Piotr Pokora

Politechnika Krakowska

Komputery kwantowe a problemy NP-zupełne.

1. Teoria komputerów kwantowych.

W dzisiejszych czasach ciężko wyobrazić sobie życie bez komputerów. Korzystamy z nich w codziennym życiu, w projektowaniu budynków, w rozwiązywaniu problemów matematycznych, czy fizycznych. Mimo iż technologia w ostatnich latach rozwijała się bardzo dynamicznie ciągle narzekamy, że komputery są za wolne. Może to być spowodowane tym, że chcemy rozwiązywać coraz to trudniejsze problemy, które wymagają coraz to większej mocy obliczeniowej. W ostatnich dwóch latach można zaobserwować jednak stagnacje w rozwoju technologii informatycznych. Nie należy z tego wysnuwać daleko idących wniosków, ale należy sobie zdawać sprawę, że możemy się zbliżać do granic technologicznych w budowie komputerów. Technologia ta wydaje się dość prosta. Naukowcy starają się zbudować z warstw półprzewodników jak najmniejsze bramki logiczne o jak najkrótszym czasie działania. Jednak jest to coraz trudniejsze. Dlatego w ostatnich latach naukowcy zaczęli poszukiwać alternatywy dla standardowych „klasycznych" komputerów. Szczególne nadzieje są związane z komputerami kwantowymi. Komputery kwantowe jak sama nazwa wskazuje są związane ze światem kwantowym, który pozwala na osiągnięcie przewagi kwantowego komputera nad klasycznym PC.

W mechanice kwantowej bardzo ważnym pojęciem jest superpozycja cząstek, w języku matematycznym możemy ją nazwać kombinacją odpowiednich stanów należących do układu kwantomechanicznego. I właśnie superpozycja jest tym zjawiskiem, które prawdopodobnie pozwoli na zwiększenie wydajności komputerów wielokrotnie.

W świecie „klasycznych" komputerów informację możemy zapisać za pomocą ciągu bitów. Jeden bit jest jednostką binarną, która może otrzymać wartość 0 albo 1. W sensie fizycznym jeden bit może być zrealizowany jako obwód elektryczny. Dla przykładu, gdy przez obwód płynie prąd to wtedy możemy obwodowi przypisać wartość 1, natomiast gdy prąd nie płynie wtedy przypisujemy 0. Z tak stworzonego układu łatwo możemy odczytać wartość bitu i nie stanowi to problemu. Stosując uproszczenie komputer możemy sobie wyobrazić jako zbiór N takich obwodów (które możemy utożsamić z N bitami). Dzięki temu wszystkich możliwych stanów mamy $2*2*2*...*2=2^N$. Komputer klasyczny może przetwarzać tylko jeden z tych stanów w danym kroku obliczeniowym.

W kwantowej teorii przetwarzania informacji rolę bitu pełni kubit (quantum bit). O ile klasyczny bit może się znaleźć w stanie „0" albo „1" tak z kubitem sytuacja się komplikuje. Wspomniałem już, że w świecie kwantowym możemy wprowadzić np. elektron w stan superpozycji. Przed superpozycją kubit przyjmuje „0" albo „1", oraz w trakcie pomiaru również przyjmuje „0" albo „1" . Natomiast w trakcie superpozycji kubit jest troszkę zerem a troszkę jedynką, a dokładnie może być dowolną kombinacją stanów odpowiadających „0" lub „1". Naukowcy często mówią, że kubit możemy sobie wyobrazić jako strzałkę, która przyjmuje dowolne położenie na

sferze, (tzw. sferze Blocha). A teraz najważniejsze. Pamiętamy, że komputer klasyczny może przetwarzać tylko jeden stan w danym czasie. Natomiast komputer kwantowy może przetwarzać aż 2^N stanów w tym samym czasie. Jest to gigantyczna różnica wynikająca właśnie z istnienia superpozycji, ponieważ mając N kubitów możemy powiedzieć, że całość znajduje się właśnie w 2^N stanów. Brzmi to bardzo zachęcająco. Czyli celem naukowców będzie zwiększanie ilości kubitów. Jednak nie jest to takie proste. Na podstawie badań wraz ze wzrostem rozmiarów układu kwantowego z jakiś powodów następuje niszczenie superpozycji. Inna hipoteza dotycząca zaniku superpozycji mówi, że wraz ze wzrostem ilości kubitów ich zdolność do przebywania w stanie superpozycji zanika. Również mówi się, że superpozycja może nie zanikać, tylko my tracimy możliwość obserwacji.

