

Studium Magisterskie

Kierunek: Analiza Danych – Big Data

Przemysław Michałowski

Nr albumu 81840

**Rozwiązywanie problemu marszrutyzacji z wykorzystaniem kwantowego algorytmu przybliżonej optymalizacji**

Praca magisterska

napisana w Instytucie

Ekonometrii

pod kierunkiem naukowym

dra Sebastiana Zająca

Warszawa 2023

Spis treści

[Wstęp 6](#_Toc139805537)

[Rozdział I. Podstawowe pojęcia 8](#_Toc139805538)

[I.1 Podstawowe cechy obliczeń kwantowych 8](#_Toc139805539)

[I.1.1 Kubit jako podstawowa jednostka informacji 8](#_Toc139805540)

[I.1.2 Kluczowe zjawiska 14](#_Toc139805541)

[I.1.3 Wykonywanie działań 15](#_Toc139805542)

[I.2 Postać ogólna problemu marszrutyzacji 19](#_Toc139805543)

[I.2.1 Postać ogólna problemu VRP 19](#_Toc139805544)

[I.2.2 Postać matematyczna VRP 22](#_Toc139805545)

[I.3 Kwantowe i hybrydowe algorytmy optymalizacyjne 27](#_Toc139805546)

[I.3.1 QAOA jako algorytm wariancyjny 27](#_Toc139805547)

[I.3.2 Założenia algorytmu QAOA 28](#_Toc139805548)

[I.3.3 Część kwantowa algorytmu QAOA 30](#_Toc139805549)

[Rozdział II. Ścieżki rozwoju algorytmu QAOA 36](#_Toc139805550)

[II.1 Rozwinięcie algorytmu QAOA 37](#_Toc139805551)

[II.1.1 Złożoność QAOA 37](#_Toc139805552)

[II.1.2 Warm-starting QAOA 39](#_Toc139805553)

[II.1.3 Hiperoptymalizacja Bayesowska 42](#_Toc139805554)

[II.2 Stan badań i aspekt komercyjny 47](#_Toc139805555)

[II.2.1 Działalność podmiotów komercyjnych 48](#_Toc139805556)

[II.2.2 Inwestycje publiczne 51](#_Toc139805557)

[II.2.3 Implementacje 52](#_Toc139805558)

[II.3 Optymalizacja logistyczna z wykorzystaniem algorytmów kwantowych 52](#_Toc139805559)

[Rozdział III. Implementacja 54](#_Toc139805560)

[III.1 Przygotowanie eksperymentu 54](#_Toc139805561)

[III.1.1 Infrastruktura, oprogramowanie i algorytmy 54](#_Toc139805562)

[III.1.2 Dane i sformułowanie problemu 57](#_Toc139805563)

[III.1.3 Postać QUBO problemu 60](#_Toc139805564)

[III.2 Wyniki 62](#_Toc139805565)

[III.2.1 Obliczenia klasyczne - hiperoptymalizacja 62](#_Toc139805566)

[III.2.2 Obliczenia klasyczne – czas obliczeń 65](#_Toc139805567)

[III.2.3 Obliczenia kwantowe 67](#_Toc139805568)

[III.3 Opis matematyczny 67](#_Toc139805569)

[III.3.1 Przykład sformułowania 67](#_Toc139805570)

[III.3.2 Sformułowanie algorytmem kwantowym (obwód?) 70](#_Toc139805571)

[Zakończenie 71](#_Toc139805572)

[LITERATURA 73](#_Toc139805573)

[Spis tabel 76](#_Toc139805574)

[Spis rysunków 76](#_Toc139805575)

[Streszczenie 77](#_Toc139805576)

[Summary 77](#_Toc139805577)

Wstęp

Wraz z rosnącym zapotrzebowaniem na moc obliczeniową, wzrasta potrzeba dostarczania coraz bardziej zaawansowanych maszyn zdolnych do wykonywania tych obliczeń. Do takich maszyn zalicza się komputery i początkowo były one wykorzystywane między innymi w obliczeniach naukowych. Pomimo dostępu do coraz silniejszego sprzętu, dokonywanie pewnych obliczeń przysparzało i nadal przysparza wielu trudności, jak choćby rozwiązywanie problemów NP-trudnych. Szczególną uwagę należy zwrócić na problemy chemików i fizyków kwantowych. Problemy z wykorzystaniem komputerów w obliczeniach dotyczących fizyki kwantowej zostały podniesione głośno przez Richarda Feynmana w 1981 na konferencji *The Physics of Computation*. W czasie swojego przemówienia poruszył on konieczność m.in.  symulowania prędkości światła oraz fal czy reprezentacji rozkładu gęstości dla zjawisk ciągłych, w tym rozkładów prawdopodobieństwa. Są to problemy z którymi ówczesne (klasyczne) komputery nie były sobie w stanie poradzić efektywnie i dokładnie mimo rosnących mocy obliczeniowych. Feynman zaproponował pomysł utworzenia komputera nowego typu – komputera kwantowego[[1]](#footnote-1). Chociaż przed Feynmanem było już kilku naukowców, którzy postulowali i proponowali utworzenie takiej maszyny (jednym z pionierów przed Feynmanem był Paul Benioff, który poświęcił temu zagadnieniu cały artykuł w 1979 roku[[2]](#footnote-2)), to dopiero wykład Feynmana spowodował wzrost zainteresowania maszynami, których działanie ma opierać się na i korzystać z zjawisk na poziomie kwantowym[[3]](#footnote-3).

Komputery kwantowe przyciągają obecnie uwagę różnych specjalistów z całego świata – już nie tylko środowiska naukowego, lecz także polityków, organów publicznych, sektora bezpieczeństwa, sektora komercyjnego i w końcu mediów. Zważywszy jednak na młodość tej technologii i związane z tym problemy, masowa implementacja komercyjna zdaje się być obecnie odległa, ale realna. Pewnym wyróżnieniem dla znaczenia i możliwości informatyki kwantowej jest przyznana w 2022 roku Nagroda Nobla z fizyki dla Alaina Aspecta, Johna F. Clausera i Antona Zeilingera za „eksperymenty ze splątanymi fotonami, potwierdzenie naruszenia nierówności Bella i pionierski wkład w teorię informacji kwantowej” (*quantum information science*)[[4]](#footnote-4).

Obecne lub niedalekie zastosowania komputerów kwantowych obejmują m.in. kryptografię, uczenie maszynowe i optymalizację, symulacje w obszarze fizyki i chemii. Należy jednak zwrócić uwagę, że obecnie są to wszystko przede wszystkim obszary badań, a nie praktycznych zastosowań – do tego są potrzebne silniejsze maszyny kwantowe – dlatego też fragment tej pracy zostanie poświęcony krótkiemu przeglądowi rynku i inwestycji w sektorze informatyki kwantowej – algorytmy bowiem wymagają odpowiedniego sprzętu do ich wykorzystania.

Celem poniższej pracy jest ukazanie komputerów kwantowych jako praktycznego narzędzia w obszarze optymalizacji, uwzględniając wczesne stadium rozwoju technologii kwantowych. Jako przykład zastosowania komputerów kwantowych zostanie przedstawiony problem optymalizacyjny, zwany problemem marszrutyzacji, ponieważ jest to problem znany i często wykorzystywany do sprawdzania możliwości algorytmów optymalizacyjnych. W szczególności zostanie opisany i przetestowany algorytm QAOA, jako kwantowy algorytm optymalizacyjny wykorzystujący bramkowy model obliczeń charakterystyczny dla większości maszyn kwantowych ogólnego zastosowania, tzn. niedostosowanych specjalnie do wykonywania nielicznych i ściśle określonych typów działań. Struktura tej pracy umożliwi czytelnikowi zapoznanie się z podstawowymi pojęciami związanymi z informatyką kwantową i problemem marszrutyzacji w Rozdziale I, dzięki czemu będzie możliwe przeprowadzenie logicznego i spójnego wywodu w dalszych rozdziałach bez popadania w dygresje. W Rozdziale II przedstawiono badaną metodę optymalizacji – algorytm QAOA, wraz propozycją hiperoptymalizacji jego parametrów z wykorzystaniem metody Bayesowskiej. Rozdział III stanowi przedstawienie procesu implementacji algorytmu dla przykładowych instancji problemu marszrutyzacji, wraz z omówieniem wyników.

# Podstawowe pojęcia

Poniższy rozdział stanowi wprowadzenie do podstawowych pojęć związanych z informatyką kwantową i problemem marszrutyzacji. Przedstawiono w nim nazewnictwo oraz oznaczenia wykorzystywane w pracy, a także zdefiniowano problem marszrutyzacji (ang. Vehicle Routing Problem – VRP).

Na początku omówiono pojęcie kubitu wprowadzając równocześnie notację matematyczną używaną w dalszej części pracy. Przedstawiono operacje matematyczne, jakie można wykonywać na kubitach bazując na przyjętych w omawianym obszarze oznaczeniach. W następnym podrozdziale wyjaśniono najważniejsze zjawiska mechaniki kwantowej, mające kluczowe znaczenie dla obliczeń kwantowych – zarówno jako źródła przewagi obliczeniowej nad klasycznymi maszynami obliczeniowymi jak i jako źródła ograniczeń i barier w obliczeniach kwantowych. Kolejna część to zapoznanie z pojęciem problemu marszrutyzacji w jego ogólnej formie, ze szczególnym uwzględnieniem warunków ograniczających. Podstawowe cechy obliczeń kwantowych

### Kubit jako podstawowa jednostka informacji

W klasycznym komputerze podstawową i najmniejszą jednostką informacji jest bit. W danym momencie może on znajdować się w dokładnie jednym z dwóch stanów: 0 lub 1. Oznacza to, że przestrzeń możliwych stanów bitu jest zdefiniowana poprzez dwa różne punkty. Skutkuje to tym, że liczba możliwych stanów systemu *n­*-bitowego wynosi [[5]](#footnote-5) Warto zauważyć, że liczebność zbioru możliwych stanów systemu jest równa liczebności przeliczalnego zbioru potęgowego zbioru możliwych stanów pojedynczego bitu, więc liczebność zbioru możliwych stanów systemu jest równa , gdzie przy pomocy uporządkowanego ciągu *n* bitów można otrzymać izomorficzne przekształcenie elementów zbioru nieuporządkowanego o liczebności .

W komputerze kwantowym podstawową jednostką informacji jest bit kwantowy zwany też kubitem[[6]](#footnote-6). Fizycznie, może być on realizowany na wiele sposobów, w zależności od architektury maszyny kwantowej, np. poprzez fotony, uwięzione jony, elektrony lub inne cząsteczki kwantowe[[7]](#footnote-7). Przestrzeń możliwych stanów systemu kwantowego jest opisywana matematycznie przez zespoloną przestrzeń wektorową z zdefiniowanym na niej iloczynem skalarnym[[8]](#footnote-8). Tak więc matematycznie kubit jest reprezentowany przez unormowany wektor w dwuwymiarowej przestrzeni Hilberta nad ciałem liczb zespolonych[[9]](#footnote-9), a system *n‑*kubitowy – przez unormowany wektor w -wymiarowej przestrzeni Hilberta.

Na mocy twierdzenia spektralnego każdy ograniczony samosprzężony operator określony na przestrzeni Hilberta może być opisywany przez kombinację liniową wektorów bazowych. Oznacza to, że każdy wektor stanu na tej przestrzeni może być przedstawiony w postaci kombinacji liniowej wektorów bazowych (będących równocześnie wektorami własnymi przyjętego operatora). W przypadku, gdy wektory własne operatora są ortogonalne, twierdzenie spektralne pozwala na dokładne wyrażenie każdego wektora stanu w postaci kombinacji liniowej tych wektorów własnych, czyli w postaci kombinacji liniowej stanów bazowych. Ograniczonymi operatorami samosprzężonymi, których wektory są ortogonalne (a nawet ortonormalne) są m.in. operatory Pauliego (opisane w dalszej części tego rozdziału). Jednym z nich jest operator Pauli-Z, który ma wektory własne postaci oraz oznaczane odpowiednio i .

Stany bazowe są to dwa dowolne ortonormalne wektory[[10]](#footnote-10). W dalszej części pracy za stany bazowe będą przyjmowane zawsze stany oraz reprezentowane odpowiednio przez pionowe wektory oraz [[11]](#footnote-11) tzn. będą one tworzyć obliczeniowy stan bazowy (*computational basis state*), pozwalając na możliwość opisania przy ich pomocy dowolnego wektora reprezentującego stan systemu. Zostały one wybrane także ze względu na ich prostotę, która znacznie ułatwi dalsze wyprowadzenia.

Stan kubitu można zapisać jako kombinację liniową stanów bazowych (ket zero) oraz (ket jeden) korzystając z notacji Diraca zwanej też notacją bra-ket[[12]](#footnote-12):

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie: i są amplitudami, tzn. liczbami zespolonymi opisującymi stan układu w fizyce, pozwalającymi na wygodne opisywanie zjawisk takich jak interferencja i spełniają warunek:

|  |  |  |
| --- | --- | --- |
|  |  |  |

co gwarantuje, że wektor będzie długości równej 1 (w normie drugiej). W ogólności dla *n*‑kubitowego systemu kwantowego można otrzymać stanów. Wówczas równanie 2 przyjmuje postać:

|  |  |  |
| --- | --- | --- |
|  |  |  |

lub równoważnie:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie oznacza iloczyn skalarny wektora z wektorem .

Korzystając z równania Eulera, można zapisać stan kubitu jako[[13]](#footnote-13):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Kąty oraz sterują kątami obrotów wektorów stanów i po jednostkowym okręgu zespolonym. Z kolei kąt steruje długością wektorów bazowych, tzn. w równaniu 2: i , a więc kąt ma wpływ na udział stanów bazowych w superpozycji stanu reprezentowanego przez wektor lub innymi słowy: wpływa na długość wektora będącego projekcją wektora rzutowanego ortogonalnie na podprzestrzeń rozpinaną przez wektor i ortogonalną do lub na odwrót (co jest możliwe na mocy twierdzenia spektralnego). Każdy z elementów powyższych wektorów wymaga reprezentacji na 2‑wymiarowej płaszczyźnie zespolonej, ponieważ wektory stanów i są dwuelementowe, dlatego łączna liczba wymiarów wymaganych do geometrycznego przedstawienia wektora wzrasta do czterech. Elementy oraz w równaniu 5 odpowiadają dokładnie amplitudom i w równaniu 1. Ponadto warto zauważyć, że oraz w równaniu 5 mają długość w normie drugiej równą 1, nie mają więc wpływu na wielkość i w równaniu 2. Zgodnie z równaniem 5 reprezentacją geometryczną wektora jest 4-wymiarowa sfera jednostkowa – po 2 wymiary na część rzeczywistą i część urojoną oraz po 2 wymiary na każdy z elementów wektora.

Jedną z własności, na które należy zwrócić uwagę jest, to że kubity reprezentowane przez wektory równoległe, są z perspektywy fizyki nierozróżnialne. Oznacza to, że w równaniu 5 znajduje się redundantna informacja. Z racji wykorzystywania wektora należącego do przestrzeni Hilberta, tzn. wykorzystującego liczby zespolone, pojęcia równoległości należy zastąpić sytuacją, w której nierówność Cauchy–Bunyakovsky–Schwarz (CBS) postaci:

|  |  |  |
| --- | --- | --- |
|  |  |  |

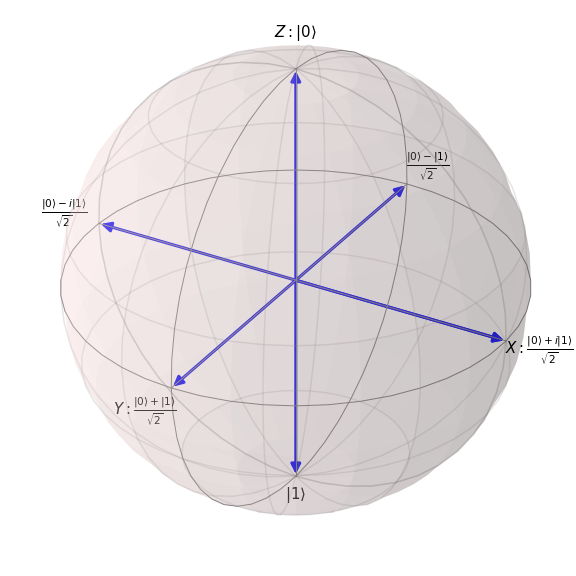
jest równością. Można zauważyć, że dwa wektory oraz będą spełniały powyższy warunek jako równość wtedy i tylko wtedy, kiedy będą miały dokładnie taką samą część , tj. identyczne kąty oraz , natomiast kąt nie będzie miał wpływu na nierówność CBS, czyli w równaniu 5 można zignorować . Korzystając z powyższego element , zwany też fazą globalną, można pominąć, ponieważ nie powoduje obserwowalnych zmian, więc z perspektywy obliczeń kwantowych będzie on nieistotny i w rezultacie otrzymamy[[14]](#footnote-14):

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie oznacza kąt obrotu po płaszczyźnie tworzonej przez część rzeczywistą wektorów bazowych, a to kąt obrotu wokół osi wyznaczanej przez wektor oraz założyć, że współczynnik przy wektorze jest rzeczywisty i nieujemny. Element jest nazywany fazą lokalną lub relatywną,

Stan kubitu można reprezentować przy pomocy wektora w 3-wymiarowej sferze (rysunek 1), zwanej sferą Blocha[[15]](#footnote-15). Warto zwrócić uwagę, że zgodnie z równaniem 1 i równaniem 2 reprezentacja graficzna powinna mieć charakter 4-wymiarowy, jednakże zgodnie z równaniem 7 możemy ograniczyć tę reprezentację do trzech wymiarów. Będzie to jednak skutkowało utratą ortogonalności stanów bazowych w wizualizacji. Stąd też w równaniu 7 kąty w funkcjach trygonometrycznych są dzielone przez dwa, ponieważ to kąt obrotu wokół osi Y w sferze Blocha. Stany bazowe znajdują się na krańcach osi Z sfery Blocha. W sferze Blocha określa kąt obrotu wokół osi Z.

Rysunek . Wizualizacja sfery Blocha.



**Źródło:** opracowanie własne na podstawie M. A. Nielsen, I.I. Chuang *Quantum Computing and Quantum Information. 10th Anniversary Edition.* Cambridge University Press, Cambridge, 2010, s. 15.

Wybór wektorów oraz na wektory bazowe wynika m.in. z powszechności ich użycia. Ta z kolei wynika z ich własności tych wektorów. Po pierwsze warto zauważyć, że po przekształceniu ich przy pomocy równania 7 otrzymano przestrzeń trójwymiarową, reprezentowaną przez sferę jednostkową nad przestrzenią liczb zespolonych. Odbicie względem osi (wektorów bazowych) rozpinających tę przestrzeń dla przypadku opisanego przez równanie 7 jest realizowane przez macierze: dla osi X, dla osi Y oraz dla osi Z, od nazw osi pochodzą oznaczenia tych macierzy, odpowiednio: , , . Macierze te są też zwane macierzami Pauliego. Wyznaczniki tych macierzy są równe 1, ślady są równe 0, a wartości własne to 1 oraz -1. Wektory własne tych macierzy (znormalizowane do 1) to:

* dla : oraz – wektory te są też oznaczane w informatyce kwantowej jako wektory stanów (ket-plus) oraz (ket-minus);
* dla : oraz ;
* dla : oraz .

Wektory własne macierzy Pauliego sugerują możliwy dobór wektorów bazowych (i stanów bazowych) – wszystkie wyżej wymienione pary wektorów mogą być stanami bazowymi systemu kwantowego, a wybierając jedną z par wektorów, stany bazowe będą się znajdować na sferze Blocha na osi odpowiadającej odpowiedniej macierzy Pauliego (patrz: rysunek 1). Macierze Pauliego są macierzami hermitowskimi, a to oznacza, że wektory własne każdej z nich tworzą układ zupełny, tzn. każdy wektor generowany przez taki operator można wyrazić jako sumę wektorów własnych, które w tym przypadku są ortonormalne[[16]](#footnote-16). Przemnożenie wektorów bazowych przez macierz Pauliego inną od tej, od której pochodzą te wektory, będzie skutkowało otrzymaniem wektora ortogonalnego, tj. drugiego wektora własnego macierzy Pauliego.

Macierze Pauliego są antykomutatorami, tzn. zamiana kolejności mnożenia dwóch różnych macierzy Pauliego skutkuje zmianą znaku wyniku na przeciwny oraz , gdzie to symbol komutacji, a A, B i C to różne macierze Pauliego. Pomnożenie każdej macierzy Pauliego przez siebie daje macierz jednostkową *I*.

Na potrzeby tej pracy systemem kwantowym będzie określany system składający się z kubitów, a rejestrem kwantowym – zbiór kubitów. Analogicznie rejestr klasyczny będzie się składał z bitów. Pod pojęciem maszyn, komputerów lub procesorów kwantowych będą rozumiane te, które są oparte na tzw. modelu bramkowym (w odróżnieniu od wyżarzaczy kwantowych – dokładne rozumienie tej różnicy nie jest konieczne dla zrozumienia tej pracy, dlatego zostanie ona pominięta). Wektor określający stan systemu, w którym wszystkie *n* kubitów znajduje się w stanie będzie oznaczany z wykorzystaniem symbolu iloczynu tensorowego (tożsamego tutaj z iloczynem Kroneckera) lub przez pogrubienie , jeśli liczba kubitów w stanie będzie jasno wynikać z kontekstu. Warto zwrócić uwagę, że symbol iloczynu Kroneckera może być zastąpiony (o ile stany nie są splątane) iloczynem , co jest czynione w niektórych publikacjach.

### Kluczowe zjawiska

Jednym ze zjawisk charakterystycznych dla mechaniki kwantowej, które jest wykorzystywane na szeroką skalę w obliczeniach kwantowych jest superpozycja. Oznacza ono, że cząsteczka kwantowa znajduje się równocześnie w dwóch przeciwstawnych stanach albo innymi słowy, że znajduje się gdzieś pomiędzy nimi. Z perspektywy informatyki kwantowej sprowadza się to do sytuacji, gdy współczynniki i definiujące bieżący stan kubitu są takie, że [[17]](#footnote-17)

Długotrwałe utrzymanie kubitu fizycznego w superpozycji jest obecnie zadaniem technologicznie trudnym. Fizycznie, interakcja z otoczeniem obiektu reprezentowanego przez kubit, doprowadza do redukcji (*collapse*) funkcji falowej. Dla obliczeń kwantowych oznacza to przejście stanu superpozycji do jednego ze stanów bazowych. Znajdując się w superpozycji, stan kubitu może się również zmieniać (tj. wektor reprezentujący stan kubitu będzie ulegał odchyleniom) bez doprowadzania do redukcji. Niekontrolowane zmiany stanu kubitu stanowią jeden z kluczowych problemów stabilności urządzeń kwantowych. Oddziaływanie kubitu z otoczeniem nosi nazwę dekoherencji. Jeżeli jest ona niezamierzona, wówczas jej skutki nazywamy szumem kwantowym (*quantum noise)*[[18]](#footnote-18) [[19]](#footnote-19).

