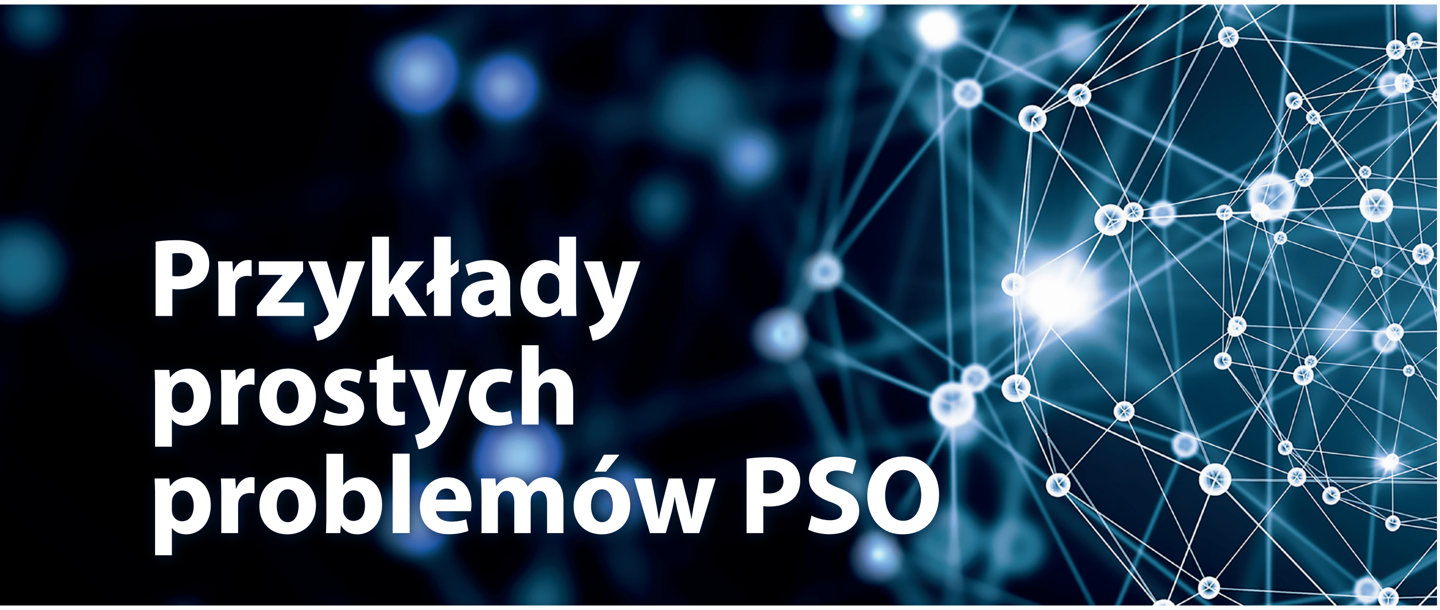

Przykłady prostych problemów PSO

Przewodnik po nauczaniu informatyki kwantowej cz. 4.

Marek Perkowski

Źródło: GetReal-WordPress.com

absolwent Wydziału Elektroniki Politechniki Warszawskiej, tu również zdobył tytuł doktora automatyki. Od 1983 r. pracuje na Wydziale Inżynierii Elektrycznej i Komputerowej w Portland State University, gdzie jest profesorem zwyczajnym i dyrektorem Laboratorium Robotów Inteligentnych. Jeden ze współautorów WARP – pierwszego kompilatora języka VHDL dla układów FPGA. Twórca Diagramów Decyzyjnych Kroneckera, struktury krat logicznych i koncepcji robotów kwantowych. Przyczynił się do powstania oprogramowania dla syntezy logicznej, używanego w przemyśle USA. Pracował jako profesor wizytujący w Holandii, Francji, Japonii, Korei Południowej i Ludowej Republice Chin. W latach 2002–2004 był profesorem zwyczajnym w KAIST – Korean Advanced Institute of Science and Technology, gdzie zajmował się robotyką humanoidalną i komputerami kwantowymi. Kierował Komitetem Logiki Wielowartościowej IEEE w latach 2003–2005 i grupą roboczą Towarzystwa Inteligencji Obliczeniowej IEEE dla Inżynierii Kwantowej w latach 2006–2007. Autor ponad 515 publikacji o automatycznym projektowaniu, syntezie logicznej, logice wielowartościowej, logice odwracalnej, uczeniu maszynowym, robotyce i informatyce kwantowej.