Z utratą superpozycji wiąże się słynna interpretacja H. Everetta z 1957 roku. Słynna hipoteza wielu światów tłumaczy co dzieje się podczas pomiaru stanu kwantowego: realizowany jest nie jeden stan lecz wszystkie możliwe, tylko że każdy z nich w innej odnodze rzeczywistości. W taki sposób możemy zrozumieć np. paradoks kota Schroedingera. Zgodnie z tą interpretacją w jednym świecie kot byłby żywy w drugim martwy, co wydaje się dość sensowne. Czy taka interpretacja jest poprawna? W świecie nauki mamy wielu zwolenników tej hipotezy, np. David Deutsch z Oxfordu, który jest niekwestionowanym autorytetem w świecie informatyki kwantowej. Przeciwnicy twierdzą, że gdyby tak było świat musiałby się rozszczepiać w każdym momencie na $10^{\wedge}{}_{10}{}^{90}$ alternatywnych wszechświatów. David Deutsch natomiast w sposób naturalny tłumaczy tą hipotezę. Jego zdaniem alternatywne wszechświaty są nieokiełznane tak jak kiedyś pole elektromagnetyczne. Kiedyś nie zdawaliśmy sobie sprawy z jego istnienia do momentu powstania teorii elektromagnetyzmu oraz licznych urządzeń, np. radia. Jego zdaniem tak samo będzie z teorią wielu alternatywnych światów. Skoro komputer kwantowy jest w stanie przeliczyć wielokrotnie więcej stanów, niż komputer klasyczny w tym samym czasie, to jego zdaniem oznacza to, że część obliczeń musi zostać wykonana w alternatywnych wszechświatach. Tak naprawdę, czy alternatywne wszechświaty istnieją, czy też nie jest sprawą na pewno trudną do udowodnienia. Jednak zgodnie z faktami komputer kwantowy jest w stanie przeprowadzać obliczenia znacznie szybciej niż komputer „klasyczny". Kluczem do wielkiego sukcesu jest tworzenie algorytmów kwantowych oraz umiejętność operowania na dużej liczbie kubitów. W sensie technologicznym stworzenie komputera kwantowego nie jest problemem. Jednak istnieje wiele problemów niezależnych, taki jak dekoherencja, która generuje liczne błędy. Również czysto matematyczne ograniczenia dotyczące możliwości obliczeniowych nadal stosowałyby się do komputerów kwantowych, nawet gdyby problem dekoherencji zostałby rozwiązany. W tym momencie warto zauważyć kolejną cechę komputera kwantowego: jest urządzeniem sekwencyjnym, tak jak i „klasyczny" komputer, tj. komputer przetwarza kubit po kubicie. Komputer klasyczny jest w stanie wykonać takich kroków setki miliardów na sekundę. W przypadku komputerów kwantowych nie jest najlepiej pod tym względem, biorąc pod uwagę to, że trudno zmienić spin szybciej niż kilkaset razy na sekundę, a ponadto superpozycja nie jest trwała. Zatem, aby pokonać komputer „klasyczny" należałoby zwiększyć liczbę kubitów oraz przetwarzać wiele więcej kubitów w każdym kroku. Jednak metoda NMR[1] pozwala tylko kilkunastu kubitom na pozostanie w superpozycji. Ostatnio pojawiła się sugestia, aby zastosować

1 NMR- Magnetyczny Rezonans Jądrowy

łańcuchy chmur elektronów, wytwarzane w ciekłym helu. Łącząc metodę NMR oraz stosując płapki jonowe można będzie utrzymać superpozycje kilkuset kubitów. Zatem pojawia się pierwsze pytanie dotyczące sensu inwestowania w rozwój takich komputerów. Problem jest ogromy, gdyż na razie posiadamy wiele osiągnięć w tej dziedzinie tylko na papierze a dotychczasowe osiągnięcia w sensie istnienia rzeczywistego komputera kwantowego są niewielkie.