Przejście ze stanu superpozycji do stanu bazowego może być również następstwem zamierzonym działań, tj. wykonania pomiaru (obserwacji). Pomiar stanowi często jeden z końcowych elementów algorytmów kwantowych bądź też ich istotnych części. W jego wyniku kubit przechodzi do jednego ze stanów bazowych[[20]](#footnote-20). W wyniku dekoherencji (niezamierzonej lub spowodowanej pomiarem) kubit znajdzie się w stanie lub z prawdopodobieństwem równym odpowiednio lub , co jest zgodne z aksjomatyczną definicją prawdopodobieństwa dzięki równaniu 2.

Ogólnie, zbiór możliwych do odczytania wartości w czasie pomiaru jest zdefiniowany przez wartości własne macierzy (dokładniej: operatora kwantowego lub obserwabli), która odpowiada za akt pomiaru. Aby dany operator był obserwablą, jego wektory własne muszą tworzyć bazę przestrzeni Hilberta. W takim ujęciu prawdopodobieństwo otrzymania w pomiarze wartości jest równe:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie jest sprzężeniem hermitowskim wektora własnego obserwabli odpowiadającego wartości własnej , a pewnym wektorem reprezentującym stan kwantowy, na którym dokonano pomiaru[[21]](#footnote-21). Przyjmując poprzednie założenia, tj., że stany bazowe to oraz , wówczas macierz jest obserwablą systemu kwantowego. W równaniu 8 można też zauważyć, że faza globalna nie będzie miała wpływu na wynik, podczas gdy faza relatywna będzie miała wpływ na wynik, ale różny w zależności od obserwabli. W informatyce kwantowej przyjęło się opisywać wynik pomiaru przy pomocy wektorów własnych odpowiadających sprawdzanej wartości własnej, tak więc w wyniku dokonania pomiaru przy pomocy obserwabli (tj. dokonania pomiaru w bazie tworzonej przez wektory własne macierzy tej obserwabli, tutaj zwanego bazą Z) możliwe do otrzymania stany to nie 1 i -1, lecz oraz , czyli oraz . Możliwe jest też dokonywanie pomiarów w innych bazach (nie tylko tych odpowiadającym macierzom Pauliego), jednak zgodnie z przyjętymi wcześniej założeniami, pomiar zawsze będzie dokonywany w bazie Z.

Kolejnym istotnym zjawiskiem jest splątanie kwantowe. Zjawisko to polega na przyczynowo-skutkowym (a nie tylko liczbowym) powiązaniu dwóch lub więcej kubitów. Fizyczny aspekt tego zjawiska nadal jest przedmiotem dyskusji. Z perspektywy obliczeń kwantowych oznacza to m.in. możliwość wpływania na jeden kubit przy pomocy innego, splątanego z nim wcześniej kubitu. Konsekwencją tego jest możliwość wnioskowania na temat stanu jednego kubitu na podstawie stanu innego kubitu. Zjawisko splątania jest również podatne na skutki dekoherencji – jeżeli jeden kubit ulegnie dekoherencji, to pozostałe splątane z nim kubity mogą również jej ulec. Podobnie jak w przypadku pomiaru, doprowadzenie do takiego zjawiska jest często intencjonalne[[22]](#footnote-22).

### Wykonywanie działań

Działania na komputerach klasycznych są wykonywane z pomocą bramek logicznych i algebry boolowskiej. Do przykładowych bramek logicznych można zaliczyć AND, OR, XOR oraz NOT[[23]](#footnote-23). Z kolei operacje na kubitach są konstruowane z wykorzystaniem macierzy i algebry liniowej. Macierze te, zwane też bramkami lub operatorami kwantowymi, są macierzami unitarnymi, tak więc, w przeciwieństwie do komputerów klasycznych, wszystkie pojedyncze operacje (tak samo jak reprezentujące je macierze) są odwracalne. Fizycznie, realizacja działań reprezentowanych przez bramki, polega na oddziaływaniu na fizyczną realizację kubitu impulsem elektromagnetycznym[[24]](#footnote-24).

Niektóre bramki kwantowe mają swoje nazwy oraz dedykowane oznaczenia. Mogą one mieć postać stałą lub parametryzowaną, działać zarówno na jednym jak i na wielu kubitach. Mają one również swoje oznaczenia graficzne, które są używane w wizualizacjach obwodów kwantowych, przy pomocy których można graficznie opisać przebieg algorytmu kwantowego. Przegląd najważniejszych bramek wraz z ich oznaczeniami używanymi w dalszej części zawiera tabela 1.

Tabela . Podstawowe operacje kwantowe i ich oznaczenia.

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa operatora | Skrót | Macierz | Symbol graficzny |
| Identity | I, ID |  |  |
| Pauli-X, NOT | X, NOT, |  | Obraz zawierający tekst  Opis wygenerowany automatycznie |
| Pauli-Y | Y, |  |  |
| Pauli-Z | Z, |  | Obraz zawierający tekst  Opis wygenerowany automatycznie |
| Hadamard | H |  |  |
| T, | T |  |  |
| Phase | S, P |  | Obraz zawierający tekst, tablica suchościerna  Opis wygenerowany automatycznie |
| S-dagger, sprzężenie hermitowskie S |  |  |  |
| Squared X | SX |  |  |
| SWAP | SWAP |  |  |
| Controlled NOT | CX, CNOT, C1NOT, C1X |  |  |
| Controlled Z | CZ |  | lub Obraz zawierający tekst, zegar  Opis wygenerowany automatycznie |
| Toffoli | CCNOT, CCX, TOFF, C1C1NOT, C1C1X |  | Obraz zawierający tekst, zegar  Opis wygenerowany automatycznie  lub |
| Ry | Ry |  |  |
| Rz | Rz |  |  |
| Rx | Rx |  |  |
| Pomiar (Measurement) | M | - | Obraz zawierający tekst, zegar, wskaźnik  Opis wygenerowany automatycznie  lub |

**Źródło:** opracowanie własne na podstawie M. A. Nielsen, I.I. Chuang *Quantum Computing and Quantum Information. 10th Anniversary Edition.* Cambridge, 2010, Cambridge University Press.

Ogólna postać kwantowej bramki unarnej (bramki działającej na jednym kubicie) ma postać[[25]](#footnote-25):

|  |  |  |
| --- | --- | --- |
|  |  |  |

lub alternatywnie:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdyż obie postaci są identyczne co do fazy globalnej, tj.:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Bramkę tę można przedstawić obrazując tym samym obrót wokół poszczególnych osi sfery Blocha[[26]](#footnote-26):

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie:

|  |  |  |
| --- | --- | --- |
|  | , |  |

co gwarantuje, że wektor, na którym zastosowano macierz, nie zmieni swojej długości. Poszczególne skalary , , reprezentują obrót wokół odpowiednich osi (patrz: równanie 16). Warto przypomnieć uogólnienie tożsamości Eulera na eksponentę macierzy[[27]](#footnote-27):

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie . Stąd, korzystając z eksponenty macierzy (równanie 14) oraz szeregów Taylora, można pokazać, że bramka Rx jest częściową rotacją wokół osi określonej przez bramkę X (analogicznie dla bramki Ry oraz Rz):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Analogicznie, po rozwinięciu równania 12 można otrzymać:

|  |  |  |
| --- | --- | --- |
|  |  |  |

co zgodnie z równaniem 14 odpowiada obrotowi na sferze jednostkowej – przestrzeni rozpinanej przez macierze Pauliego.

Bramki kwantowe dzięki swojej postaci, powodują, że wektor (reprezentujący system kwantowy) na którym są stosowane, zachowuje swoją długość (*length-preserving operator*), więc poprawnie zdefiniowana bramka kwantowa zastosowana na poprawnie zdefiniowanym stanie kwantowym zwróci poprawnie zdefiniowany stan kwantowy o niezmienionej długości.

Podobnie jak w przypadku wektorów, również operatory można łączyć wykorzystując iloczynem tensorowy (Kroneckera). Wówczas można do bramki kwantowej dodać indeks dolny oznaczający kubit lub kubity, na które bramka kwantowa będzie oddziaływać. Indeks dolny zapisany w postaci liczby w systemie dziesiętnym należy przekodować na system dwójkowy – miejsca w których znajdują się jedynki odpowiadają kubitom, na które opisywana bramka będzie oddziaływać, natomiast pozostałe kubity nie zmienią swojego stanu, dlatego zostanie wprowadzona notacja:

gdzie:

Zostanie również wprowadzony symbol iloczynu Hadamarda (zwanego też iloczynem Schura lub iloczynem po współrzędnych), tj. mnożenia macierzy o takich samych wymiarach element po elemencie:

## Postać ogólna problemu marszrutyzacji

### Postać ogólna problemu VRP

Problem komiwojażera (Travelling Salesman Problem – TSP) jest jednym z najbardziej znanych problemów optymalizacyjnych. Polega on na znalezieniu jak najkrótszej trasy pozwalającej na odwiedzenie wszystkich określonych miejsc (reprezentowanych przez wierzchołki grafu), pomiędzy którymi odległości są znane. Wprowadzony został on w 1934 przez Hasslera Whitneya[[28]](#footnote-28), ujęty matematycznie przez Merilla Flooda w 1955 roku[[29]](#footnote-29) [[30]](#footnote-30), a następnie uogólniony pod nazwą Truck Dispatching Problem przez Georga Dantziga i Johna Ramsera poprzez dodanie wymogu powrotu do pozycji startowej po odwiedzeniu określonej liczby lokacji[[31]](#footnote-31). Udowodniono, że problem TSP jest NP-trudny[[32]](#footnote-32), przy czym wykazano także, że jego decyzyjna wersja jest także NP-zupełna przez sprowadzenie do problemu grafu hamiltonowskiego, który jest problemem NP-zupełnym[[33]](#footnote-33).

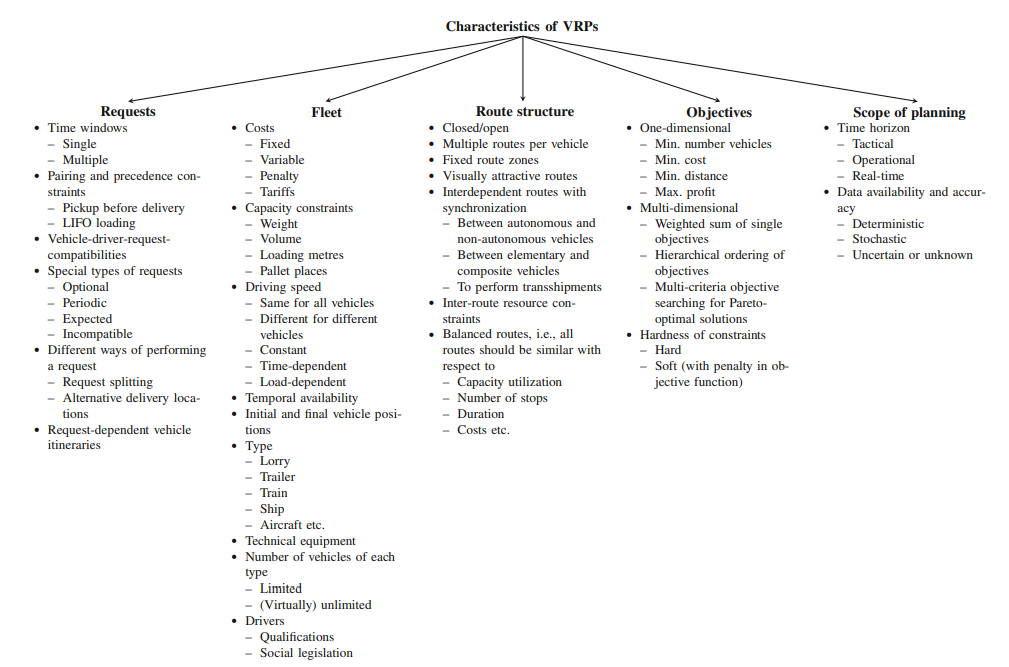
Już Dantzig i Ramser we wspomnianym wcześniej artykule zauważyli, że problem opisany przez Flooda można rozszerzyć[[34]](#footnote-34). Z czasem rzeczywiście pojawiało się coraz więcej rozszerzeń, rozmyło się również nazewnictwo[[35]](#footnote-35). Uogólnienie problemu TSP zaczęło nosić nazwę problemu marszrutyzacji (Vehicle Routing Problem – VRP[[36]](#footnote-36) lub General Routing Problem - GRP[[37]](#footnote-37)). Ich cechą wspólną jest również to, że podobnie jak problem TSP, są problemami NP-trudnymi z wersjami decyzyjnymi NP-zupełnymi[[38]](#footnote-38) [[39]](#footnote-39). Niezależnie od nazwy, ogólna idea pozostaje ta sama: pojazd (agent) musi odwiedzić wszystkie miejsca (wierzchołki grafu lub krawędzie grafu) w jak najkrótszym czasie (minimalizując funkcję kosztu).

Definicja problemu VRP jest często rozwijana lub modyfikowana w zależności od postawionego zadania, tworząc kolejne odmiany problemu VRP. Do przykładowych odmian można zaliczyć:

* problem VRP z *m* pojazdami,
* problem VRP z *k* różnymi typami pojazdów, każdy w liczebności *mi*, ;
* problem VRP z ograniczoną pojemnością pojazdów;
* problem VRP o różnej wartości kosztu przejścia pomiędzy tymi samymi miejscami, tzn.  – wówczas graf reprezentujący problem będzie grafem skierowanym;
* problem VRP z oknami czasowymi (VRPTW);
* problem VRP z wieloma punktami zaopatrzenia.

Powyższa lista nie wyczerpuje listy możliwych modyfikacji problemu VRP. Możliwe jest również łączenie tychże modyfikacji. Jak zauważył M. Drexl, mnogość publikacji na temat problemu VRP wynika z kilka powodów: pierwszy to jego znaczenie dla logistyki (chociaż warto zauważyć, że problemy logistyczne nie dotyczą wyłącznie transportu, co pokazano w Rozdziale II.3.2), drugi to wyzwanie intelektualne, jakie stanowi ten problem[[40]](#footnote-40). Na rysunku 2 przedstawiono przykładowy podział problemów VRP zaproponowany przez Drexla, ukazując tym samym mnogość możliwych modyfikacji.

Rysunek 2: Odmiany problemu VRP.



**Źródło:** M.Drexl, *Rich vehicle routing in theory and practice*, Logistics Research, 5, 47–63 (2012), s. 49.

### Postać matematyczna VRP

Będąc świadomym różnych ujęć i rozwinięć problemu VRP Raff zaproponował ogólne ujęcie matematyczne problemu[[41]](#footnote-41), które przedstawiono poniżej (zastosowano zmiany w oznaczeniach).

Niech graf składa się z V wierzchołków (*vertices*), A krawędzi (*edges)* oraz macierzy kosztów C, która przyjmujemy, że jest symetryczna, tzn. . Celem jest minimalizacja funkcji kosztu:

|  |  |  |
| --- | --- | --- |
|  |  |  |

pod warunkami:

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie:

– koszt przejazdu z *i­-*tego do *j­*-tego wierzchołka,

– informacja binarna czy przejazd odbywa się przez krawędź łączącą *i*-ty wierzchołek z *j*‑tym (0 – nie odbywa się, 1 – odbywa się),

– liczba przyjazdów do *j*-tego wierzchołka,

– liczba wyjazdów z *i*-tego wierzchołka,

*n* – liczba wierzchołków, ,

– zbiór możliwych rozwiązań o długości *n,*

– zbiór wszystkich możliwych kombinacji połączeń wierzchołków.

Warunki określone równaniami 18 i 19 gwarantują, że każdy wierzchołek zostanie odwiedzony dokładnie jeden raz oraz, że z każdego wierzchołka pojazd wyjedzie dokładnie jeden raz. Warunek związany ze zbiorem dotyczy powstawania zamkniętych cykli niepołączonych z punktem startowym. Można go również interpretować jako nakaz wykonania kolejnego kroku startując z miejsca, w którym zakończono poprzedni krok, bądź też rozwiązanie, w którym istnieje dokładnie jeden graf hamiltonowski łączący wszystkie wierzchołki (**patrz Rysunek)**. W przeciwieństwie do poprzednich dwóch ograniczeń lokalnych, warunek dotyczący zamkniętych cykli jest warunkiem o charakterze globalnym. W zależności od konkretnego problemu VRP może on ulegać istotnym modyfikacjom. Poniżej zostanie przedstawionych kilka najważniejszych metod i podejść do sformułowania tego ograniczenia, które daje się zastosować w większości problemów VRP (założono optymalizację na grafie nieskierowanym).

Bektas zaproponował następującą formułę[[42]](#footnote-42):

|  |  |  |
| --- | --- | --- |
|  |  |  |

która jest równoważna:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie oznacza wszystkie wierzchołki z wyjątkiem tego, który oznacza punkt startowy (tzn. ), a oznacza liczbę podtras będących częścią opracowanej trasy. Takie ujęcie warunku nazywa się też jawnym ograniczeniem Dantziga-Fulkersona-Johnsona (chociaż nie zostało przez nich opracowane tylko przez Tolgę Bektasa). Powoduje ono, że liczba warunków ograniczających rośnie w tempie wykładniczym wraz ze wzrostem liczby wierzchołków[[43]](#footnote-43). Wynika to z tego, że zbiór , jest zbliżony liczebnością do zbioru potęgowego zbioru , a konkretnie , ze względu na wykluczenie zbioru pustego. Głównym zamysłem równania 20 jest to, że liczba krawędzi w każdym z podzbiorów zbioru różnym od zbioru pustego powinna być o co najmniej 1 mniejsza niż liczba wierzchołków wchodzących w skład danego podzbioru, ponieważ jedynym grafem cyklicznym w zbiorze potęgowym zbioru , jest właśnie graf . Równanie 21 oznacza, że dla każdego podgrafu istnieje taki wierzchołek, do którego prowadzi co najmniej jedna krawędź spoza danego podgrafu (przy czym wyklucza się tutaj punkt startowy, więc ta nierówność jest spełniona też dla zbioru *V*).

Innym z podejść jest zastosowanie niejawnych ograniczeń Dantziga-Fulkersona-Johnsona (Implicit Dantzig-Fulkerson-Johnson formulation). Algorytm szukający optimum rozpoczyna działanie bez żadnych ograniczeń dotyczących zamkniętych cykli. Po otrzymaniu rozwiązania w każdej iteracji, trasa jest sprawdzana pod względem posiadania zamkniętych cykli. Jeśli rozwiązanie okaże się niedopuszczalne, wówczas do zestawu ograniczeń zostaje dodany warunek, który nie zezwala na ponowne powstanie otrzymanego w danej iteracji zamkniętego cyklu. Metoda ta pozostawia dowolność w kwestii definiowania warunku ograniczającego. Takie ujęcie pozwala zmniejszyć liczbę warunków ograniczających i okazało się efektywne, dopóki nie wynaleziono kolejnych metod[[44]](#footnote-44).

Grupę metod bazujących na podziale przestrzeni rozwiązań dla dużych problemów TSP/VRP omawia Applegate et al. Omawiana tam tzw. metoda cięcia płaszczyzny jest częścią algorytmów, których celem jest zmniejszenie rozmiaru przeszukiwanej przestrzeni. Ogólny zamysł polega na wydzieleniu grup wierzchołków leżących blisko siebie (co samo w sobie może być oddzielnym zadaniem) a następnie tworzenie warunków ograniczających w odniesieniu do powstałych grup wierzchołków, a nie całego zbioru [[45]](#footnote-45) Efektywność czasowa tego podejścia opiera się na tym, że suma zbiorów potęgowych zbiorów rozłącznych jest mniejszym zbiorem niż zbiór potęgowy sumy tych zbiorów.

Miller, Tucker i Zemlin zaproponowali sformułowanie tych ograniczeń dla problemu CVRP (Capacitated VRP – problem VRP z ograniczoną pojemnością pojazdu – po odwiedzeniu określonej liczby miejsc, agent musi wrócić do punktu startowego uzupełnić zapas), dla którego liczba ograniczeń związanych z zamkniętymi cyklami rośnie w tempie wielomianowym [[46]](#footnote-46)Kulkarni et al. te ograniczenia uogólnił[[47]](#footnote-47), a Kara et al. poprawił i przeformułował do postaci dwóch zestawów następujących nierówności[[48]](#footnote-48):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | |
|  |  | | |  |

gdzie:

– numer odwiedzonego wierzchołka – numer nadawany jest przy pierwszym przyjechaniu do wierzchołka, za każdym razem coraz większy. Punkt startowy nigdy nie ma nadawanego numeru.

– maksymalna pojemność pojazdu,

– zapotrzebowanie zgłaszane w i-tym wierzchołku.

Warto zauważyć, że warunki opisane przez równania 22 i 23 nie obowiązują, gdy pojazd jedzie do punktu startowego, ze względu na *j* dla których obowiązuje ograniczenie. Dla problemu VRP bez ograniczeń związanych z pojemnością pojazdu można przyjąć, że  oraz .

Powyższe ujęcia warunku niepowstawania zamkniętych cykli chociaż są często przywoływane w literaturze dotyczącej problemu komiwojażera, to jednak mogą okazać się problematyczne w implementacji na maszynach kwantowych ze względu na brak lub trudne przekształcenie zrelaksowanych wersji tychże warunków do postaci funkcji kwadratowej oraz dodanie dodatkowych zmiennych, co zwiększy liczbę potrzebnych jednostek obliczeniowych niezbędnych do ujęcia ograniczenia. Dlatego zostaną przywołane także inne metody.

Andrew Lucas rozpatrujący ujęcie różnych problemów NP-trudnych w sposób możliwy do przeniesienia na algorytmy kwantowe zaproponował postać dla grafów skierowanych[[49]](#footnote-49):

Gdzie:

– zmienna binarna określająca fakt odwiedzenia *u*-tego wierzchołka w czasie *j*

*E* – zbiór krawędzi należących do zbioru aktywnych (używanych w rozwiązaniu) krawędzi.

