla operointi vaikuttaa kvanttitilaan. Erotellaan alkioit kahdeksi joukoksi: X_1 :een kuuluvat alkioit toteuttavat ehdon $P(x)=1$ ja X_0 :aan kuuluvat eivät. Kirjoitetaan lisäksi b auki.

Saadaan

$$\begin{aligned}
 & Q\left(\frac{1}{\sqrt{2^n}}\left(\sum_{x \in X_0} |x, b\rangle + \sum_{x \in X_1} |x, b\rangle\right)\right) \\
 &= \frac{1}{\sqrt{2^{n+1}}} Q\left(\sum_{x \in X_0} |x, 0\rangle + \sum_{x \in X_1} |x, 0\rangle - \sum_{x \in X_0} |x, 1\rangle - \sum_{x \in X_1} |x, 1\rangle\right) \\
 &= \frac{1}{\sqrt{2^{n+1}}}\left(\sum_{x \in X_0} |x, 0+0\rangle + \sum_{x \in X_1} |x, 0+1\rangle - \sum_{x \in X_0} |x, 1+0\rangle - \sum_{x \in X_1} |x, 1+1\rangle\right) \\
 &= \frac{1}{\sqrt{2^{n+1}}}\left(\sum_{x \in X_0} |x, 0\rangle + \sum_{x \in X_1} |x, 1\rangle - \sum_{x \in X_0} |x, 1\rangle - \sum_{x \in X_1} |x, 0\rangle\right) \\
 &= \frac{1}{\sqrt{2^n}}\left(\sum_{x \in X_0} |x, b\rangle - \sum_{x \in X_1} |x, b\rangle\right) \tag{4}
 \end{aligned}$$

Huomataan, että joukon X_1 alkioit on saatu erottumaan X_0 :n alkioista, koska niiden amplitudien arvot ovat muuttuneet vastakkaismerkkisiksi, mutta itseisarvot ovat kuitenkin pysyneet samana. Tulosbitti b on sen sijaan pysynyt ennallaan.

VAIHE 3

Suoritetaan operaatio, jolla muutetaan ehdon P täyttävien tilojen amplitudin itseisarvoa.

Tämä vaihe on olennainen, koska todennäköisyys löytää oikea alkio kasvaa (aluksi).

Saatua superpositiotilaa operoidaan unitaarisella matriisilla D:

$$D = \begin{pmatrix} \frac{2-N}{N} & \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} & \frac{2}{N} & & \\ \dots & & & & \\ \frac{2}{N} & \frac{2}{N} & & & \frac{2-N}{N} \end{pmatrix} \quad (5)$$

Matriisi D saadaan matriisitulona WRW , jossa W on Walsh-Hadamardin muunnos ja R on kierto-operaation suorittava matriisi. Walsh-Hadamardin muunnos on tosin määritelty vain jos $N = 2^n$. Muissa tapauksissa voi käyttää mitä tahansa unitaarista matriisiä, jolla päästään samaan alkutilaan (3) [BBH96]. Kun matriisilla D kerrotaan superpositiotila, niin amplitudit a_i muuttuvat seuraavasti: $a_i \rightarrow 2A - a_i$. Symboli A merkitsee lausekkeessa amplitudien keskiarvoa. On huomattava, että muutosten jälkeenkin ehto

$$\sum a^2 = 1 \text{ pitää paikkansa.}$$

3.2 Esimerkki algoritmin käytöstä

Olkoon $N=16$. Nyt on siis käytössä kantatilat $|0000\rangle, |0001\rangle \dots |1111\rangle$. Oletetaan aluksi, että on vain yksi ehdon P täyttävä alkio. Merkitään sitä vastaavan tilan amplitudia a :lla ja muita b :llä. Algoritmi toimii seuraavasti:

Vaiheessa 1 Saadaan kaikille amplitudeille sama arvo, joka on $1/4$, koska $16 \cdot (1/4)^2 = 1$.

Vaiheessa 2 etsityn alkion amplitudi muuttuu arvoon $-1/4$. Tällöin amplitudien keskiarvo on muuttunut arvost $1/4$ arvoon $7/32$.

Vaiheessa 3 saadaan uusiksi arvoiksi $a = 2 \cdot (7/32) - (-1/4) = 22/32$ ja $b = 6/32$.

Jos nyt suoritetaan mittaus, löydettäisiin alkio todennäköisyydellä $(22/32)^2 = 0,47$. Kolmannen kierroksen jälkeen todennäköisyys olisi jo $0,96$, mutta neljännen jälkeen enää $0,58$. Seuraavilla kierroksilla se edelleen laskee ja alkaa sitten nousta uudestaan.

