



Studium Magisterskie

Kierunek: Analiza Danych – Big Data

Przemysław Michałowski

Nr albumu 81840

Wykorzystanie algorytmów kwantowych w optymalizacji kombinatorycznej

Praca magisterska
pod kierunkiem naukowym
dra Sebastiana Zająca
Instytut Ekonometrii

Warszawa 2023

SPIS TREŚCI

| | |
|--|-----------|
| Wstęp | 4 |
| ROZDZIAŁ I. Podstawowe pojęcia | 6 |
| I.1 Wprowadzenie do informatyki kwantowej | 6 |
| I.1.1 Kubit jako podstawowa jednostka informacji | 6 |
| I.1.2 Kluczowe zjawiska mechaniki kwantowej | 12 |
| I.1.3 Wykonywanie działań na maszynach kwantowych | 13 |
| I.2 Wprowadzenie do problemu marszrutyzacji | 17 |
| I.2.1 Postać ogólna problemu VRP | 17 |
| I.2.2 Postać matematyczna problemu VRP | 20 |
| I.3 Rozwój i komercjalizacja informatyki kwantowej | 25 |
| ROZDZIAŁ II. Algorytm QAOA | 32 |
| II.1 Zasady działania algorytmu QAOA | 32 |
| II.1.1 QAOA jako algorytm wariacyjny | 32 |
| II.1.2 Założenia algorytmu QAOA | 33 |
| II.1.3 Część kwantowa algorytmu QAOA | 35 |
| II.2 Rozwinięcie algorytmu QAOA | 40 |
| II.2.1 Złożoność obliczeniowa i ograniczenia QAOA | 40 |
| II.2.2 Warm-starting QAOA | 43 |
| II.2.3 Hiperoptrymalizacja Bayesowska | 46 |
| ROZDZIAŁ III. Implementacja algorytmu | 53 |
| III.1 Przygotowanie eksperymentu | 53 |
| III.1.1 Infrastruktura, oprogramowanie i algorytmy | 53 |
| III.1.2 Dane i sformułowanie problemu | 56 |
| III.1.3 Postać QUBO problemu | 59 |
| III.2 Wyniki | 60 |
| III.2.1 Obliczenia klasyczne – hiperoptrymalizacja | 60 |
| III.2.2 Obliczenia klasyczne – czas obliczeń | 63 |
| III.2.3 Obliczenia kwantowe | 65 |
| III.3 Wnioski i dalsze prace | 70 |
| Zakończenie | 74 |
| Bibliografia | 77 |
| Spis tabel | 85 |
| Spis rysunków | 85 |
| Streszczenie | 87 |

WSTĘP

Wraz z rosnącym zapotrzebowaniem na moc obliczeniową, wzrasta potrzeba dostarczania coraz bardziej zaawansowanych maszyn zdolnych do wykonywania tych obliczeń. Do takich urządzeń zalicza się komputery. Początkowo były one wykorzystywane między innymi w obliczeniach naukowych. Pomimo dostępu do coraz silniejszego sprzętu, dokonywanie pewnych obliczeń przysparzało i nadal przysparza wielu trudności – za przykład może posłużyć rozwiązywanie problemów NP-trudnych. Uwagę należy zwrócić na ograniczenia zgłaszane przez chemików i fizyków kwantowych - komplikacje związane z wykorzystaniem komputerów w obliczeniach dotyczących fizyki kwantowej zostały podniesione głośno przez Richarda Feynmana w 1981 na konferencji *The Physics of Computation*. Podczas swojego przemówienia poruszył on konieczność m.in. symulowania prędkości światła oraz fal czy reprezentacji rozkładu gęstości dla zjawisk ciągłych, w tym rozkładów prawdopodobieństwa¹. 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 kwantowego². Choć wcześniej było już kilku naukowców, którzy postulowali i proponowali utworzenie takiej maszyny (jednym z pionierów był Paul Benioff, który poświęcił temu zagadnieniu cały artykuł w 1979 roku³), to dopiero wspomniany wykład spowodował wzrost zainteresowania maszynami, których działanie ma opierać się na i korzystać z zjawisk zachodzących na poziomie kwantowym⁴.

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,

¹ R. P. Feynman, *Simulating Physics with Computers*, "International Journal of Theoretical Physics", czerwiec 1982, nr 21, (s. 467–488), s. 476–477.

² Ibidem.

³ P. Benioff, *The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines*, "Journal of Statistical Physics", czerwiec 1980, nr 22, s. 563-591.

⁴ J. D. Hidary, *A Brief History of Quantum Computing*, w: *Quantum Computing: An Applied Approach* (s. 11-16), red. J. D. Hidary, 2019, wyd. Springer Cham, s. 11–12.

potwierdzenie naruszenia nierówności Bella i pionierski wkład w teorię informacji kwantowej” (*quantum information science*)⁵.

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 aktualnie są to przede wszystkim obszary badań, a nie praktycznych zastosowań – do nich 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 tej technologii. Jako przykład zastosowania komputerów kwantowych zostanie przedstawione zadanie optymalizacyjne zwane problemem marszrutyzacji, ponieważ jest ono często wykorzystywane do sprawdzania możliwości algorytmów optymalizacyjnych ze względu na wyzwanie jakie stanowi. 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 niniejszej pracy umożliwi czytelnikowi zapoznanie w się w I rozdziale z podstawowymi pojęciami związanymi z informatyką kwantową i problemem marszrutyzacji, dzięki czemu będzie możliwe przeprowadzenie logicznego i spójnego wywodu w dalszych rozdziałach bez popadania w dygresje. Ostatnia część rozdziału I została poświęcona rynkowi komercyjnemu oraz inwestycjom publicznym w obszarze informatyki kwantowej; przy okazji przedstawiono przykłady użycia kwantowych algorytmów optymalizacyjnych oraz omówiono szanse na rozwój tej technologii. W rozdziale II opisano badaną metodę optymalizacji – algorytm QAOA, wraz z propozycją hiperoptymalizacji jego parametrów z wykorzystaniem metody Bayesowskiej. Rozdział III stanowi przedstawienie procesu implementacji algorytmu dla przykładowych instancji problemu marszrutyzacji, wraz z omówieniem wyników.

⁵ *The Nobel Prize in Physics 2022*, <https://www.nobelprize.org/prizes/physics/2022/summary>.

ROZDZIAŁ I. Podstawowe pojęcia

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

Na początku omówiono pojęcie kubit, 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. Ostatnia część została poświęcona dokonaniom podmiotów komercyjnych w zakresie obliczeń kwantowych, wraz z szansami na wykorzystanie i rozwój informatyki kwantowej, w szczególności kwantowych algorytmów optymalizacyjnych.

I.1 Wprowadzenie do informatyki kwantowej

I.1.1 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 $N = 2^n$. 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 2^n , gdzie przy pomocy uporządkowanego ciągu n bitów można otrzymać izomorficzne przekształcenie elementów zbioru nieuporządkowanego o liczebności 2^n .

⁶ T. G Wong, *Introduction to Classical and Quantum Computing*, 1 maja 2022, s. 6–7.

W komputerze kwantowym podstawową jednostką informacji jest bit kwantowy zwany też kubit⁷. Fizycznie, może być on realizowany na wiele sposobów, w zależności od architektury maszyny kwantowej, np. poprzez fotony, uwięzione jony, elektrony lub inne cząsteczki kwantowe⁸. Przestrzeń możliwych stanów systemu kwantowego jest opisywana matematycznie przez zespoloną przestrzeń wektorową ze zdefiniowanym na niej iloczynem skalarnym⁹. Tak więc matematycznie kubit jest reprezentowany przez unormowany wektor w dwuwymiarowej przestrzeni Hilberta \mathcal{H} nad ciałem liczb zespolonych¹⁰, a system n -kubitowy – przez unormowany wektor w 2^n -wymiarowej przestrzeni Hilberta.

Na mocy twierdzenia spektralnego każdy ograniczony samosprężony 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 bazowych¹¹. Ograniczonymi operatorami samosprężonymi, 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 $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ oraz $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ oznaczane odpowiednio $|0\rangle$ i $|1\rangle$.

Stany bazowe są to dwa dowolne ortonormalne wektory¹². W dalszej części pracy za stany bazowe będą przyjmowane zawsze stany $|0\rangle$ oraz $|1\rangle$ reprezentowane odpowiednio przez pionowe wektory $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ oraz $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ¹³, tzn. będą one tworzyć obliczeniowy stan bazowy (*computational basis state*), pozwalając na możliwość opisanie przy ich pomocy dowolnego wektora reprezentującego stan systemu. Zostały one wybrane także ze względu na ich prostotę, która znacznie ułatwi dalsze wyprowadzenia.

⁷ M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, wyd. Cambridge University Press, 9 grudnia 2010, s. 13.

⁸ T.G. Wong, op. cit., s. 97.

⁹ M. Nagy, S. G. Akl, *Quantum computation and quantum information*, "The International Journal of Parallel, Emergent and Distributed Systems", luty 2006, 21(1), (s. 1-59), s. 5.

¹⁰ L. Susskind, A. Friedeman, *Mechanika kwantowa. Teoretyczne minimum*, wyd. Prószyński Media Sp. z o.o., Warszawa, 2016, s. 19.

¹¹ R. Walter, *Functional Analysis*, wyd. McGraw-Hill Book Company, 1991, s. 306.

¹² M. A. Nielsen, I. L. Chuang, op. cit., s. 13.

¹³ T. G. Wong, op. cit., s. 115.

Stan kubitów można zapisać jako kombinację liniową stanów bazowych $|0\rangle$ (ket zero) oraz $|1\rangle$ (ket jeden) korzystając z notacji Diraca zwanej też notacją bra-ket¹⁴:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \alpha, \beta \in \mathbb{C}, \quad |\psi\rangle \in \mathcal{H} \quad (1)$$

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

$$|\alpha|^2 + |\beta|^2 = 1, \quad (2)$$

co gwarantuje, że wektor $|\psi\rangle$ będzie długości równej 1 (w normie drugiej). W ogólności dla n -kubitowego systemu kwantowego można otrzymać $N = 2^n$ stanów. Wówczas równanie 2 przyjmuje postać:

$$\sum_{j=0}^{N-1} |\alpha_j|^2 = 1 \quad (3)$$

lub równoważnie:

$$\langle\psi|\psi\rangle = 1, \quad (4)$$

gdzie $\langle\mathbf{u}|\mathbf{v}\rangle$ oznacza iloczyn skalarny wektora \mathbf{u} z wektorem \mathbf{v} .

Korzystając z równania Eulera, można zapisać stan kubitów jako¹⁵:

$$\begin{aligned} |\psi\rangle &= e^{i\lambda} \cos \frac{\theta}{2} |0\rangle + e^{i\gamma} \sin \frac{\theta}{2} |1\rangle \\ &= e^{i\lambda} \left(\cos \frac{\theta}{2} |0\rangle + e^{i(\gamma-\lambda)} \sin \frac{\theta}{2} |1\rangle \right) \\ &= e^{i\lambda} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right), \\ \lambda, \gamma, \phi &\in [0, 2\pi], \quad \theta \in [0, \pi] \end{aligned} \quad (5)$$

Kąty λ oraz γ sterują kątami obrotów wektorów stanów $|0\rangle$ i $|1\rangle$ po jednostkowym okręgu zespolonym. Z kolei kąt $\frac{\theta}{2}$ steruje długością wektorów bazowych, tzn. w równaniu 2: $|\alpha| = \cos \frac{\theta}{2}$ i $|\beta| = \sin \frac{\theta}{2}$, a więc kąt $\frac{\theta}{2}$ ma wpływ na udział stanów bazowych w superpozycji stanu reprezentowanego przez wektor $|\psi\rangle$ lub innymi słowy: wpływa na długość wektora będącego projekcją wektora $|\psi\rangle$ rzutowanego ortogonalnie na podprzestrzeń rozpinaną przez wektor $|0\rangle$ i ortogonalną do $|1\rangle$ 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 $|0\rangle$ i $|1\rangle$ są dwuelementowe, dlatego łączna liczba wymiarów wymaganych do geometrycznego

¹⁴ P. A. M. Dirac, *A New Notation for Quantum Mechanics*, "Mathematical Proceedings of the Cambridge Philosophical Society", lipiec 1939, 35(3), s. 416–418.