Suma ta będzie równa zero, jeżeli nie będzie istniało połączenie pomiędzy wierzchołkami nie należącymi do rozwiązania. Kłopotliwe w tym ujęciu jest natomiast zdefiniowanie zbioru *E*, ponieważ takie ujęcie problemu zakłada obliczanie funkcji kary po wyznaczeniu rozwiązania – takie rozwiązanie nie będzie odpowiednie dla optymalizacji znanymi algorytmami kwantowymi.

W artykule Salehi, Glosa i Miszczaka dla problemu VRP-TW (VRP z oknami czasowymi – aby rozwiązanie było dopuszczalne, wierzchołki muszą zostać odwiedzone w określonych ramach czasowych) ograniczenia zostały wcielone do funkcji celu. Funkcja ta nie zabrania osiągania rozwiązań niedopuszczalnych, natomiast nakłada karę na funkcję celu za niespełnienie warunku. Warunek ten ma postać następującego sformułowania[[50]](#footnote-50):

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie:

Pierwszy i trzeci składnik funkcji odpowiadają odpowiednio za wyjazd z i powrót do wierzchołka początkowego, a drugi składnik – za przejazd przez pozostałe wierzchołki. W tej funkcji za każdym razem są sprawdzane 2 przejazdy – bieżący i następujący bezpośrednio po nim. Jeśli w kroku następnym ((-szym) pojazd będzie startował z innego wierzchołka niż ten, w którym skończył w kroku , wówczas iloczyn będzie równy zero. Sformułowanie ograniczenia w języku funkcji kary, powoduje, że zapis tego sformułowania można łączyć z innymi ograniczeniami, łącząc w ten sposób kilka ograniczeń w postaci jednej funkcji, która jest krótsza, niż dwie oddzielne funkcje kary[[51]](#footnote-51). Dobór wielkości kary stanowi kompromis – zbyt wysoka kara skutkuje zachłannością algorytmu i wpadnięciem w pułapkę lokalnego minimum (które nie musi być globalnym minimum funkcji celu). Zbyt niska kara może skutkować otrzymaniem rozwiązania niespełniającego ograniczeń.

## Kwantowe i hybrydowe algorytmy optymalizacyjne

### QAOA jako algorytm wariancyjny

Jednym z algorytmów kwantowych dedykowanych optymalizacji kombinatorycznej jest kwantowy algorytm przybliżonej optymalizacji (*Quantum Approximate Optimization Algorithm* – QAOA), zaproponowany w 2014 przez Edwarda Farhiego, Jeffreya Goldstone’a i Sama Gutmana do rozwiązywania problemu max-cut[[52]](#footnote-52), który podobnie jak problem VRP, jest problemem kombinatorycznym (tzn. dyskretnym, o relatywnie dużej przestrzeni rozwiązań). W Rozdziałach I.3.2 oraz I.3.3 przedstawiono algorytm QAOA zgodnie z założeniami jego twórców. Algorytm ten należy do klasy kwantowych algorytmów wariancyjnych (*Variational Quantum Algorithm* - VQA). Algorytmy VQA są algorytmami hybrydowymi – łączą one obliczenia dokonywane na kubitach z wykorzystaniem maszyn kwantowych z obliczeniami na maszynach klasycznych (bitowych). Łącznikami są pomiary, które skutkują przeniesieniem pewnej informacji z kubitu na bit oraz kwantowe bramki parametryzowane, gdzie wartość parametrów jest wyznaczana przy pomocy klasycznych algorytmów[[53]](#footnote-53).

Udowodniono, że dla problemu MAX-3LIN-EQN algorytm QAOA przez pewien czas osiągał lepszą jakość przybliżeń niż jakikolwiek znany klasyczny algorytm[[54]](#footnote-54) [[55]](#footnote-55). Problem MAX-3LIN-EQN jest problemem NP-trudnym, gdzie danymi wejściowymi jest układ równań liniowych (w arytmetyce modulo 2), każdy z maksymalnie 3 zmiennymi. Celem jest znalezienie takich wartości zmiennych, aby jak największa liczba równań została spełniona[[56]](#footnote-56).

Jedną z głównych wad algorytmu QAOA jest jego silna zależność od stosunku liczby ograniczeń do liczby zmiennych[[57]](#footnote-57). Algorytm QAOA może być wykorzystany również jako część innych algorytmów, które, przynajmniej teoretycznie, mogą rozwiązać problem w najkrótszym możliwym czasie, np. algorytm Grovera[[58]](#footnote-58) [[59]](#footnote-59) [[60]](#footnote-60).

### Założenia algorytmu QAOA

W optymalizacji z wykorzystaniem algorytmów kwantowych jednym z kluczowych pojęć jest Hamiltonian (zwany też funkcją lub operatorem Hamiltona). Opisuje on ilość energii w systemie w zależności od stanu systemu. Z perspektywy informatyki kwantowej służy on do przekształcenia wektora stanu kwantowego do pewnej postaci. Hamiltonian oznaczany   (dla przypadku ogólnego) lub (dla algorytmu QAOA) jest macierzową reprezentacją funkcji celu, a Hamiltonian pewną funkcję mieszającą stosowaną w algorytmie.

Zadania optymalizacyjne tłumaczy się na język informatyki kwantowej jako zadanie polegające na znalezieniu takiego stanu (wektora stanu), dla którego ilość energii jest w systemie jest najniższa (*lowest energy state, ground state*) lub najwyższa, w zależności od celu optymalizacji. Energię systemu w stanie opisuje wartość oczekiwana, która jest wartością funkcji celu dla zmiennych określanych przez wektor [[61]](#footnote-61):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Zadanie odnalezienia stanu podstawowego , tzn. stanu, w którym ilość energii w systemie jest najmniejsza, można traktować jako zadanie odnalezienia minimum globalnego funkcji celu, zadanego jako[[62]](#footnote-62):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Algorytm polega na znalezieniu pewnego stanu, który zależy od parametru (tutaj: zestawu parametrów) , tj. . Należy więc znaleźć taką wartość parametru , która pozwoli na minimalizację funkcji celu postaci:

|  |  |  |
| --- | --- | --- |
|  |  |  |

więc:

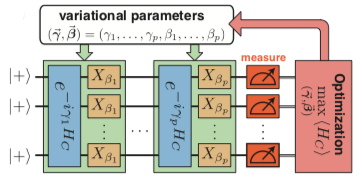
|  |  |  |
| --- | --- | --- |
|  |  |  |

Znaleziony stan będzie przybliżeniem wartości zmiennych odpowiadającym wartości poszukiwanego optimum globalnego oznaczanego [[63]](#footnote-63).

W trakcie działania algorytmu stan będzie otrzymywany poprzez zastosowanie na stanie początkowym parametryzowalnego operatora kwantowego (bramki kwantowej), tj. . Parametr jest aktualizowany co iterację przez algorytm klasyczny. Dane dla algorytmu klasycznego są przekazywane jako wynik pomiaru na końcu kwantowej części algorytmu w danej iteracji (rysunek 3)[[64]](#footnote-64).

W opisany powyżej sposób, część kwantowa algorytmu QAOA będąca kwantowym obwodem wariancyjnym, zwraca wektor reprezentujący pewien stan kwantowy, którego wartości zależą od wartości parametrów. Wartości parametrów są aktualizowane przez część klasyczną na podstawie wartości skalarnej (kodowanej binarnie w postaci ciągu bitów) zwracanej przez część kwantową do rejestru klasycznego. Parametry te są przekazywane z powrotem do części kwantowej, po czym rozpoczyna się kolejna iteracja algorytmu[[65]](#footnote-65). Można zauważyć, że sposób działania algorytmu QAOA jest podobny do sposobu działania klasycznych sieci neuronowych[[66]](#footnote-66), gdzie wagi kombinacji liniowych w warstwach sieci są aktualizowane przez algorytm propagacji wstecznej na podstawie wyników zwróconych przez sieć, po czym iteracja jest powtarzana.

Rysunek : Schemat algorytmu QAOA.



**Źródło:** https://medium.com/mit-6-s089-intro-to-quantum-computing/qaoa-bench-marking-7dfdd8a31e54, dostęp: 12 października 2022.

### Część kwantowa algorytmu QAOA

Iteracja algorytmu QAOA zaczyna się od wprowadzenia wszystkich kubitów w stan równej superpozycji (konkretnie w stan ket plus: ), poprzez zastosowanie na nich bramek Hadamarda[[67]](#footnote-67):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Stanowi to etap przygotowawczy, który pozwoli na wykorzystanie zjawisk kwantowych w kolejnych etapach, w szczególności: zastosowaniu bramek bazujących na Hamiltonianach funkcji celu oraz bramek bazujących na Hamiltonianach mieszających. Etap stosowania operatorów bazujących na Hamiltonianach może być powtarzany wiele razy (tzw. troteryzacja) – kolejne powtórzenia tego etapu w ramach jednej iteracji będą dalej indeksowane małą literą *k*, a ich łączna liczba – dużą literą *K*. O ile sama postać Hamiltonianów pozostanie niezmienna, o tyle sparametryzowane bramki na nich bazujące będą się zmieniały, ponieważ parametry mogą być inne dla każdego *k-*tego powtórzenia.

Przekształcenie funkcji celu z postaci analitycznej do postaci macierzy operatora kwantowego wymaga dokonania na niej przekształceń. Przed przekształceniem funkcji celu w Hamiltonian należy ją najpierw sprowadzić do postaci problemu QUBO, bowiem problem wyrażony w ten sposób daje się łatwo wyrazić w postaci bramek kwantowych, a komputery kwantowe potrafią szybko rozwiązywać problemy QUBO[[68]](#footnote-68).

QUBO (Quadratic Unconstrained Binary Optimization – kwadratowa optymalizacja binarna bez ograniczeń) jest NP-trudnym problemem optymalizacji kombinatorycznej.[[69]](#footnote-69) Metoda sprowadzania problemu optymalizacyjnego do postaci QUBO jest inspirowana modelem Isinga[[70]](#footnote-70) i jest często wykorzystywana w problemach tego typu. Aby sprowadzić problem do postaci QUBO należy wszystkie ograniczenia wcielić do funkcji celu, a następnie wyrazić tę funkcję w postaci formy kwadratowej, która daje się łatwo zamienić na postać macierzową. Funkcja celu w QUBO w przypadku ogólnym jest postaci[[71]](#footnote-71).

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie:

– współczynniki funkcji celu,

– macierz współczynników (wag),

– wektor zmiennych decyzyjnych binarnych,

– funkcja kosztu zależna od ,

Znalezienie Hamiltonianu kosztu odbywa się poprzez rozwiązanie następującego równania[[72]](#footnote-72) [[73]](#footnote-73):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Przyjmując, że wektor jest długości (przypominając: – liczba kubitów, – liczba możliwych stanów bazowych), wówczas otrzymamy:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie to bramka Pauli-Z zastosowana na *i*-tym kubicie. W rezultacie przejdziemy od wektora, dla którego , do wektora, dla którego . W równaniu 32 zamiana miejscami elementów pozwoli otrzymać[[74]](#footnote-74) [[75]](#footnote-75):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Korzystając z równania 33 można zapisać równanie 30 jako:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Nowa postać funkcji nie jest bezpośrednio zależna od , może więc ona być poszukiwanym Hamiltonianem kosztu, który będzie macierzą diagonalną:

|  |  |  |
| --- | --- | --- |
|  |  |  |

W algorytmie QAOA występują bramki parametryzowane. Parametryzacja odbywa się poprzez zastosowanie eksponenty macierzy oraz transformację przy pomocy wektora parametrów gdzie [[76]](#footnote-76):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Należy również zdefiniować Hamiltonian mieszający oraz go sparametryzować

Hamiltonian mieszający powinien być sumą prostych unarnych bramek (tzn. bramką stosowaną na wszystkich kubitach oddzielnie, bez dokonywania splątania). Taką bramką będzie bramka Pauli‑X, bowiem stosując ją na kubicie otrzymamy wartość własną 1, zgodnie z:

|  |  |  |
| --- | --- | --- |
|  |  |  |

oraz -1 dla :

|  |  |  |
| --- | --- | --- |
|  | , |  |

przy czym warto zauważyć, że jeżeli zachodzi potrzeba zmiany problemu minimalizacji na maksymalizacji, można albo rozpoczynać algorytm od zastosowaniu bramki Hadamarda na kubitach w stanie i otrzymać w ten sposób stany , albo zmienić znak funkcji celu. Hamiltonian mieszający będzie postaci[[77]](#footnote-77):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Po parametryzacji otrzymuje się bramkę[[78]](#footnote-78):

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Można więc zauważyć, że dzięki parametryzacji, zamiast zwykłego obrotu o kąt wokół osi OX w sferze Blocha (i przejście do stanu ortogonalnego, np. z do , ale można też tę operację wykonywać na kubitach w superpozycji), następuję obrót częściowy wokół tej osi. Ponadto, należy zauważyć, że eksponenta macierzy Pauli-X zachowuje jej wektory własne.

Celem Hamiltonianu mieszającego jest umożliwienie wyjście z tzw. pułapki lokalnego minimum, wywołanej nadmierną zachłannością algorytmu. Wektory własne macierzy (w tym macierzy i oraz i , ponieważ eksponenta macierzy nie zmienia wektorów własnych) reprezentują pewne stany, dla których funkcje odpowiadające przekształceniom dokonywanym przez macierze, osiągają lokalne ekstremum. W wyniku zastosowania bramki (niezależnie od wartości parametru ) system będzie dążył do znalezienia się w stanie reprezentowanym przez wektor własny macierzy , poprzez wzmacnianie i osłabianie odpowiednich amplitud w systemie kwantowym (*amplitude amplification*), tak aby były one możliwe podobne do tych znajdujących się w wektorach własnych. Skutkiem powyższego procesu jest dążenie do lokalnego ekstremum, które nie musi być ekstremum globalnym, więc algorytm wpadnie w pułapkę lokalnego ekstremum. Wraz z kolejnymi zastosowaniami bramki na stanie będącym wektorem własnym , wektor ten pozostanie niezmienny, tj. , co wynika wprost z definicji wektora własnego. Hamiltonian mieszający musi być anty-komutatorem Hamiltonianu celu , aby mieć pewność, że te dwie macierze nie będą miały takich samych wartości własnych. Operator Pauli‑X jest dobrym wyborem, ponieważ bazująca na nim bramka za wektor własny przyjmuje stan , który jest stanem przyjmowanym na początku algorytmu. W takim przypadku, jeżeli operator za wektor własny przyjmuje , wówczas wykrycie pułapki lokalnego ekstremum będzie trywialne, ponieważ system cały czas będzie się w tym ekstremum znajdował, tj. w stanie .

Wykonane powyżej operacje można zapisać w postaci jednego działania matematycznego, którego wynikiem będzie wektor [[79]](#footnote-79):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Część kwantowa algorytmu QAOA zostaje zakończona serią pomiarów kubitów tworzących wektor . Wynik pomiaru jest rzutowany na bity, po czym przekazywany do maszyny klasycznej w postaci uporządkowanego łańcucha znaków, składającego się z zer i jedynek. Otrzymane w ten sposób łańcuchy znaków stanowią istotny element obliczania wartości oczekiwanej w równaniu 27[[80]](#footnote-80). Sposób obliczania tej wartości oraz wykorzystanie algorytmów klasycznych zostanie szerzej omówione w Rozdziale II – możliwość optymalizacji parametrów i algorytmami klasycznymi, nie została od razu dostrzeżona przez twórców algorytmu QAOA i nie istnieje sztywno ustalona część klasyczna[[81]](#footnote-81).

Stosowanie operatorów , na zmianę więcej niż jeden raz oznacza wspomnianą wcześniej troteryzację, a pojedyncze wykorzystanie tych dwóch bramek tworzy warstwę (*circuit layer)*. Miarą troteryzacji jest liczba warstw. Technika ta jest inspirowana procesami adiabatycznymi, które są wykorzystywane między innymi w wyżarzaniu symulowanym, tyle że w QAOA w ramach poszukiwania stanu o najniższej energii wykonywane są znacznie dłuższe kroki (czyli stopień troteryzacji w QAOA jest znacznie niższy niż liczba iteracji w algorytmie wyżarzania symulowanego)., tj. mniejsza jest liczba iteracji[[82]](#footnote-82). Przewidywanym pozytywnym skutkiem troteryzacji jest przybliżenie się do rozwiązania będącego optimum globalnym. W szczególności mamy gwarancję, że[[83]](#footnote-83):

|  |  |  |
| --- | --- | --- |
|  |  |  |

co oznacza, że dla troteryzacji na poziomie (tzn. z warstwami) wartość funkcji celu jest bliżej szukanej wartości (bliżej minimum dla minimalizacji, bliżej maksimum dla maksymalizacji) niż dla troteryzacji na poziomie *K* (przy pozostałych założeniach niezmienionych). Symbol pochodzi z teorii użyteczności i należy go czytać jako „jest bardziej preferowany niż”. Ponadto:

|  |  |  |
| --- | --- | --- |
|  |  |  |

przy czym należy pamiętać, że jest to własność graniczna i dla skończonych *K,* nie ma gwarancji otrzymania dokładnego rozwiązania (stąd algorytm „przybliżonej optymalizacji”)[[84]](#footnote-84). Własność opisana równaniem 43 jest tłumaczona nawiązaniem do procesu adiabatycznego, którego stroteryzowaną wersję stanowi QAOA[[85]](#footnote-85).

Skutkiem zwiększenia liczby warstw (a więc troteryzacji) jest zwiększenie głębokości obwodu (*circuit depth*), czyli największej liczby bramek zastosowanych na jednym kubicie. W związku z zwiększeniem głębokości obwodu, troteryzacja skutkuje zwiększeniem się szumu (*quantum noise)*[[86]](#footnote-86)*.* Wraz ze zwiększaniem się jakości maszyn kwantowych oraz rozwojem technik korekcji błędów, należy się spodziewać, że powyższy problem będzie miał coraz mniejsze znaczenie.

Dobór parametrów oraz stopnia troteryzacji stanowi istotny element efektywności algorytmu. Im niższe (bliższe zeru) wielkości parametrów (kątów) w wektorach oraz , tym dokładniejsze przybliżenie optymalnego rozwiązania można otrzymać, ponieważ dokładniej jest przeszukiwana przestrzeń rozwiązań, przy czym zbyt mały kąt , będzie skutkował trudnościami z wyjściem z pułapki minimum lokalnego Z kolei, aby móc się przybliżyć do tego rozwiązania, potrzeba dłuższego czasu przetwarzania, czyli większego stopnia troteryzacji *K*[[87]](#footnote-87)*.* Należy więc zauważyć, że zwiększony stopień troteryzacji będzie zmniejszał dokładność (przez szum) oszacowania optymalnego rozwiązania, zmniejszenie kątów pozwoli tą dokładność zwiększyć. Równocześnie małe kąty będą wymagały większego stopnia troteryzacji, aby przeszukać odpowiednio dużą przestrzeń rozwiązań. Tak więc dobór kąta i stopnia troteryzacji stoją w opozycji nie tylko z perspektywy czasu obliczeń, ale też ilości i skali popełnianych błędów (wielkości szumu), a co za tym idzie – dokładności oszacowań. Rozwiązaniem może być zmiana stopnia troteryzacji po pewnej liczbie iteracji algorytmu. O ile jak na razie brakuje artykułów wprowadzających zmienny stopień troteryzacji, o tyle wartości parametrów i poddawano już klasycznej optymalizacji, co zostanie przybliżone w Rozdziale II.

**Podsumowanie**

W powyższym rozdziale przedstawiono pojęcia składające się na tytuł poniższej pracy. Wprowadzono i omówiono podstawowe pojęcia mechaniki kwantowej wykorzystywane w obliczeniach kwantowych oraz przedstawiono oznaczenia powszechnie wykorzystywane w tej dziedzinie, które są wykorzystywane w opisach w dalszych rozdziałach. Następnie pokazano ogólny zarys problemu VRP, zwracając uwagę na mnogość jego definicji i rozwinięć, a także podejść do sformułowań matematycznych. Szczególną uwagę poświęcono problemowi zamkniętych cykli, który jest często pomijany w publikacjach, a jego sformułowanie stanowi jedno z wyzwań rozwiązywania problemu VRP. Trzecia część tego rozdziału została poświęcona algorytmowi QAOA. Ukazano podstawową wersję tego algorytmu, a także ukazano logikę jego działania, omawiając poszczególne operacje matematyczne i bramki kwantowe wykorzystywane w tym algorytmie, odnosząc się głównie do pracy, w której po raz pierwszy przedstawiono ten algorytm, ale też zasygnalizowano możliwe modyfikacje oraz niektóre słabe punkty, którym więcej miejsca zostanie poświęcone w następnym rozdziale.

# Ścieżki rozwoju algorytmu QAOA

Celem tego rozdziału jest przedstawienie możliwych działań i kierunków rozwoju, które pozwolą na rozwój algorytmu QAOA oraz jego praktyczne zastosowanie, co będzie miało przełożenie także na zwiększenie możliwości stosowania innych algorytmów kwantowych. Pierwsza część dotyczy wyłącznie algorytmu QAOA. Druga jest poświęcona branży informatyki kwantowej, której rozwój jest warunkiem koniecznym do możliwości stosowania algorytmów kwantowych w praktyce.

## Rozwinięcie algorytmu QAOA

W pierwszej części tego rozdziału zwrócono uwagę na problemy jakie można napotkać w użytkowaniu algorytmu QAOA. Następnie przedstawiono pomysł zamiany stanu początkowego równej superpozycji na stan obciążony. Ostatnia część dotyczy klasycznej hiperoptymalizacji parametrów algorytmu QAOA.