2. Pierwsze komputery kwantowe.

W roku 1996 Gershenfeld, Chuang i Kubiniec pierwsi pokazali działający komputer kwantowy. Zbudowali go zaledwie z 2 kubitów, którymi były jądra węgla i wodoru w cząsteczce chloroformu $CHCL_3$. Zatem pierwszy komputer kwantowy był cieczą. Największym problemem było stworzenie układu, który pozwoliłby na programowanie cząsteczki i odczytywanie danych. W tym celu wykorzystano zjawisko magnetycznego rezonansu jądrowego. Probówkę z cieczą umieszczono w zewnętrznym polu magnetycznym (NMR), które ustawiało spin jądrowe, a następnie komputer programowano oddziałując na spiny impulsami radiowymi. Komputer kwantowy działał, gdy spin jądra wodoru był skierowany w górę, a spin jądra węgla w dół. Tak stworzono bramkę typu XOR, czyli alternatywy wykluczającej. Urządzenie to potrafiło odnaleźć wyróżniony element w zbiorze czteroelementowym. Dość marnie jak na komputer, który kosztował prawie milion dolarów. Do wyszukania elementu zastosowano algorytm Grovera. Algorytm pozwala na znaczne przyśpieszenie przeszukiwania bazy danych, jednak jego działanie nie da się wytłumaczyć w sposób klasyczny. Ilość kubitów w komputerach kwantowych nie rośnie w sposób zastraszający. W 2001 dla przykładu stworzono komputer 7 kubitowy, na którym zaimplementowano kwantowy algorytm Shora. Zespół z Almenden Research Center dokonał po raz pierwszy faktoryzacji liczby za pomocą komputera kwantowego, tj. komputer rozłożył liczbę 15 na iloczyn liczb 3 i 5. Wynik pod względem matematycznym wydaje się co najmniej rozczarowująco trywialny, jednak w tym przypadku ważny był sposób osiągnięcia tego wyniku. Zapis sposobu otrzymania tego wyniku zajmuje kilkanaście stron.

Rok 2007 stał się przełomowy. Kanadyjska firma D-Wave System stworzyła adiabatyczny komputer kwantowy (istnieją adiabatyczne i obwodowe komputery kwantowe) na początku 16, później 28 kubitowy. Firma zapewnia, że jest w stanie zwiększyć ilość kubitów w sposób znaczący. Do końca roku ma powstać komputer 1024 kubitowy, co byłoby wielkim osiągnięciem. Procesory D-Wave system jako kubity wykorzystują złącza Josephsona, których budowa jest dość skomplikowana. Ponadto sygnały muszą być przesyłane za pomocą nadprzewodników, a zatem wymagają temperatury w okolicach zera bezwzględnego, dlatego komputer może działać tylko w warunkach laboratoryjnych. Komputer podobno ma być udostępniony przez internet i będzie można na nim robić symulacje metodą Monte Carlo, tworzyć programy w rozszerzonej wersji języka SQL. Ale tak naprawdę jak komputer wygląda i czy na pewno jest komputerem kwantowym tego nie wiedzą nawet naukowcy. Na jego temat posiadamy bardzo niewiele informacji, które wskazują, że w przynajmniej małym zakresie może być to jednak komputer kwantowy.