¹⁵ M. A. Nielsen, I. L. Chuang, op. cit., s. 13.

przedstawienia wektora $|\psi\rangle$ wzrasta do czterech. Elementy $e^{i\lambda} \cos \frac{\theta}{2}$ oraz $e^{i\gamma} \sin \frac{\theta}{2}$ w równaniu 5 odpowiadają dokładnie amplitudom α i β w równaniu 1. Ponadto warto zauważyć, że $e^{i\lambda}$ oraz $e^{i\gamma}$ w równaniu 5 mają długość w normie drugiej równą 1, nie mają więc wpływu na wielkość $|\alpha|^2$ i $|\beta|^2$ w równaniu 2. Zgodnie z równaniem 5 reprezentacją geometryczną wektora $|\psi\rangle$ 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) postaci¹⁶:

$$\langle \mathbf{u} | \mathbf{v} \rangle^2 \leq \langle \mathbf{u} | \mathbf{u} \rangle \langle \mathbf{v} | \mathbf{v} \rangle \quad (6)$$

jest równością. Można zauważyć, że dwa wektory $|\psi_1\rangle$ oraz $|\psi_2\rangle$ będą spełniały powyższy warunek jako równość wtedy i tylko wtedy, kiedy będą miały dokładnie taką samą część $(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle)$, tj. identyczne kąty θ oraz ϕ , natomiast kąt λ nie będzie miał wpływu na nierówność CBS, czyli w równaniu 5 można zignorować $e^{i\lambda}$. Korzystając z powyższego, element $e^{i\lambda}$, zwany też fazą globalną, można pominąć, ponieważ nie powoduje on obserwowalnych zmian, więc z perspektywy obliczeń kwantowych będzie on nieistotny i w rezultacie otrzymano¹⁷:

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle, \quad \phi \in [0, 2\pi], \quad \theta \in [0, \pi] \quad (7)$$

gdzie θ oznacza kąt obrotu po płaszczyźnie tworzonej przez część rzeczywistą wektorów bazowych, a ϕ to kąt obrotu wokół osi wyznaczonej przez wektor $|1\rangle$. Dla uproszczenia notacji założono, że współczynnik przy wektorze $|0\rangle$ jest rzeczywisty i nieujemny. Element $e^{i\phi}$ jest nazywany fazą lokalną lub relatywną.

Stan kubitów można reprezentować przy pomocy wektora w 3-wymiarowej sferze (rysunek 1), zwanej sferą Blocha¹⁸. Warto zwrócić uwagę, że zgodnie z równaniem 1 i równaniem 2 reprezentacja graficzna powinna mieć charakter 4-wymiarowy, jednakże zgodnie z równaniem 7 można ograniczyć tę reprezentację do trzech wymiarów. Będzie

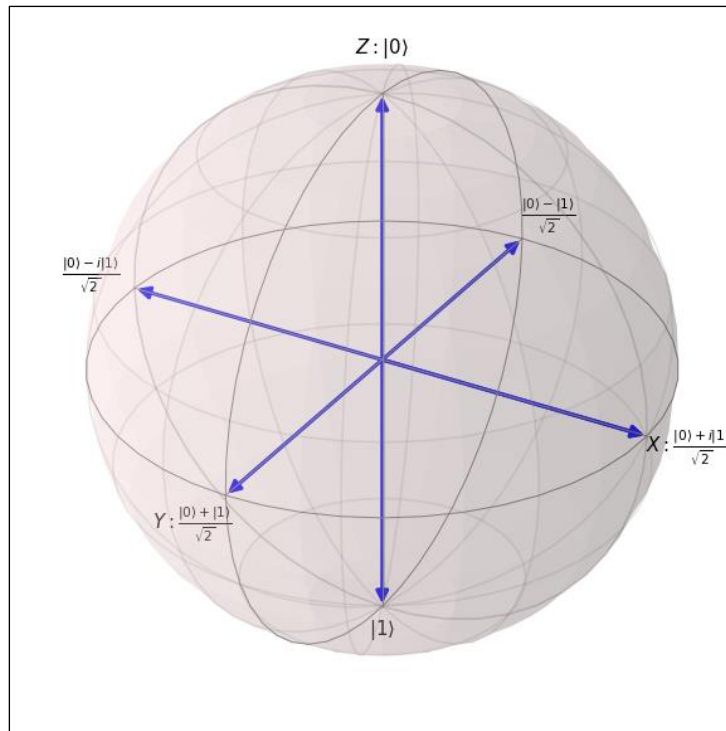
¹⁶ D. Joyce, *Cauchy's Inequality*, wyd. Clark University, 2015.

¹⁷ T. G. Wong, op. cit., s. 91-93.

¹⁸ Ibidem, s. 76.

to jednak skutkowało utratą ortogonalności stanów bazowych w wizualizacji. Stąd też w równaniu 7 kąty w funkcjach trygonometrycznych są dzielone przez dwa, ponieważ θ to kąt obrotu wokół osi Y w sferze Blocha. Stany bazowe znajdują się na krańcach osi Z sfery Blocha. W sferze Blocha ϕ określa kąt obrotu wokół osi Z.

Rysunek 1. Wizualizacja sfery Blocha.



Źródło: opracowanie własne na podstawie M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, wyd. Cambridge University Press, 9 grudnia 2010, s. 15.

Wybór wektorów $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ oraz $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ na wektory bazowe wynika m.in. z powszechności ich użycia. Ta z kolei jest następstwem 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: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ dla osi X, $\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ dla osi Y oraz $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ dla osi Z, od nazw osi pochodzą oznaczenia tych macierzy, odpowiednio: σ_x , σ_y , σ_z . Macierze te są też zwane macierzami Pauliego¹⁹. 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:

¹⁹ M. A. Nielsen, I. L. Chuang, op. cit., s. 65.

- dla σ_x : $\begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}$ oraz $\begin{bmatrix} 1 \\ -\sqrt{2} \end{bmatrix}$ – wektory te są też oznaczane w informatyce kwantowej jako wektory stanów $|+\rangle$ (ket-plus) oraz $|-\rangle$ (ket-minus);
- dla σ_y : $\begin{bmatrix} 1 \\ i \end{bmatrix}$ oraz $\begin{bmatrix} 1 \\ -i \end{bmatrix}$;
- dla σ_z : $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ oraz $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

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 ortonormalne²⁰. 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.

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). Obwodem kwantowym będzie nazywany zbiór rejestrów klasycznych i kwantowych wraz z dokonywanymi na nich operacjami. Wektor określający stan systemu, w którym wszystkie n kubitów znajduje się w stanie $|0\rangle$ będzie oznaczany z wykorzystaniem symbolu iloczynu tensorowego (tożsamego tutaj z iloczynem Kroneckera, wyjaśnionego dokładniej w rozdziale I.1.3) $|0\rangle^{\otimes n}$ lub przez pogrubienie $|0\rangle$, jeśli liczba kubitów w stanie $|0\rangle$ będzie jasno wynikać z kontekstu. Warto zwrócić uwagę, że symbol iloczynu Kroneckera może być zastąpiony (o ile stany nie są splątane – patrz rozdział I.1.2) iloczynem $\prod_{i=1}^n |0\rangle$.

²⁰ L. Susskind, A. Friedeman, op. cit., s. 73–74.

I.1.2 Kluczowe zjawiska mechaniki kwantowej

Jednym ze zjawisk charakterystycznych dla mechaniki kwantowej, które jest wykorzystywane na szeroką skalę w obliczeniach kwantowych jest superpozycja. Oznacza ona, 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 α i β definiujące bieżący stan kubitów są takie, że $|\alpha|, |\beta| \notin \{0,1\}$ ²¹.

Długotrwałe utrzymanie kubitów fizycznych 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 ze stanu superpozycji do jednego ze stanów bazowych. Znajdując się w superpozycji, stan kubitów (a więc i wektora go reprezentującego) również może się zmieniać, ale bez doprowadzania do redukcji. Niekontrolowane zmiany stanu kubitów stanowią jeden z kluczowych problemów stabilności urządzeń kwantowych. Oddziaływanie kubitów z otoczeniem nosi nazwę dekoherencji. Jeżeli jest ona niezamierzona, wówczas jej skutki nazywa się szumem kwantowym (*quantum noise*)^{22 23}.

Przejście ze stanu superpozycji do stanu bazowego może być również następstwem zamierzonym działań, tj. wykonania pomiaru (obserwacji). Pomiary stanowią często jeden z końcowych elementów algorytmów kwantowych bądź też ich istotnych części. W jego wyniku kubit przechodzi do jednego ze stanów bazowych²⁴. W wyniku dekoherencji (niezamierzonej lub spowodowanej pomiarem) kubit znajdzie się w stanie $|0\rangle$ lub $|1\rangle$ z prawdopodobieństwem równym odpowiednio $|\alpha|^2$ lub $|\beta|^2$, co jest zgodne z aksjomatyczną definicją prawdopodobieństwa dzięki równaniu 2.

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 obserwabłą, jego wektory własne muszą tworzyć bazę przestrzeni Hilberta. W takim ujęciu prawdopodobieństwo otrzymania w pomiarze wartości \hat{a}_i jest równe:

$$P_i = |\langle a_i | \psi \rangle|^2 = \langle a_i | \psi \rangle \langle \psi | a_i \rangle, \quad (8)$$

²¹ M. A. Nielsen, I. L. Chuang, op. cit., s. 13-14.

²² T. G. Wong, op. cit., s. 190.

²³ M. A. Nielsen, I. L. Chuang, op. cit. s. 278, 353–394.

²⁴ Ibidem, s. 13–14.

gdzie $\langle a_i |$ jest sprzężeniem hermitowskim wektora własnego obserwabli odpowiadającego wartości własnej \hat{a}_i , a $|\psi\rangle$ wektorem reprezentującym stan kwantowy, na którym dokonano pomiaru²⁵. Przyjmując poprzednie założenia, tj., że stany bazowe to $|0\rangle$ oraz $|1\rangle$, wówczas macierz σ_z jest obserwabłą systemu kwantowego. 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 σ_z (tj. dokonania pomiaru w bazie tworzonej przez wektory własne macierzy tej obserwabli, tutaj zwanej bazą Z) możliwe do otrzymania stany to nie 1 i -1, lecz $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ oraz $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, czyli $|0\rangle$ oraz $|1\rangle$. 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. Polega ono 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 kubit. Konsekwencją tego jest możliwość wnioskowania na temat stanu jednego kubit na podstawie stanu innego kubit. 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 intencjonalne²⁶.

I.1.3 Wykonywanie działań na maszynach kwantowych

Działania na komputerach klasycznych są wykonywane z pomocą bramek logicznych i algebry boolowskiej. Do przykładowych bramek logicznych można zaliczyć AND, OR, XOR oraz NOT²⁷. 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. Realizacja działań reprezentowanych przez bramki, polegać może

²⁵ L. Susskind, A. Friedeman, op. cit., s. 79-81.

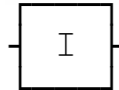
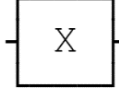

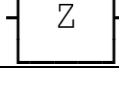
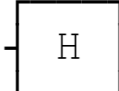
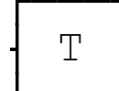
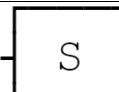

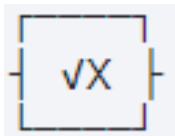
²⁶ T. G. Wong, op. cit., s. 237-251.

²⁷ Ibidem, s. 11-15.

(w zależności od budowy maszyny) np. na oddziaływaniu na fizyczną realizację kubitu impulsem elektromagnetycznym²⁸.

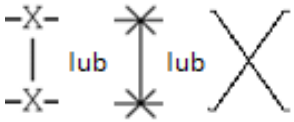
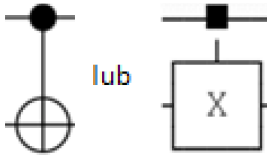
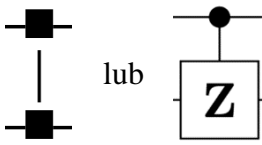
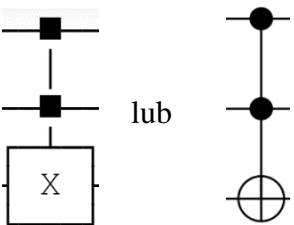
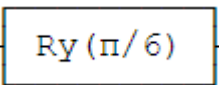
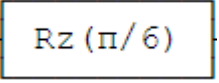
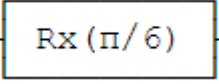
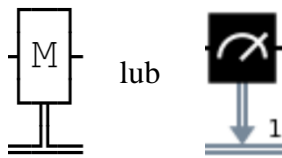
Niektóre bramki kwantowe mają swoje nazwy oraz dedykowane oznaczenia. Mogą one mieć postać stałą lub parametryzowaną, działać zarówno na jednym jak i na wielu kubitach. Mają one również swoje oznaczenia graficzne, które są używane w wizualizacjach obwodów kwantowych, przy pomocy których można graficznie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

Tabela 1. Podstawowe operacje kwantowe i ich oznaczenia (cz. 1/2).

| Nazwa operatora | Skrót | Macierz | Symbol graficzny |
|-------------------------------------|--------------------|---|---|
| Identity | I, ID | $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ |  |
| Pauli-X, NOT | X, NOT, σ_x | $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ |  |
| Pauli-Y | Y, σ_y | $\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ |  |
| Pauli-Z | Z, σ_z | $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ |  |
| Hadamard | H | $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$ |  |
| T, $\pi/8$ | T | $\begin{bmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{bmatrix}$ |  |
| Phase | S, P | $\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ |  |
| S-dagger, sprzężenie hermitowskie S | S^\dagger | $\begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$ |  |
| Squared X | SX | $\begin{bmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{bmatrix}$ |  |

²⁸ M. A. Nielsen, I. L. Chuang, op. cit., s. 13-14.