### Złożoność QAOA

Jednym z elementów wyróżniających algorytm QAOA na tle innych kwantowych algorytmów optymalizacyjnych jest udowodniony matematycznie fakt, że nawet w najgorszym przypadku poradzi on sobie lepiej z problemem optymalizacji kombinatorycznej niż algorytm losowo wybierający rozwiązania[[88]](#footnote-88). Dla klasycznych algorytmów obliczenie wyrażenia w równaniu 25 jest zadaniem, dla którego czas obliczeń rośnie dla komputerów klasycznych w tempie podwójnie wykładniczym wraz ze wzrostem głębokości algorytmu[[89]](#footnote-89) (przekładającego się na rosnącą liczbę kątów w i ) oraz w tempie wielomianowym wraz ze wzrostem liczby wierzchołków[[90]](#footnote-90). Dla podejścia kwantowego, głębokość obwodu rośnie w tempie liniowym – powoduje to, że dla niewielkich głębokości może być możliwe przeprowadzenie niewielkich obliczeń praktycznych[[91]](#footnote-91). Przy czym warto tutaj wspomnieć o początkowym skomplikowaniu o troteryzacji równej 1 (tzn. głębokości obwodu, gdzie i są jednoelementowe), które w porównaniu do obecnych możliwości technicznych pozostaje duże. Należy zwrócić uwagę, że można policzyć klasycznie wartość oczekiwaną w równaniu 25, ale niemożliwe jest policzenie metodami klasycznymi rozkładu wyników składających się na tę wartość oczekiwaną, zakładając, że oraz, że przejście do stanów o niższym stopniu skomplikowania wielomianowego nie jest możliwe[[92]](#footnote-92). Tak więc, o ile algorytm QAOA nie może zagwarantować, że zawsze będzie dawał lepsze wyniki (przynajmniej dla małych głębokości, dla których policzenie wartości oczekiwanej jest wystarczająco proste dla znanych algorytmów klasycznych), o tyle dla dużych głębokości QAOA ma szansę działać lepiej niż klasyczne algorytmy.

Na poważne ograniczenie algorytmu QAOA zwraca uwagę Akshay et al[[93]](#footnote-93). Zwraca on uwagę na problem deficytu osiągalności (*reachability deficit)*. Polega on na tym, że jeżeli stopień troteryzacji jest zbyt mały, to optimum globalne jest całkowicie nieosiągalne, niezależnie od początkowych ustawień parametrów. Wymagany stopień troteryzacji rośnie wraz z zwiększeniem się gęstości problemu (*problem density)*, tzn. liczby ograniczeń w stosunku do liczby zmiennych. Jest to problem znacznie poważniejszy od problemu tzw. jałowego płaskowyżu (*barren plateau)*, gdzie globalne optimum jest osiągalne, chociaż dobór parametrów nie ma dużego wpływu na jego osiągnięcie, więc może on być losowy. Dowód deficytu osiągalności algorytmu QAOA został przeprowadzony dla zastosowania QAOA w algorytmie Grover’a oraz numerycznie dla problemów z grupy MAX-SAT[[94]](#footnote-94) (na których początkowy był testowany algorytm QAOA)[[95]](#footnote-95).

Warto tutaj zwrócić uwagę w artykule Akshay’a et al. na założenie o stałości stopnia troteryzacji algorytmu QAOA oraz na fakt, że niemożność osiągnięcia pewnych ekstremów lokalnych (w tym ekstremów globalnych) jest typowym wyzwaniem dla algorytmów poszukujących optimum. Dla problemów NP-trudnych o dużej złożoności sprawdzenie czy algorytm odnalazł optimum globalne jest równie trudne jak jego znalezienie, tak więc wspomniane wcześniej oskarżenie, chociaż zdaje się być poważne, nie jest krytyczne, jako że większość algorytmów optymalizacji kombinatorycznej jest podatna na pułapkę lokalnego ekstremum i posiada jedynie metody wspomagające opuszczenie tego stanu (a pod tym względem algorytm QAOA znacznie od nich nie odbiega).

Trudność policzenia wartości oczekiwanej rośnie wraz ze wzrostem stopnia troteryzacji oraz liczby ograniczeń, ale nie wraz z liczbą wierzchołków [[96]](#footnote-96). Informacja ta zdaje się przemawiać za poszukiwaniem rozwiązaniem dotyczących QAOA, które nie będą zwiększały troteryzacji algorytmu, ale pozwoli na jego skalowanie. Przyjmując założenie, że wyższy stopień troteryzacji oznacza lepsze przybliżenie wartości optymalnej, można stwierdzić, że algorytm QAOA będzie sobie względnie radził z problemami nie wymagającymi dokładnych przybliżeń, za to wymagających uwzględnienia wielu badanych punktów, co jest charakterystyczne dla problemów optymalizacji kombinatorycznej.

Dokonywano także porównań z metodami klasycznymi (szczególnie wyżarzaniem symulowanym – SA – simulated annealing) oraz wyżarzaniem kwantowym (QA – quantum annealing). Hastings wnioskuje, że dla problemów o małej złożoności algorytm QAOA (tzn. takich, dla których nie jest wymagany wysoki stopień troteryzacji) przypomina w działaniu niektóre ze znanych algorytmów klasycznych, dlatego też nie będzie on w stanie osiągnąć znaczącej przewagi[[97]](#footnote-97). Katzbrager et al. z kolei porównuje algorytm SA i QA. Zauważa, że dla problemów optymalizacyjnych, gdzie bariery pomiędzy minimami (zakładając minimalizację jako cel) są szerokie, algorytmy SA i QA radzą sobie relatywnie słabo (w szczególności algorytm QA jest wrażliwy na szerokość barier), za to QA znacznie lepiej radzi sobie z barierami wysokimi niż SA. Należy jednak zwrócić uwagę, że założono tam, iż problemy o szerokich i wąskich barierach mają porównywalne wysokości tych barier (o czym autorzy napisali wprost)[[98]](#footnote-98). Streif i Leib do tych badań dołączają algorytm QAOA, tworząc porównanie algorytmów kwantowych inspirowanych procesami adiabatycznymi – opisane wyżej przez Katzbragera et al. trudne dla SA i QA problemy optymalizacyjne są względnie łatwe do rozwiązania dla algorytmu QAOA, z zastrzeżeniem, że lokalne minima nie są stanowią satysfakcjonujących rozwiązań[[99]](#footnote-99). W innej pracy podjęli się oni wykorzystania algorytmu QAOA do zoptymalizowania schematu działania algorytmu QA, otrzymując zadowalające wyniki[[100]](#footnote-100).

### Warm-starting QAOA

Jedną z metod hiperoptymalizacji działania algorytmu QAOA jest tzw. ciepłe wystartowanie (*warm starting*). Powstał on z myślą o QAOA z niskim poziomem troteryzacji. Technika ta polega przygotowaniu stanu początkowego innego niż . W ramach tego podejścia, Hamiltonian mieszający pozostaje bez zmian, jednak zmienia się stan początkowy, przez co równanie 41 nie jest zachowane[[101]](#footnote-101).

Gradientowe metody optymalizacji są gorsze od numerycznych, szczególnie ze względu na fakt, że nawet niewielki szum ma bardzo duży wpływ na wyniki dla metod gradientowych[[102]](#footnote-102). W wyjściowej wersji WS-QAOA (warm-starting QAOA) system jest na początku iteracji w stanie[[103]](#footnote-103):

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

tzn. kubity zmieniają w swojej reprezentacji matematycznej obrócą się w stronę stanu o kąt nie większy, niż robi to bramka Hadamarda. Spowoduje to, że nowy stan początkowy systemu nie będzie już reprezentowany przez wektor będący wektorem własnym Hamiltonianu mieszającego, dlatego należy też zmienić jego postać:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie przyjmuje za wektor własny . Modyfikując równanie 40 otrzymamy:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Po takim przekształceniu, gdzie stan początkowy jest wektorem własnym , równanie 44 pozostaje w mocy, jeżeli , a więc algorytm WS-QAOA będzie zbiegał do optymalnego rozwiązania. Jeżeli , wówczas kubit nie zostanie obrócony lub przejdzie do stanu . Będzie to skutkowało pozostaniem danego kubitu nienaruszonego przez cały czas trwania iteracji algorytmu, a więc nie będzie możliwe zbliżenie się do rozwiązania odpowiadającego przeciwległego stanowi tego kubitu (czyli jeżeli kubit rozpocznie w stanie , a optimum będzie odpowiadać temu kubitowi w stanie (i vice versa), to nie będzie możliwe osiągniecie tego stanu)[[104]](#footnote-104).

Aby zapobiec wybraniu , można posłużyć się parametrem regulującym . Wówczas otrzymamy:

* ,
* ,
* .

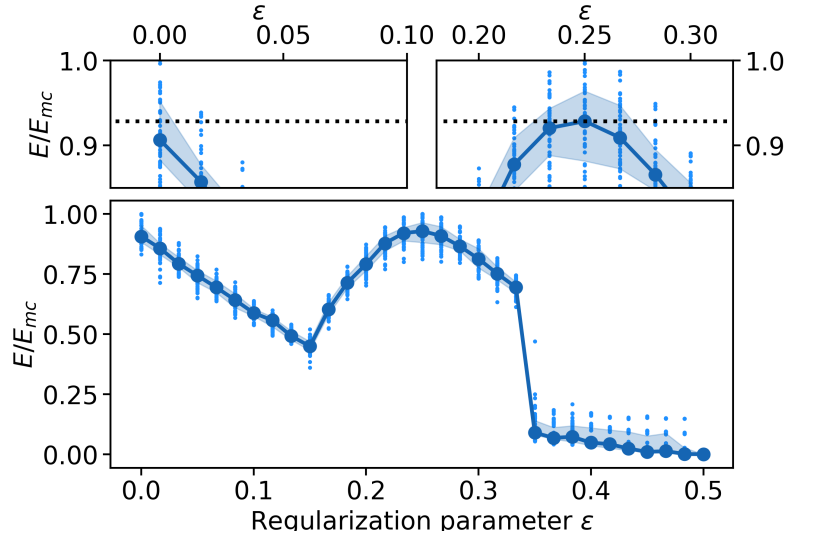
Następnie należy odpowiednio zaktualizować , oraz. Jeżeli , wówczas zostanie wymuszony na wszystkich kubitach stan , będzie to więc odpowiadało standardowemu rozpoczęciu algorytmu QAOA.

Egger et al.[[105]](#footnote-105) dostrzegli możliwość dodatkowej modyfikacji macierzy mieszającej. Zauważyli oni, że bramkę bazującą na macierzy mieszającej można zmodyfikować do następującej postaci:

|  |  |  |
| --- | --- | --- |
|  |  |  |

tzn. zmienią się znaki kątów; wówczas nastąpi kilka zmian. Po pierwsze, nowa macierz nie będzie posiadała wektora początkowego jako wektora własnego, co uniemożliwi zachowanie własności asymptotycznej zbieżności do optymalnego rozwiązania w równaniu 44[[106]](#footnote-106). Z drugiej strony, wyniki numeryczne pokazały, że przy wartości parametru oraz troteryzacji na poziomie 1 algorytm znacznie szybciej odnajduje coraz lepsze rozwiązania, co nie zachodzi przy poprzedniej wersji bramki . Sytuację tę przedstawia rysunek 4 – najszybsza zbieżność do optimum (tutaj: maksimum) globalnego jest osiągana dla . Eksperyment był powtarzany wiele razy, grube punkty wskazują medianę, zacieniowany obszar – rozwiązania leżące między pierwszym a trzecim kwartylem[[107]](#footnote-107). Widać, że zależność pomiędzy tempem zbieżności a jest niemonotoniczna, w przeciwieństwie do podejścia, w którym stan początkowy jest wektorem własnym . Należy jednak zwrócić uwagę, że autorzy przedstawili jedynie wynik numeryczny, a więc wynik manipulacji oraz mógł okazać się pozytywny wyłącznie ze względu na dobrany problem. Autorzy dostrzegli również, że algorytm WS‑QAOA zbiega do optimum szybciej niż zwykły algorytm QAOA.

Rysunek . Wartość rozwiązania po jednej iteracji WS-QAOA z parametrem .



**Źródło:** D. J. Egger, J. Mareček, S. Woerner, *Warm-starting quantum optimization,* Quantum 5, 479 (2021), s. 8.

Warto wspomnieć jedną z prac, w której podjęto się badań numerycznych wpływu hiperparametrów WS-QAOA – praca J. Obsta, gdzie numerycznie wykazał, że im bliższe , tym gwałtowniej i częściej zmieniają się wyniki przy coraz to mniejszych zmianach parametrów i (przy troteryzacji pojedynczej)[[108]](#footnote-108).

### Hiperoptymalizacja Bayesowska

Jedną z proponowanych metod poszukiwania optymalnego zestawu parametrów i jest zastosowanie podejścia Bayesowskiego. Jest ono szczególnie polecane dla problemów o trudnej w ewaluacji funkcji celu[[109]](#footnote-109). Podejście Bayesowskie opiera się na doborze parametrów bazując na pewnej wiedzy, zwanej wiedzą *a priori* oraz pewnych danych liczbowych (np. próby), w rezultacie jako wynik zostaną otrzymane rozkłady *a posteriori* parametrów. Należy zwrócić uwagę, że poszukiwane parametry są tutaj traktowane jako zmienne losowe, zwracana więc będzie wartość oczekiwana tych parametrów, a pozostałe informacje na temat ich rozkładu będzie można traktować jako dane pomocnicze lub diagnostyczne.

Argumentem (poza deklarowanym wzrostem szybkości przetwarzania i dokładności szacowania) przemawiającym za tą metodą jest to, że niepewność jest wykorzystywana i uwzględniania w metodach Bayesowkich, a w algorytmach kwantowych mamy obecnie dwa rodzaje niepewności: pierwszy, związany z dokonywaniem pomiaru oraz drugi, związany z szumem kwantowym. Innym argumentem jest też relatywnie niska liczba potrzebnych iteracji części kwantowej, aby móc dokonać hiperoptymalizacji metodami Bayesowskimi. Ponadto, dla problemów, gdzie istotne jest intensywne przeszukiwanie bliskiego sąsiedztwa (*local problems*), metody te radzą sobie relatywnie dobrze[[110]](#footnote-110) (warto zaznaczyć, że problem VRP akurat do tych problemów się nie zalicza).

Informacją liczbową w optymalizacji Bayesowskiej może być kilka oszacowań optymalizowanej funkcji w losowych (co do rozkładu jednostajnego) punktach dziedziny określonej przez możliwe zbiory wartości parametrów przyjmowanych za zmienne. Wartości parametrów (oznaczane **,** gdzie *i* to indeks danego zestawu parametrów wykorzystywanych w danym oszacowaniu) oraz odpowiadające im oszacowania wartości funkcji tworzą zbiór ⅅ[[111]](#footnote-111). Jednak, aby uniknąć błądzenia losowego po przestrzeni rozwiązań oraz wielokrotnego obliczania wartości skomplikowanej funkcji, można optymalizować funkcję użyteczności[[112]](#footnote-112) (zwaną dalej funkcją akwizycji).

Jednym z problemów jest wyznaczenie używanej w optymalizacji Bayesowskiej funkcji akwizycji (*acquisition function*) bazującej na funkcji oczekiwanej poprawy (*expected improvement*)[[113]](#footnote-113). Funkcja ta korzysta z rozkładu *a posteriori* oraz pewnej wartości będącej największą znalezioną wartością optymalizowanej funkcji w ramach iteracji algorytmu. Funkcja poprawy jest zdefiniowana jako:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Gdzie:

– postać funkcji a posteriori,

– największa znaleziona wartość funkcji *f*,

*n* – numer iteracji,

czyli przyjmuje wartość 0, jeśli w nowym punkcie *x* wartość optymalizowanej funkcji a posteriori jest niższa niż dotychczas znalezionej największej wartości lub jest równie różnicy między tymi dwiema wartościami, w przypadku znalezienia wartości większej. Problemem jest jednak znalezienie takiego *x*, aby wzrost był jak największy. Jako krok pośredni można dokonać maksymalizacji oczekiwanej wartości poprawy. Korzystając z wzoru na obliczenie wartości oczekiwanej przez całkowanie, tj.:

|  |  |  |
| --- | --- | --- |
|  |  |  |

można wyznaczyć funkcję oczekiwanej poprawy postaci (korzystając z całkowania przez części)[[114]](#footnote-114) [[115]](#footnote-115):

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie:

- oczekiwana różnica w jakości pomiędzy proponowanym punktem *x*, a poprzednim najlepszym wynikiem ,

– moduł z ,

– wartość funkcji gęstości rozkładu normalnego w zadanym punkcie,

– wartość dystrybuanty rozkładu normalnego w zadanym punkcie,

– odchylenie standardowe a posteriori.

Na koniec wyznaczany jest punkt:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie:

– wyznaczony punkt

– sprawdzany punkt.

Rozwiązanie równania 54 jest nieskomplikowane, bowiem funkcje w równaniu 53 są łatwo-różniczkowalne, a maksimum funkcji można wyznaczyć analitycznie wykorzystując drugą pochodną lub (bardziej praktycznie z perspektywy obliczeń komputerowych) korzystając np. z algorytmu quasi-Newtonowskiego[[116]](#footnote-116). Podejście takie stanowi trzecią wyjście w problemie kompromisu pomiędzy zachłannością a losowością (*exploration vs exploitation tradeoff*)[[117]](#footnote-117).

Przyjęcie rozkładu *a priori* bazuje na przyjęciu pewnych założeń dotyczących rozkład. Konstrukcja rozkładu *a priori* stanowi również najbardziej złożony element hiperoptymalizacji metodą Bayesowską. Tibaldi et al. proponuje, aby traktować funkcję celu jako zmienną losową, ponieważ nie znamy jej dokładnej postaci[[118]](#footnote-118). Konkretnie, proponuje, aby uznać tę funkcję za część procesu losowego Gaussowskiego, czyli przyjąć, że łączny rozkład zmiennych ma wielowymiarowy rozkład normalny określony parametrami postaci:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

gdzie to średnia wartość funkcji , a to kowariancja (lub jądro) funkcji . Wartość średnia odzwierciedla posiadane (także w ramach założeń) informacje o funkcji , podczas gdy funkcja jądrowa odzwierciedla niepewność posiadanych informacji[[119]](#footnote-119) (w sposób analogiczny do wariancji w jednowymiarowym rozkładzie normalnym, przy czym należy również uwzględnić kowariancję). Jeżeli wartości w są względnie niskie, oznacza to wysoką pewność co do tego, że przyjęta przez nas postać funkcji *f* jest bliska prawdziwej.

Funkcja jądrowa musi być również podana przez użytkownika. Tibaldi et al. proponuje przyjęcie funkcji jądrowej Matérna postaci:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie oraz to hiperparametry charakteryzujące proces Gaussowski. Hiperparametr określa wariancję zmiennych losowych, a to czynnik kontrolujący korelację między zmiennymi. Granicznie, , wszystkie punkty są ze sobą skorelowane w tym samym stopniu, a dla wszystkie punkty są ze sobą nieskorelowane. W wyniku obliczenia wartości funkcji dla każdej pary punktów ze zbioru (wektora) otrzymamy macierz wariancji-kowariancji 𝕂.[[120]](#footnote-120) Wartości hiperparametrów oraz są wyznaczane przy pomocy zbioru ⅅ.

Hiperparametry jądra są też zwane hiperparametrami *a* *priori[[121]](#footnote-121).* Frazier proponuje trzy metody ich wyznaczania. Pierwsza, to wyznaczenie hiperparametrów metodą największej wiarygodności (*maximum likelihood estimator –* MLE), tzn., jeśli przyjąć za zbiór hiperparametrów, wówczas należy wyznaczyć takie, że:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie ma wielowymiarowy rozkład normalny[[122]](#footnote-122).

Drugie podejście jest podobne do pierwszego, ale wykorzystuje nie rozkład *a priori*, lecz *a posteriori* (*maximum a posteriori* – MAP). Dokonuje się tego przez wykorzystanie wzoru Bayesa (gdzie mianownik wzoru Bayesa można zignorować, ponieważ nie zawiera on elementów zależnych od zmiennych i pełni on rolę czynnika skalującego tym samym nie wpływając na ) i otrzymując wzór postaci:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Przewaga MAP nad MLE ujawnia się, gdy funkcja *f* zmienia swoje wartości zbyt szybko lub zbyt wolno[[123]](#footnote-123).

Trzecia metoda opiera się w pełni na podejściu Bayesowskim (*fully Bayesian approach*). W podejściu tym wyznacza się rozkład *a posteriori* funkcji *f* poprzez obliczenie wartości brzegowych wiarygodności po wszystkich hiperparametrach:

|  |  |  |
| --- | --- | --- |
|  |  |  |

co można przybliżyć wybierając *J* punktów (np. metodą MCMC)[[124]](#footnote-124):

|  |  |  |
| --- | --- | --- |
|  |  |  |

Rozkład *a posteriori* jest wyznaczany jako funkcja określona rozkładem *a priori* warunkowa na wartościach pochodzącymi ze zbioru ⅅ. Otrzymany rozkład *a posteriori* jest nadal wielowymiarowy i przyjmuje parametry[[125]](#footnote-125):

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

gdzie to wektor kolumnowy określający kowariancję funkcji *k* w punkcie a wszystkimi pozostałym punktami. Warto zauważyć, że nowa średnia jest liniową kombinacją oszacowań funkcji, tj. elementów wektora .

W ramach algorytmu optymalizacji Bayesowskiej należy przyjąć pewną liczbę iteracji . Każda z iteracji rozpoczyna się od zaktualizowania rozkładu *a posteriori f*poprzez aktualizację raz przyjętego rozkładu *a priori* danymi ze zbioru ⅅ. Na podstawie zaktualizowanego rozkładu *a posteriori* należy obliczyć funkcję akwizycji oraz znaleźć wektor parametrów maksymalizujący wartość tej funkcji. Następnie należy obliczyć wartość funkcji dla znalezionego wektora, tj. obliczyć Jeżeli , wówczas należy zaktualizować (w przeciwnym przypadku należy nie robić nic). Niezależnie od wyniku, należy dołączyć parę do zbioru ⅅ. Iterację należy zakończyć obliczeniem nowego zestawu hiperparametrów funkcji jądrowej i , po czym rozpocząć kolejną iterację do czasu osiągnięcia iteracji o numerze [[126]](#footnote-126) Znaleziony w ten sposób wektor *2K*‑elementowy (gdzie *K* to stopień troteryzacji) zawiera zoptymalizowane metodą bayesowską hiperparametry i algorytmu QAOA.

Na koniec omówienia tego podrozdziału warto przyjrzeć się kilku założeniom poczynionym przez autora zastosowania tej metody w QAOA. Pierwsze z nich dotyczy uznania rozkładu za wielowymiarowy rozkład normalny. Ponieważ nie znamy postaci założenie to jest dość arbitralne, jednak pozwala ono wykorzystać metodę MCMC w losowaniu punktów. Zakładając bardzo wysoką wariancję (co jest typowym założeniem, jeśli postać funkcji nie jest znana), można przyjąć to założenie za nieszkodliwe.