Jak pokazała firma D-Wave Systems sposobów na budowę komputera kwantowego jest wiele. Można stosować NMR, złącza Josephsona, elektrony ciekłego helu itp. Każdy sposób związany jest i innymi problemami. Jednak istnieje wiele problemów wspólnych, których pokonanie jest konieczne. Zapewne warto stworzyć mechanizm korekty błędów, które mogłyby się pojawić. Aby tego dokonać należy zbadać

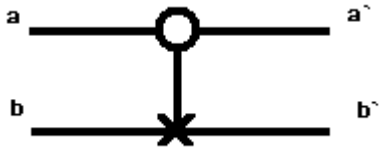
poszczególne kubity. W momencie pomiaru jak pamiętamy superpozycja zanika, a zatem znika wynik. Jest to bardzo istotny problem, którego rozwiązanie pojawiło się całkiem niedawno. Mianowicie naukowcy chcą skorzystać z efektu kwantowego-splątania. Dzięki niemu będziemy mogli tak jakby połączyć ze sobą dwa kubity, przy czym jeden z nich byłby kopią oryginału, oczywiście zawierającą te same informacje. W wyniku pomiaru zniszczona zostałaby superpozycja kopii, a nie oryginału i dzięki splątaniu moglibyśmy otrzymać stan kubita.

3. **Algorytmy kwantowe.**

W przypadku komputerów klasycznych, aby rozwiązać pewien problem musimy skorzystać z odpowiednich algorytmów, które nazywamy algorytmami klasycznymi. W świecie komputerów kwantowych ze względu na fizykę kwantową nie możemy stosować algorytmów klasycznych. Dlatego naukowcy zaczęli tworzyć nowe, odmienne algorytmy, które nazywamy algorytmami kwantowymi. Nazwa tych algorytmów wiąże się z środowiskiem w jakim funkcjonują oraz ich własnościami. Nie wchodząc w szczegóły budowy można powiedzieć, że stworzenie takiego algorytmu jest bardzo trudne. Biorąc pod uwagę, że idea algorytmów kwantowych pojawiła się blisko 30 lat temu do tej pory powstało zaledwie kilkanaście algorytmów.

Każdy algorytm można zapisać za pomocą sekwencji elementarnych bramek logicznych: AND, OR, NOT. Liczba rodzajów bramek nie jest duża, natomiast możemy je ułożyć w dowolnie długą sekwencję. Wiemy, że klasyczne bramki nie są odwracalne, tj. na podstawie danych wychodzących nie jesteśmy w stanie ustalić danych wchodzących do bramki (w tym zdaniu zastosowałem bardzo duże uproszczenie tego procesu, aby pokazać samo sedno sprawy, a nie szczegóły, a jest to dość ważne). Rozważając komputery kwantowe oraz algorytmy kwantowe spotykamy się z nowymi bramkami, które są jako jedyne odwracalne. Są to następujące bramki C-NOT (bramka kontrolowanego zaprzeczenia), bramka Hadamarda. W związku z tym, że wszystkie bramki kwantowe są odwracalne bramki klasyczne nie posiadają kwantowego odpowiednika. Jest to bardzo ważne w budowie algorytmów. Teraz przedstawię budowę bramki C-NOT.

Dane wchodzące: Dane wychodzące:



a	b	a`	b`
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

W praktyce, przy dużych zbiorach danych wejściowych ważne znaczenie ma kwestia szybkości działania algorytmu. Można ją scharakteryzować badając jako funkcję zależności kroków f do zakończenia algorytmu od ilości danych początkowych n, więc jest to funkcja f(n). Gdy funkcja występuje w postaci $n^{k,}$ (gdzie k to liczba naturalna) to mówimy, że algorytm jest wielomianowy. Takiego typu

algorytmy traktujemy jako szybkie, mimo iż mogą się pojawić w tej rodzinie algorytmy wolne jak i szybkie w zależności od k. Natomiast gdy f(n) przyjmuje postać $f=2^n$ to mówimy, że algorytm jest wykładniczy (wolny). Powyższe rozważania dotyczyły algorytmów klasycznych, jednak podobnie można obliczyć szybkość algorytmu kwantowego, Struktura tych algorytmów jest podobna: klasyczne wejście w postaci n kubitów przechodzi przez ciąg operacji, a następnie w wyniku pomiaru kwantowego otrzymujemy klasyczną informację w postaci ciągu bitowego. W przypadku algorytmów kwantowych w związku z większą ilością składowych („klocków"), czyli bramek oraz tego, że algorytm jest odwracalny pojawiło się pytanie, czy algorytmy kwantowe są wolniejsze od klasycznych.