„Przewodnik po nauczaniu informatyki kwantowej" przedstawia metodologię rozwiązywania decyzyjnych Problemów ze Spełnianiem Ograniczeń (PSO) i problemów optymalizacyjnych z wykorzystaniem hybrydowego systemu komputera klasycznego i komputera kwantowego z algorytmem Grovera. Po wprowadzeniu układów odwracalnych jako rozszerzenia układów boolowskich pokazujemy superpozycję i splątanie kwantowe w sposób prosty, ale ścisły. Następnie przedstawiamy podstawowe dla wielu algorytmów kwantowych pojęcie wyroczni. Omawiamy, w jaki sposób wyrocznie są stosowane do rozwiązywania problemów decyzyjnych i optymalizacyjnych. Przykład znalezienia wszystkich „Optymalnych Zbiorów Suportujących" dla funkcji boolowskiej, który znajduje zastosowania w uczeniu maszynowym, dokładnie ilustruje proponowaną metodologię. Na koniec wyjaśniamy, jak działa algorytm Grovera. Po przeczytaniu tego cyklu uważny Czytelnik powinien być w stanie tworzyć podobne systemy kwantowe dla nowych, podobnych do przedstawionych, problemów.

Ponieważ problemy optymalizacyjne są redukowalne do Problemów ze Spełnianiem Ograniczeń (PSO), dlatego skoncentrujmy się na nich. PSO mają wiele zastosowań w syntezie logicznej, logice, kryptografii, robotyce, uczeniu się maszynowym czy sterowaniu.

Kolorowanie grafu

Dany jest graf G ze zbiorem węzłów N i zbiorem krawędzi E, krawędzie są parami węzłów $e_{ij} = (n_i, n_j)$. Węzły należy pokolorować funkcją KOLOR: N $\longrightarrow$ C, gdzie C jest zbiorem K kolorów. KOLOR (n_1) = *czerwony* oznacza pokolorowanie węzła n_1 kolorem *czerwonym*. Dla każdej krawędzi $e_{ij} = (n_i, n_j)$ musi być zachowane ograniczenie KOLOR (n_i) ≠ KOLOR (n_j), co oznacza, że każde dwa sąsiednie węzły muszą mieć różne kolory. Prawidłowe pokolorowanie grafu to takie, w którym to ograniczenie jest spełnione dla każdej krawędzi.

Należy znaleźć rozwiązanie dla następującego problemu decyzyjnego: *Czy jest możliwe pokolorowanie grafu G przy użyciu K kolorów? Jeśli tak, to pokaż to pokolorowanie. Graf jest K-kolorowalny, a liczba chromatyczna tego grafu jest K lub mniej.*

Algorytm PSO konstruktywnie znajduje: rozwiązanie tego problemu, więcej niż jedno rozwiązanie albo udowadnia, że problem ten nie ma rozwiązania. Użytkownik może łatwo zweryfikować, czy problem został dobrze rozwiązany. Choć znalezienie rozwiązania dla dużego problemu jest trudne, weryfikacja jest łatwa. Podkreślmy raz jeszcze, że powyższy problem jest problemem decyzyjnym, a nie optymalizacyjnym. Odpowiedzią na wszystkie problemy PSO jest albo „tak", albo „nie".

Sudoku

Wiele problemów można zredukować do problemu kolorowania grafu. Na przykład problem 4*4 Sudoku reprezentowany jest przez 16 kratek w macierzy 4*4, w której pewne kratki zawierają liczby 1, 2, 3, lub 4, a inne kratki są puste. Rozwiązanie polega na znalezieniu takich liczb ze zbioru {1, 2, 3, 4} dla pustych kratek w macierzy, żeby w każdym wierszu, w każdej kolumnie i w każdej z 4 narożnych 2*2 macierzy wszystkie liczby były różne.