Drugie to przyjęcie funkcji jądrowej Matérna. Można je usprawiedliwić tym, że w przeciwieństwie do często przyjmowanej funkcji jądrowej kwadratu wykładniczego, funkcja jądrowa Matérna ma bardziej nieregularny kształt, co lepiej odzwierciedla złożoność rzeczywistych problemów[[127]](#footnote-127).

## Stan badań i aspekt komercyjny

Podrozdział ten służy ukazaniu, w jaki sposób podmioty zainteresowane rozwiązaniami proponowanymi przez informatykę kwantową zapatrują się na rozwój tej dziedziny. Powyższe zostanie przedstawione zarówno z perspektywy producentów jak i odbiorców rozwiązań. Szczególna uwaga zostanie poświęcona optymalizacji z wykorzystaniem algorytmów kwantowych (nie tylko QAOA, lecz także wyżarzaniu kwantowemu) – większość źródeł będą stanowiły doniesienia medialne, dlatego należy je traktować z rezerwą. Celem przywołania tych źródeł jest ukazanie, że informatyka kwantowa, chociaż jest dziedziną młodą, to wzbudza zainteresowanie rozmaitych grup. Należy równocześnie pamiętać, że okazywany entuzjazm może być częścią kreowania marki firmy (szczególnie z perspektywy dostawców rozwiązań), jednak z drugiej strony firmy (także odbiorcy) są przede wszystkim podmiotami nastawionymi na zysk – bezpiecznie jest przyjąć zatem założenie, że inwestycja lub testowanie rozwiązań kwantowych jest opłacalne długoterminowo dla niektórych podmiotów.

### Działalność podmiotów komercyjnych

DOKOŃCZYĆ – POCZEKAC NA RAPORT ZA 2022

Do największych graczy na rynku technologii kwantowych zaliczają się giganci technologiczni o ugruntowanej pozycji na rynku, w szczególności IBM, Microsoft, Google oraz Amazon – każdy z tych graczy dysponuje dużymi udziałami na szeroko pojętym rynku informatycznym, jednak tylko IBM może się pochwalić działalnością badawczo-rozwojową, która zaowocowała przyjęciem rozwiązań na poziomie sprzętu (hardware’u) na poziomie globalnym – pozostali gracze skupiają się raczej na rozwijaniu oprogramowania lub działalności XaaS (Anything as a Service – zdalne udostępnianie zasobów).

Warto również wspomnieć, że IBM jako jedyny umożliwia bezpłatny dostęp indywidualnym osobom do infrastruktury kwantowej. Robi to za pomocą swojej platformy IBM Quantum. IBM stoi również za pakietem programistycznym (SDK – Software Development Kit) Qiskit o otwartym kodzie źródłowym udostępnianym w formie bibliotek do języka Python, umożliwiając prowadzenie symulacji kwantowych na maszynach klasycznych[[128]](#footnote-128).

W zależności od źródła, można znaleźć informacje o wartości rynku technologii kwantowych w 2021 zawierającym się w przedziale 400-600 mln USD rocznie. Różnie podaje się również spodziewane tempo wzrostu rok do roku, na ogół zawierające się w przedziale 25‑55% rocznie[[129]](#footnote-129) [[130]](#footnote-130). Różnica w tych szacunkach wynika z różnego potraktowania gwałtownego wzrostu wartości rynku technologii kwantowych i inwestycjami w te technologie w czasie pandemii COVID-19[[131]](#footnote-131) i trudnościami w wyizolowaniu faktycznego trendu od chwilowych wzrostów spowodowanych inwestycjami w technologie informatyczne w trakcie pandemii, przyspieszenia transformacji cyfrowej, a także zawirowań gospodarczych, skutkujących spowolnieniem przyrostu lub spadkiem PKB w czasie pandemii lub w okresie po niej, kiedy pisana jest ta praca[[132]](#footnote-132).

Głównym ograniczeniem stojącym obecnie na drodze do rozwoju technologii informatyki kwantowej są ograniczenia związane z samym sprzętem – problem zarówno dotyczy wielkości (możliwości obliczeniowych) obecnych maszyn kwantowych jak i ich jakości. Największa maszyna kwantowa, którą można nazwać procesorem kwantowym ogólnego zastosowania liczy (pod koniec 2022 roku) 433 kubitów – rekord ten należy do komputera Osprey (pl. rybołów) firmy IBM[[133]](#footnote-133). Zgodnie z zapowiedziami tej firmy, w 2023 ma zostać osiągnięty próg 1000 kubitów, co należy uznać za realny plan, zważywszy na fakt, że zgodnie z planami czynionymi od 2019, dotychczasowy postulat rokrocznego zwiększania liczby kubitów około 2,5-krotnie rocznie, był zawsze spełniany[[134]](#footnote-134). Należy zaznaczyć, że samo zwiększanie liczby kubitów będzie bezcelowe, jeśli nie będą one odpowiedniej jakości, tzn. sposób ich konstrukcji oraz struktura układu powinna zapewnić jego odpowiednią stabilność zapobiegając m. in. dekoherencji, a szerzej – tzw. błędom maszynowym. Zwiększanie liczby kubitów należy uznać za szczególnie istotne dla możliwości algorytmu QAOA.

Pod zwiększeniem jakości można też rozumieć zwiększenie liczby operacji na sekundę (CLOPS - Circuit Layer Operations Per Second – liczba operacji na poziomie obwodu na sekundę – miara analogiczna do klasycznych FLOPS-ów, przy czym należy pamiętać, że pojedyncza operacja na komputerze kwantowym (CLO) będzie wolniejsza, ale niosła ze sobą więcej informacji, niż klasyczna (FLO)). Liczba CLOPS-ów również ulega stopniowemu zwiększaniu, w tempie nadproporcjonalnym do zwiększania liczby kubitów – jest to możliwe nie tylko dzięki lepszej jakości sprzętowi, lecz także dzięki wspomnianemu wcześniej zjawisku splątania kwantowego, które dzięki lepszej infrastrukturze może być intensywniej eksploatowane przy zachowaniu zadowalającej stabilności. Miara ta została wprowadzona przez IBM, chcąc wyznaczyć podstawową metodologię określania benchmarków, ustaleniu kierunku rozwoju, a także pokazać, że rozwój maszyn kwantowych ma rzeczywiście miejsce i jest mierzalnym zjawiskiem, a nie jedynie szumnym hasłem marketingowym[[135]](#footnote-135). Można się też tutaj doszukiwać prób ugruntowania pozycji lidera na rynku poprzez ogłaszanie pewnych miar standardem. W ten sposób IBM określił 3 podstawowe (i współzależne) miary mierzące zdolność procesora kwantowego do dokonywania obliczeń[[136]](#footnote-136):

* wielkość (scale) - liczba kubitów,
* szybkość (speed) - liczba CLOPS-ów – wprowadzone przez IBM w 2021,
* jakość (quality) – rozumiana jako maksymalna liczba kubitów, która może być naraz w użyciu (Quantum Volume – objętość kwantowa), zależne od liczby kubitów oraz skali popełnianych błędów, przy czym szczegółowy sposób obliczania ulegał zmianie[[137]](#footnote-137) [[138]](#footnote-138) [[139]](#footnote-139). Metryka ta jest kluczowa, aby określić czy algorytm implementowany przez obwód o określonej złożoności (rozumianej przez głębokość – *circuit depth*) może zostać wykorzystany na danej maszynie[[140]](#footnote-140).

O ile rozwój sprzętu jest główną domeną gigantów technologicznych, o tyle w rozwoju oprogramowania dla obliczeń kwantowych przodują start-upy – 68% start-upów założonych w sektorze technologii kwantowych w 2018 roku było związanych z rozwojem oprogramowania[[141]](#footnote-141). Z racji niskiej jakości i wielkości obecnych maszyn kwantowych, za główny obszar działalności producentów oprogramowania do obliczeń kwantowych można uznać dwa obszary:

* dotyczący tworzenia oprogramowania umożliwiającego przeprowadzanie symulacji obliczeń kwantowych na maszynach klasycznych (m.in. biblioteki do języka Python takie jak Qiskit od IBM lub Cirq od Google). Najpopularniejsze z nich są tworzone przez duże firmy;
* dotyczący optymalizacji algorytmów kwantowych (np. FireOpal od Q-Ctrl) – główną motywacją powstawania takiego oprogramowania jest przezwyciężenie problemów związanych z niską jakością maszyn kwantowych poprzez dodawanie do algorytmów operacji mających na celu zmniejszenie skali błędów maszynowych, automatyczną optymalizację kodu. To ta część została zdominowana przez start-upy.

FIRE OPAL

Przy okazji wspomnienia o oprogramowaniu, warto zwrócić uwagę na rosnącą liczbę języków, które wspierają symulacje kwantowe. Dla wspomnianego wcześniej Pythona zostały utworzone m.in. pakiety Qiskit od IBM, pyQuil od Rigetti Computing[[142]](#footnote-142) (firma zajmująca się budowaniem i udostępnianiem dostępu do obliczeń na maszynach kwantowych za pośrednictwem chmury)[[143]](#footnote-143), PennyLane od Xanadu (firma o podobnym profilu do Rigetti Computing[[144]](#footnote-144)) Cirq od Google[[145]](#footnote-145), jednak takie pakiety są tworzone także dla produktów komercyjnych takich jak Matlab od firmy MathWorks, dla którego pakiet został opublikowany w 2023 roku[[146]](#footnote-146). Istnieją także języki dedykowane symulacjom kwantowym jak Q# od Microsoftu.[[147]](#footnote-147) Wiele z tych firm oferuje także usługi chmurowe w zakresie obliczeń kwantowych; poza wyżej wymienionymi warto wymienić także usługę Amazon Braket[[148]](#footnote-148) oraz bezpłatną platformę IBM Quantum, umożliwiającą bezpłatny dostęp do niewielkich maszyn kwantowych[[149]](#footnote-149).

### Inwestycje publiczne

Na uwagę w Polsce zasługuje inwestycja ze strony Komisji Europejskiej w ramach projektu European High-Performance Computing Joint Undertaking (EuroHPC JU). Inicjatywa ta, z finansowaniem w wysokości ponad 100 milionów euro ma na celu zbudowanie pierwszego w Unii Europejskiej komputera kwantowego oraz dołączenie do technologicznego wyścigu – jednym z sześciu krajów zaangażowanych w tę inicjatywę jest Polska. Projekt ten jest częścią 10-letniego, trwającego od 2018 programu Quantum Technologies Flagship, w ramach którego UE ma wydać 10 miliardów euro na działalność badawczo-rozwojową w zakresie technologii kwantowych[[150]](#footnote-150).

Wśród pierwszych osiągnięć Quantum Technologies Flagship wymienia się: zbudowanie komputera kwantowego konkurencyjnej jakości, wykonanie początkowych testów sprawdzających, czy jest możliwe powstanie kwantowej sieci internetowej, zbudowanie co najmniej 100-kubitowego urządzenia kwantowego do celów symulacji. Równocześnie w ramach inicjatywy do końca 2021 sfinansowano powstanie 25 start-upów w obszarze technologii kwantowych, udało się również przyznać 64 patenty oraz opublikować ponad 1300 artykułów naukowych[[151]](#footnote-151).

### Implementacje

DWAVE Australia?

## Optymalizacja logistyczna z wykorzystaniem algorytmów kwantowych

Rozdział ten stanowi przegląd rozwiązań zaproponowanych w pracach naukowych w zakresie wykorzystania technologii kwantowych w optymalizacji logistycznej, w szczególności dla problemu marszrutyzacji. Zostaną przedstawione nie tylko wykorzystania algorytmu QAOA, ale też wyżarzania kwantowego – wynika to z chęci wyrażenia problemu VRP w formie QUBO.

Jako pierwszy zostanie ponownie przywołany artykuł Salehi et al. dotyczący problemu VRP-TW. W równaniu 24 zaproponowano sformułowanie warunku na brak zamkniętych cykli. Sformułowanie to jednak nie nadaje się do sformułowania w postaci QUBO, ponieważ pojawiają się tam czwarte potęgi (dokładnie: iloczyn kwadratów dwóch zmiennych). Aby to zmienić należy przekształcić składniki równania 24 w następujący sposób:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Przekształcenie w równaniu 64 tłumaczy się następująco: dla wierzchołka *j*, wartość jest równa 1 dokładnie dla jednego wierzchołka docelowego *k*. Z racji tego, że wyrażenie w sumie może być równe wyłącznie 0 lub 1, podnoszenie do drugiej potęgi nie jest potrzebne. W ten sposób równanie 24 można przekształcić do następującej postaci:

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Podsumowanie**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Implementacja

Poniższy rozdział stanowi opis serii eksperymentów mających na celu sprawdzenie czy zastosowanie algorytmu QAOA z Bayesowką hiperoptymalizacją parametrów stanowi opłacalną lub możliwą metodę rozwiązywania problemu komiwojażera. Pierwsza część stanowi opis przygotowania do eksperymentów w postaci przedstawienia grafów, na których zostanie dokonana optymalizacja, opis infrastruktury oraz algorytmów służących za podstawę do oceny algorytmu QAOA względem innych metod (tzw. benchmark).

## Przygotowanie eksperymentu

### Infrastruktura, oprogramowanie i algorytmy

Do obliczeń (zarówno klasycznych i jak kwantowych) wykorzystano język Python platformę IBM Quantum Lab. Obecne tam środowisko Jupyter Lab umożliwia pisanie kodu w języku Python w edytorze Jupyter Notebook. Dzięki zastosowaniu platformy IBM Quantum Lab użytkownik ma dostęp do środowiska z zainstalowanymi niezbędnymi pakietami, w szczególności z stabilnymi wersjami modułu qiskit i pokrewnych (szczegóły dotyczące użytych pakietów zawiera **Tabela),**, co w chwili obecnej jest istotne ze względu na brak wersji v. 1.0.0. – funkcje w modułach są między sobą powiązane, a zmiana w wersji jednego modułu może spowodować problemy z działaniem innego modułu. Do przechowywania postaci matematycznej problemu wykorzystano obiekt *TSP* z pakietu qiskit-optimization, jednak wymagał on wprowadzenia modyfikacji (opisanych w rozdziale III.1.3). Pakiety quimb pozwalał na zintegrowanie algorytmu QAOA z algorytmem hiperoptymalizacji Bayesowskiej wdrożonej przy pomocy scikit-optimize. Pakiety networkx oraz cotengra służyły do tworzenia struktur grafowych odpowiednio dla pakietów qiskit oraz quimb. Do wizualizacji wykorzystano matplotlib oraz pandas, pakiety scipy oraz numpy pozwoliły na wdrożenie innych, klasycznych optymalizatorów, opisanych dalej.

Tabela . Wersje oprogramowania użyte w czasie eksperymentów.

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa oprogramowania / modułu | Wersja | Nazwa oprogramowania / modułu | Wersja |
| Python | 3.10.8 | numpy | 1.23.5 |
| qiskit | 0.43.1 | pandas | 1.5.2 |
| qiskit-optimization | 0.5.0 | networkx | 2.8.8 |
| qiskit-aer | 0.12.0 | matplotlib | 3.6.3 |
| cotengra | 0.2.0 | scikit-optimize (skopt) | 0.9.0 |
| quimb | 1.5.0 | scipy | 1.9.3 |

**Źródło**: opracowanie własne

Do obliczeń kwantowych wykorzystano system *ibm\_jakarta* o 7 kubitach, objętości kwantowej 16, 2400 CLOPS-ach, bazujący na procesorze kwantowym Falcon r5.11H. Falcon był to pierwszy procesor kwantowy od którego IBM rozpoczął plan podwajania liczby kubitów co rok – pierwsza wersja procesora Falcon ukazała się w 2019 roku – obecnie do bezpłatnego użytku są dostępne tylko te procesory. r5.11 stanowi określenie wersji procesora, a ostatnia litera określa kształt układu tworzony przez połączenia kubitów w procesorze (w tym przypadku są one ułożone w kształt litery H), tworząc w ten sposób segment procesora. Bramki, które można stosować z użyciem tego procesora to CX, I, Rz, SX, X, dlatego też konieczna będzie dekompozycja stosowanych bramek do zestawu powyższych[[152]](#footnote-152).

W ramach eksperymentów porównawczych zostaną wykorzystane 2 algorytmy klasyczne. Pierwszy z nich to algorytm wyszukiwania wyczerpującego (*brute force*), dla którego ograniczenia zostaną zaimplementowane bez rozluźniania – zagwarantuje to wyszukanie optymalnego rozwiązania, co jest możliwe także dzięki niewielkiej skali problemów.

Drugi algorytm to NumPyMinimumEigensolver będący klasycznym algorytmem zaimplementowanym w bibliotece qiskit, ale zbudowanym na podstawie implementacji w bibliotece numpy. Algorytm ten szuka wektorów własnych macierzy problemu, tzn. Hamiltonianu kosztu uwzględniającego ograniczenia w postaci zrelaksowanej. NumPyMinimumEigensolver bazuje na procedurze geevz pakietu LAPACE zaimplementowanego w języku Fortran (z którego korzysta biblioteka numpy). Procedura geev polega na przekształceniu macierzy wejściowej do postaci macierzy Hessenberga metodą Householdera, a następnie wielokrotnej dekompozycji tej macierzy metodą QR do czasu osiągnięcia postaci macierzy zadowalająco bliskiej macierzy trójkątnej – w rezultacie otrzyma się przekształcenie pozwalające łatwo określić wektory i wartości własne macierzy wejściowej[[153]](#footnote-153).

Trzeci algorytm to symulacja działania algorytmu QAOA z wykorzystaniem maszyny klasycznej oraz obiektu klasy QAOA dostępnego w pakiecie qiskit. Użytkownik przekazuje do obiektu problem w postaci QUBO. Można także dobrać rodzaj optymalizatora części klasycznej algorytmu QAOA. Na potrzeby eksperymentu zostanie wykorzystany optymalizator rekomendowany przez qiskit w dokumentacji, tj. optymalizator numeryczny COBYLA. Iteracje algorytmu QAOA były wykonywane do czasu osiągnięcia zbieżności. Do wad używania obiektu QAOA należą: niewielka możliwość jego modyfikacji (w szczególności duże restrykcje co do możliwości zastosowania optymalizatora klasycznego), niewielka ilość danych, które można wyciągnąć z procesu optymalizacji, w szczególności brak możliwości otrzymania bramek kwantowych implementujących algorytm QAOA[[154]](#footnote-154).

Kluczowym eksperymentem jest przeprowadzony na własnoręcznie napisanym kodzie implementującym QAOA. Zdefiniowano m.in. funkcje tworzące bramki kwantowe, implementujące je symulacje, (klasyczny) optymalizator Bayesowski do hiperoptymalizacji klasycznej oraz funkcja scalająca część kwantową algorytmu QAOA z klasyczną hiperoptymalizacją. W ramach przygotowań najpierw przeprowadzono klasyczne symulacje, które również zostaną omówione. W ramach klasycznych symulacji zezwolono na maksymalnie 20 iteracji algorytmu QAOA oraz po 2 iteracje iteracje na każdy etap hiperoptymalizacji. Proces mógł być przedwcześnie przerwany, jeżeli 5 razy z rzędu otrzymano identyczny wynik

Ostatnim eksperymentem jest uruchomienie algorytmu QAOA na maszynie kwantowej – brak bezpłatnego dostępu do maszyn o więcej niż 7 kubitach stanowił jedną z barier tego przedsięwzięcia, nie bez znaczenia były również długie czasy oczekiwania na wykonanie pojedynczego programu ze względu na kolejkę oczekujących (czas oczekiwania na wykonanie jednej iteracji jest liczony w tygodniach). Wprawdzie narzędzie Qiskit Runtime (ogłoszone jesienią 2022 roku razem z komputerem Osprey) umożliwia implementację algorytmu QAOA (i innych algorytmów wariancyjnych wymagających środowiska hybrydowego w każdej iteracji, jednak algorytm QAOA stanowił główną motywację stojącą za wprowadzeniem tego narzędzia) na maszynie kwantowej przy wsparciu maszyn klasycznych, jest to jednak rozwiązanie płatne[[155]](#footnote-155).

### Dane i sformułowanie problemu

Ze względu na ograniczenia w zakresie dostępnej infrastruktury kwantowej, analizowane problemy będą niewielkie. Z tego powodu przygotowano cztery grafy – dwa 3-wierzchołkowe (oznaczane T1 oraz T2) oraz dwa 4-wierzchołkowe (oznaczane F1 oraz F2). W jednej parze grafów (para T1 oraz F1) odległości pomiędzy wierzchołkami grafów są identyczne – posłużą one za test warunków skrajnych. Warto zwrócić uwagę, że dla ewentualne naruszenie twierdzenie o naruszeniu nierówności trójkąta dla zestawu F1, ponieważ czas przejazdu (uznawany tutaj za odległość) nie musi odpowiadać długości metrycznej trasy, a ta w rzeczywistości również nie zawsze będzie miała postać odcinka. Pozostałe dwa zestawy (T2 oraz F2) stanowią problem oparty o dane podane przez Główny Urząd Statystyczny (GUS) i dotyczy problemu przejazdu pomiędzy czterema najludniejszymi miastami w Polsce. Dane na temat czasu przejazdu pomiędzy centroidami miast zostały opublikowane przez GUS w 2019 roku, skąd też pobrano dane dotyczące odległości między Warszawą, Krakowem, Wrocławiem i Łodzią.[[156]](#footnote-156) Macierze odległości dla poszczególnych zestawów przedstawiono w **TABELA**

Tabela . Macierze odległości dla grafów używanych w eksperymentach.

|  |  |  |  |
| --- | --- | --- | --- |
| Numer zestawu | Macierz odległości | Numer zestawu | Macierz odległości |
| T1 |  | F1 |  |
| T2 |  | F2 |  |

**Źródło**: opracowanie własne, wykorzystano dane Głównego Urzędu Statystycznego: https://stat.gov.pl/statystyki-eksperymentalne/obszary-funkcjonalne-oraz-dostepnosc-terytorialna/oszacowanie-odleglosci-i-czasu-przejazdu-pomiedzy-wybranymi-gminami-w-polsce-w-2016-roku,11,1.html.