W 1973 roku C. Bennett z laboratorium IBM udowodnił, że algorytm kwantowy złożony z bramek kwantowych wykonywany na kubitach nie jest gorszy od klasycznego, czyli algorytmy kwantowe rozwiązują problemy tak samo skutecznie jak algorytmy klasyczne.

Naukowcy zastanawiali się czy algorytmy kwantowe są szybsze od klasycznych. Okazuje się, że tak może być. Wiemy, że każdy kubit można wprowadzić w superpozycje stanów. Jeżeli mamy n kubitów, i wszystkie n wprowadzimy w stan superpozycji to w takim przypadku nastąpi kwantowy paralelizm. W poszczególnych bramkach mówiąc kolokwialnie będzie trochę jedynki i trochę zera, a zatem będzie alternatywa przetwarzanych danych na każdej z bramek. Dzięki takiej sytuacji obliczenia będą szybsze.

Podstawę algorytmów kwantowych stworzył David Deutsch z Oxfordu. Wraz z R. Jozsą skonstruował algorytm, który rozwiązywał pewien problem w czasie wielomianowym, podczas gdy jego odpowiednik klasyczny rozwiązywał w czasie wykładniczym. Algorytm ten jest dość specyficzny bo polega na odróżnianiu funkcji stałej na dyskretnym zbiorze punktów od funkcji zrównoważonej. Przełom nastąpił w 1994 roku, kiedy to Peter Shor przedstawił kwantowy algorytm służący do faktoryzacji liczb. Jak się okazało jest to bardzo efektywny algorytm, ponieważ liczba jego kroków rośnie jak trzecia potęga liczby cyfr badanego iloczynu.

ALGORYTM SHORA (klasyczny):

Na wejściu mamy liczbę N. Następnie:

1. Wylosuj liczbę a < N
2. Oblicz NWD(a, N) – na przykład za pomocą algorytmu Euklidesa.
3. Jeśli NWD(a, N) ≠ 1, to znaleźliśmy nietrywialny dzielnik N i możemy zakończyć.
4. W przeciwnym wypadku używamy podprocedury znajdującej okres funkcji (poniżej) dla znalezienia r – okresu następującej funkcji:
   $f(x) = a^x \bmod N$,
   (czyli znajdujemy najmniejsze r takie że f(x + r) = f(x).
5. Jeśli r jest nieparzyste, wróć do punktu 1.
6. Jeśli $a^{r/2} \equiv -1 \pmod N$, wróć do punktu 1.
7. Dzielnikiem N jest NWD($a^{r/2} \pm 1$, N)
8. Koniec algorytmu.

ANALIZA ALGORYTMU SHORA

Liczby naturalne mniejsze od *N* i względnie pierwsze z *N* z mnożeniem modulo *N* tworzą pewną grupę skończoną. Każdy element *a* należący do tej grupy ma więc jakiś skończony rząd *r* – najmniejszą liczbę dodatnią taką że:

$$a^r \equiv 1 \bmod N.$$

Zatem $N \mid (a^r - 1)$. Jeśli potrafimy obliczyć r i jest ono parzyste, to:

$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \bmod N$$
$$\Rightarrow N \mid (a^{r/2} - 1)(a^{r/2} + 1).$$

skoro r jest *najmniejszą* liczbą taką że $a^r \equiv 1$, to N nie może dzielić $(a^{r/2} - 1)$. Jeśli N nie dzieli również $(a^{r/2} + 1)$, to N musi mieć nietrywialny wspólny dzielnik z obiema liczbami: $(a^{r/2} - 1)$ i $(a^{r/2} + 1)$.