Tabela 1. Podstawowe operacje kwantowe i ich oznaczenia (cz. 2/2).

| | | | |
|-------------------------|--|--|---|
| SWAP | SWAP | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |  |
| Controlled NOT | CX, CNOT, C ₁ NOT, C ₁ X | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ |  |
| Controlled Z | CZ | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$ |  |
| Toffoli | CCNOT, CCX, TOFF, C ₁ C ₁ NOT, C ₁ C ₁ X | $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ |  |
| Ry | Ry | $\begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}$ |  |
| Rz | Rz | $\begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$ |  |
| Rx | Rx | $\begin{bmatrix} \cos \frac{\theta}{2} & -e^{i\frac{\pi}{2}} \sin \frac{\theta}{2} \\ e^{i\frac{\pi}{2}} \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}$ |  |
| Pomiar (Measurement) | M | - |  |

Źródło: opracowanie własne na podstawie opracowanie własne na podstawie M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, wyd. Cambridge University Press, 9 grudnia 2010.

Ogólna postać kwantowej bramki unarnej (bramki działającej na jednym kubicie) jest reprezentowana przez macierz²⁹:

$$U(\omega, \phi, \lambda, \theta) = \begin{bmatrix} e^{i(\omega - \frac{\phi}{2} - \frac{\lambda}{2})} \cos \frac{\theta}{2} & -e^{i(\omega - \frac{\phi}{2} + \frac{\lambda}{2})} \sin \frac{\theta}{2} \\ e^{i(\omega + \frac{\phi}{2} - \frac{\lambda}{2})} \sin \frac{\theta}{2} & e^{i(\omega + \frac{\phi}{2} + \frac{\lambda}{2})} \cos \frac{\theta}{2} \end{bmatrix} \quad (9)$$

lub alternatywnie:

$$U(\lambda, \phi, \theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\lambda+\phi)} \cos \frac{\theta}{2} \end{bmatrix}, \quad (10)$$

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

$$U(\alpha, \phi, \lambda, \theta) = e^{i(\omega - \frac{\beta}{2} - \frac{\gamma}{2})} U(\lambda, \phi, \theta). \quad (11)$$

Bramkę tę można przedstawić w sposób umożliwiający odczytanie obrotu wokół poszczególnych osi sfery Blocha³⁰:

$$e^{\alpha} [\cos \left(\frac{\theta}{2} \right) I - i \sin \left(\frac{\theta}{2} \right) (n_x X + n_y Y + n_z Z)], \quad (12)$$

gdzie:

$$n_x^2 + n_y^2 + n_z^2 = 1, \quad n_x, n_y, n_z \in \mathbb{R}, \quad (13)$$

co gwarantuje, że wektor, na którym zastosowano macierz, nie zmieni swojej długości. Poszczególne skalary n_x , n_y , n_z reprezentują obrót wokół odpowiednich osi (patrz: równanie 16). Warto przypomnieć uogólnienie tożsamości Eulera na eksponentę macierzy³¹:

$$e^{i\tau A} = \cos(\tau) I + i \sin(\tau) A, \quad (14)$$

gdzie $A^2 = I$. 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):

$$e^{-i\theta X} = e^{-i\theta \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}} \approx \begin{bmatrix} \cos \frac{\theta}{2} & -e^{i\frac{\pi}{2}} \sin \frac{\theta}{2} \\ -e^{i\frac{\pi}{2}} \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} = R_x. \quad (15)$$

Analogicznie, po rozwinięciu równania 12 można otrzymać:

$$e^{\alpha} [\cos \left(\frac{\theta}{2} \right) I - i \sin \left(\frac{\theta}{2} \right) (n_x X + n_y Y + n_z Z)] = e^{\alpha} e^{i\frac{\theta}{2}(n_x X + n_y Y + n_z Z)}, \quad (16)$$

²⁹ Ibidem, s. 98–110.

³⁰ T. G. Wong, op. cit., s. 108–110.

³¹ G. E. Crooks, *Gates, States, and Circuits*, 1 marca 2023, s. 10.

co zgodnie z równaniem 12 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 niezmięnionej długości.

Podobnie jak w przypadku wektorów, również operatory można łączyć wykorzystując iloczyn tensorowy (Kroneckera). Wówczas do symbolu bramki kwantowej można 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 zmieniają swojego stanu, dlatego zostanie wprowadzona notacja:

$$A_k = \begin{cases} A \otimes \bigotimes_{i=k+1}^{N-1} I & \text{jeżeli } k = 0 \\ \bigotimes_{i=0}^{k-1} I \otimes A \otimes \bigotimes_{i=k+1}^{N-1} I & \text{jeżeli } 1 < k < N - 1, \\ \bigotimes_{i=0}^{k-1} I \otimes A & \text{jeżeli } k = N - 1 \end{cases} \quad (17)$$

gdzie: $k, N \in \mathbb{N} \wedge N > 0 \wedge k \in [0, N - 1] \wedge A \in \mathbb{C}^{2 \times 2}$.

I.2 Wprowadzenie do problemu marszrutyzacji

I.2.1 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 Whitneya³², ujęty matematycznie przez

³² M. M. Flood, *The Traveling-Salesman Problem*, "Operations Research", luty 1956, 4(1), (s. 61-75), s. 61.

Merilla Flooda w 1955 roku^{33 34}, a następnie uogólniony pod nazwą Truck Dispatching Problem przez Georga Dantzig i Johna Ramsera poprzez dodanie wymogu powrotu do pozycji startowej po odwiedzeniu określonej liczby lokacji³⁵. Udowodniono, że problem TSP jest NP-trudny³⁶, przy czym wykazano także, że jego decyzyjna wersja jest także NP-zupełna przez sprowadzenie do problemu grafu hamiltonowskiego, który jest problemem NP-zupełnym³⁷.

Już Dantzig i Ramser we wspomnianym wcześniej artykule zauważyli, że problem opisany przez Flooda można rozszerzyć³⁸. Z czasem rzeczywiście pojawiało się coraz więcej rozszerzeń, rozmyło się również nazewnictwo³⁹. Uogólnienie problemu TSP zaczęło nosić nazwę problemu marszrutyzacji (Vehicle Routing Problem – VRP⁴⁰ lub General Routing Problem - GRP⁴¹). Ich cechą wspólną jest również to, że podobnie jak problem TSP, są problemami NP-trudnymi z wersjami decyzyjnymi NP-zupełnymi^{42 43}. 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 m_i , $i \in \{1, 2, \dots, k\}$;
- problem VRP z ograniczoną pojemnością pojazdów;

³³ Ibidem.

³⁴ G. B. Dantzig, J. H. Ramser, *The Truck Dispatching Problem*, "Management Science", październik 1959, 6(1), s. 80–81.

³⁵ Ibidem.

³⁶ V. R. Pratt, *An $n \log n$ Algorithm to Distribute n Records Optimally in a Sequential Access File*, w: *Complexity of Computer Computations. The IBM Research Symposia Series*, (s. 111-118), red. R. E. Miller, J. W. Thatcher, J. D. Bohlinger, wyd. Springer / Plenum Press, Nowy Jork, 1972, s. 112-118.

³⁷ R. M. Karp, *Reducibility among Combinatorial Problems*, w: *Complexity of Computer Computations. The IBM Research Symposia Series*, (s. 85-103), red. R. E. Miller, J. W. Thatcher, J. D. Bohlinger, wyd. Springer / Plenum Press, Nowy Jork, 1972, s. 94.

³⁸ G. B. Dantzig, J. H. Ramser, op. cit., s. 81.

³⁹ C. S. Orloff, *A Fundamental Problem in Vehicle Routing*, "Networks", 1974, 4(1), (s. 35-64), s. 35–36.

⁴⁰ J. K. Lenstra, A. H. G. R. Kan, *Complexity of Vehicle Routing and Scheduling Problems*, "Networks", 1 czerwca 1981, 11(2), s. 221–227.

⁴¹ C. S. Orloff, op. cit.

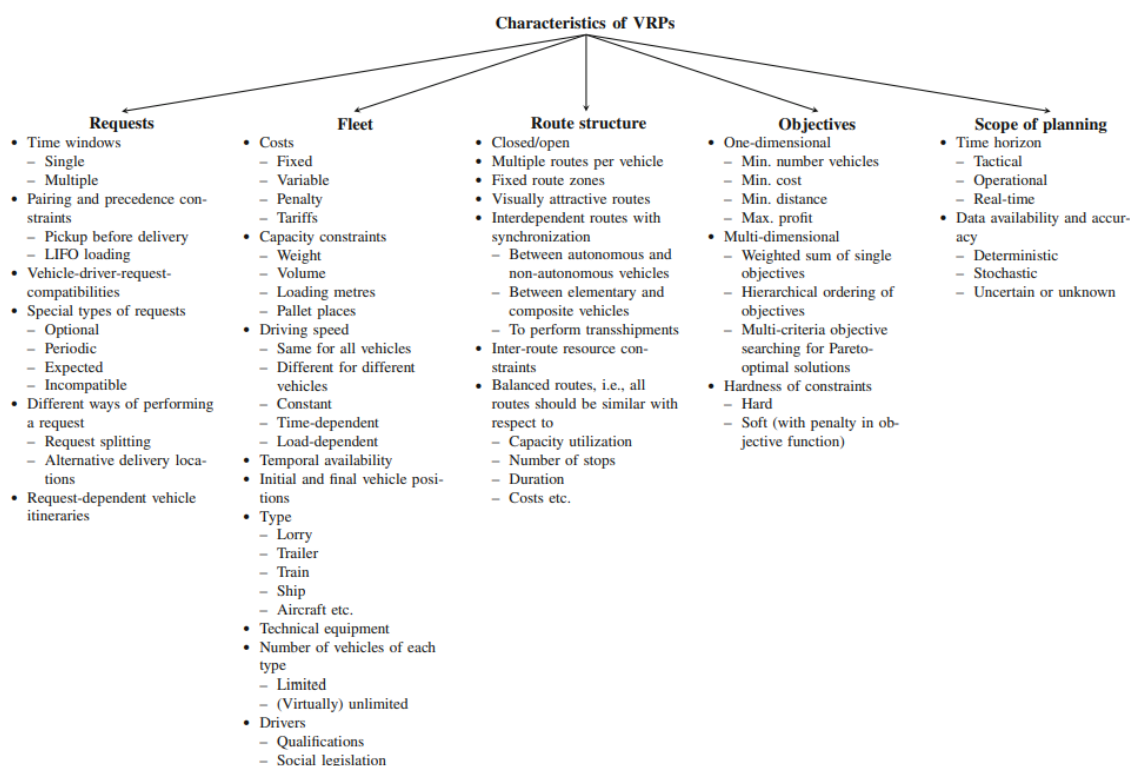
⁴² R. M. Karp, op. cit.

⁴³ J. K. Lenstra, A. H. G. R. Kan, op. cit., s. 223.

- problem VRP o różnej wartości kosztu przejścia pomiędzy tymi samymi miejscami, tzn. $d(a,b) \neq d(b,a)$ – wówczas graf reprezentujący problem będzie grafem skierowanym;
- problem VRP z oknami czasowymi (VRPTW);
- problem VRP z wieloma punktami zaopatrzenia.

Powyższa lista nie wyczerpuje możliwych modyfikacji problemu VRP. Możliwe jest również łączenie tych modyfikacji. Jak zauważył M. Drexel, mnogość publikacji na temat problemu VRP wynika z kilka powodów: pierwszy to jego znaczenie dla logistyki (choć warto zauważyć, że problemy logistyczne nie dotyczą wyłącznie transportu); drugi to wyzwanie intelektualne, jakie stanowi ten problem⁴⁴. Na rysunku 2 przedstawiono przykładowy podział problemów VRP zaproponowany przez Drexla, ukazując tym samym mnogość możliwych modyfikacji.

Rysunek 2: Odmiany problemu VRP.



Źródło: M. Drexel, *Rich Vehicle Routing in Theory and Practice*, "Logistics Research", 1 sierpnia 2012, 5(1), s. 49.

⁴⁴ M. Drexel, *Rich Vehicle Routing in Theory and Practice*, "Logistics Research", 1 sierpnia 2012, 5(1), (s. 47-63), s. 47.

I.2.2 Postać matematyczna problemu VRP

Będąc świadomym różnych ujęć i rozwinięć problemu VRP Raff zaproponował ogólne ujęcie matematyczne problemu⁴⁵, które przedstawiono poniżej (zastosowano zmiany w oznaczeniach).