Przyjęte zmienne decyzyjne są zmiennymi binarnymi postaci , , gdzie oznacza, że połączenie (krawędź) jest aktywne, a , że krawędź jest nieaktywna, tzn. nie jest częścią rozwiązania. Takie zdefiniowanie zmiennych binarnych pozwala na rozwiązanie problemu VRP dla grafu nieskierowanego i wymaga mniej mocy obliczeniowej. Ponadto nie konstruowano zmiennych binarnych dla tożsamych wierzchołków – w rozwiązywanym przypadku nie zachodzi potrzeba pozostawania w wierzchołku lub przebywania zerowego dystansu przez agenta. Powyższe zabiegi pozwoliły na zmniejszenie skali problemu zdefiniowanego w domyślnej wersji obiektu *TSP*. W rezultacie zmniejszono liczbę warunków ograniczających określanych przez równania 18 i 19. Należało jednak dodać warunek ograniczający odpowiadający za warunek zakazu istnienia zamkniętych cykli. **DOPISAC o CYKLACH TABELA X zawiera zestawienie liczby wymaganych kubitów.**

Tabela . Redukcja rozmiaru problemu VRP.

|  |  |  |
| --- | --- | --- |
|  | Liczba możliwych rozwiązań | Wymagana liczba kubitów |
|  | Obiekt *TSP* - wersja domyślna | |
| Przypadek ogólny |  |  |
| Dla 3 wierzchołków | 512 | 9 |
| Dla 4 wierzchołków | 65536 | 16 |
|  | Po usunięciu krawędzi o tożsamych wierzchołkach | |
| Przypadek ogólny |  |  |
| Dla 3 wierzchołków | 64 | 6 |
| Dla 4 wierzchołków | 4096 | 12 |
|  | Po przekształceniu do grafu nieskierowanego | |
| Przypadek ogólny |  |  |
| Dla 3 wierzchołków | 8 | 3 |
| Dla 4 wierzchołków | 64 | 6 |

**Źródło**: opracowanie własne.

Algorytm QAOA jest algorytmem parametryzowalnym. Pierwszy, stały parametr dotyczy wag funkcji kary. Brakuje kompleksowej literatury w tym zakresie dla algorytmów kwantowych. Poniżej przedstawiono propozycję formułowania wag dla funkcji kary.

Hamiltonian funkcji celu można sformułować następująco:

|  |  |  |
| --- | --- | --- |
|  |  |  |

gdzie:

– Hamiltonian funkcji celu,

– Hamiltonian funkcji kosztu,

– Hamiltonian -tej funkcji kary,

– Hamiltonian wszystkich funkcji kar,

– mnożnik funkcji kosztu,

– mnożnik -tej funkcji kary,

– mnożnik łączny wszystkich funkcji kar.

Dla uproszczenia przyjęto . Dobranie wartości dla ograniczeń lokalnych (oznaczanych ; ograniczenia globalne będą indeksowane odpowiednio ) realizujących równania 18 i 19 będzie bazowało na macierzy odległości, przy czym liczba ograniczeń lokalnych jest równa liczbie wierzchołków w grafie, więc -ty warunek lokalny będzie powiązany z -tym wierzchołkiem. Należy zauważyć, że -ty mnożnik powinien być nie mniejszy niż najkrótsze możliwe połączenie prowadzące do wierzchołka, którego dotyczy mnożnik, aby algorytm nie preferował złamania ograniczenia nad utworzeniem trasy krótszej, ale nie spełniającej warunków zadania. Z drugiej strony zbyt wysoki mnożnik będzie powstrzymał algorytm przed eksploracją przestrzeni rozwiązań z powodu zbyt wysokiej zachłanności – zagwarantuje to znalezienie rozwiązania spełniającego warunki ograniczające, maleje jednak szansa na znalezienie rozwiązania optymalnego. Z tego powodu postuluje się ustalanie -tego mnożnika jako wartości nie przewyższającej znacznie wagi (długości) najważniejszej (najdłuższej) krawędzi prowadzącej do wierzchołka, którego dotyczy ograniczenie lokalne, tzn.:

gdzie to wektor krawędzi mających wierzchołek w -tym wierzchołku. Należy zwrócić uwagę, że dla grafów nieskierowanych warunki lokalne (równania 18 i 19) można sprowadzić do jednego warunku, tj.:

tzn. dla każdego wierzchołka liczba aktywnych krawędzi z nim związanych powinna być równa 2 – wówczas jedną krawędź należy interpretować jako krawędź prowadzącą do -tego wierzchołka, a drugą jako krawędź wychodzącą (kolejność nie jest istotna, bo ). W postaci QUBO wraz z mnożnikiem warunek ten można zapisać jako:

W ramach przeprowadzanych eksperymentów zostaną przypisane wartości mnożników równe długości najdłuższej krawędzi powiązanej z danym wierzchołkiem, tzn.:

gdzie oznacza indeks wierzchołka, którego dotyczy ograniczenie, co pokazano w **Tabeli**

Tabela . Mnożniki funkcji kary dla lokalnych warunków ograniczających.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T1 | | | F1 | | | |
|  |  |  |  |  |  |  |
| 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| T2 | | | F2 | | | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Źródło**: opracowanie własne.

### Postać QUBO problemu

Dla poszczególnych problemów otrzymano następujące Hamiltoniany celu stanowiące postać QUBO:

* T1 z ograniczeniami lokalnymi:
* F1 z ograniczeniami lokalnymi:
* T2 z ograniczeniami lokalnymi:

* T1 z ograniczeniami lokalnymi:

Warto zauważyć, że (zgodnie z intuicją) dla problemów T1 i F1, gdzie odległości między wierzchołkami są identyczne, również wagi dla poszczególnych elementów Hamiltonianów będą identyczne (ale nadal różne pomiędzy elementami reprezentującymi koszt przejazdu a elementami reprezentującymi funkcje kary – dla problemu F1 ich identyczność wyniku wyłącznie z nietypowego doboru liczb i należy go uznać za kwestię pomijalną). Warto również zwrócić uwagę na fakt, że mnożniki elementów funkcji kary stanowią połowę mnożników , co wynika z dzielenia w równaniu 33. Przygotowane w ten sposób Hamiltoniany pozostają niezmienne w czasie przetwarzania algorytmu i wraz z służą do skonstruowania bramki , która wraz z bramką zawiera podstawowe dane wejściowe algorytmu QAOA. Z kolei w postaic nieprzetworzonej, powyższe Hamiltoniany przekazano do algorytmu NumPyMinimumEigensolver.

## Wyniki

### Obliczenia klasyczne - hiperoptymalizacja

Na rysunkach X-Y przedstawiono przebieg procesu hiperoptymalizacji parametrów i . Algorytm szukał takiego zestawu hiperparametrów, aby zminimalizować funkcję celu, tj. minimalizować wartość oczekiwaną długości trasy po uwzględnieniu kar za niespełnienie warunków ograniczających. Na każdym z wykresów na rysunku **555** widoczna jest znacznie większa zmienność wartości funkcji celu ze względu na wartości parametru ; potwierdzają to wykresy zależności częściowej (PDP – Partial Dependence Plot), na których widać dużą nieregularność i skomplikowanie wykresu dla , podczas gdy dla wykresy te mają kształt zbliżony do linii poziomej. Podobny wniosek można wyciągnąć patrząc na mapy ciepła. Może to sugerować, że dla przedstawionego problemu znacznie istotniejszy jest właściwy dobór wartości parametru niż dobór wartości parametru , tj. należy się bardziej skupić na hiperoptymalizacji parametrów związanych z bramką implementującą hamiltonian celu niż na hiperparametrach związanych z bramką implementującą hamiltonian mieszający .

Rysunek 5. Hiperoptymalizacja parametrów metodą Bayesowską optymalizacji Bayesowskiej dla problemu T1.

Obraz zawierający tekst, diagram, Wykres, linia

Opis wygenerowany automatycznie

**Źródło**: opracowanie własne.

Rysunek . Hiperoptymalizacja parametrów metodą Bayesowską dla problemu T2.

Obraz zawierający tekst, diagram, linia, Równolegle

Opis wygenerowany automatycznie

**Źródło**: opracowanie własne.

Rysunek 7. Hiperoptymalizacja parametrów metodą Bayesowską dla problemu F1.

Obraz zawierający tekst, diagram, Wykres, linia

Opis wygenerowany automatycznie

**Źródło**: opracowanie własne.

Rysunek 8. Hiperoptymalizacja parametrów metodą Bayesowską dla problemu F2.

Obraz zawierający tekst, diagram, Wykres, linia

Opis wygenerowany automatycznie

**Źródło**: opracowanie własne.

### Obliczenia klasyczne – czas obliczeń

W **tabeli** przedstawiono wyniki czasowe obliczeń klasycznych. Szybki czas przetwarzania dla metody przeszukiwania wyczerpującego wynika z niewielkiego rozmiaru problemów. Z kolei długi czas przetwarzania dla symulacji algorytmów kwantowych również nie powinien być zaskoczeniem, ponieważ są to klasyczne symulacje algorytmów, które z założenia mają być stosowane na maszynach kwantowych. Dla symulacji algorytmu QAOA wykorzystujących metody Bayesowskie można zauważyć niższy odsetek sukcesów dla problemów T2 i F2 – wyjaśnienia można szukać na rysunku 6 gdzie klasyczny optymalizator miał problem ze znalezieniem minimum funkcji celu i przerywał przedwcześnie działanie. Pomimo wielokrotnie większego czasu przetwarzania, metody QAOA ze wsparciem metod Bayesowkich w 3 na 4 przypadkach był w stanie sobie lepiej poradzić niż QAOA z optymalizatorem COBYLA z pakietu qiskit. Równocześnie w każdym przypadku symulacja algorytmu QAOA wolniej kończyła przetwarzanie niż deterministyczne algorytmy klasyczne – zarówno przeszukiwanie wyczerpujące, jak i bazujące na postaci QUBO metody zaimplementowane w bibliotece NumPy było w stanie znaleźć poprawne rozwiązanie w krótkim czasie.

Tabela . Wyniki optymalizacji metodami klasycznymi.

|  |  |  |
| --- | --- | --- |
| Algorytm (klasyczny) | Czas obliczeń [ms] | Szczegóły |
| T1 z ograniczeniami lokalnymi | | |
| Przeszukiwanie wyczerpujące | 0.003 | Sukces |
| NumPy Eigensolver | 14.8 | Sukces |
| Qiskit QAOA | 159 | 32 iteracje, p-ństwo odczytania najlepszego wyniku = 80% |
| QAOA z hiperoptymalizacją Bayesowską | 1560 | Sukces dla 10/10 testów |
| F1 z ograniczeniami lokalnymi | | |
| Przeszukiwanie wyczerpujące | 0.077 | Sukces |
| NumPy Eigensolver | 31 | Sukces |
| Qiskit QAOA | 123 | 34 iteracje, p-ństwo odczytania jednego z najlepszych wyników = 83% |
| QAOA z hiperoptymalizacją Bayesowską | 819 | Sukces dla 6/10 testów. |
| T2 z ograniczeniami lokalnymi | | |
| Przeszukiwanie wyczerpujące | 0.0031 | Sukces |
| NumPy Eigensolver | 15.5 | Sukces |
| Qiskit QAOA | 170 | 28 iteracji, p-ństwo odczytania najlepszego wyniku = 63% |
| QAOA z hiperoptymalizacją Bayesowską | 1831.2 | Sukces dla 8/10 testów |
| F2 z ograniczeniami lokalnymi | | |
| Przeszukiwanie wyczerpujące | 0.078 | Sukces |
| NumPy Eigensolver | 23.5 | Sukces |
| Qiskit QAOA | 450 | 29 iteracji, p-ństwo odczytania najlepszego wyniku = 48% |
| QAOA z hiperoptymalizacją Bayesowską | 1281 | Sukces dla 8/10 testów |

**Źródło**: opracowanie własne.

### Obliczenia kwantowe

Instrukcje dla maszyny kwantowej przekazano z wykorzystaniem języka maszynowego OpenQASM 2.0. Należy przy tym pamiętać, że nie jest to język dostosowany do obliczeń klasycznych. Z tego powodu działania takie jak konstruowanie macierzy operatorów oraz musi się odbywać w oddzielnym środowisku. Ponadto warto mieć na uwadze, że język OpenQASM 2.0 oferuje mniej sposobów konstruowania bramek niż qiskit. Z tego powodu należy dokonać transpilacji obwodu zawierającego bramki oraz , tzn. dokonać dekompozycji bramek zdefiniowanych z pomocą pakietu qiskit na prostsze, które można wpisać do kodu języka maszynowego. Proces transpilacji obwodu jest podatny na klasyczne błędy numeryczne związane z przybliżeniami – po dokonaniu dekompozycji następuje próba optymalizacji liczby prostych bramek. Funkcja *transpile* z biblioteki qiskit umożliwia sterowanie stopniem optymalizacji obwodu w procesie transpilacji – im wyższy poziom optymalizacji, tym wyższe ryzyko popełnienia błędów numerycznych w trakcie procesu. W ramach powyższych eksperymentów zostanie przyjęty średni poziom optymalizacji (drugi z trzech), a docelowe bramki to CX, I, Rz, SX, X, czyli te, które są wykorzystywane przez procesor Falcon r5.11H. Zadania dekompozycji, optymalizacji i uodparniania na zaszumienie stanowią oddzielny przedmiot badań w zakresie informatyki kwantowej, ich efektywne rozwiązanie jest uważane za jedno z wyzwań. Przykładowo, na **RYSUNKU** przedstawiono obwód kwantowy po transpilacji dla problemu T1 (szara linia (bariera) oddziela zdekomponowaną bramkę od zdekomponowanej bramki ).

Rysunek . Obwód kwantowy dla problemu T1 po transpilacji.

Obraz zawierający zrzut ekranu, linia, tekst, diagram

Opis wygenerowany automatycznie

**Źródło**: opracowanie własne.

Ze względu na charakterystykę (i ograniczenia) używanej infrastruktury i oprogramowania należało zmienić metodę testowania algorytmu. W ramach każdego z problemów wykonano następujące kroki, wykonując łącznie 3 iteracje algorytmu QAOA (ostatnia iteracja bez części klasycznej algorytmu QAOA):

1. Zdefiniowanie problemu w języku Python.
2. Zdefiniowanie bramek kwantowych i w języku Python, w bibliotece qiskit.
3. Zdefiniowanie obwodu kwantowego w języku qiskit, zawierającego pojedyczną iterację części kwantowej algorytmu QAOA (czyli zastosowanie bramek i oraz dokonanie pomiaru).
4. Transpilacja obwodu z poprzedniego punktu.
5. Ręczne przekazanie definicji obwodu po transpilacji do języka OpenQASM 2.0.
6. Uruchomienie części kwantowej algorytmu QAOA 100 razy.
7. Przekazanie wyników pomiaru do pliku.
8. Zaimportowanie pliku z wynikami do środowiska z językiem Python.
9. Dokonanie hiperoptymalizacji parametrów metodą Bayesowską z .
10. Powtórzenie kroków 2-9.
11. Powtórzenie kroków 2-7 (ostatnia iteracja bez części klasycznej algorytmu).
12. Analiza końcowych wyników algorytmu QAOA.

W tabeli **XXX** przedstawiono czasy przetwarzania poszczególnych etapów:

Tabela . Czasy przetwarzania algorytmu QAOA w sekundach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteracja | Etap | T1 | T2 | F1 | F2 |
| 1 | Przygotowanie bramek (2-4) |  |  |  |  |
| Część kwantowa (6) |  |  |  |  |
| Klasyczna hiperoptymalizacja (9) |  |  |  |  |
| 2 | Przygotowanie bramek (2-4) |  |  |  |  |
| Część kwantowa (6) |  |  |  |  |
| Klasyczna hiperoptymalizacja (9) |  |  |  |  |
| 3 | Przygotowanie bramek (2-4) |  |  |  |  |
| Część kwantowa (6) |  |  |  |  |

**Źródło**: opracowanie własne.

## Wnioski dla wyników

W pierwszej kolejności można zauważyć, że dla symulacji klasycznych hiperoptymalizacja metodami Bayesowskimi

Zakończenie

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

LITERATURA

**Książki**

1. Bradley K., *Podstawy metodyki PRINCE 2*, Warszawa 2002.
2. Bukowski M., *Korzyści z wdrożenia metodyki PRINCE2 w wybranej jednostce administracji publicznej*, op. naukowy dr inż. W. Dąbrowski, Warszawa, czerwiec 20.
3. Flasiński M., *Zarządzanie projektami informatycznymi,* Wydawnictwo Naukowe PWN, Warszawa 2006.
4. Frączkowski K., *Zarządzanie projektem informatycznym. Projekty w środowisku wirtualnym. Czynniki sukcesu i niepowodzeń projektów*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2003, s.125-126.
5. Gunia G., *Implementacja zintegrowanych systemów informatycznych w małych i średnich przedsiębiorstwach*, „Zarządzanie Przedsiębiorstwem”, 2009.
6. Gunia G., *Wdrażanie zintegrowanych systemów informatycznych,* Wydawnictwo Fundacji Centrum Nowych Technologii, Bielsko Biała 2009.
7. Januszewski A., *Funkcjonalność informatycznych systemów zarządzania. Tom 1*, Wydawnictwo Naukowe PWN, Warszawa 2008.
8. Kisielnicki J., *Zintegrowane systemy informatyczne, dobre praktyki wdrożeń systemów klasy ERP*, Wydawnictwo Naukowe PWN
9. Kisielnicki J., *Systemy informacyjne biznesu,* Placet, Warszawa 1999.
10. Koszlajda A.*, Zarządzanie projektami IT*, Gliwice 2010.
11. Lech P., Zintegrowane Systemy Zarządzania ERP/ERP2. *Wykorzystanie w biznesie wdrażanie,* Difin, Warszawa 2003.
12. Miłosz M. (red.), *Wdrażanie i eksploatacja systemów informatycznych. Wybrane problemy,* Polskie Towarzystwo Informatyczne, Lublin 2002.
13. Office of Government Commerce, *Skuteczne zarządzanie projektami PRINCE2*, London: TSO, wydanie 2005.
14. Rokicka-Broniatowska A., *Wstęp do informatyki gospodarczej*, Szkoła Główna Handlowa, Warszawa 2006.
15. Stefanowicz B., *Informacyjne systemy zarządzania. Przewodnik,* Szkoła Główna Handlowa, Warszawa 2007.
16. Szyjewski Z., *Metodyki zarządzania projektami informatycznymi,* PLACET, Warszawa 2004.
17. Trocki M., *Metodyki zarządzania projektami*, Biblioteka Project Managera, Warszawa 2011.
18. Waćkowski K., Chmielewski J., *Wspomaganie zarządzania projektami informatycznymi. Poradnik dla menadżera*, Helion, Gliwice 2007.

**Artykuły i studia**

1. Dębowski L.,*ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICE’ 2006*, Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 22
2. IE. 1991. Competition in manufacturing leads to MRP II. 23 (July) 10-13
3. WJ Hopp, ML Spearman Commissioned Paper To Pull or Not to Pull: What Is the Question, Manufacturing & Service Operations Management, 2004

**Strony internetowe**

1. Poksiński P., *PRINCE2*, materiał dostępny w wersji elektronicznej pod adresem internetowym: http://www.poksinski.com/pdf\_files/PRINCE2.pdf [18-09-2012]
2. *Słownik APICS*, http://www.apics.org/dictionary/dictionary-information?ID=2399 [4-01-2013]
3. Schmidt P., *PRINCE2 i techniki planowania*, materiał dostępny w wersji elektronicznej pod adresem internetowym: http://ww.4pm.pl/artykul/prince2\_i\_techniki\_planowania-56-1247.html [17-10-2012]
4. *Cloud Computing*, materiał zaczerpnięty ze strony internetowej: http://www.it.integro.pl/pl/aplikacje-w-chmurze-cloud-computing [4-01-2013]
5. Trąbka J., *Zarządzanie projektem wdrożeniowym systemu klasy ERP – autorska metodyka*, materiał dostępny w wersji elektronicznej pod adresem internetowym: http://kkio2012.agh.edu.pl/presentations/KKIO2012-C1\_3.pdf [10-11-2012]
6. *Skuteczne wdrożenia systemu ERP dzięki metodyce Microsoft Sure Step;* http://www.it.integro.pl/pl/metodyka-wdrozeniowa. [28-09-2012]
7. *Nasza metodyka na wdrożenie ERP w twojej firmie;* http://www.profidata.com.pl/erp/metodyka-pit-stop.html. [28-09-2012]
8. *Manifesto for Agile Software Development*; Cytat ze strony http://www.agilemanifesto.org/ [10-10-2012]
9. *Metodyka wdrożenia QlikView- Business Intelligence;* , http://www.businessintelligence.pl/pl/metodyka-wdrozenia. [28-09-2012]
10. *QlikView przełamuje bariery rynku – Business Intelligence;* http://www.businessintelligence.pl/pl/a/QlikView-przelamuje-bariery-rynku [11-10-2012]
11. *ERP- centrum wiedzy o systemach ERP na decyzje-IT.pl;* http://decyzje-it.pl/centrum-wiedzy/erp.html [10-10-2012]
12. *The history of Prince2- Project Smart*; http://www.projectsmart.co.uk/history-of-prince2.html [18-09-2012]
13. *Oracle- Hardware and Software, Enginnered to work together*; http://www.oracle.com/index.html [23-09-2012]
14. CRP for Oracle R12- Welcom to Oracle ERP http://www.oracleerp4u.com/2010/06/crp-for-oracle-r12.html [189-10-2012]
15. *Przykłady zastosowań technologii RFID w magazynach* http://rfid-lab.pl/przyklady-zastosowan-technologii-rfid-w-magazynach [07-01-2013]

Spis tabel

[Tabela 1. Lorem ipsum dolor sit amet **Błąd! Nie zdefiniowano zakładki.**](#_Toc355852181)

[Tabela 2. Lorem ipsum dolor sit amet, consectetur adipisicing elit **Błąd! Nie zdefiniowano zakładki.**](#_Toc355852182)

Spis rysunków

[Rysunek 1. Wizualizacja sfery Blocha 11](#_Toc116386948)

[Rysunek 2. Odmiany problemu VRP. Źródło: M. Drexl, Rich Vehicle Routing in theory and practice. POPRAWIC TEN podpis **Błąd! Nie zdefiniowano zakładki.**](#_Toc116386949)

[Rysunek 3. Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet. **Błąd! Nie zdefiniowano zakładki.**](#_Toc116386950)

[Rysunek 4. Lorem ipsum dolor sit amet, consectetur. 60](#_Toc116386951)

Streszczenie

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Summary

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Akshay, V., H. Philathong, M. E. S. Morales, i J. Biamonte. „Reachability Deficits in Quantum Approximate Optimization”. *Physical Review Letters* 124, nr 9 (5 marzec 2020): 090504. https://doi.org/10.1103/PhysRevLett.124.090504.