Otrzymujemy w ten sposób jakąś faktoryzację N. Jeśli N jest iloczynem dwóch liczb pierwszych, jest to jego jedyna faktoryzacja.

Faktoryzacja liczb „dużych", składających się np. z 100 cyfr odgrywa zasadniczą rolę w protokołach szyfrowania informacji. Jeden z protokołów RSA bazuje właśnie na fakcie, że nie jest znany żadny efektywny algorytm, który z zadanego iloczynu dwóch liczb pierwszych potrafiłby szybko odgadnąć oba mnożniki. Dlatego ogłoszenie algorytmu Shora spowodowało zainteresowanie teorią obliczeń kwantowych z wiadomych powodów. Nauczenie się rozkodowania szyfru RSA pozwoliłoby na łamanie szyfrów kont bankowych itp., dlatego też mimo ogromnego znaczenia dla rozwoju obliczeń kwantowych algorytm Shora może być niebezpieczny z punktu łamania szyfrów, a zatem bezpieczeństwa wielu osób. Jednak do łamania takich szyfrów potrzebny jest komputer kwantowy o dużej ilości kubitów, co jak na razie jest nieosiągalne.

4. Problemy NP-zupełne, a komputery kwantowe.

W informatyce bardzo ważne znaczenie odgrywa czas rozwiązywania pewnych problemów. Matematycy czy informatycy tworzą algorytmy, które mimo dużego skomplikowania są rozwiązywanie w dość przyzwoitym czasie. Już wcześniej wspomniałem , że algorytmy wielomianowe są właśnie dość efektywne, tj. gdy czas rośnie w postaci n^k, gdzie k jest liczbą naturalną. Problemy rozwiązywane w sposób efektywny należą do klasy złożoności zwanej P (polynominal time -czas wielomianowy). Jest wiele problemów klasy P, które są dobrze rozwiązywane przez klasyczne, np. mając mapę drogową i podaną drogę sprawdzić czy dane miejscowości są połączone. Problemy te są dość przyjazne również pod względem sprawdzenia. Istnieją również inne problemy, trudniejsze, których rozwiązanie nie jest już takie oczywiste. Czy da się pomalować mapę 3 kolorami w taki sposób, aby by każde sąsiednie kraje miały różny kolor? Czy w końcu da się podróżować między poszczególnymi wyspami, które są połączone mostami w taki sposób, żeby odwiedzić każdą z nich tylko raz?. Istnieje wiele podobnych trudnych problemów, które łączą pewne cechy. Są to problemy, które potrzebują czasu rosnącego wykładniczo. Wymyślono już wiele algorytmów, które nie dość, że potrzebują wykładniczego czasu, to jeszcze liczą trochę po omacku, mało efektywnie. Gdyby jednak udało się stworzyć efektywny algorytm klasyczny, bądź kwantowy, który rozwiązałby jeden z tych problemów to pociągałoby to za sobą rozwiązanie wszystkich pozostałych problemów. Te bardzo ważne stwierdzenie pojawiło się już w latach siedemdziesiątych, kiedy tworzono podstawy teorii problemów NP. NP jest skrótem od „nondeterministic polynominal time", czyli niedeterministyczny czas wielomianowy (algorytm posiada krok, który jest niedeterministyczny). Są to