Niech graf $G = [V, A, C]$ składa się z V wierzchołków (*vertices*), A krawędzi (*edges*) oraz macierzy kosztów C , która przyjmujemy, że jest symetryczna, tzn. $c_{ij} = c_{ji}$. Celem jest minimalizacja funkcji kosztu:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (18)$$

pod warunkami:

$$\sum_{i=1}^n x_{ij} = b_j = 1 \quad (j = 1, \dots, n), \quad (19)$$

$$\sum_{j=1}^n x_{ij} = a_i = 1 \quad (i = 1, \dots, n), \quad (20)$$

$$X = (x_{ij}) \in S, \quad (21)$$

gdzie:

$$x_{ij} = \begin{cases} 1 & \text{jeśli krawędź łącząca wierzchołki } i \text{ oraz } j \text{ należy do rozwiązania} \\ 0 & \text{w przeciwnym przypadku} \end{cases},$$

c_{ij} – koszt przejazdu z i -tego do j -tego wierzchołka,

x_{ij} – informacja binarna czy przejazd odbywa się przez krawędź łączącą i -ty wierzchołek z j -tym (0 – nie odbywa się, 1 – odbywa się),

b_j – liczba przyjazdów do j -tego wierzchołka,

a_i – liczba wyjazdów z i -tego wierzchołka,

n – liczba wierzchołków, $n = |V|$,

X – zbiór możliwych rozwiązań o długości n ,

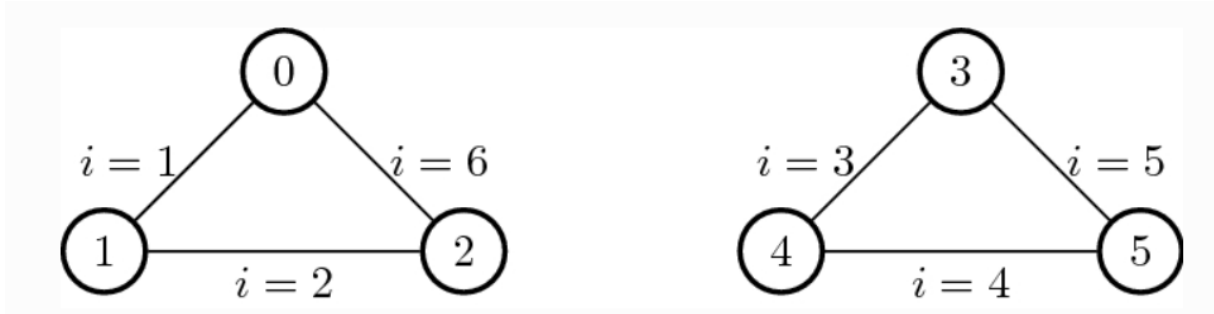
S – zbiór wszystkich możliwych kombinacji połączeń wierzchołków.

Warunki określone równaniami 19 i 20 gwarantują, że każdy wierzchołek zostanie odwiedzony dokładnie jeden raz oraz, że z każdego wierzchołka pojazd wyjedzie dokładnie

⁴⁵ S. Raff, *Routing and Scheduling of Vehicles and Crews: The State of the Art*, "Computers & Operations Research", 1983, 10(2), s. 83–86, [https://doi.org/10.1016/0305-0548\(83\)90030-8](https://doi.org/10.1016/0305-0548(83)90030-8).

jeden raz. Warunek związany ze zbiorem S w równaniu 21 dotyczy powstawania zamkniętych cykli niepołączonych z punktem startowym, tzn. dla każdego wierzchołka musi istnieć połączenie z wierzchołkiem 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 3). W przeciwieństwie do poprzednich dwóch ograniczeń lokalnych, warunek dotyczący istnienia jednego zamkniętego cyklu jest warunkiem o charakterze globalnym. W zależności od konkretnego problemu VRP może on ulegać istotnym modyfikacjom. Poniżej przedstawiono kilka najważniejszych metod i podejść do sformułowania tego ograniczenia, które daje się zastosować w większości problemów VRP (założono optymalizację na grafie nieskierowanym). Warto także zauważyć, że dla grafu nieskierowanego, każdy wierzchołek może być grafem startowym, a więc można powiedzieć, że opisywany warunek będzie spełniony, jeżeli graf wynikowy będzie grafem spójnym. Z tego powodu, tam gdzie to możliwe, w dalszej części pracy ograniczenie z równania 21 będzie nazywane warunkiem spójności grafu.

Rysunek 3. Przykład rozwiązania problemu VRP, dla którego nie jest spełniony warunek spójności grafu.



Źródło: Ö. Salehi, A. Glos, J. A. Mischak, *Unconstrained Binary Models of the Travelling Salesman Problem Variants for Quantum Optimization*, "Quantum Information Processing", 22 stycznia 2022, 21(2), s. 7.

Bektas zaproponował następującą formułę dla warunku z równania 21⁴⁶:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V \setminus \{1\}, \quad S \neq \emptyset, \quad (22)$$

która jest równoważna:

⁴⁶ T. Bektas, *The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures*, "Omega", czerwiec 2006, 34(3), (s. 63-211), s. 83-84.

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq 1, \quad \forall S \subseteq V \setminus \{1\}, \quad S \neq \emptyset, \quad (23)$$

gdzie $V \setminus \{1\}$ oznacza wszystkie wierzchołki z wyjątkiem tego, który oznacza punkt startowy (tzn. $i, j \neq 1$), a $|S|$ oznacza liczbę podtras będących częścią opracowanej trasy. Takie ujęcie warunku nazywa się też jawnym ograniczeniem Dantzig-Fulkersona-Johnsona (choć nie zostało przez nich opracowane tylko przez Tolgę Bektasa). Powoduje ono, że liczba warunków ograniczających rośnie w tempie wykładniczym wraz ze wzrostem liczby wierzchołków⁴⁷. Wynika to z tego, że zbiór S jest zbliżony liczebnością do zbioru potęgowego zbioru V , a konkretnie $|S| = |V|^2 - 1$, ze względu na wykluczenie zbioru pustego. Głównym zamysłem równania 22 jest to, że liczba krawędzi w każdym z podzbiorów zbioru V 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 V , jest właśnie graf V . Równanie 23 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 Dantzig-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 spełnienia warunku spójności grafu. Jeśli rozwiązanie okaże się niedopuszczalne, wówczas do zestawu ograniczeń zostaje dodany warunek, który nie zezwala na ponowne powstanie otrzymanego w danej iteracji grafu. Metoda ta pozostawia dowolność w kwestii definiowania warunku ograniczającego. Takie ujęcie pozwala zmniejszyć liczbę warunków ograniczających i okazało się efektywne, dopóki nie wynaleziono kolejnych metod⁴⁸.

Grupę metod bazujących na podziale przestrzeni rozwiązań dla dużych problemów TSP/VRP omawia Applegate et al. Przedstawiona 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

⁴⁷ Ibidem.

⁴⁸ G. Dantzig, R. Fulkerson, S. Johnson, *Solution of a Large-Scale Traveling-Salesman Problem*, "Journal of the Operations Research Society of America", 1 listopada 1954, 2(4), (s. 393-410), s. 397-398.

w odniesieniu do powstałych grup wierzchołków, a nie całego zbioru V ⁴⁹. 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 wierzchołków, agent musi wrócić do punktu startowego uzupełnić zapas), dla którego liczba ograniczeń związanych ze spójnością grafu rośnie w tempie wielomianowym $O(n^2 + n)$ ⁵⁰. Kulkarni et al. te ograniczenia uogólnił⁵¹, a Kara et al. poprawił i przeformułował do postaci dwóch zestawów następujących nierówności⁵²:

$$u_i - u_j + Qx_{ij} \leq Q - q_j \quad (i, j = 2, \dots, n, \quad i \neq j), \quad (24)$$

$$q_i \leq u_i \leq Q \quad (i = 2, \dots, n), \quad (25)$$

gdzie:

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

Q – maksymalna pojemność pojazdu,

q_i – zapotrzebowanie zgłaszane w i -tym wierzchołku.

Warto zauważyć, że warunki opisane przez nierówności 24 i 25 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 $Q = |V|$ oraz $q_j = 1$.

Powyższe ujęcia warunku spójności grafu, 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

⁴⁹ D. Applegate et. al., *Implementing the Dantzig-Fulkerson-Johnson Algorithm for Large Traveling Salesman Problems*, "Mathematical Programming", lipiec 2003, 97(1), s. 91–153.

⁵⁰ C. E. Miller, A. W. Tucker, R. A. Zemlin, *Integer Programming Formulation of Traveling Salesman Problems*, "Journal of the ACM", październik 1960, 7(4), s. 326–329.

⁵¹ R. V. Kulkarni, P. R. Bhawe, *Integer Programming Formulations of Vehicle Routing Problems*, "European Journal of Operational Research", kwiecień 1985, 20(1), (s. 58–67), s. 63–66.

⁵² I. Kara, G. Laporte, T. Bektas, *A Note on the Lifted Miller–Tucker–Zemlin Subtour Elimination Constraints for the Capacitated Vehicle Routing Problem*, "European Journal of Operational Research", listopad 2004, 158(3), s. 793–795.

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 skierowanych⁵³:

$$h(x) = \sum_{(uv) \notin E} \sum_{j=1}^N x_{u,j} x_{v,j+1}, \quad (26)$$

gdzie:

$x_{u,j}$ – zmienna binarna określająca fakt odwiedzenia u -tego wierzchołka w czasie j ,

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

Suma w równaniu 26 będzie równa zero, jeżeli nie będzie istniało połączenie pomiędzy wierzchołkami nie należącymi do rozwiązania. Kłopotliwe w tym ujęciu jest natomiast zdefiniowanie zbioru E , ponieważ metoda ta zakłada obliczanie funkcji kary po wyznaczeniu rozwiązania.

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 osiągnięcia rozwiązań niedopuszczalnych, natomiast nakłada karę na funkcję celu za niespełnienie warunku. Warunek ten ma postać następującego sformułowania⁵⁴:

$$\begin{aligned} & \sum_{j=2}^n \left(1 - \sum_{k=2}^n x_{1,j,1} x_{j,k,2} \right)^2 \\ & + \sum_{t=2}^{n-2} \sum_{j=2}^n \left(1 - \sum_{\substack{i,k=2 \\ i \neq k}}^n x_{i,j,t} x_{j,k,t+1} \right)^2 \\ & + \sum_{j=2}^n \left(1 - \sum_{i=2}^n x_{i,j,n-1} x_{j,1,n} \right)^2, \end{aligned} \quad (27)$$

gdzie:

$$x_{i,j,t} = \begin{cases} 1 & \text{jeśli agent przechodzi z } i - \text{tego do } j - \text{tego wierzchołka w } t - \text{tym kroku} \\ 0 & \text{w przeciwnym przypadku} \end{cases},$$

⁵³ A. Lucas, *Ising formulations of many NP problems*, "Frontiers in Physics", 2014, nr 2, s. 17.

⁵⁴ Ö. Salehi, A. Glos, J. A. Miszczak, *Unconstrained Binary Models of the Travelling Salesman Problem Variants for Quantum Optimization*, "Quantum Information Processing", 22 stycznia 2022, 21(2), s. 7–8.

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 $((t + 1)$ -szym) pojazd będzie startował z innego wierzchołka niż ten, w którym skończył w kroku t , 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 kary⁵⁵. 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 warunków ograniczających.

I.3 Rozwój i komercjalizacja informatyki kwantowej

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 – część 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.

Do najbardziej znaczących podmiotów na rynku technologii kwantowych zaliczają się giganci technologiczni o ugruntowanej pozycji na rynku, w szczególności IBM, Microsoft, Google oraz Amazon – każda z tych firm dysponuje dużymi udziałami na szeroko pojętym rynku informatycznym. Warto również wspomnieć, że IBM jako jedyny umożliwia bezpłatny dostęp indywidualnym osobom do infrastruktury kwantowej. Robi to za pomocą swojej

⁵⁵ Ö. Salehi, A. Glos, J. A. Miszczak, op. cit., s. 8–11.

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 klasycznych⁵⁶. Fakt ten zostanie wykorzystany w eksperymentach w rozdziale III.

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^{57 58 59}. Różnica w tych szacunkach wynika z różnego potraktowania gwałtownego wzrostu wartości rynku technologii kwantowych i inwestycjami w te technologie w czasie pandemii COVID-19⁶⁰ 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 praca⁶¹.

Głównym wyzwaniem dla rozwoju technologii informatyki kwantowej są obecnie 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 (po polsku: rybołów) firmy IBM⁶². Zgodnie z zapowiedziami, w 2023 ma zostać osiągnięty próg 1000 kubitów w jednym procesorze, co należy uznać za realny plan, zważywszy na fakt, że zgodnie z planami czynionymi od 2019, dotychczasowy postulat rokrocznego zwiększania liczby

⁵⁶ Qiskit, <https://qiskit.org/>.

⁵⁷ Fortune Business Insights, *Quantum Computing Market Size, Share. Growth Forecast 2029*, <https://www.fortunebusinessinsights.com/quantum-computing-market-104855>.

⁵⁸ Prescient & Strategic Intelligence, *Quantum Computing Market Size & Share Forecast Report 2030*, <https://www.psmarketresearch.com/market-analysis/quantum-computing-market>.

⁵⁹ McKinsey & Company, *Quantum Technology Monitor*, <https://www.mckinsey.com/~media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/quantum%20technology%20sees%20record%20investments%20progress%20on%20talent%20gap/quantum-technology-monitor-april-2023.pdf>.

⁶⁰ McKinsey & Company, *Quantum computing: An emerging ecosystem and industry use cases*, https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/quantum-computing-use-cases-are-getting-real-what-you-need-to-know#.

⁶¹ Dane pochodzą ze strony Banku Światowego: *World Development Indicators | DataBank*, <https://databank.worldbank.org/reports.aspx?source=2&series=NY.GDP.MKTP.CD&country=>.