Amazon Web Services, Inc. „Quantum Cloud Service - Quantum Computing Service - Amazon Braket - AWS”. Dostęp 19 marzec 2023. https://aws.amazon.com/braket/.

Applegate, David, Robert Bixby, Vašek Chvátal, i William Cook. „Implementing the Dantzig-Fulkerson-Johnson Algorithm for Large Traveling Salesman Problems”. *Mathematical Programming* 97, nr 1 (lipiec 2003): 91–153. https://doi.org/10.1007/s10107-003-0440-4.

Baldwin, Charles H., Karl Mayer, Natalie C. Brown, Ciarán Ryan-Anderson, i David Hayes. „Re-examining the quantum volume test: Ideal distributions, compiler optimizations, confidence intervals, and scalable resource estimations”. *Quantum* 6 (9 maj 2022): 707. https://doi.org/10.22331/q-2022-05-09-707.

Barak, Boaz, Ankur Moitra, Ryan O’Donnell, Prasad Raghavendra, Oded Regev, David Steurer, Luca Trevisan, Aravindan Vijayaraghavan, David Witmer, i John Wright. „Beating the random assignment on constraint satisfaction problems of bounded degree”. arXiv, 11 sierpień 2015. http://arxiv.org/abs/1505.03424.

Bektas, Tolga. „The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures”. *Omega* 34, nr 3 (1 czerwiec 2006): 209–19. https://doi.org/10.1016/j.omega.2004.10.004.

Benioff, Paul. „The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines”. *Journal of Statistical Physics* 22, nr 5 (maj 1980): 563–91. https://doi.org/10.1007/BF01011339.

Bennett, Charles H., Ethan Bernstein, Gilles Brassard, i Umesh Vazirani. „Strengths and Weaknesses of Quantum Computing”. *SIAM Journal on Computing* 26, nr 5 (październik 1997): 1510–23. https://doi.org/10.1137/S0097539796300933.

Crooks, Gavin E. „Gates, States, and Circuits”, b.d., 79.

Dantzig, G. B., i J. H. Ramser. „The Truck Dispatching Problem”. *Management Science* 6, nr 1 (październik 1959): 80–91. https://doi.org/10.1287/mnsc.6.1.80.

Dantzig, G., R. Fulkerson, i S. Johnson. „Solution of a Large-Scale Traveling-Salesman Problem”. *Journal of the Operations Research Society of America* 2, nr 4 (1954): 393–410.

Date, Prasanna, Davis Arthur, i Lauren Pusey-Nazzaro. „QUBO Formulations for Training Machine Learning Models”. *Scientific Reports* 11, nr 1 (11 maj 2021): 10029. https://doi.org/10.1038/s41598-021-89461-4.

Dirac, P. A. M. „A New Notation for Quantum Mechanics”. *Mathematical Proceedings of the Cambridge Philosophical Society* 35, nr 3 (lipiec 1939): 416–18. https://doi.org/10.1017/S0305004100021162.

Drexl, Michael. „Rich Vehicle Routing in Theory and Practice”. *Logistics Research* 5, nr 1 (1 sierpień 2012): 47–63. https://doi.org/10.1007/s12159-012-0080-2.

Egger, Daniel J., Jakub Marecek, i Stefan Woerner. „Warm-starting quantum optimization”. *Quantum* 5 (17 czerwiec 2021): 479. https://doi.org/10.22331/q-2021-06-17-479.

European Commission - European Commission. „EU Deploys First Quantum Technology in 6 Sites across Europe”. Text. Dostęp 6 luty 2023. https://ec.europa.eu/commission/presscorner/detail/en/IP\_22\_5914.

Farhi, E., J. Goldstone, S. Gutmann, i H. Neven. „Quantum Algorithms for Fixed Qubit Architectures”. arXiv, 17 marzec 2017. http://arxiv.org/abs/1703.06199.

Farhi, Edward, Jeffrey Goldstone, i Sam Gutmann. „A Quantum Approximate Optimization Algorithm”. arXiv, 14 listopad 2014. http://arxiv.org/abs/1411.4028.

———. „A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem”. arXiv, 25 czerwiec 2015. http://arxiv.org/abs/1412.6062.

Farhi, Edward, i Aram W. Harrow. „Quantum Supremacy through the Quantum Approximate Optimization Algorithm”. arXiv, 20 październik 2019. https://doi.org/10.48550/arXiv.1602.07674.

Feynman, Richard P. „Simulating Physics with Computers”. *International Journal of Theoretical Physics* 21, nr 6 (1 czerwiec 1982): 467–88. https://doi.org/10.1007/BF02650179.

Flood, Merrill M. „The Traveling-Salesman Problem”. *Operations Research* 4, nr 1 (1956): 61–75.

Frazier, Peter I. „A Tutorial on Bayesian Optimization”. arXiv, 8 lipiec 2018. https://doi.org/10.48550/arXiv.1807.02811.

Glover, Fred, Gary Kochenberger, i Yu Du. „A Tutorial on Formulating and Using QUBO Models”. arXiv, 4 listopad 2019. https://doi.org/10.48550/arXiv.1811.11538.

Google Quantum AI. „Cirq”. Dostęp 19 marzec 2023. https://quantumai.google/cirq.

Grover, Lov K. „A fast quantum mechanical algorithm for database search”. arXiv, 19 listopad 1996. https://doi.org/10.48550/arXiv.quant-ph/9605043.

GUS. „Oszacowanie odległości i czasu przejazdu pomiędzy wybranymi gminami w Polsce w 2016 roku”. stat.gov.pl. Dostęp 26 czerwiec 2023. https://stat.gov.pl/statystyki-eksperymentalne/obszary-funkcjonalne-oraz-dostepnosc-terytorialna/oszacowanie-odleglosci-i-czasu-przejazdu-pomiedzy-wybranymi-gminami-w-polsce-w-2016-roku,11,1.html.

Ha, Johan. „Some Optimal Inapproximability Results”, b.d., 62.

Hastings, M. B. „Classical and Quantum Bounded Depth Approximation Algorithms”. arXiv, 1 sierpień 2019. https://doi.org/10.48550/arXiv.1905.07047.

Hidary, Jack D. „A Brief History of Quantum Computing”. *Quantum Computing: An Applied Approach*, 2019, 11–16. https://doi.org/10.1007/978-3-030-23922-0\_2.

IBM Quantum. „IBM Quantum”. Dostęp 1 lipiec 2023. https://quantum-computing.ibm.com/services/resources.

IBM Quantum. „Qiskit Runtime Overview”. Dostęp 1 lipiec 2023. https://quantum-computing.ibm.com/lab/docs/iql/runtime.

„IBM Quantum Computing”, 1 październik 2015. https://www.ibm.com/www.ibm.com/quantum.

„IBM Quantum Computing | Roadmap”, 1 październik 2015. https://www.ibm.com/quantum/www.ibm.com/quantum/roadmap.

„IBM Unveils 433-Qubit Osprey Chip - IEEE Spectrum”. Dostęp 5 luty 2023. https://spectrum.ieee.org/ibm-quantum-computer-osprey.

Jones, Donald R., Matthias Schonlau, i William J. Welch. „Efficient Global Optimization of Expensive Black-Box Functions”. *Journal of Global Optimization* 13, nr 4 (1 grudzień 1998): 455–92. https://doi.org/10.1023/A:1008306431147.

Kadowaki, Tadashi, i Hidetoshi Nishimori. „Quantum Annealing in the Transverse Ising Model”, 25 kwiecień 1998. https://doi.org/10.1103/PhysRevE.58.5355.

Kara, Imdat, Gilbert Laporte, i Tolga Bektas. „A Note on the Lifted Miller–Tucker–Zemlin Subtour Elimination Constraints for the Capacitated Vehicle Routing Problem”. *European Journal of Operational Research* 158, nr 3 (1 listopad 2004): 793–95. https://doi.org/10.1016/S0377-2217(03)00377-1.

Karp, Richard M. „Reducibility among Combinatorial Problems”. W *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and Sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, zredagowane przez Raymond E. Miller, James W. Thatcher, i Jean D. Bohlinger, 85–103. The IBM Research Symposia Series. Boston, MA: Springer US, 1972. https://doi.org/10.1007/978-1-4684-2001-2\_9.

Katzgraber, Helmut G., Firas Hamze, Zheng Zhu, Andrew J. Ochoa, i H. Munoz-Bauza. „Seeking Quantum Speedup Through Spin Glasses: The Good, the Bad, and the Ugly”. *Physical Review X* 5, nr 3 (1 wrzesień 2015): 031026. https://doi.org/10.1103/PhysRevX.5.031026.

Kulkarni, R. V., i P. R. Bhave. „Integer Programming Formulations of Vehicle Routing Problems”. *European Journal of Operational Research* 20, nr 1 (1 kwiecień 1985): 58–67. https://doi.org/10.1016/0377-2217(85)90284-X.

Lavrijsen, Wim, Ana Tudor, Juliane Müller, Costin Iancu, i Wibe de Jong. „Classical Optimizers for Noisy Intermediate-Scale Quantum Devices”. W *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 267–77, 2020. https://doi.org/10.1109/QCE49297.2020.00041.

Lenstra, J. K., i A. H. G. Rinnooy Kan. „Complexity of Vehicle Routing and Scheduling Problems”. *Networks* 11, nr 2 (1981): 221–27. https://doi.org/10.1002/net.3230110211.

Liu, Dong C., i Jorge Nocedal. „On the Limited Memory BFGS Method for Large Scale Optimization”. *Mathematical Programming* 45, nr 1 (1 sierpień 1989): 503–28. https://doi.org/10.1007/BF01589116.

Lucas, Andrew. „Ising formulations of many NP problems”. *Frontiers in Physics* 2 (2014). https://www.frontiersin.org/articles/10.3389/fphy.2014.00005.

MathWorks. „Quantum Computing for Matlab”. Dostęp 19 marzec 2023. https://www.mathworks.com/products/quantum-computing.html.

McKinsey & Company. „Quantum computing: An emerging ecosystem and industry use cases”, grudzień 2021. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/quantum-computing-use-cases-are-getting-real-what-you-need-to-know#/.

Miller, C. E., A. W. Tucker, i R. A. Zemlin. „Integer Programming Formulation of Traveling Salesman Problems”. *Journal of the ACM* 7, nr 4 (1 październik 1960): 326–29. https://doi.org/10.1145/321043.321046.

Miller, Keith, Charles Broomfield, Ann Cox, Joe Kinast, i Brandon Rodenburg. „An Improved Volumetric Metric for Quantum Computers via more Representative Quantum Circuit Shapes”. arXiv, 14 lipiec 2022. http://arxiv.org/abs/2207.02315.

Moll, Nikolaj, Panagiotis Barkoutsos, Lev S. Bishop, Jerry M. Chow, Andrew Cross, Daniel J. Egger, Stefan Filipp, i in. „Quantum optimization using variational algorithms on near-term quantum devices”. *Quantum Science and Technology* 3, nr 3 (lipiec 2018): 030503. https://doi.org/10.1088/2058-9565/aab822.

Morales, Mauro E. S., Timur Tlyachev, i Jacob Biamonte. „Variational learning of Grover’s quantum search algorithm”. *Physical Review A* 98, nr 6 (27 grudzień 2018): 062333. https://doi.org/10.1103/PhysRevA.98.062333.

Nagy, Marius. „Quantum computation and quantum information”. *IJPEDS* 21 (1 luty 2006): 1–59. https://doi.org/10.1080/17445760500355678.

Nielsen, Michael A., i Isaac L. Chuang. „Quantum Computation and Quantum Information: 10th Anniversary Edition”. Higher Education from Cambridge University Press. Cambridge University Press, 9 grudzień 2010. https://doi.org/10.1017/CBO9780511976667.

NobelPrize.org. „The Nobel Prize in Physics 2022”. Dostęp 18 luty 2023. https://www.nobelprize.org/prizes/physics/2022/summary/.

„NumPyMinimumEigensolver — Qiskit 0.43.2 documentation”. Dostęp 1 lipiec 2023. https://qiskit.org/documentation/stubs/qiskit.algorithms.NumPyMinimumEigensolver.html.

Obst, Julian. „Parameter Initialization for Warm-Starting QAOA”, 2022. https://doi.org/10.18419/opus-12366.

Orloff, C. S. „A Fundamental Problem in Vehicle Routing”. *Networks* 4, nr 1 (1974): 35–64. https://doi.org/10.1002/net.3230040105.

Otterbach, J. S., R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, i in. „Unsupervised Machine Learning on a Hybrid Quantum Computer”. arXiv, 15 grudzień 2017. https://doi.org/10.48550/arXiv.1712.05771.

Pratt, Vaughan R. „An n Log n Algorithm to Distribute n Records Optimally in a Sequential Access File”. W *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and Sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, zredagowane przez Raymond E. Miller, James W. Thatcher, i Jean D. Bohlinger, 111–18. The IBM Research Symposia Series. Boston, MA: Springer US, 1972. https://doi.org/10.1007/978-1-4684-2001-2\_11.

P&S Intelligence. „Quantum Computing Market Size & Share Forecast Report 2030”. Dostęp 5 luty 2023. https://www.psmarketresearch.com/market-analysis/quantum-computing-market.

„PyQuil: Quantum programming in Python”. Python. 2017. Reprint, Rigetti Computing, 18 marzec 2023. https://github.com/rigetti/pyquil.

„QAOA documentation”, b.d. https://qiskit.org/documentation/stable/0.30/stubs/qiskit.aqua.algorithms.QAOA.html.

„Qiskit”. Dostęp 5 luty 2023. https://qiskit.org/.

„Quantum Annealing and Related Optimization Methods | SpringerLink”. Dostęp 13 październik 2022. https://link.springer.com/book/10.1007/11526216.

„Quantum Computing Market Size, Share | Growth Forecast 2029”. Dostęp 5 luty 2023. https://www.fortunebusinessinsights.com/quantum-computing-market-104855.

Raff, Samuel. „Routing and Scheduling of Vehicles and Crews: The State of the Art”. *Computers & Operations Research*, Routing and Scheduling of Vehicles and Crews. The State of the Art, 10, nr 2 (1 styczeń 1983): 63–211. https://doi.org/10.1016/0305-0548(83)90030-8.

Rigetti Computing. „Quantum Computing”. Dostęp 19 marzec 2023. https://www.rigetti.com/.

Salehi, Özlem, Adam Glos, i Jarosław Adam Miszczak. „Unconstrained Binary Models of the Travelling Salesman Problem Variants for Quantum Optimization”. *Quantum Information Processing* 21, nr 2 (22 styczeń 2022): 67. https://doi.org/10.1007/s11128-021-03405-5.

Shaydulin, Ruslan, i Stefan M. Wild. „Exploiting Symmetry Reduces the Cost of Training QAOA”. *IEEE Transactions on Quantum Engineering* 2 (2021): 1–9. https://doi.org/10.1109/TQE.2021.3066275.

Snoek, Jasper, Hugo Larochelle, i Ryan P Adams. „Practical Bayesian Optimization of Machine Learning Algorithms”. W *Advances in Neural Information Processing Systems*, T. 25. Curran Associates, Inc., 2012. https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html.

SoniaLopezBravo. „Introduction to Q# & Quantum Development Kit - Azure Quantum”, 4 luty 2023. https://learn.microsoft.com/en-us/azure/quantum/overview-what-is-qsharp-and-qdk.

Streif, Michael, i Martin Leib. „Comparison of QAOA with Quantum and Simulated Annealing”. arXiv, 7 styczeń 2019. https://doi.org/10.48550/arXiv.1901.01903.

———. „Training the quantum approximate optimization algorithm without access to a quantum processing unit”. *Quantum Science and Technology* 5, nr 3 (12 maj 2020): 034008. https://doi.org/10.1088/2058-9565/ab8c2b.

Susskind, Leonard, i Art Friedeman. *Mechanika kwantowa. Teoretyczne minimum*. Warszawa: Prószyński Media Sp z o.o., 2016.

„The EU’S Quantum Technologies Flagship | Shaping Europe’s Digital Future”, 26 styczeń 2023. https://digital-strategy.ec.europa.eu/en/library/eus-quantum-technologies-flagship.

Tibaldi, Simone, Davide Vodola, Edoardo Tignone, i Elisa Ercolessi. „Bayesian Optimization for QAOA”. arXiv, 30 wrzesień 2022. https://doi.org/10.48550/arXiv.2209.03824.

Tom’s Hardware. „IBM Introduces CLOPS Performance Standard for Quantum Computing”, 3 listopad 2021. https://www.tomshardware.com/news/ibm-introduces-clops-performance-standard-for-quantum-computing.

Verdon, Guillaume, Michael Broughton, i Jacob Biamonte. „A quantum algorithm to train neural networks using low-depth circuits”. arXiv, 14 grudzień 2017. https://doi.org/10.48550/arXiv.1712.05304.

Wecker, D., M. B. Hastings, i M. Troyer. „Towards Practical Quantum Variational Algorithms”. *Physical Review A* 92, nr 4 (2 październik 2015): 042303. https://doi.org/10.1103/PhysRevA.92.042303.

Wong, Thomas G. *Introduction to Classical and Quantum Computing*, 2022.

„World Development Indicators | DataBank”. Dostęp 5 luty 2023. https://databank.worldbank.org/reports.aspx?source=2&series=NY.GDP.MKTP.CD&country=.

Xanadu. „Welcome to Xanadu”. Dostęp 19 marzec 2023. https://www.xanadu.ai/.

Zhu, D., N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, i in. „Training of quantum circuits on a hybrid quantum computer”. *Science Advances* 5, nr 10 (18 październik 2019): eaaw9918. https://doi.org/10.1126/sciadv.aaw9918.