problemy dość trudne, bo łatwo sobie wyobrazić, że znalezienie trasy łączące np. 100, 200 wysp jest trudne i trwa bardzo długo. Jednak jeśli udałoby się znaleźć taką trasę to łatwo sprawdzić jest poprawność. O problemach właśnie takiej postaci mówimy, że są to problemy NP. Najtrudniejsze z problemów NP noszą nazwę problemów NP-zupełnych. Również znalezienie efektywnego algorytmu dawałoby możliwość rozwiązania innych problemów tej klasy. To ostatnie stwierdzenie pociąga za sobą pewien wniosek. Mianowicie gdyby znaleźć efektywny, tj. wielomianowy algorytm dla problemów NP to wtedy każdy problem NP i NP-zupełny byłby problemem klasy P, czyli klasa NP byłaby równa klasie P, co zapisujemy P=NP. Znalezienie takiego superalgorytmu jest bardzo cenne oraz jak się okazuje bardzo dochodowe, bowiem za znalezienie takiego algorytmu przewidziana jest nagroda miliona dolarów. Wiemy obecnie, że nie został znaleziony klasyczny algorytm, który **efektywnie** rozwiązywałby problemy NP-zupełne. Obecnie istniejące algorytmy, które rozwiązują część takich problemów i są bardzo czasochłonne, bowiem są to algorytmy niewielomianowe (wykładnicze i wolniejsze). Również nie znaleziono żadnego algorytmu kwantowego, które byłby pomocny. Zatem ze zdecydowaniem można stwierdzić, że obecnie tylko algorytmy siłowe rozwiązują wszystkie problemy NP-zupełne. Wiemy, że algorytmy siłowe są algorytmami wolnymi (nie wielomianowymi). Jest to mało satysfakcjonujące, ponieważ człowiek chciałby rozwiązywać coraz to trudniejsze problemy matematyczne. Jednak obecny technika na to nie pozwala. Czy komputery kwantowe pozwoliłyby na rozwiązywanie problemów klasy NP? To trudne pytanie. Po pierwsze wyobraźmy sobie komputer kwantowy i algorytm kwantowy. Już wcześniej pisałem, w jaki sposób taki algorytm działa. Wiemy, że po dokonaniu pomiaru otrzymujemy tylko jeden wynik, jeden z 2^n, w zależności od n. Powoduje to, że pozostałe wyniki znikają, i tak naprawdę nie wiemy, czy ten wynik, który uzyskalibyśmy jest najlepszy. Nie będziemy w stanie przeanalizować poprawności wyniku, co jest niezadowalające. Jednak mechanika kwantowa daję możliwość stworzenia takiego algorytmu, który sam eliminowałby złe odpowiedzi. Ta własność nosi nazwę interferencji destruktywnej. Jest to jednak dość skomplikowane do łatwego wytłumaczenia. Dla nas jednak ważna jest sama możliwość skonstruowania algorytmu, który sam eliminowałby złe odpowiedzi.

W ostatnich latach w związku z małymi i niewystarczającymi osiągnięciami w dziedzinie komputerów kwantowych pojawiło się bardzo wiele nowych pomysłów dotyczących rozwoju informatyki. Po pierwsze komputery klasyczne rozwiązują w sposób zadowalający pewne problemy matematyczne . Należy pamiętać, że w naszym świecie nie da się podzielić czasu na krótsze przedziały czasu niż 10^{-43} s, czyli tzw. czas Plancka. Także nie można uzyskać wyniku w nieskończenie krótkim czasie, co powinno pozostać na uwadze. Również pojawił się pomysł, aby dokonać podróży w czasie. Pamiętając o paradoksie braci Einsteina można wiązać z tą metodą pewne nadzieje. Jednak zbliżenie się w jakikolwiek sposób do prędkości światła jest w obecnych czasach nieosiągalna, ponadto wiąże się to z ogromnymi poborami mocy, ale to jest już inne zagadnienie teoretyczne. Trzeba pamiętać, że zbliżenie się do prędkości światła nie będzie miało tak dużego znaczenia dla informatyki, jak dla astronomii i eksploracji wszechświata. W 1991 roku David Deutsch z Oxfordu wysnuł hipotezę pętli czasowej, w której to moglibyśmy zostawić komputer i uzyskać wyniki szybciej dzięki temu, że czas wewnątrz takiej pętli płynie również szybciej . Przy obecnym stanie wiedzy nie jesteśmy w stanie stwierdzić, czy coś takiego w ogóle istnieje. Jak na razie jest to bardziej hipotetyczne i brzmi

nieprawdopodobnie. Niektórzy naukowcy twierdzą, że warto rozwijać komputery klasyczne. Ja osobiście uważam, że odkrycie grafenu daje nowe możliwości dla rozwoju komputerów osobistych PC. Świetne własności przewodzenia prądu elektrycznego powodują duże zainteresowanie tego materiału. Jednak bardzo wysoka cena produkcji blisko 1000 dolarów za kilka miligramów stanowi dużą barierę.