⁶² IEEE Spectrum, *IBM Unveils 433-Qubit Osprey Chip*, <https://spectrum.ieee.org/ibm-quantum-computer-osprey>.

kubitów około 2,5-krotnie rocznie, był zawsze spełniany⁶³. Warto 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 – błędom maszynowym. Zwiększanie liczby kubitów należy uznać za szczególnie istotne dla możliwości omawianego w rozdziale II 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 operacja klasyczna). 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, celem wyznaczenia podstawowej metodologii określania benchmarków, ustalenia kierunku rozwoju, a także pokazania, że rozwój maszyn kwantowych ma rzeczywiście miejsce i jest mierzalnym zjawiskiem, a nie jedynie szumnym hasłem marketingowym⁶⁴. 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 obliczeń⁶⁵:

- 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^{66 67 68}. Metryka ta jest kluczowa, aby określić czy algorytm implementowany

⁶³ IBM, *IBM Quantum Computing. Roadmap*,
<https://www.ibm.com/quantum/www.ibm.com/quantum/roadmap>.

⁶⁴ F. Pires, *IBM Introduces CLOPS Performance Standard for Quantum Computing*, 3 listopada 2021,
<https://www.tomshardware.com/news/ibm-introduces-clops-performance-standard-for-quantum-computing>.

⁶⁵ Ibidem.

⁶⁶ N. Moll et. al., *Quantum optimization using variational algorithms on near-term quantum devices*, "Quantum Science and Technology", lipiec 2018, 3(3), 030503, s. 3-6.

przez obwód o określonej złożoności (rozumianej przez głębokość – *circuit depth*) może zostać wykorzystany na danej maszynie⁶⁹.

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 oprogramowania⁷⁰. 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.

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 Computing⁷¹ (firma zajmująca się budowaniem i udostępnianiem dostępu do obliczeń na maszynach kwantowych za pośrednictwem chmury), PennyLane od Xanadu (firma o podobnym profilu do Rigetti Computing⁷²) Cirq od Google⁷³, jednak takie pakiety są tworzone także dla produktów komercyjnych takich jak Matlab od firmy MathWorks, dla którego pakiet został opublikowany w 2023 roku⁷⁴. Istnieją także języki dedykowane symulacjom kwantowym

⁶⁷ Ch. H. Baldwin et. al., *Re-examining the quantum volume test: Ideal distributions, compiler optimizations, confidence intervals, and scalable resource estimations*, "Quantum", 9 maja 2022, 6(9), s. 707.

⁶⁸ K. Miller et. al., *An Improved Volumetric Metric for Quantum Computers via more Representative Quantum Circuit Shapes*, 14 lipca 2022, <http://arxiv.org/abs/2207.02315>.

⁶⁹ N. Moll et. al., op. cit., s. 3-4.

⁷⁰ McKinsey & Company, *Quantum computing: An emerging ecosystem and industry use cases*.

⁷¹ Rigetti Computing, *PyQuil: Quantum programming in Python*, <https://github.com/rigetti/pyquil>.

⁷² Xanadu, <https://www.xanadu.ai/>.

⁷³ Google, *Google Quantum AI*, <https://quantumai.google/cirq>.

⁷⁴ MathWorks, *Quantum Computing with Matlab*, <https://www.mathworks.com/products/quantum-computing.html>.

jak Q# od Microsoftu⁷⁵. Wiele z tych firm oferuje także usługi chmurowe w zakresie obliczeń kwantowych; poza wyżej wymienionymi warto wymienić także usługę Amazon Braket⁷⁶ oraz platformę IBM Quantum, umożliwiającą bezpłatny dostęp do niewielkich maszyn kwantowych⁷⁷.

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 kwantowych⁷⁸.

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 naukowych⁷⁹.

W kwestii stosowania obliczeń kwantowych na chwilę obecną nie ma żadnego, jednoznacznego przypadku, w którym maszyny kwantowe są w stanie wykonać obliczenia szybciej niż klasyczne komputery. Odkrycie i skuteczna prezentacja takiego zastosowania oznaczałaby osiągnięcie tzw. supremacji kwantowej. W szerszym znaczeniu oznacza to skuteczne rozwiązanie problemu obliczeniowego przez maszynę kwantową szybciej niż jest to w stanie zrobić najszybsza maszyna klasyczna. W węższym znaczeniu termin ten oznacza skuteczne rozwiązanie na komputerze kwantowym problemu, którego żaden klasyczny komputer nie byłby w stanie rozwiązać w „praktycznym” czasie. We wrześniu 2019 roku firma Google ogłosiła osiągnięcie kwantowej supremacji używając własnego

⁷⁵ Microsoft, *Introduction to Q# & Quantum Development Kit - Azure Quantum*, <https://learn.microsoft.com/en-us/azure/quantum/overview-what-is-qsharp-and-qdk>.

⁷⁶ Amazon Web Services, Inc., *Quantum Cloud Service - Quantum Computing Service - Amazon Braket - AWS*, <https://aws.amazon.com/braket/>.

⁷⁷ IBM, *IBM Quantum*, www.ibm.com/quantum.

⁷⁸ Komisja Europejska, *EU Deploys First Quantum Technology in 6 Sites across Europe*, 4 października 2022, https://ec.europa.eu/commission/presscorner/detail/en/IP_22_5914.

⁷⁹ Komisja Europejska, *The EU'S Quantum Technologies Flagship | Shaping Europe's Digital Future*, 28 października 2022, <https://digital-strategy.ec.europa.eu/en/library/eus-quantum-technologies-flagship>.

procesora kwantowego Sycamore-53⁸⁰, jednak eksperyment został obalony, zarówno poprzez podważenie metodologii^{81 82} jak i dokonanie obliczeń na komputerze klasycznym, które zostały zakończone szybciej niż na procesorze Sycamore-53⁸³. Wprawdzie w wiosną 2023 roku Google ponownie ogłosił kwantową supremację⁸⁴, warto jednak te doniesienia traktować z dystansem do czasu potwierdzenia doniesień przez osoby niezaangażowane w eksperyment, a także pamiętać, że dokonanie jednego typu obliczenia szybciej może nie stanowić wystarczającego argumentu za inwestycjami w maszyny kwantowe.

Zainteresowanie kwantowymi algorytmami optymalizacyjnymi jest dostrzegalne wśród firm prowadzących działalność na skalę globalną. Metody takie jak wyżarzanie kwantowe czy omawiany w rozdziale II niniejszej pracy algorytm QAOA mogą dołączyć do zestawu technik używanych do rozwiązywania problemów optymalizacyjnych. Deklarowanym przykładem szansy na użyteczne wdrożenie algorytmów kwantowych jest ustalanie przebiegu przesyłu danych transakcyjnych dla operacji finansowych (*transaction routing*) wymagające częstego rozwiązywania instancji problemu marszrutyzacji w bardzo krótkim czasie, choćby kosztem uzyskania nie zawsze najlepszego rozwiązania⁸⁵. Poza porównywaniem zdolności optymalizacyjnych algorytmów klasycznych z kwantowymi, należy więc także zwrócić na czynniki związane z wdrażaniem tych algorytmów na środowisko produkcyjne – na podstawie powyższego przykładu można zauważyć, że czynnik ten może mieć w przyszłości decydujące znaczenie przy wyborze metody optymalizacji. Na chwilę obecną natomiast można znaleźć jedynie implementacje typu *Proof of Concept*, mające wykazać praktyczną użyteczność obliczeń kwantowych^{86 87 88}, nie stanowią one jednak dowodu wyższości komputerów kwantowych nad klasycznymi.

⁸⁰ F. Arute et. al., *Quantum Supremacy Using a Programmable Superconducting Processor*, "Nature", 23 października 2019, 574(7779), (s. 505–510).

⁸¹ G. Kalai, Y. Rinott, T. Shoham, *Google's Quantum Supremacy Claim: Data, Documentation, and Discussion*, 25 stycznia 2023, <http://arxiv.org/abs/2210.12753>.

⁸² A. Reddy, et. al., *Comment on the Quantum Supremacy Claim by Google*, 29 sierpnia 2021, <http://arxiv.org/abs/2108.13862>.

⁸³ F. Pan, K. Chen, P. Zhang, *Solving the sampling problem of the Sycamore quantum circuits*, "Physical Review Letters", 22 sierpnia 2022, 129(9), 090502, s. 4.

⁸⁴ A. Morvan et. al., *Phase transition in Random Circuit Sampling*, 21 kwietnia 2023, <http://arxiv.org/abs/2304.11119>.

⁸⁵ Classiq, *Quantum Computing's use cases*, 21 lipca 2022, <https://www.classiq.io/insights/quantum-computing-use-cases?fbclid=IwAR3E12Ooc3OEMQyIVrO2jrl1DIav4wkgkJTVHiYW9uPUgc73JB3KwA2JxrU>.

⁸⁶ M. Machin, *Air cargo optimization solution ready for freighters in D-Wave quantum computers*, 12 maja 2022, https://quantum-south.com/airlinescargo/air-cargo-optimization-d-wave-quantum-computers/?fbclid=IwAR2NHkPhqHVRhsBcOgWJxNGgY2F7Gm8wdzzeEZpKdwaaN_lpWj3w2quY4qk.

⁸⁷ A. Bhasin, *NEC, D-Wave, and the Australian Department of Defence Collaborate on Quantum Computing Initiative*, 20 kwietnia 2021, <https://www.dwavesys.com/company/newsroom/press-release/nec-d-wave-and-the-australian-department-of-defence-collaborate-on-quantum-computing-initiative/>.

Kolejną kwestią, którą warto zasygnalizować, chociaż nie jest ona często podnoszona ze względu na brak danych, jest zużycie energii i wytwarzanie tzw. śladu węglowego. Ubocznym rezultatem eksperymentu firmy Google z 2019 roku było zmierzenie, ile energii zużył procesor Sycamore-53 na dokonanie obliczeń – było to zaledwie 0,002% energii zużytej przez maszynę klasyczną⁸⁹. Należy przy tym zwrócić uwagę, że energię zużywa się nie tylko do dokonywania obliczeń, lecz także do chłodzenia układu – co w przypadku maszyn kwantowych stanowi nadal istotną barierę w rozwoju technologii, można jednak zauważyć postęp w tej dziedzinie. Chociaż kwestia energochłonności maszyn kwantowych jest rozważana obecnie raczej w kategoriach stabilizacji układów obliczeniowych, o tyle warto mieć na uwadze, że wraz ze wzrostem nacisku na zrównoważoną politykę klimatyczną, porównywanie wpływu maszyn klasycznych i kwantowych na środowisko może stać się częścią debaty publicznej i istotnie wpłynąć na rozwój technologii kwantowych. Warto także dodać, że symulacje kwantowe (wykraczające poza zakres tej pracy) również mogą się przyczynić do wspierania idei zrównoważonego rozwoju, poprzez pomoc w odkrywaniu nowych materiałów i związków chemicznych.

Podsumowanie

W powyższym rozdziale przedstawiono podstawowe pojęcia z dziedzin obejmowanych przez słowa użyte w tytule pracy. Wprowadzono i omówiono kluczowe zagadnienia z zakresu mechaniki kwantowej wykorzystywane w obliczeniach kwantowych takie jak superpozycja i splątanie. Przedstawiono oznaczenia, które są wykorzystywane w opisach w dalszych rozdziałach. Następnie pokazano ogólny zarys problemu marszrutyzacji, zwracając uwagę na mnogość jego definicji i rozwinięć, a także podejść do sformułowań matematycznych. Trzecia część tego rozdziału została poświęcona rynkowi technologii kwantowych oraz ogólnemu stanowi badań w zakresie informatyki kwantowej, wraz z przedstawieniem kilku przypadków użycia.

⁸⁸ *Rolls-Royce and Classiq Collaborate on Quantum Algorithm Design for Computational Fluid Dynamics*, 18 października 2022, <https://www.businesswire.com/news/home/20221018005044/en/Rolls-Royce-and-Classiq-Collaborate-on-Quantum-Algorithm-Design-for-Computational-Fluid-Dynamics>.

⁸⁹ B. Villalonga et. al., *Establishing the Quantum Supremacy Frontier with a 281 Pflop/s Simulation*, "Quantum Science and Technology", 27 kwietnia 2020, 5(3), 034003, <https://doi.org/10.1088/2058-9565/ab7eeb>.

ROZDZIAŁ II. Algorytm QAOA

Poniższy rozdział jest w całości poświęcony algorytmowi QAOA. Najpierw zostały omówione jego podstawowe założenia, następnie skupiono się na części kwantowej. W drugiej części omówiono kluczowe problemy i wyzwania związane z omawianym algorytmem, w tym dobór metody optymalizacji w części klasycznej. Wybrany sposób optymalizacji metodą Bayesowską omówiono na końcu rozdziału.

II.1 Zasady działania algorytmu QAOA

II.1.1 QAOA jako algorytm wariacyjny

Jednym z algorytmów kwantowych dedykowanych optymalizacji kombinatorycznej jest kwantowy algorytm QAOA (*Quantum Approximate Optimization Algorithm*), zaproponowany w 2014 przez Edwarda Farhiego, Jeffreya Goldstone’a i Sama Gutmana do rozwiązywania problemu max-cut⁹⁰, który podobnie jak problem VRP, jest problemem kombinatorycznym (tzn. dyskretnym, o relatywnie dużej przestrzeni rozwiązań). W rozdziałach II.1.2 oraz II.1.3 przedstawiono algorytm QAOA zgodnie z założeniami jego twórców.