1. Richard P. Feynman, „Simulating Physics with Computers”, *International Journal of Theoretical Physics* 21, nr 6 (1 czerwiec 1982): 467–69, 476–77, https://doi.org/10.1007/BF02650179. [↑](#footnote-ref-1)
2. Paul Benioff, „The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines”, *Journal of Statistical Physics* 22, nr 5 (maj 1980): 563–91, https://doi.org/10.1007/BF01011339. [↑](#footnote-ref-2)
3. Jack D. Hidary, „A Brief History of Quantum Computing”, *Quantum Computing: An Applied Approach*, 2019, 11–12, https://doi.org/10.1007/978-3-030-23922-0\_2. [↑](#footnote-ref-3)
4. „The Nobel Prize in Physics 2022”, NobelPrize.org, dostęp 18 luty 2023, https://www.nobelprize.org/prizes/physics/2022/summary/. [↑](#footnote-ref-4)
5. Thomas G Wong, *Introduction to Classical and Quantum Computing*, 2022, 6–7. [↑](#footnote-ref-5)
6. Michael A. Nielsen i Isaac L. Chuang, „Quantum Computation and Quantum Information: 10th Anniversary Edition”, Higher Education from Cambridge University Press (Cambridge University Press, 9 grudzień 2010), 13, https://doi.org/10.1017/CBO9780511976667. [↑](#footnote-ref-6)
7. Wong, *Introduction to Classical and Quantum Computing*, 97. [↑](#footnote-ref-7)
8. Marius Nagy, „Quantum computation and quantum information”, *IJPEDS* 21 (1 luty 2006): 5, https://doi.org/10.1080/17445760500355678. [↑](#footnote-ref-8)
9. Leonard Susskind i Art Friedeman, *Mechanika kwantowa. Teoretyczne minimum* (Warszawa: Prószyński Media Sp z o.o., 2016), 39. [↑](#footnote-ref-9)
10. Nielsen i Chuang, „Quantum Computation and Quantum Information”, 13. [↑](#footnote-ref-10)
11. Wong, *Introduction to Classical and Quantum Computing*, 115. [↑](#footnote-ref-11)
12. P. A. M. Dirac, „A New Notation for Quantum Mechanics”, *Mathematical Proceedings of the Cambridge Philosophical Society* 35, nr 3 (lipiec 1939): 416–18, https://doi.org/10.1017/S0305004100021162. [↑](#footnote-ref-12)
13. Nielsen i Chuang, „Quantum Computation and Quantum Information”, 13. [↑](#footnote-ref-13)
14. Wong, *Introduction to Classical and Quantum Computing*, 91-3. [↑](#footnote-ref-14)
15. Wong, 76. [↑](#footnote-ref-15)
16. Susskind i Friedeman, *Mechanika kwantowa. Teoretyczne minimum*, 73–74. [↑](#footnote-ref-16)
17. Nielsen i Chuang, „Quantum Computation and Quantum Information”, 13–14. [↑](#footnote-ref-17)
18. Wong, *Introduction to Classical and Quantum Computing*, 190. [↑](#footnote-ref-18)
19. Nielsen i Chuang, „Quantum Computation and Quantum Information”, 278, 353–94. [↑](#footnote-ref-19)
20. Nielsen i Chuang, 13–14. [↑](#footnote-ref-20)
21. Susskind i Friedeman, *Mechanika kwantowa. Teoretyczne minimum*, 79–81. [↑](#footnote-ref-21)
22. Wong, *Introduction to Classical and Quantum Computing*, 237–51. [↑](#footnote-ref-22)
23. Wong, 11–15. [↑](#footnote-ref-23)
24. Nielsen i Chuang, „Quantum Computation and Quantum Information”, 49. [↑](#footnote-ref-24)
25. Nielsen i Chuang, 98–110. [↑](#footnote-ref-25)
26. Wong, *Introduction to Classical and Quantum Computing*, 108–10. [↑](#footnote-ref-26)
27. Gavin E Crooks, „Gates, States, and Circuits”, b.d., 10. [↑](#footnote-ref-27)
28. Merrill M. Flood, „The Traveling-Salesman Problem”, *Operations Research* 4, nr 1 (1956): 61. [↑](#footnote-ref-28)
29. Flood, „The Traveling-Salesman Problem”. [↑](#footnote-ref-29)
30. G. B. Dantzig i J. H. Ramser, „The Truck Dispatching Problem”, *Management Science* 6, nr 1 (październik 1959): 80–81, https://doi.org/10.1287/mnsc.6.1.80. [↑](#footnote-ref-30)
31. Dantzig i Ramser, „The Truck Dispatching Problem”. [↑](#footnote-ref-31)
32. Vaughan R. Pratt, „An n Log n Algorithm to Distribute n Records Optimally in a Sequential Access File”, w *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and Sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, red. Raymond E. Miller, James W. Thatcher, i Jean D. Bohlinger, The IBM Research Symposia Series (Boston, MA: Springer US, 1972), 112–18, https://doi.org/10.1007/978-1-4684-2001-2\_11. [↑](#footnote-ref-32)
33. Richard M. Karp, „Reducibility among Combinatorial Problems”, w *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and Sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, red. Raymond E. Miller, James W. Thatcher, i Jean D. Bohlinger, The IBM Research Symposia Series (Boston, MA: Springer US, 1972), 94, https://doi.org/10.1007/978-1-4684-2001-2\_9. [↑](#footnote-ref-33)
34. Dantzig i Ramser, „The Truck Dispatching Problem”, 81. [↑](#footnote-ref-34)
35. C. S. Orloff, „A Fundamental Problem in Vehicle Routing”, *Networks* 4, nr 1 (1974): 35–36, https://doi.org/10.1002/net.3230040105. [↑](#footnote-ref-35)
36. J. K. Lenstra i A. H. G. Rinnooy Kan, „Complexity of Vehicle Routing and Scheduling Problems”, *Networks* 11, nr 2 (1981): 221–27, https://doi.org/10.1002/net.3230110211. [↑](#footnote-ref-36)
37. Orloff, „A Fundamental Problem in Vehicle Routing”. [↑](#footnote-ref-37)
38. Karp, „Reducibility among Combinatorial Problems”. [↑](#footnote-ref-38)
39. Lenstra i Kan, „Complexity of Vehicle Routing and Scheduling Problems”, 223. [↑](#footnote-ref-39)
40. Michael Drexl, „Rich Vehicle Routing in Theory and Practice”, *Logistics Research* 5, nr 1 (1 sierpień 2012): 47, https://doi.org/10.1007/s12159-012-0080-2. [↑](#footnote-ref-40)
41. Samuel Raff, „Routing and Scheduling of Vehicles and Crews: The State of the Art”, *Computers & Operations Research*, Routing and Scheduling of Vehicles and Crews. The State of the Art, 10, nr 2 (1 styczeń 1983): 83–86, https://doi.org/10.1016/0305-0548(83)90030-8. [↑](#footnote-ref-41)
42. Tolga Bektas, „The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures”, *Omega* 34, nr 3 (1 czerwiec 2006): 212–13, https://doi.org/10.1016/j.omega.2004.10.004. [↑](#footnote-ref-42)
43. Bektas, 212–13. [↑](#footnote-ref-43)
44. G. Dantzig, R. Fulkerson, i S. Johnson, „Solution of a Large-Scale Traveling-Salesman Problem”, *Journal of the Operations Research Society of America* 2, nr 4 (1954): 397–98. [↑](#footnote-ref-44)
45. David Applegate i in., „Implementing the Dantzig-Fulkerson-Johnson Algorithm for Large Traveling Salesman Problems”, *Mathematical Programming* 97, nr 1 (lipiec 2003): 91–153, https://doi.org/10.1007/s10107-003-0440-4. [↑](#footnote-ref-45)
46. C. E. Miller, A. W. Tucker, i R. A. Zemlin, „Integer Programming Formulation of Traveling Salesman Problems”, *Journal of the ACM* 7, nr 4 (1 październik 1960): 326–29, https://doi.org/10.1145/321043.321046. [↑](#footnote-ref-46)
47. R. V. Kulkarni i P. R. Bhave, „Integer Programming Formulations of Vehicle Routing Problems”, *European Journal of Operational Research* 20, nr 1 (1 kwiecień 1985): 58–67, https://doi.org/10.1016/0377-2217(85)90284-X. [↑](#footnote-ref-47)
48. Imdat Kara, Gilbert Laporte, i Tolga Bektas, „A Note on the Lifted Miller–Tucker–Zemlin Subtour Elimination Constraints for the Capacitated Vehicle Routing Problem”, *European Journal of Operational Research* 158, nr 3 (1 listopad 2004): 793–95, https://doi.org/10.1016/S0377-2217(03)00377-1. [↑](#footnote-ref-48)
49. Andrew Lucas, „Ising formulations of many NP problems”, *Frontiers in Physics* 2 (2014): 17, https://www.frontiersin.org/articles/10.3389/fphy.2014.00005. [↑](#footnote-ref-49)
50. Özlem Salehi, Adam Glos, i Jarosław Adam Miszczak, „Unconstrained Binary Models of the Travelling Salesman Problem Variants for Quantum Optimization”, *Quantum Information Processing* 21, nr 2 (22 styczeń 2022): 7–8, https://doi.org/10.1007/s11128-021-03405-5. [↑](#footnote-ref-50)
51. Salehi, Glos, i Miszczak, 8–11. [↑](#footnote-ref-51)
52. Edward Farhi, Jeffrey Goldstone, i Sam Gutmann, „A Quantum Approximate Optimization Algorithm” (arXiv, 14 listopad 2014), http://arxiv.org/abs/1411.4028. [↑](#footnote-ref-52)
53. D. Wecker, M. B. Hastings, i M. Troyer, „Towards Practical Quantum Variational Algorithms”, *Physical Review A* 92, nr 4 (2 październik 2015): 1–2, https://doi.org/10.1103/PhysRevA.92.042303. [↑](#footnote-ref-53)
54. Edward Farhi, Jeffrey Goldstone, i Sam Gutmann, „A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem” (arXiv, 25 czerwiec 2015), http://arxiv.org/abs/1412.6062. [↑](#footnote-ref-54)
55. Boaz Barak i in., „Beating the random assignment on constraint satisfaction problems of bounded degree” (arXiv, 11 sierpień 2015), http://arxiv.org/abs/1505.03424. [↑](#footnote-ref-55)
56. Johan Ha, „Some Optimal Inapproximability Results”, b.d., 2 http://www.cs.umd.edu/~gasarch/BLOGPAPERS/max3satl.pdf., s. 814 [↑](#footnote-ref-56)
57. V. Akshay i in., „Reachability Deficits in Quantum Approximate Optimization”, *Physical Review Letters* 124, nr 9 (5 marzec 2020): 090504, https://doi.org/10.1103/PhysRevLett.124.090504. [↑](#footnote-ref-57)
58. Mauro E. S. Morales, Timur Tlyachev, i Jacob Biamonte, „Variational learning of Grover’s quantum search algorithm”, *Physical Review A* 98, nr 6 (27 grudzień 2018): 062333, https://doi.org/10.1103/PhysRevA.98.062333. [↑](#footnote-ref-58)
59. Lov K. Grover, „A fast quantum mechanical algorithm for database search” (arXiv, 19 listopad 1996), https://doi.org/10.48550/arXiv.quant-ph/9605043. [↑](#footnote-ref-59)
60. Charles H. Bennett i in., „Strengths and Weaknesses of Quantum Computing”, *SIAM Journal on Computing* 26, nr 5 (październik 1997): 1510–23, https://doi.org/10.1137/S0097539796300933. [↑](#footnote-ref-60)
61. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm”, 3. [↑](#footnote-ref-61)
62. Farhi, Goldstone, i Gutmann, 3. [↑](#footnote-ref-62)
63. Farhi, Goldstone, i Gutmann, 3. [↑](#footnote-ref-63)
64. Farhi, Goldstone, i Gutmann, 3. [↑](#footnote-ref-64)
65. Farhi, Goldstone, i Gutmann, 3–4, 6–10. [↑](#footnote-ref-65)
66. E. Farhi i in., „Quantum Algorithms for Fixed Qubit Architectures” (arXiv, 17 marzec 2017), 4, http://arxiv.org/abs/1703.06199. [↑](#footnote-ref-66)
67. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem”, 2. [↑](#footnote-ref-67)
68. Prasanna Date, Davis Arthur, i Lauren Pusey-Nazzaro, „QUBO Formulations for Training Machine Learning Models”, *Scientific Reports* 11, nr 1 (11 maj 2021): 1–2, https://doi.org/10.1038/s41598-021-89461-4. [↑](#footnote-ref-68)
69. Lucas, „Ising formulations of many NP problems”, 1–3. [↑](#footnote-ref-69)
70. Tadashi Kadowaki i Hidetoshi Nishimori, „Quantum Annealing in the Transverse Ising Model”, 25 kwiecień 1998, 3, https://doi.org/10.1103/PhysRevE.58.5355. [↑](#footnote-ref-70)
71. Fred Glover, Gary Kochenberger, i Yu Du, „A Tutorial on Formulating and Using QUBO Models” (arXiv, 4 listopad 2019), 5–7, https://doi.org/10.48550/arXiv.1811.11538. [↑](#footnote-ref-71)
72. „Quantum Annealing and Related Optimization Methods | SpringerLink”, 13, 21–22, dostęp 13 październik 2022, https://link.springer.com/book/10.1007/11526216. [↑](#footnote-ref-72)
73. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm”, 4. [↑](#footnote-ref-73)
74. Farhi, Goldstone, i Gutmann, 4. [↑](#footnote-ref-74)
75. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem”, 5–6. [↑](#footnote-ref-75)
76. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm”, 2,4. [↑](#footnote-ref-76)
77. Farhi, Goldstone, i Gutmann, 2. [↑](#footnote-ref-77)
78. Farhi, Goldstone, i Gutmann, 2–3. [↑](#footnote-ref-78)
79. Farhi, Goldstone, i Gutmann, 3. [↑](#footnote-ref-79)
80. Farhi, Goldstone, i Gutmann, 6–7. [↑](#footnote-ref-80)
81. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem”, 12. [↑](#footnote-ref-81)
82. Guillaume Verdon, Michael Broughton, i Jacob Biamonte, „A quantum algorithm to train neural networks using low-depth circuits” (arXiv, 14 grudzień 2017), 3, https://doi.org/10.48550/arXiv.1712.05304. [↑](#footnote-ref-82)
83. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm”, 3. [↑](#footnote-ref-83)
84. Farhi, Goldstone, i Gutmann, 3. [↑](#footnote-ref-84)
85. Farhi, Goldstone, i Gutmann, 7. [↑](#footnote-ref-85)
86. Farhi i in., „Quantum Algorithms for Fixed Qubit Architectures”, 13–14. [↑](#footnote-ref-86)
87. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm”, 10–11. [↑](#footnote-ref-87)
88. Edward Farhi i Aram W. Harrow, „Quantum Supremacy through the Quantum Approximate Optimization Algorithm” (arXiv, 20 październik 2019), 4, https://doi.org/10.48550/arXiv.1602.07674. [↑](#footnote-ref-88)
89. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm”, 15. [↑](#footnote-ref-89)
90. Farhi i Harrow, „Quantum Supremacy through the Quantum Approximate Optimization Algorithm”, 15. [↑](#footnote-ref-90)
91. Farhi i Harrow, 4. [↑](#footnote-ref-91)
92. Farhi i Harrow, 15. [↑](#footnote-ref-92)
93. Akshay i in., „Reachability Deficits in Quantum Approximate Optimization”, 3. [↑](#footnote-ref-93)
94. Akshay i in., 2–4. [↑](#footnote-ref-94)
95. Farhi, Goldstone, i Gutmann, „A Quantum Approximate Optimization Algorithm”. [↑](#footnote-ref-95)
96. Ruslan Shaydulin i Stefan M. Wild, „Exploiting Symmetry Reduces the Cost of Training QAOA”, *IEEE Transactions on Quantum Engineering* 2 (2021): 5, https://doi.org/10.1109/TQE.2021.3066275. [↑](#footnote-ref-96)
97. M. B. Hastings, „Classical and Quantum Bounded Depth Approximation Algorithms” (arXiv, 1 sierpień 2019), 16, https://doi.org/10.48550/arXiv.1905.07047. [↑](#footnote-ref-97)
98. Helmut G. Katzgraber i in., „Seeking Quantum Speedup Through Spin Glasses: The Good, the Bad, and the Ugly”, *Physical Review X* 5, nr 3 (1 wrzesień 2015): 5–8, https://doi.org/10.1103/PhysRevX.5.031026. [↑](#footnote-ref-98)
99. Michael Streif i Martin Leib, „Comparison of QAOA with Quantum and Simulated Annealing” (arXiv, 7 styczeń 2019), https://doi.org/10.48550/arXiv.1901.01903. [↑](#footnote-ref-99)
100. Michael Streif i Martin Leib, „Training the quantum approximate optimization algorithm without access to a quantum processing unit”, *Quantum Science and Technology* 5, nr 3 (12 maj 2020): 034008, https://doi.org/10.1088/2058-9565/ab8c2b. [↑](#footnote-ref-100)
101. Daniel J. Egger, Jakub Marecek, i Stefan Woerner, „Warm-starting quantum optimization”, *Quantum* 5 (17 czerwiec 2021): 479, https://doi.org/10.22331/q-2021-06-17-479. [↑](#footnote-ref-101)
102. Wim Lavrijsen i in., „Classical Optimizers for Noisy Intermediate-Scale Quantum Devices”, w *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2020, 267–77, https://doi.org/10.1109/QCE49297.2020.00041. [↑](#footnote-ref-102)
103. Egger, Marecek, i Woerner, „Warm-starting quantum optimization”, 2. [↑](#footnote-ref-103)
104. Egger, Marecek, i Woerner, 3. [↑](#footnote-ref-104)
105. Egger, Marecek, i Woerner, 4–5. [↑](#footnote-ref-105)
106. Egger, Marecek, i Woerner, 4. [↑](#footnote-ref-106)
107. Egger, Marecek, i Woerner, 7–8. [↑](#footnote-ref-107)
108. Julian Obst, „Parameter Initialization for Warm-Starting QAOA” (masterThesis, 2022), https://doi.org/10.18419/opus-12366. [↑](#footnote-ref-108)
109. Simone Tibaldi i in., „Bayesian Optimization for QAOA” (arXiv, 30 wrzesień 2022), 2, https://doi.org/10.48550/arXiv.2209.03824. [↑](#footnote-ref-109)
110. D. Zhu i in., „Training of quantum circuits on a hybrid quantum computer”, *Science Advances* 5, nr 10 (18 październik 2019): 3, https://doi.org/10.1126/sciadv.aaw9918. [↑](#footnote-ref-110)
111. Tibaldi i in., „Bayesian Optimization for QAOA”, 2–5. [↑](#footnote-ref-111)
112. J. S. Otterbach i in., „Unsupervised Machine Learning on a Hybrid Quantum Computer” (arXiv, 15 grudzień 2017), 3, https://doi.org/10.48550/arXiv.1712.05771. [↑](#footnote-ref-112)
113. Peter I. Frazier, „A Tutorial on Bayesian Optimization” (arXiv, 8 lipiec 2018), 6–7, https://doi.org/10.48550/arXiv.1807.02811. [↑](#footnote-ref-113)
114. Donald R. Jones, Matthias Schonlau, i William J. Welch, „Efficient Global Optimization of Expensive Black-Box Functions”, *Journal of Global Optimization* 13, nr 4 (1 grudzień 1998): 471, https://doi.org/10.1023/A:1008306431147. [↑](#footnote-ref-114)
115. Frazier, „A Tutorial on Bayesian Optimization”, 7. [↑](#footnote-ref-115)
116. Dong C. Liu i Jorge Nocedal, „On the Limited Memory BFGS Method for Large Scale Optimization”, *Mathematical Programming* 45, nr 1 (1 sierpień 1989): 503–28, https://doi.org/10.1007/BF01589116. [↑](#footnote-ref-116)
117. Frazier, „A Tutorial on Bayesian Optimization”, 7. [↑](#footnote-ref-117)
118. Tibaldi i in., „Bayesian Optimization for QAOA”, 3. [↑](#footnote-ref-118)
119. Tibaldi i in., 3. [↑](#footnote-ref-119)
120. Tibaldi i in., 3–4. [↑](#footnote-ref-120)
121. Frazier, „A Tutorial on Bayesian Optimization”, 6. [↑](#footnote-ref-121)
122. Frazier, 5–6. [↑](#footnote-ref-122)
123. Frazier, 6. [↑](#footnote-ref-123)
124. Frazier, 6. [↑](#footnote-ref-124)
125. Tibaldi i in., „Bayesian Optimization for QAOA”, 4. [↑](#footnote-ref-125)
126. Tibaldi i in., 4. [↑](#footnote-ref-126)
127. Jasper Snoek, Hugo Larochelle, i Ryan P Adams, „Practical Bayesian Optimization of Machine Learning Algorithms”, w *Advances in Neural Information Processing Systems*, t. 25 (Curran Associates, Inc., 2012), 3, https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html. [↑](#footnote-ref-127)
128. „Qiskit”, dostęp 5 luty 2023, https://qiskit.org/. [↑](#footnote-ref-128)
129. „Quantum Computing Market Size, Share | Growth Forecast 2029”, dostęp 5 luty 2023, https://www.fortunebusinessinsights.com/quantum-computing-market-104855. [↑](#footnote-ref-129)
130. „Quantum Computing Market Size & Share Forecast Report 2030”, P&S Intelligence, dostęp 5 luty 2023, https://www.psmarketresearch.com/market-analysis/quantum-computing-market. [↑](#footnote-ref-130)
131. McKinsey & Company, „Quantum computing: An emerging ecosystem and industry use cases”, grudzień 2021, https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/quantum-computing-use-cases-are-getting-real-what-you-need-to-know#/. [↑](#footnote-ref-131)
132. „World Development Indicators | DataBank”, dostęp 5 luty 2023, https://databank.worldbank.org/reports.aspx?source=2&series=NY.GDP.MKTP.CD&country=. [↑](#footnote-ref-132)
133. „IBM Unveils 433-Qubit Osprey Chip - IEEE Spectrum”, dostęp 5 luty 2023, https://spectrum.ieee.org/ibm-quantum-computer-osprey. [↑](#footnote-ref-133)
134. „IBM Quantum Computing | Roadmap”, 1 październik 2015, https://www.ibm.com/quantum/www.ibm.com/quantum/roadmap. [↑](#footnote-ref-134)
135. „IBM Introduces CLOPS Performance Standard for Quantum Computing”, Tom’s Hardware, 3 listopad 2021, https://www.tomshardware.com/news/ibm-introduces-clops-performance-standard-for-quantum-computing. [↑](#footnote-ref-135)
136. „IBM Introduces CLOPS Performance Standard for Quantum Computing”. [↑](#footnote-ref-136)
137. Nikolaj Moll i in., „Quantum optimization using variational algorithms on near-term quantum devices”, *Quantum Science and Technology* 3, nr 3 (lipiec 2018): 030503, https://doi.org/10.1088/2058-9565/aab822. [↑](#footnote-ref-137)
138. Charles H. Baldwin i in., „Re-examining the quantum volume test: Ideal distributions, compiler optimizations, confidence intervals, and scalable resource estimations”, *Quantum* 6 (9 maj 2022): 707, https://doi.org/10.22331/q-2022-05-09-707. [↑](#footnote-ref-138)
139. Keith Miller i in., „An Improved Volumetric Metric for Quantum Computers via more Representative Quantum Circuit Shapes” (arXiv, 14 lipiec 2022), http://arxiv.org/abs/2207.02315. [↑](#footnote-ref-139)
140. Moll i in., „Quantum optimization using variational algorithms on near-term quantum devices”. [↑](#footnote-ref-140)
141. McKinsey & Company, „Quantum computing: An emerging ecosystem and industry use cases”. [↑](#footnote-ref-141)
142. „PyQuil: Quantum programming in Python”, Python (2017; repr., Rigetti Computing, 18 marzec 2023), https://github.com/rigetti/pyquil. [↑](#footnote-ref-142)
143. „Quantum Computing”, Rigetti Computing, dostęp 19 marzec 2023, https://www.rigetti.com/. [↑](#footnote-ref-143)
144. „Welcome to Xanadu”, Xanadu, dostęp 19 marzec 2023, https://www.xanadu.ai/. [↑](#footnote-ref-144)
145. „Cirq”, Google Quantum AI, dostęp 19 marzec 2023, https://quantumai.google/cirq. [↑](#footnote-ref-145)
146. MathWorks, „Quantum Computing for Matlab”, dostęp 19 marzec 2023, https://www.mathworks.com/products/quantum-computing.html. [↑](#footnote-ref-146)
147. SoniaLopezBravo, „Introduction to Q# & Quantum Development Kit - Azure Quantum”, 4 luty 2023, https://learn.microsoft.com/en-us/azure/quantum/overview-what-is-qsharp-and-qdk. [↑](#footnote-ref-147)
148. „Quantum Cloud Service - Quantum Computing Service - Amazon Braket - AWS”, Amazon Web Services, Inc., dostęp 19 marzec 2023, https://aws.amazon.com/braket/. [↑](#footnote-ref-148)
149. „IBM Quantum Computing”, 1 październik 2015, https://www.ibm.com/www.ibm.com/quantum. [↑](#footnote-ref-149)
150. „EU Deploys First Quantum Technology in 6 Sites across Europe”, Text, European Commission - European Commission, dostęp 6 luty 2023, https://ec.europa.eu/commission/presscorner/detail/en/IP\_22\_5914. [↑](#footnote-ref-150)
151. „The EU’S Quantum Technologies Flagship | Shaping Europe’s Digital Future”, 26 styczeń 2023, 3, https://digital-strategy.ec.europa.eu/en/library/eus-quantum-technologies-flagship. [↑](#footnote-ref-151)
152. „IBM Quantum”, IBM Quantum, dostęp 1 lipiec 2023, https://quantum-computing.ibm.com/services/resources. [↑](#footnote-ref-152)
153. „NumPyMinimumEigensolver — Qiskit 0.43.2 documentation”, dostęp 1 lipiec 2023, https://qiskit.org/documentation/stubs/qiskit.algorithms.NumPyMinimumEigensolver.html. [↑](#footnote-ref-153)
154. „QAOA documentation”, b.d., https://qiskit.org/documentation/stable/0.30/stubs/qiskit.aqua.algorithms.QAOA.html. [↑](#footnote-ref-154)
155. „Qiskit Runtime Overview”, IBM Quantum, dostęp 1 lipiec 2023, https://quantum-computing.ibm.com/lab/docs/iql/runtime. [↑](#footnote-ref-155)
156. GUS, „Oszacowanie odległości i czasu przejazdu pomiędzy wybranymi gminami w Polsce w 2016 roku”, stat.gov.pl, dostęp 26 czerwiec 2023, https://stat.gov.pl/statystyki-eksperymentalne/obszary-funkcjonalne-oraz-dostepnosc-terytorialna/oszacowanie-odleglosci-i-czasu-przejazdu-pomiedzy-wybranymi-gminami-w-polsce-w-2016-roku,11,1.html. [↑](#footnote-ref-156)