Wyrocznię budujemy następująco. W kolorowaniu grafu dla każdej pary węzłów połączonych krawędzią komparator nierówności powinien dawać wartość 1 na swym wyjściu decyzyjnym, gdy na jego wejściach są różne kolory. Podobnie tworzymy graf układu wyroczni dla 4*4 Sudoku z 16 węzłami. Dla każdych dwóch węzłów z każdej kolumny umieszczamy w wyroczni komparator nierówności. Podobnie: dla każdych dwóch węzłów z każdego wiersza tworzymy komparator nierówności; dla każdego z czterech narożnych kwadratów 2*2 i dla każdej pary kratek w nich tworzymy komparator nierówności. Wyjście wielo-argumentowej bramki iloczynu logicznego I, której wejściami są wyjścia ze wszystkich komparatorów musi być = 1 dla każdego poprawnego rozwiązania problemu 4*4 Sudoku. Podobne wyrocznie łatwo jest utworzyć dla wielu znanych łamigłówek.

SAT

Innym problemem PSO jest słynny problem spełnialności (problem SAT). Dana jest formuła F w pewnej logice (na przykład boolowskiej). Pytamy: *czy ta formuła może być spełniona?* Co znaczy: *czy można znaleźć takie wartości jej zmiennych wejściowych, że F=1?* Na przykład, formuła $F(a,b) = (a+b) \cdot (a'+b) \cdot (a+b') \cdot (a'+b')$ w logice boolowskiej nie jest spełnialna. Formuła ta jest jednak 3-spełnialna, co znaczy, że jeśli usuniemy jeden z czterech termów sumacyjnych z powyższego iloczynu F(a,b), to formuła będzie spełnialna. Na przykład, formuła $F_1(a,b) = (a+b) \cdot (a'+b) \cdot (a+b') = (a+bb') \cdot (a'+b) = a(a'+b) = ab$, zatem formuła jest spełnialna dla wartości wejść a=b=1. Taki problem nazywamy MAXSAT. Problemy SAT i MAXSAT mają setki zastosowań w praktycznych inżynierskich problemach, które są redukowalne do nich. Są to np. problemy optymalizacji układów cyfrowych czy analogowych, layoutu czy organizacji systemów sterowania. Również takie problemy, jak znajdowanie najlepszej drogi ewakuacji ludzi z terytorium katastrofy elektrowni atomowej. Problem spełnialności może być zredukowany do PSO. Podobnie problemy PSO mogą być zredukowane do SAT.

Problemy kryptoarytmetyczne

Jeszcze inny przykład problemu PSO to problem kryptoarytmetyczny: SEND+MORE=MONEY, w którym należy znaleźć podstawienie cyfr 0–9 za litery S,E,N,D,M,O,R,Y w jednoznaczny (jeden-na-jeden) sposób tak, by powyższe symboliczne równanie stało się prawidłowym równaniem na liczbach. Ta prosta łamigłówka jest maksymalnym uproszczeniem podobnych problemów w kryptografii, dziedzinie o ważnych zastosowaniach militarnych i komputerowego bezpieczeństwa. Podobnie do problemów SAT, kolorowania grafu czy innych problemów PSO, problem ten może być rozwiązany przez wyrocznię. Niekwantowe wyrocznie tego typu są budowane z bramek I, LUB, NIE, bloków logicznych takich, jak predykaty (A=B) lub (A>C) oraz bloków arytmetycznych takich, jak sumatory czy układy mnożące. Predykaty w naszym systemie logiczno-arytmetyczno-predykatowym są realizowane jako komparatory lub inne bloki biorące dowolne argumenty, ale zwracające wartości logiczne. Podobnie jak inne bloki, budujemy je z elementarnych bramek logicznych, stosując metody syntezy logicznej, specjalne metody i metody arytmetyki komputerowej.