Algorytm QAOA należy do klasy kwantowych algorytmów wariacyjnych (*Variational Quantum Algorithm* - VQA). Algorytmy VQA są algorytmami hybrydowymi – łączą one obliczenia wykonywane na maszynach kwantowych z obliczeniami wykonywanymi na maszynach klasycznych. Łącznikami są pomiary, które skutkują przeniesieniem pewnej informacji z kubitu na bit oraz kwantowe bramki parametryzowane, gdzie wartość parametrów jest wyznaczana przy pomocy klasycznych algorytmów⁹¹.

Udowodniono, że dla problemu MAX-3LIN-EQN algorytm QAOA przez pewien czas osiągał lepszą jakość przybliżeń niż jakikolwiek znany klasyczny algorytm^{92 93}. Problem

⁹⁰ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm*, 14 listopada 2014, <http://arxiv.org/abs/1411.4028>.

⁹¹ D. Wecker, M. B. Hastings, M. Troyer, *Towards Practical Quantum Variational Algorithms*, "Physical Review A", 2 października 2015, 92(4), 042303 s: 1–2.

⁹² E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem*, 25 czerwca 2015, <http://arxiv.org/abs/1412.6062>.

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 spełniona⁹⁴.

Jedną z głównych wad algorytmu QAOA jest jego silna zależność od stosunku liczby ograniczeń do liczby zmiennych⁹⁵. Algorytm QAOA może być wykorzystany również jako część innych algorytmów, które, przynajmniej teoretycznie, mogą rozwiązać problem w najkrótszym możliwym czasie, np. algorytm Grovera^{96 97 98}.

II.1.2 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 \hat{H} (dla przypadku ogólnego) lub \hat{H}_C (dla algorytmu QAOA) jest macierzową reprezentacją funkcji kosztu, a Hamiltonian \hat{H}_M pewną funkcją 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 w systemie jest najniższa (*lowest energy state, ground state*) lub najwyższa, w zależności od celu optymalizacji. Energię systemu w stanie $|\psi\rangle$ opisuje wartość oczekiwana, która jest wartością funkcji celu \hat{H} dla zmiennych określanych przez wektor $|\psi\rangle$ ⁹⁹:

$$E(|\psi\rangle) = \langle\psi|\hat{H}|\psi\rangle. \quad (28)$$

⁹³ B. Barak, et. al., *Beating the random assignment on constraint satisfaction problems of bounded degree*, 11 sierpnia 2015.

⁹⁴ J. Ha, *Some Optimal Inapproximability Results*, "Journal of the ACM", 1 stycznia 2001, 48(4), (s. 798-859), s. 814.

⁹⁵ V. Akshay et. al., *Reachability Deficits in Quantum Approximate Optimization*, "Physical Review Letters", 5 marca 2020, 124(9), 090504, s. 1-2.

⁹⁶ M. E. S. Morales, T. Tlyachev, J. Biamonte, *Variational learning of Grover's quantum search algorithm*, "Physical Review A", 27 grudnia 2018, 98(6), 062333.

⁹⁷ L. K. Grover, *A fast quantum mechanical algorithm for database search*, 19 listopada 1996, <https://doi.org/10.48550/arXiv.quant-ph/9605043>.

⁹⁸ Ch. H. Bennett et. al., *Strengths and Weaknesses of Quantum Computing*, "SIAM Journal on Computing", 5 października 1997, 26(5), s. 1510-1523, <https://doi.org/10.1137/S0097539796300933>.

⁹⁹ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm*, s. 3.

Zadanie odnalezienia stanu podstawowego $|\psi^*\rangle$, tzn. stanu, w którym ilość energii w systemie jest najmniejsza, można traktować jako zadanie odnalezienia minimum globalnego funkcji celu, zadanego jako¹⁰⁰:

$$|\psi^*\rangle = \underset{|\psi\rangle \in \mathcal{H}}{\operatorname{argmin}}(E(|\psi\rangle)). \quad (29)$$

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

$$E(\theta) = \langle \psi(\theta) | \hat{H} | \psi(\theta) \rangle, \quad (30)$$

więc:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}(E(\theta)). \quad (31)$$

Znaleziony stan $|\psi(\theta^*)\rangle$ będzie przybliżeniem wartości zmiennych $|\psi^*\rangle$ odpowiadającym wartości poszukiwanego optimum globalnego oznaczanego C_{opt} ¹⁰¹.

W trakcie działania algorytmu stan $|\psi(\theta)\rangle$ jest otrzymywany poprzez zastosowanie na stanie początkowym $|\psi_0\rangle$ parametryzowanego operatora kwantowego (bramki kwantowej), tj. $|\psi(\theta)\rangle = U(\theta)|\psi_0\rangle$. Parametr θ 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 4)¹⁰².

W opisanym powyżej sposób, część kwantowa algorytmu QAOA, zwraca wektor reprezentujący pewien stan kwantowy, który zależy od wartości parametrów wejściowych. 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 algorytmu¹⁰³. Można zauważyć, że sposób działania algorytmu QAOA jest podobny do sposobu działania klasycznych sieci neuronowych¹⁰⁴, gdzie wagi kombinacji liniowych w warstwach sieci są aktualizowane przez algorytm propagacji wstecznej na podstawie wyników zwróconych przez sieć, po czym iteracja jest powtarzana.

¹⁰⁰ Ibidem.

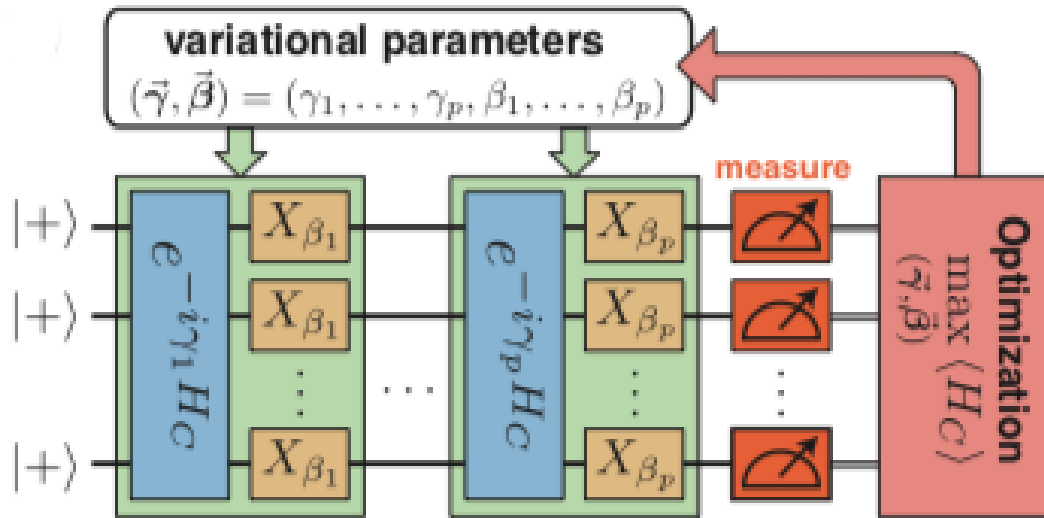
¹⁰¹ Ibidem.

¹⁰² Ibidem.

¹⁰³ Ibidem, s. 3–4, 6–10.

¹⁰⁴ E. Farhi et. al., *Quantum Algorithms for Fixed Qubit Architectures*, 17 marca 2017, <http://arxiv.org/abs/1703.06199>, s. 4.

Rysunek 4: Schemat algorytmu QAOA.



Źródło: H. Ramirez et.al., *QAOA Bench-Marking*, <https://medium.com/mit-6-s089-intro-to-quantum-computing/qaoa-bench-marking-7dfdd8a31e54>.

II.1.3 Część kwantowa algorytmu QAOA

Iteracja algorytmu QAOA zaczyna się od wprowadzenia wszystkich kubitów w stan równej superpozycji (konkretnie w stan ket plus: $|+\rangle$), poprzez zastosowanie na nich bramek Hadamarda¹⁰⁵:

$$(H|0\rangle)^{\otimes n} = \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes n} = |+\rangle^{\otimes n}. \quad (32)$$

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 (zwana też stopniem troteryzacji) – dużą literą K . O ile sama postać Hamiltonianów pozostanie niezmienna, o tyle sparametryzowane bramki na nich bazujące (patrz: równania 39 i 43) będą się zmieniały, ponieważ parametry mogą być inne w każdej iteracji algorytmu QAOA, jak i w ramach jednej iteracji przy $K > 1$.

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

¹⁰⁵ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem*, s. 2.

wyrażony w ten sposób daje się łatwo wyrazić w postaci bramek kwantowych, a komputery kwantowe potrafią szybko rozwiązywać problemy QUBO¹⁰⁶.

QUBO (Quadratic Unconstrained Binary Optimization – kwadratowa optymalizacja binarna bez ograniczeń) jest NP-trudnym problemem optymalizacji kombinatorycznej¹⁰⁷. Metoda sprowadzania problemu optymalizacyjnego do postaci QUBO jest inspirowana modelem Isinga¹⁰⁸ 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 postaci¹⁰⁹.

$$C(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^i q_{ij} x_i x_j = \mathbf{x}^T Q \mathbf{x}, \quad (33)$$

gdzie:

q_{ij} – współczynniki funkcji celu,

$Q \in \mathbf{R}^{n \times n}$ – macierz współczynników (wag),

\mathbf{x} – wektor zmiennych decyzyjnych binarnych,

$C(\mathbf{x})$ – funkcja kosztu zależna od \mathbf{x} ,

Znalezienie Hamiltonianu kosztu \hat{H}_C odbywa się poprzez rozwiązanie następującego równania^{110 111}:

$$\hat{H}_C |\mathbf{x}\rangle = C(\mathbf{x}) |\mathbf{x}\rangle. \quad (34)$$

Przyjmując, że wektor $|\mathbf{x}\rangle$ jest długości $N = 2^n$ (przypominając: n – liczba kubitów, N – liczba możliwych stanów bazowych), otrzymano równanie:

$$Z_i |\mathbf{x}\rangle = (-1)^{x_i} = 1 - 2x_i, \quad (35)$$

gdzie Z_i to bramka Pauli-Z zastosowana na i -tym kubicie. W rezultacie możliwe jest przejście od wektora, dla którego $x_i \in \{0,1\}$, do wektora, dla którego $x_i \in \{1,-1\}$. W równaniu 35 zamiana miejscami elementów pozwoli otrzymać^{112 113}:

¹⁰⁶ P. Date, D. Arthur, L. Pusey-Nazzaro, *QUBO Formulations for Training Machine Learning Models*, "Scientific Reports", 11 maja 2021, 11(1), 10029, s. 1–2.

¹⁰⁷ A. Lucas, op. cit., s. 1–3.

¹⁰⁸ T. Kadowaki, H. Nishimori, *Quantum Annealing in the Transverse Ising Model*, "Physical Review E", 25 kwietnia 1998, 58(5), (s. 5355–5363), s. 3.

¹⁰⁹ F. Glover, G. Kochenberger, Y. Du, *A Tutorial on Formulating and Using QUBO Models*, 4 listopada 2019, <https://arxiv.org/abs/1811.11538>, s. 5–7.

¹¹⁰ A. Das, B. Chakrabarti, *Quantum Annealing and Related Optimization Methods*, 2005, wyd. Springer, s. 21–22.

¹¹¹ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm*, s. 4.

¹¹² Ibidem.

$$x_i = \frac{I_i - Z_i}{2} |x\rangle. \quad (36)$$

Korzystając z równania 36 można zapisać równanie 34 jako:

$$C(x) = \sum_{i=1}^n \sum_{j=1}^i q_{ij} \frac{I_i - Z_i}{2} \frac{I_j - Z_j}{2}. \quad (37)$$

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

$$\hat{H}_C = \sum_{i=1}^n \sum_{j=1}^i q_{ij} \frac{I_i - Z_i}{2} \frac{I_j - Z_j}{2}. \quad (38)$$

W algorytmie QAOA występują bramki parametryzowane. Parametryzacja \hat{H}_C odbywa się poprzez zastosowanie eksponenty macierzy oraz transformację przy pomocy wektora parametrów γ , gdzie $\forall_{k \in \{1, \dots, K\}} \gamma_k \in [0, 2\pi]^{114}$:

$$U_k^C(\gamma_k) = e^{-i\gamma_k \hat{H}_C}. \quad (39)$$

Należy również zdefiniować Hamiltonian mieszający \hat{H}_M 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 $|+\rangle$ zostanie otrzymany skalar 1, będący wartością własną macierzy reprezentującą bramkę Pauli-X, zgodnie z:

$$X|+\rangle = 1|+\rangle \quad (40)$$

oraz analogicznie -1 dla $|-\rangle$:

$$X|-\rangle = -1|-\rangle, \quad (41)$$

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 $|1\rangle$ i otrzymać w ten sposób stany $|-\rangle$, albo zmienić znak funkcji celu.

Hamiltonian mieszający \hat{H}_M będzie postaci¹¹⁵:

$$\hat{H}_M = \sum_{i=1}^n X_i. \quad (42)$$

Po parametryzacji otrzymuje się bramkę¹¹⁶:

¹¹³ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem*, s. 5–6.