Gdyby jednak udało się stworzyć komputer kwantowy, to zapewne nie służyłby do łamania kodów, raczej do modelowania komputerowego i do symulacji analitycznych. Gdyby taki komputer powstał, służyłby do badania fluktuacji gospodarczych, analizy ryzyka i opłacalności gier giełdowych, inwestycji.

5. Analiza drzewa Steinera, jako problemu NP-zupełnego.

Na początku chciałbym trochę powiedzieć o samym drzewie Steinera. Załóżmy, że dany jest graf $G=(V,E)$ oraz funkcja wagowa $w:E\rightarrow R$. Podzbiór wierzchołków grafów nazywamy terminalami. Poprzez drzewo Steinera $T_s=(V_s,E_s)$ dla grafu G oraz zbiór terminali X to spójny podgraf grafu G, taki że X zawiera się ściśle w V_s, zwierający się w V oraz, $|E_s|=|V_s|-1$.

Problem GSTP (Graph Steiner Tree Problem) polega na znalezieniu takiego dla danego grafu G i zbioru terminali X drzewa T_s, którego suma wag byłaby minimalna. Wierzchołki $V_s\backslash X$ nazywamy punktami Steinera. Matematycy uznają powyższy problem jako NP-zupełny (NP-trudny). Do rozwiązania tego problemu stworzono wiele algorytmów. Od takich najprostszych (w sensie subiektywnego punktu widzenia) polegających na wypisaniu wszystkich możliwych drzew i wybraniu tego o najmniejszej wadze po bardziej skomplikowane.

Algorytm dokładny Hakimi jest wykładniczy. Mówiąc zwięźle polega na wypisaniu wszystkich możliwych drzew a następnie wybranie tego, którego waga jest najmniejsza. Algorytm działa wystarczająco szybko dla $n<9$ (wierzchołków).

W rozwiązywaniu problemów NP można stosować algorytmy siłowe (Brute Force), jednak wiemy, że jest to algorytm wolny (nie wielomianowy).

Jak się ma do tego komputer kwantowy. Obecnie nie stworzono żadnego algorytmu kwantowego, który rozwiązywałby problem drzewa Steinera. Pojawiły się pewne pomysły rozwiązania tego problemu dla geometrycznego drzewa Steinera w przestrzeniach Euklidesowych, lub z zastosowaniem metryki Manhattan jednak jak na razie nie znam rezultatów tych prac. Wiadomo jest natomiast, że użycie któregokolwiek algorytmu rozwiązującego GSTP okazałoby się wystarczające. Także znalezienie jednego algorytmu kwantowego dla komputera kwantowego daje możliwość rozwiązania możliwie w szybszy sposób problemu GSTP oraz problemów pokrewnych.

6. Bibliografia

- Paweł Hordecki, Karol Życzkowski, *Kwanty, które liczą,* Wiedza i Życie 2/2008, Prószyński Media 2008.
- Scott Aaronson, *NP-complete problems and physical reality*, www.scottaaronson.com/papers/npcomplete.pdf (pobrano 24 lipca 2008).
- Jarosław Chrostowski, *Komputer inny niż wszystkie*, Wiedza i Życie, Prószyński Media 2/2008.
- Algorytm Shora, www.wikipedia.pl (pobrano 24 lipca 2008).

7. Lista tematów związanych z referatem:

- Algorytmy Grovera.
- Zastosowanie fulerenów i grafenu w nanotechnologii.
- Paradoks „Brata Einsteina”.
- Magnetyczny Rezonans Jądrowy.
- Zasada superpozycji.
- Zjawisko tunelowania.
- Atomy rydbergowskie i złącza Josephsona.
- Kod RSA.
- Artur Ekert -pionier kryptografii kwantowej.