Wszystko to pozostaje prawdą, gdy konstruujemy wyrocznię kwantową z bramek kwantowych. Metody konstrukcji są inne, ale podstawowa zasada zostaje ta sama. Umiejętność projektowania układów FPGA okazuje się bardzo przydatna dla programisty kwantowego. Inne omówienie problemów z wyroczniami można znaleźć w pracy Bshouty'ego i Jacksona[1].

Konstruowanie wyroczni jako kwantowych układów odwracalnych

Układy kwantowe są naturalnie odwracalne, zarówno poznane już układy permutacyjne, jak i dowolne układy opisane bramkami unitarnymi. Wynika to z własności macierzy unitarnych. Dla bramek „istotnie kwantowych" odwracalność ta istnieje w szerszym sensie – dla każdej macierzy unitarnej istnieje macierz odwrotna. Jest to wykorzystywane do budowania „układów zwierciadlanych", omówionych w następnej części. Istnieją również układy logicznie odwracalne, które nie zapewniają superpozycji i splątania, i nimi się nie zajmujemy. W tym tekście układy odwracalnych są układami kwantowymi.

Konstruowanie kwantowych układów odwracalnych jest podstawą konstrukcji wyroczni dla algorytmu Grovera. Ogólne wyrocznie w naszej metodologii są zależne od problemów i nawet od rozmiarów tych problemów. Algorytmy kwantowe to układy kwantowe budowane z bramek kwantowych, choć nie tylko z binarnych odwracalnych bramek kwantowych. Wyrocznia w algorytmie Grovera jest sercem tego algorytmu i jest zbudowana jedynie z bramek odwracalnych. Projektant nowych algorytmów bazujących na algorytmie Grovera pozostaje więc praktycznie w domenie boolowskiej. Inne bramki i bloki w tym algorytmie są niezależne od problemu, zawsze te same, łatwe do projektowania i dobrze znane. Nie ma potrzeby ich optymalizować. Użytkownik może znaleźć dla nich gotowe rozwiązania w języku QISKIT. Bramki Hadamarda na początku całego układu tworzą przestrzeń rozwiązań, a bramki Hadamarda łącznie z bramkami odwracalnymi służą do implementacji układów dyfuzji, które transformują krok po kroku niemierzalną informację z wyroczni do mierzalnej kwantowo informacji po szeregu powtórzeń pętli Grovera.

Zastosowanie algorytmu Grovera do nowego problemu to po prostu umiejętność sfomułowania tego problemu jako PSO, a następnie skonstruowanie układu kwantowego dla wyroczni i opisanie go w języku kwantowym. Dlatego zgrubny opis funkcjonalności tej ogólnej wyroczni może być wystarczający, a całościowy proces tworzenia programu dla algorytmu Grovera kompilowanego do poziomu realizowalnych bramek z pewnej biblioteki kwantowej – zaadaptowanego do naszego problemu – może być całkowicie zautomatyzowany. Systemów takich jeszcze nie ma, trwają nad nimi prace. Kiedy projektant-programista wie, jak zbudować wyrocznię w logice boolowskiej, to używając standardowych bloków arytmetycznych, może przetransformować swój abstrakcyjny opis do realizowalnych bramek odwracalnych – poprzez konwersję bramek boolowskich do bramek odwracalnych i użycie kwantowych bloków arytmetycznych, np. sumatora kwantowego z biblioteki takich bloków.

Metody translacji z logiki boolowskiej do odwracalnej są znane i w przyszłości zostaną w pełni zautomatyzowane. Przyszłe translator/symulatory języków kwantowych, podobnie jak obecne języki opisu sprzętu, takie jak System Verilog, będą wyposażone w bardzo złożone i inteligentne metody syntezy i optymalizacji na poziomie systemów, procesorów, bloków, układów i layoutu. Z punktu widzenia programisty będzie to jak przejście z poziomu programowania w assemblerze do poziomu programowania w PROLOGU.