¹¹⁴ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm*, s. 2, 4.

¹¹⁵ Ibidem, s. 2.

¹¹⁶ Ibidem, s. 2-3.

$$U_k^M = e^{-i\beta_k \hat{H}_M} = \prod_{i=1}^n R_{X_i}(2\beta_k), \quad (43)$$

gdzie $\forall_{k \in \{1, \dots, K\}} \beta_k \in [0, \pi]$.

Można zauważyć, że dzięki parametryzacji, zamiast zwykłego obrotu o kąt π wokół osi OX w sferze Blocha (i przejście do stanu ortogonalnego, np. z $|0\rangle$ do $|1\rangle$), ale można też tę operację wykonywać na kubitach w superpozycji), następuje częściowy obrót wokół tej osi. Ponadto, należy zwrócić uwagę, że eksponenta macierzy będących bramkami kwantowymi (w tym Pauli-X) nie zmienia wektorów własnych.

Celem Hamiltonianu mieszającego jest umożliwienie wyjście z tzw. pułapki lokalnego minimum, wywołanej nadmierną zachłannością algorytmu. Wektory własne macierzy (w tym macierzy \hat{H}_C i \hat{H}_M oraz U_k^C i U_k^M , 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 U_k^C (niezależnie od wartości parametru γ_k) system będzie dążył do znalezienia się w stanie reprezentowanym przez wektor własny macierzy U_k^C , poprzez wzmacnianie i osłabianie odpowiednich amplitud w systemie kwantowym (*amplitude amplification*), tak aby były one możliwie 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 U_k^M na stanie $|\psi\rangle$ będącym wektorem własnym U_k^M , wektor ten pozostanie niezmienny, tj. $U_k^M |\psi\rangle = |\psi\rangle$, co wynika wprost z definicji wektora własnego. Stąd wynika, że Hamiltonian mieszający \hat{H}_M nie powinien współdzielić z Hamiltonianem kosztu \hat{H}_C wektorów własnych, nie jest to jednak możliwe do zagwarantowania w prosty sposób, dlatego przyjmuje się, że Hamiltonian mieszający powinien być prosty. Operator Pauli-X jest dobrym wyborem, ponieważ bazująca na nim bramka U_k^M za wektor własny przyjmuje stan $|+\rangle^{\otimes n}$, który jest stanem przyjmowanym na początku algorytmu. W takim przypadku, jeżeli operator U_k^C za wektor własny przyjmuje $|+\rangle^{\otimes n}$, wówczas wykrycie pułapki lokalnego ekstremum będzie trywialne, ponieważ system cały czas będzie się w tym ekstremum znajdował, tj. w stanie $|+\rangle^{\otimes n}$.

Wykonane powyżej operacje można zapisać w postaci jednego działania matematycznego, którego wynikiem będzie wektor $|\psi\rangle$ ¹¹⁷:

¹¹⁷ Ibidem, s. 3.

$$|\psi\rangle = U_K^M U_K^C \dots U_0^M U_0^C H |\mathbf{0}\rangle. \quad (44)$$

Część kwantowa algorytmu QAOA zostaje zakończona serią pomiarów kubitów tworzących wektor $|\psi\rangle$. Wynik pomiaru jest rzutowany na bity, po czym następuje przekazanie ich do maszyny klasycznej pod postacią uporządkowanego łańcucha znaków, składającego się z zer i jedynek. Otrzymane w ten sposób łańcuchy znaków stanowią istotny element obliczania wartości oczekiwanej w równaniu 30¹¹⁸. Sposób obliczania tej wartości oraz wykorzystanie algorytmów klasycznych zostanie szerzej omówione w rozdziale II.2 – możliwość optymalizacji parametrów γ i β algorytmami klasycznymi nie została od razu dostrzeżona przez twórców algorytmu QAOA i nie istnieje sztywno ustalona część klasyczna¹¹⁹.

Stosowanie operatorów U_k^C, U_k^M 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 iteracji¹²⁰. Przewidywanym pozytywnym skutkiem troteryzacji jest przybliżenie się do rozwiązania będącego optimum globalnym. W szczególności w przestrzeni ciągłej (dla problemu minimalizacji) istnieje gwarancja, że¹²¹:

$$C_{K+1} < C_K, \quad (45)$$

co oznacza, że dla troteryzacji na poziomie $K + 1$ (tzn. z $K + 1$ 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 niezmiennych). Ponadto:

$$\lim_{K \rightarrow \infty} C_K = C_{opt}, \quad (46)$$

przy czym należy pamiętać, że dla skończonych K nie ma gwarancji otrzymania dokładnego rozwiązania (stąd w nazwie algorytmu „approximate optimization”)¹²². Własność opisana

¹¹⁸ Ibidem, s. 6–7.

¹¹⁹ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem*, s. 12.

¹²⁰ G. Verdon, M. Broughton, J. Biamonte, *A quantum algorithm to train neural networks using low-depth circuits*, 14 grudnia 2017, <https://arxiv.org/abs/1712.05304>, s. 3.

¹²¹ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm*, s. 3.

¹²² Ibidem.

równaniem 45 jest tłumaczona nawiązaniem do procesu adiabatycznego, którego stroteryzowaną wersję stanowi QAOA¹²³. Należy także zwrócić uwagę, że równania 45 i 46 odnoszą się do zjawisk fizycznych, a nie faktycznego działania algorytmu.

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*)¹²⁴. 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 jeden z elementów efektywności algorytmu. Im niższe (bliższe zeru) wielkości parametrów (kątów) w wektorach γ oraz β , tym dokładniejsze przybliżenie optymalnego rozwiązania można otrzymać, ponieważ dokładniej jest przeszukiwana przestrzeń rozwiązań, przy czym zbyt mały kąt β , 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 ¹²⁵. 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 do siebie 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 γ i β poddawano już klasycznej optymalizacji, co zostanie przybliżone w Rozdziale II.2.

II.2 Rozwinięcie algorytmu QAOA

II.2.1 Złożoność obliczeniowa i ograniczenia QAOA

Jednym z elementów wyróżniających algorytm QAOA na tle innych kwantowych algorytmów optymalizacyjnych jest udowodniony matematycznie fakt, że nawet

¹²³ Ibidem, s. 7.

¹²⁴ E. Farhi et. al., *Quantum Algorithms for Fixed Qubit Architectures*, s. 13–14.

¹²⁵ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm*, s. 10–11.

w najgorszym przypadku poradzi on sobie lepiej z problemem optymalizacji kombinatorycznej niż algorytm losowo wybierający rozwiązania¹²⁶. Dla klasycznych algorytmów obliczenie wyrażenia w równaniu 30 jest zadaniem, dla którego czas obliczeń rośnie dla komputerów klasycznych w tempie podwójnie wykładniczym wraz ze wzrostem głębokości algorytmu¹²⁷ (przekładającego się na rosnącą liczbę kątów w γ i β) oraz w tempie wielomianowym wraz ze wzrostem liczby wierzchołków¹²⁸. Dla podejścia kwantowego, poziom złożoności obwodu rośnie w tempie liniowym wraz ze wzrostem głębokości obwodu – powoduje to, że dla niewielkich głębokości może być możliwe przeprowadzenie obliczeń praktycznych¹²⁹. Należy zwrócić uwagę, że można policzyć klasycznie wartość oczekiwaną w równaniu 30, ale niemożliwe jest policzenie metodami klasycznymi rozkładu wyników składających się na tę wartość oczekiwaną, zakładając, że $P \neq NP$ oraz, że przejście do stanów o niższym stopniu skomplikowania wielomianowego nie jest możliwe¹³⁰. 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 al¹³¹. Porusza on problem deficytu osiągalności (*reachability deficit*). Polega on na tym, że jeżeli stopień troteryzacji K jest zbyt mały, to optimum globalne jest całkowicie nieosiągalne, niezależnie od początkowych ustawień parametrów. Wymagany stopień troteryzacji K^* 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 płaskowyzu (*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-SAT¹³² (na których początkowy był testowany algorytm QAOA)¹³³.

¹²⁶ E. Farhi, A. W. Harrow, *Quantum Supremacy through the Quantum Approximate Optimization Algorithm*, 20 października 2019, <https://arxiv.org/abs/1602.07674>, s. 4.

¹²⁷ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm*, s. 15.

¹²⁸ E. Farhi, A. W. Harrow, op. cit., s. 15.

¹²⁹ Ibidem, s. 4.

¹³⁰ Ibidem, s. 15.

¹³¹ V. Akshay, et. al., op. cit., s. 3.

¹³² Ibidem, s. 2–4.

¹³³ E. Farhi, J. Goldstone, S. Gutmann, *A Quantum Approximate Optimization Algorithm*, s. 1-2.

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 optymalizacyjnych. 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 metodami kwantowymi rośnie wraz ze wzrostem stopnia troteryzacji K ¹³⁴. Informacja ta zdaje się przemawiać za poszukiwaniem rozwiązań dotyczących QAOA, które nie będą zwiększały troteryzacji algorytmu. Przyjmując założenie, że wyższy stopień troteryzacji oznacza lepsze przybliżenie wartości optymalnej, można stwierdzić, że algorytm QAOA będzie sobie radził dobrze 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 (tzn. takich, dla których nie jest wymagany wysoki stopień troteryzacji) algorytm QAOA przypomina w działaniu niektóre ze znanych algorytmów klasycznych, dlatego też nie będzie on w stanie osiągnąć znaczącej przewagi¹³⁵. Katzgraber 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)¹³⁶. Streif i Leib do tych badań dołączają algorytm QAOA,

¹³⁴ R. Shaydulin, S. M. Wild, *Exploiting Symmetry Reduces the Cost of Training QAOA*, "IEEE Transactions on Quantum Engineering", 17 marca 2021, 2(5), <https://doi.org/10.1109/TQE.2021.3066275>.

¹³⁵ M. B. Hastings, *Classical and Quantum Bounded Depth Approximation Algorithms*, 1 sierpnia 2019, <https://arxiv.org/abs/1905.07047>, s. 16.

¹³⁶ H. G. Katzgraber et. al., *Seeking Quantum Speedup Through Spin Glasses: The Good, the Bad, and the Ugly*, "Physical Review X", 1 września 2015, 5(3), 031026, s. 5–8.

tworząc porównanie algorytmów klasycznych i kwantowych inspirowanych procesami adiabatycznymi – opisane wyżej przez Katzbragera et al. trudne dla SA i QA problemy optymalizacyjne są względnie łatwe do rozwiązania dla algorytmu QAOA¹³⁷. W innej pracy podjęli się oni wykorzystania algorytmu QAOA do zoptymalizowania schematu działania algorytmu QA, otrzymując zadowalające wyniki¹³⁸.

II.2.2 Warm-starting QAOA

Jedną z metod poprawy efektywności działania algorytmu QAOA jest tzw. ciepłe wystartowanie (*warm starting*). Powstało ono z myślą o QAOA z niskim poziomem troteryzacji. Technika ta polega przygotowaniu stanu początkowego innego niż $|+\rangle^{\otimes n}$. W ramach tego podejścia, Hamiltonian mieszający pozostaje bez zmian, jednak zmienia się stan początkowy, przez co własność opisana równaniem 46 nie jest zachowana¹³⁹, chociaż należy też zauważyć, że jest to jedynie własność graniczna, która nie musi mieć przełożenia na eksperymenty ze skończoną liczbę iteracji.

W wyjściowej wersji WS-QAOA (warm-starting QAOA) system jest na początku iteracji w stanie¹⁴⁰:

$$|\psi_{c^*}\rangle = \bigotimes_{i=0}^{n-1} Ry(\theta_i) |0\rangle = \bigotimes_{i=0}^{n-1} (\sqrt{1-c_i^*}|0\rangle + \sqrt{c_i^*}|1\rangle), \quad (47)$$

gdzie:

$$\theta_i = 2 \sin^{-1}(\sqrt{c_i^*}), \quad \theta \in \left[0, \frac{\pi}{2}\right]^n, \quad c^* \in [0,1]^n, \quad (48)$$

tzn. kubity w swojej reprezentacji matematycznej obróćą się w stronę stanu $|1\rangle$ o kąt nie większy niż powodowany przez bramkę 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:

$$\hat{H}_{M,c^*} = \sum_{i=1}^n \begin{bmatrix} 2c_i^* - 1 & -2\sqrt{c_i^*(1-c_i^*)} \\ -2\sqrt{c_i^*(1-c_i^*)} & -(2c_i^* - 1) \end{bmatrix}, \quad (49)$$

¹³⁷ M. Streif, M. Leib, *Comparison of QAOA with Quantum and Simulated Annealing*, 7 stycznia 2019, <https://doi.org/10.48550/arXiv.1901.01903>, s. 4.

¹³⁸ M. Streif, M. Leib, *Training the quantum approximate optimization algorithm without access to a quantum processing unit*, "Quantum Science and Technology", 12 maja 2020, 5(3), 034008.

¹³⁹ D. J. Egger, J. Marecek, S. Woerner, *Warm-starting quantum optimization*, "Quantum", 17 czerwca 2021, 5(479), s. 5.