Jos ehdon täyttäviä alkioita olisi 4 , niin huomataan mielenkiintoinen asia. Nyt keskiarvo on vaiheen 2 jälkeen $1/8$ ja uudet amplitudit ovat: $a = 1/2$ ja $b=0$. Tämä tarkoittaa, että heti ensimmäisellä mittauskerralla löydetään jokin etsityistä alkioista (ja seuraavilla kierroksilla todennäköisyys alkaa jälleen pienentyä). Sääntö pätee yleisesti: Jos ehdon P täyttäviä alkioita on $1/4 \cdot N$, niin ratkaisu löytyy 1 . iteraatiokierroksen jälkeen [BBH96].

Jos mikään alkio ei täytä ehtoa, ei Groverin algoritmi ei muuta amplitudeja.

3.3 Algoritmin toimivuus käytännössä

Edellisen kappaleen esimerkit antoivat hyvän kuvan Groverin algoritmin hyvistä ja huonoista puolista. Algoritmi löytää ratkaisun nopeasti, mutta on tiedettävä tarkkaan, montako kertaa sitä toistetaan. Tällöin olisi tiedettävä myös oikeiden ratkaisujen määrä. Yleisessä tapauksessa tätä tietoa ei välttämättä ole käytettävissä. Tutkitaan seuraavaksi erikseen tapauksia, joissa ratkaisujen määrä tunnetaan ja niitä, joissa sitä ei tiedetä.

Ratkaisujen lukumäärä on tunnettu

Olkoon k ehdon P täyttävien alkioden lukumäärä, $a(r)$ on niitä vastaavien kantatilojen ja $b(r)$ on muiden kantatilojen amplitudi r :n iteraatiokierroksen jälkeen. Silloin pätee aina ehto (6):

$$k a(r)^2 + (N-k) b(r)^2 = 1 \quad (6)$$

Voidaan osoittaa [Hir00], että amplitudeille pätevät seuraavat rekursiiviset yhtälöt:

$$a(r+1) = \frac{N-2k}{N} a(r) + 2 \frac{N-k}{N} b(r)$$

$$b(r+1) = \frac{N-2k}{N} b(r) - \frac{2k}{N} a(r) \quad (7)$$

Merkitään $k/N = \sin^2 \theta$, jossa $\theta \leq \pi/2$. Nyt voidaan johtaa kaavasta (7) kaavat (8):

$$\begin{aligned} a(r) &= \frac{1}{\sqrt{k}} \sin(2r+1)\theta \\ b(r) &= \frac{1}{\sqrt{N-k}} \cos(2r+1)\theta \end{aligned} \quad (8)$$

Mittauksella löydetään varmasti haluttu alkio, jos $|b(r)| = 0$. Tämä tapahtuu, kun $\cos(2r+1)\theta = \pi/2$ eli $r = -1/2 + \pi/(4\theta)$. Jos $k \ll N$, niin $\sin\theta \approx \theta$. Tällöin saamme seuraavan arvion iteraatiokerroille r :

$$r \approx -\frac{1}{2} + \frac{\pi}{4} \sqrt{\frac{N}{k}} \approx \left\lfloor \frac{\pi}{4} \sqrt{\frac{N}{k}} \right\rfloor \quad (9)$$

Kaava (9) tarkoittaa nyt, että pyöristetään r alaspäin kokonaislukuun, joka on mahdollisimman lähellä ihannetulosta. Mikä sitten on pyöristysvirheen vaikutus? Voidaan osoittaa [BBH96], että todennäköisyys saada väärä alkio r :n iteraatiokierroksen jälkeen on $(N-k)b^2 \leq k/N$. Jos $k \ll N$, niin virhe on hyvin pieni. Toisaalta millä tahansa k :n arvolla voidaan näyttää [Hir00], että onnistumisen todennäköisyys $ka^2 \geq 1/4$. Groverin algoritmi siis toimii mainiosti, kun etsitään muutamaa alkioita. Jos kuitenkin k/N on suuri, ei algoritmiä välttämättä kannata käyttää orjallisesti, vaan yhdistää osaksi laajempaa algoritmiä. Esimerkki tästä nähdään, kun tutkitaan tapausta, jossa ei tunneta k :ta.

Ratkaisujen lukumäärää ei tunneta

Todennäköisyys, että ratkaisu löytyy r :nnellä iteraatiokierroksella, on $\sin^2(2r+1)\theta$.