Jak widzimy, wyrocznia to po prostu binarny układ kombinacyjny, a więc twórca wyroczni musi myśleć o swym problemie w terminach dekompozycji wszystkich abstrakcyjnych ograniczeń problemu PSO do poziomu znanych binarnych bloków i realizowalnych bramek. Funkcje odwracalne są matematycznymi odwzorowaniami (jeden-do-jednego) jednych binarnych wektorów w inne binarne wektory. Jeśli abstrakcyjna funkcja, którą chcemy zakodować w naszym układzie lub jego podukładzie, nie jest funkcją jeden-do-jednego, co zwykle ma miejsce, to ta funkcja nadal może być zmapowana do bramek odwracalnych, ale należy dodać tak zwane kubity dodatkowe (*ancilla qubits*). Problemem jest, że chcemy dodawać minimalną niezbędną liczbę takich kubitów, co jest związane ze sposobem kodowania naszego problemu do kubitów. Zwróćmy też uwagę, że większość funkcji boolowskich, używanych w typowych blokach arytmetycznych czy komparatorach w wyroczniach, nie jest odwracalna. Projektując bloki, musimy zrealizować je z kwantowych bramek odwracalnych. Oznacza to, że w każdym bloku musimy dodać pewną liczbę dodatkowych kubitów inicjalizowanych do stałych 0, aby móc zrealizować to odwzorowanie. Na przykład boolowski operator, dwuwejściowy iloczyn logiczny I(a, b) jest realizowany jako ($A = a$, $B = b$, $C = ab \oplus c$) przy użyciu bramki Toffoliego z dodatkowym bitem $c = 0$. Zatem to, co nazywamy tutaj bramkami odwracalnymi czy blokami odwracalnymi, nie musi odpowiadać funkcjom odwracalnym w sensie matematycznym, gdyż w praktycznych

1 Bshouty, J., Jackson, J.: *Learning DNF over the uniform distribution using a quantum example Oracle*. Proceedings of the Eighth Annual Workshop on Computational Learning Theory, New York, 1995, s. 118–127.

wyroczniach wiele bloków nie jest matematycznie funkcjami odwracalnymi i posiada jeden lub więcej kubitów dodatkowych. Te bloki odpowiadają funkcjom boolowskim lub kodowanym binarnie dowolnym funkcjom niebędącymi odwzorowaniami jeden-do-jednego. Ten punkt często sprawia kłopot początkującym w językach kwantowych programistom. Jednak synteza takich bloków z dodatkowymi kubitami pozwala projektantowi posiadającemu doświadczenie w konstruowaniu klasycznych układów cyfrowych na natychmiastowe użycie swojej wiedzy i znajomości narzędzi projektowania EDA do budowania zoptymalizowanych i chytrych wyroczni kwantowych. Występują tu problemy znane z klasycznej syntezy układów cyfrowych i automatów, takie jak kodowanie[2] czy synteza w specyficznych bazach logicznych typu AND/EXOR czy – zwłaszcza – ESOP[3].

Należy mocno podkreślić, że idea realizowanej kwantowo wyroczni to znacznie więcej niż tylko wyrocznia Grovera. Można budować wyrocznię dla relacji (funkcji niezupełnie określonych). Można też projektować wyrocznie uogólnione, które poza wyjściem tak/nie i powtórzonymi wartościami zmiennych wejściowych dla rozwiązania, zwracają także inne dane, odpowiadające rozwiązaniom czy ich zbiorom, co może być przydatne do automatycznej konstrukcji następnych wyroczni kwantowych przez klasyczny komputer sterujący. Wyrocznie są więc układami cyfrowymi do rozwiązywania szerokiej klasy problemów odwrotnych.

Wielu autorów tworzy wyrocznie jedynie jako matematyczne koncepty, bez troski o ich praktyczną realizację z praktycznie realizowalnych bramek kwantowych. Zakładają oni, że każda macierz unitarna może być dekomponowana do jedno- i dwukubitowych macierzy unitarnych[4], co jest matematycznie prawdziwe, ale praktycznie bardzo trudne do policzenia. Na dodatek te wynikowe bramki mogą być nierealizowalne w praktycznym sprzęcie kwantowym. Dlatego propagowane tutaj podejście konstrukcji układów wyroczni „od dołu do góry" i to jedynie z bramek znanych i przebadanych jest bardziej praktyczne, choć czasem płacimy znaczną cenę – konieczność zastosowania dodatkowych kubitów.

Wymaganie realizowalności układów wyroczni jest bardzo ważne ze względu na dekoherencję, a także wtedy, gdy chcemy ocenić złożoność układu na podstawie liczby elementarnych bramek, a nie tylko na podstawie liczby ewaluacji układu wyroczni. Jak wiemy, algorytm Grovera daje kwadratowe przyspieszenie nad klasycznym algorytmem pełnego poszukiwania dla tego samego problemu, liczone w liczbie wywołań wyroczni. Ważne jest jednak, by brać pod uwagę nie tylko tę liczbę, lecz także czas i koszty układu poświęcone na jedno wywołanie wyroczni, na co pozwala przedstawiona metodologia.

Co możemy zrobić z algorytmem Grovera?

Fundamentalna idea algorytmu Grovera polega na znalezieniu rozwiązania dla Problemu Odwrotnego. Inne algorytmy kwantowe, np. algorytm faktoryzacji Shora, dają wykładnicze przyspieszenie, ale często mają ograniczone zastosowanie. Natomiast algorytm Grovera pozwala na zredukowanie do niego bardzo wielu praktycznych problemów typu PSO i optymalizacyjnych. Inne ważne algorytmy kwantowe, które mogą być przez nas użyte jako bloki czy podprogramy dla nowych algorytmów wysokiego poziomu, to algorytm estymacji fazy (phase estimation)[5], algorytm zliczania kwantowego (quantum counting)[6], algorytm szybkiej transformacji Fouriera, algorytm symulacji kwantowej i algorytm problemów algebry liniowej HHL (Harrow, Hasidim, Lloyd). Zwróćmy uwagę, że algorytm estymacji fazy używa algorytmu szybkiej transformacji Fouriera. Algorytm zliczania kwantowego wymaga algorytmu Grovera i algorytmu estymacji fazy. Algorytm Shora wymaga algorytmu szybkiej transformacji Fouriera i algorytmu estymacji fazy. Wiele algorytmów kwantowych sieci neurono-

2 Dhawan, S., Perkowski, M.: *Comparison of influence of two data-encoding methods for grover algorithm on quantum costs*. 41st IEEE International Symposium on Multiple-Valued Logic, 2011, s. 176–181; doi: 10.1109/ISMVL.2011.29.
Tsai, E., Perkowski, M.: A quantum algorithm for automata encoding. *Facta Universitatis, Series: Electronics and Energetics*, 2020, 33(2), s. 169–215.

3 Mishchenko, A., Perkowski, M.: Fast heuristic minimization of exclusive sums-of-products. Reed Muller 2001, Workshop.

4 Vartiainen, J.J., Möttönen, M., Salomaa, M.M.: Efficient decomposition of quantum gates. Phys. Rev. Lett. 2004, 92, 177902.
Khan, F., Perkowski, M.: *Synthesis of Ternary Quantum Logic Circuits by Decomposition*. Proceedings of 7th International Symposium on Representations and Methodology of Future Computing Technologies, RM 2005, s. 114–118.

5 Nielsen, M., Chuang, I.: *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

6 Brassard, G., Høyer, P., Tapp, A.: Quantum counting. W: K.G. Larsen, S. Skyum, G. Winskel (red.), *Automata, Languages and Programming* (s. 820–831). Berlin, Heidelberg 1998; doi: 10.1007/BFb0055105.

wej wymaga algorytmu Grovera. W konkluzji – wiele ciekawych nowych algorytmów kwantowych możemy utworzyć za pomocą jedynie algorytmu Grovera. Jeśli jednak dodatkowo zrozumiemy algorytm szybkiej transformacji Fouriera i algorytm estymacji fazy, a także algorytm symulacji kwantowej, to otwierają się przed nami duże możliwości tworzenia algorytmów kwantowych dla nowych problemów.

Wiele użytecznych informacji o kwantowych odwracalnych bramkach i układach, syntezie i optymalizacji takich układów można znaleźć w cytowanych artykułach i książkach. Algorytm Grovera ma wiele wariantów takich jak wzmocnienie amplitudy (amplitude amplification) czy inicjalizacja stanu kwantowego (quantum initialization) do stanów innych niż $|0\rangle^n$ czy $(|0\rangle + |1\rangle)^n$.