Voidaan osoittaa [Hir00], että jos valitaan luku $m \geq \sqrt{N}$ ja ratkaisujen määrä k on pienempi kuin $3N/4$, niin pätee keskimääräistä todennäköisyyttä kuvaava kaava (10).

$$\frac{1}{m} \sum_{i=0}^{m-1} \sin^2(2i+1)\theta \geq \frac{1}{4} \quad (10)$$

Kun siis valitaan jokin luku r väliltä $[0, m-1]$, niin keskimäärin onnistumisen todennäköisyys suoritettaessa r kierrosta Groverin algoritmia on suurempi kuin $1/4$. Yhdistetään Groverin algoritmi osaksi seuraavaa algoritmia [Hir00].

1. Valitaan jokin alkio x satunnaisesti ja testataan, päteekö ehto $P(x)=1$. Jos pätee, niin lopetetaan, mutta muuten jatketaan seuraavista vaiheista.
2. Valitaan satunnaisesti jokin luku väliltä $[0, m-1]$
3. Suoritetaan Groverin algoritmin alustusvaihe (vaihe 1) kerran ja muut r kertaa.
4. Mitataan tulos. Jos $P(x)=0$, suoritetaan koko algoritmi uudestaan.

Ylläoleva algoritmi vaatii keskimäärin 4 toistoa kunnes ratkaisu löytyy. Jos $k \geq 3N/4$, niin vaihe 3 ei takaa, että onnistumistodennäköisyys on suurempi kuin $1/4$. Silloin kuitenkin jo vaiheessa 1 päästään vähintään todennäköisyyteen $3/4$.

Tapaus, jossa $k=0$, voidaan hoitaa toistamalla algoritmia riittävän monta kertaa. Jos alkioita ei löydy, voidaan tehdä päätelmä, että sellaista ei ole.

3.4 Algoritmin sovelluskohteet

Mielenkiintoinen käyttökohte Groverin algoritmille voisi olla salausavainten selvittäminen. Esimerkiksi DES-salakirjoitusstandardi perustuu siihen, että selväteksti muutetaan 56-bittisellä avaimella salakirjoitukseksi. Jos ulkopuolinen taho saa selville sekä selvätekstin että salatun tekstin, avain voidaan ratkaista kokeilemalla salakirjoittaa selväteksti erilaisilla avaimilla ja tutkia vastaako lopputulos siepattua salatekstiä. Koska erilaisia avaimia on $2^{56} \approx 3,2 * 10^{16}$ kappaletta, tähän menee hirveästi aikaa. Groverin menetelmässä erilaiset avaimet voitaisiin koodata 56-bittisen kvanttisysteemin kantatiloiksi. Nyt avaimen selvittämiseen vaadittava algoritmin toistomäärä olisi verrannollinen lukuun $2^{28} \approx 2,7 * 10^9$. Se tekisi salausmenetelmästä käytännössä turvattoman. Ongelmana on vain tietysti se, että ei tiedetä voiko kvanttietokonetta rakentaa.

4 Yhteenveto

Tässä työssä on esitelty Groverin ratkaisualgoritmi järjestämättömän joukon etsintäongelmaan. Algoritmi pystyy löytämään kvanttietokoneella etsityn alkion $O(\sqrt{N})$ askeleella. Parhaiten algoritmi toimii, jos ehdon täyttävien alkioden suhteellinen määrä on pieni ja se tunnetaan. Tällöin pystytään määrittämään hyvin tarkasti suoritusaskelten määrä. Jos määrää ei tunneta, voidaan Groverin algoritmi liittää osaksi laajempaa algoritmiä. Käytännössä algoritmiä voisi käyttää esimerkiksi salakirjoitusavainten selvittämiseen.

Lähteet

- BBH96 Boyer, M., Brassard, G., Hoyer, P., Tapp, A., *Tight Bound on Quantum Searching*, Fourth Workshop on Physics and Computation. PhysComp' 96 Ed. T. Toffoli, M. Biaford, J. Lean, New England Complex Systems Institute, 1996, 36-43
- Gro96 Grover, Lov K., *A Fast Quantum Mechanical Algorithm for Database Search* Proceedings of 28th Annual ACM Symposium on Theory of Computing, 1996, 212-219
- Hir00 Hirvensalo, M., *Quantum Computing*. Springer-Verlag, Saksa 2001, 73-90
- RiP00 Rieffel, E., Polak, W., *An Introduction for Quantum Computing for Non-Physicists*. ACM Computing Surveys, 32(3) 2000, 300-335