Interesująca praca[7] rozszerza idee Grovera na problemy, które mają strukturę. Różne uogólnienia są na przykład używane w kwantowych sieciach neuronowych[8] i sieciach bayesowskich[9]. Przedstawione idee projektowania wyroczni czy układów stosują się też do wielu z tych rozszerzeń.

Algorytmy kwantowe realizowane z wyroczniami są potencjalnie szybsze niż klasyczne układy boolowskie, ponieważ wykorzystują równoległość kwantową wynikającą z superpozycji. Algorytmy te operują na wszystkich wektorach przestrzeni rozwiązań równolegle (tak jakby dla każdego mintermu funkcji boolowskiej w wyroczni istniał osobny klasyczny procesor wyliczający wartość tej funkcji). Minterm to binarny wektor reprezentujący iloczyn wszystkich zmiennych tej funkcji lub negacji zmiennych. Na przykład dla funkcji 4 zmiennych jednym z 16 mintermów jest *a'bc'd* reprezentowany przez 0101.

Podczas gdy klasyczna wyrocznia potrzebowałaby N razy sprawdzać elementy przestrzeni rozwiązań o N elementach jeden po drugim, równoważna wyrocznia w algorytmie Grovera jest wywoływana tylko $\sqrt{(N)}$ razy, co oznacza kwadratowe przyspieszenie. Zatem wyrocznia kwantowa, iterowana wystarczającą liczbę razy jako część pętli Grovera, znacznie zwiększa prawdopodobieństwo znalezienia jednego z rozwiązań problemu w pojedynczym pomiarze. Pomiar ten jest dla wszystkich kubitów zmiennych wejściowych, a czasem mierzone są też dodatkowe kubity.

Dotychczas omawialiśmy wariant algorytmu Grovera z jednym rozwiązaniem. W następnym artykule z cyklu omówimy problemy z wieloma rozwiązaniami. Zauważmy, że $N \leq 2^n$, gdzie n jest liczbą kubitów potrzebnych do zakodowania rozwiązywanego problemu. Jak wspomniano, algorytm Grovera daje kwadratowe przyspieszenie w porównaniu do tego samego problemu rozwiązywanego klasycznie. Zauważmy jednak, że to kwadratowe przyspieszenie odnosi się jedynie do problemów ślepego pełnego przeszukiwania przestrzeni rozwiązań. Algorytm Grovera nie jest zatem panaceum. Kiedy chcemy zastosować ten algorytm do rozwiązania nowego problemu, musimy się najpierw zapytać o złożoność najlepszego znanego obecnie algorytmu klasycznego dla rozwiązania tego problemu.

Konkludując, algorytm Grovera jest stosowalny do większej klasy problemów niż inne algorytmy kwantowe, jest używany przez wiele innych algorytmów kwantowych. Istnieje wiele rozszerzeń i modyfikacji algorytmu Grovera. Zaznajomienie się z tym algorytmem jest idealnym początkiem uczenia się informatyki kwantowej.

---

7 Cerf, N.J., Grover, L.K., Williams, C.P.: Nested quantum search and NP-hard problems. *Applicable Algebra in Engineering, Communication and Computing*, 2000, 10(4/5), s. 311–338.

8 Lagaris, I.E., Likas, A., Fotiadis, D.I.: Artificial neural network methods in quantum mechanics. *Computer Physics Communications*, 1997, vol. 104, s. 1–14.
Beer, K., Bondarenko, D., Farrelly, D., Osborne, T.J., Salzmann, R., Scheiermann, D., Wolf, R.: Training deep quantum neural networks. *Nature Communications*, 2020, 11, 808.
Ezhov, A., Ventura, D.: Quantum Neural Networks. W: N. Kasabov (red.), *Future Directions for Intelligent Systems and Information Science* (s. 213–235). Haidelberg 2000, Physica-Verlag.

9 Tucci, R.: Quantum bayesian nets. *Int. J. Modern Phys,* 1995, vol. B9, s. 295–337.