¹⁴⁰ Ibidem, s. 2.

gdzie \hat{H}_{M,c^*} przyjmuje za wektor własny $|\psi_{c^*}\rangle$. Modyfikacja równania 43 pozwoli na otrzymanie:

$$U_k^{M,c^*} = e^{-i\beta_k \hat{H}_{M,c^*}} = \prod_{i=1}^n Ry(\theta_i)Rz_i(-2\beta_k)Ry(-\theta_i). \quad (50)$$

Po takim przekształceniu, gdzie stan początkowy jest wektorem własnym U_k^{M,c^*} , równanie 46 pozostaje w mocy, jeżeli $c^* \in (0,1)^n$, a więc algorytm WS-QAOA będzie zbiegał do optymalnego rozwiązania. Jeżeli $c_i^* \in \{0,1\}$, wówczas kubit $|\psi_{c^*,i}\rangle$ nie zostanie obrócony lub przejdzie do stanu $|1\rangle$. Będzie to skutkowało pozostaniem danego kubitu nienaruszonego przez cały czas trwania iteracji algorytmu, a więc nie będzie możliwe zbliżenie się do rozwiązania odpowiadającego przeciwnego stanowi tego kubitu (czyli jeżeli kubit rozpocznie w stanie $|0\rangle$, a optimum będzie odpowiadać temu kubitowi w stanie $|1\rangle$ (i vice versa), to nie będzie możliwe osiągnięcie tego stanu)¹⁴¹.

Aby zapobiec wybraniu $c_i^* \in \{0,1\}$, można posłużyć się parametrem regulującym $\varepsilon \in [0, \frac{1}{2}]$. Wówczas otrzyma się:

- $\theta_i = 2 \arcsin(\sqrt{c_i^*})$ jeżeli $c_i^* \in [\varepsilon, 1 - \varepsilon]$,
- $\theta_i = 2 \arcsin(\sqrt{\varepsilon})$ jeżeli $c_i^* \leq \varepsilon$,
- $\theta_i = 2 \arcsin(\sqrt{1 - \varepsilon})$ jeżeli $c_i^* \geq \varepsilon$.

Następnie należy odpowiednio zaktualizować $|\psi_{c^*}\rangle$, \hat{H}_{M,c^*} oraz U_k^{M,c^*} . Jeżeli $\varepsilon = \frac{1}{2}$, wówczas zostanie wymuszony na wszystkich kubitach stan $|+\rangle$, będzie to więc odpowiadało standardowemu rozpoczęciu algorytmu QAOA.

Egger et al.¹⁴² dostrzegli możliwość dodatkowej modyfikacji macierzy mieszającej. Zauważyli oni, że bramkę bazującą na macierzy mieszającej można zmodyfikować do następującej postaci:

$$U_k^{M,c^*} = e^{-i\beta_k \hat{H}_{M,c^*}} = \prod_{i=1}^n Ry(-\theta_i)Rz_i(-2\beta_k)Ry(\theta_i), \quad (51)$$

tzn. zmieniają 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 46¹⁴³. Z drugiej strony, wyniki numeryczne pokazały, że przy wartości

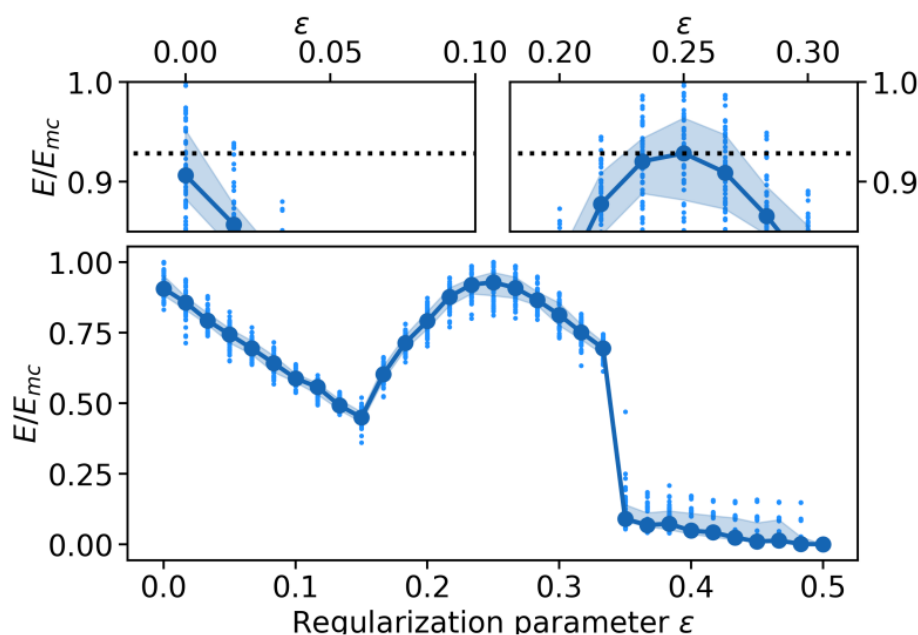
¹⁴¹ Ibidem, s. 3.

¹⁴² Ibidem, s 4–5.

¹⁴³ Ibidem, s. 4.

parametru $\varepsilon = \frac{1}{4}$ oraz troteryzacji na poziomie 1 algorytm znacznie szybciej odnajduje coraz lepsze rozwiązania, co nie zachodzi przy poprzedniej wersji bramki $U_k^{M,c*}$. Sytuację tę przedstawia rysunek 5 – najszybsza zbieżność do optimum (tutaj: maksimum) globalnego jest osiągnięta dla $\varepsilon = \frac{1}{4}$. Eksperyment był powtarzany wiele razy, grube punkty wskazują medianę, zacieniowany obszar – rozwiązania leżące między pierwszym a trzecim kwantylem¹⁴⁴. Widać, że zależność pomiędzy tempem zbieżności a ε ma charakter niemonotoniczny, w przeciwieństwie do podejścia, w którym stan początkowy jest wektorem własnym $U_k^{M,c*}$. Należy jednak zwrócić uwagę, że autorzy przedstawili jedynie wynik numeryczny, a więc wynik manipulacji ε oraz $U_k^{M,c*}$ 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 5. Wartość rozwiązania po jednej iteracji WS-QAOA z parametrem ε .



Źródło: D. J. Egger, J. Marecek, S. Woerner, *Warm-starting quantum optimization*, "Quantum", 17 czerwca 2021, 5(479), 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 ε bliższe $\frac{1}{2}$,

¹⁴⁴ Ibidem, s. 7–8.

tym gwałtowniej i częściej zmieniają się wyniki przy coraz to mniejszych zmianach parametrów γ i β (przy troteryzacji pojedynczej)¹⁴⁵.

II.2.3 Hiperoptymalizacja Bayesowska

Gradientowe metody optymalizacji są podatne na niewielkie zmiany w funkcji celu - nawet niewielki szum ma bardzo duży wpływ na wyniki dla metod gradientowych¹⁴⁶. Z racji specyfiki omawianego w rozdziale III eksperymentu zostanie przedstawiona inna metoda optymalizacji. Dla celów eksperymentalnych zostaną także uwzględnione optymalizatory numeryczne. Jedną z proponowanych metod poszukiwania optymalnego zestawu parametrów γ i β jest zastosowanie podejścia Bayesowskiego. Jest ono szczególnie polecane dla problemów o trudnej w ewaluacji funkcji celu¹⁴⁷. 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 uwzględniana w metodach Bayesowskich, 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 dobrze¹⁴⁸ (warto zaznaczyć, że problem marszrutyzacji akurat do tych problemów się nie zalicza).

Informacją liczbową (*a posteriori*) w optymalizacji Bayesowskiej może być kilka oszacowań optymalizowanej funkcji w losowych (co do rozkładu jednostajnego) punktach

¹⁴⁵ J. Obst, *Parameter Initialization for Warm-Starting QAOA*, 2022, <https://doi.org/10.18419/opus-12366>, s. 33-34.

¹⁴⁶ W. Lavrijsen et. al., *Classical Optimizers for Noisy Intermediate-Scale Quantum Devices*, "2020 IEEE International Conference on Quantum Computing and Engineering (QCE)", 2020, s. 267–277, s. 273.

¹⁴⁷ S. Tibaldi, et. al., *Bayesian Optimization for QAOA*, 30 września 2022, <https://arxiv.org/abs/2209.03824>, s. 2.

¹⁴⁸ D. Zhu, et. al., *Training of quantum circuits on a hybrid quantum computer*, "Science Advances", 18 października 2019, 5(10), s. 3.

dziedziny określonej przez możliwe zbiory wartości parametrów przyjmowanych za zmienne. Wartości parametrów (oznaczane θ_i , gdzie i to indeks danego zestawu parametrów wykorzystywanych w danym oszacowaniu) oraz odpowiadające im oszacowania wartości funkcji y tworzą zbiór D ¹⁴⁹. Jednak, aby uniknąć błędzenia losowego po przestrzeni rozwiązań oraz wielokrotnego obliczania wartości skomplikowanej funkcji, można optymalizować funkcję użyteczności¹⁵⁰ (zwaną dalej funkcją akwizycji).

Jednym z problemów jest wyznaczenie używanej w optymalizacji Bayesowskiej funkcji akwizycji (*acquisition function*) bazującej na funkcji oczekiwanej poprawy (*expected improvement*)¹⁵¹. Funkcja ta korzysta z rozkładu *a posteriori* oraz pewnej wartości f_n^* będącej największą znaną wartością optymalizowanej funkcji w ramach iteracji algorytmu. Funkcja poprawy jest zdefiniowana jako:

$$I_n(x) = \max(f(x) - f_n^*, 0), \quad (52)$$

Gdzie:

$f(x)$ – postać funkcji *a posteriori*,

f_n^* – największa znaleziona wartość funkcji f ,

n – numer iteracji,

czyli $I_n(x)$ 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.:

$$E(f(x)) = \int x f(x) dx, \quad (53)$$

można wyznaczyć funkcję oczekiwanej poprawy postaci (korzystając z całkowania przez części)^{152 153}:

$$EI_n(x) = \max(\Delta_n(x), 0) + \sigma_n(x) \varphi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) - |\Delta_n(x)| \Phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right), \quad (54)$$

gdzie:

¹⁴⁹ S. Tibaldi, et. al., op. cit., s. 2–5.

¹⁵⁰ J. S. Otterbach, et. al., *Unsupervised Machine Learning on a Hybrid Quantum Computer*, 15 grudnia 2017, <https://arxiv.org/abs/1712.05771>, s. 3.

¹⁵¹ P. I. Frazier, *A Tutorial on Bayesian Optimization*, 8 lipca 2018, <https://arxiv.org/abs/1807.02811>, s. 6-7.

¹⁵² D. R. Jones, M. Schonlau, W. J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions*, "Journal of Global Optimization", 1 grudnia 1998, 13(4), (s. 452-492), s. 471.

¹⁵³ P. I. Frazier, op. cit., s. 7.

$\Delta_n(x) = \mu_n(x) - f_n^*$ - oczekiwana różnica w jakości pomiędzy proponowanym punktem x , a poprzednim najlepszym wynikiem f_n^* ,

$|\Delta_n(x)|$ – moduł z $\Delta_n(x)$,

$\varphi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right)$ – wartość funkcji gęstości rozkładu normalnego w zadanym punkcie,

$\Phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right)$ – wartość dystrybuanty rozkładu normalnego w zadanym punkcie,

$\sigma_n(x)$ – odchylenie standardowe a posteriori.

Na koniec wyznaczany jest punkt:

$$x_{n+1} = \operatorname{argmax} EI_n(x), \quad (55)$$

gdzie:

x_{n+1} – wyznaczony punkt

x – sprawdzany punkt.

Rozwiązanie równania 55 jest nieskomplikowane, bowiem funkcje w równaniu 54 są łatwo-różniczkowalne, a maksimum funkcji $EI_n(x)$ można wyznaczyć analitycznie wykorzystując drugą pochodną lub (bardziej praktycznie z perspektywy obliczeń komputerowych) korzystając np. z algorytmu quasi-Newtonowskiego¹⁵⁴. Podejście takie stanowi trzecie wyjście w problemie kompromisu pomiędzy zachłannością a losowością (*exploration vs exploitation tradeoff*)¹⁵⁵.

Przyjęcie rozkładu *a priori* bazuje na przyjęciu pewnych założeń dotyczących rozkładu. Konstrukcja rozkładu *a priori* stanowi również najbardziej złożony element hiperoptymalizacji metodą Bayesowską. Tibaldi et al. proponuje, aby traktować funkcję celu $f(\theta)$ jako zmienną losową, ponieważ nie znamy jej dokładnej postaci¹⁵⁶. Konkretnie, proponuje, aby uznać tę funkcję za część procesu losowego Gaussowskiego, czyli przyjąć, że łączny rozkład zmiennych $f(\theta)$ ma wielowymiarowy rozkład normalny określony parametrami postaci:

$$\mu(\theta) = E(f(\theta)), \quad (56)$$

$$k(\theta, \theta') = E(f(\theta) - \mu(\theta))(f(\theta') - \mu(\theta')), \quad (57)$$

gdzie $\mu(\theta)$ to średnia wartość funkcji $f(\theta)$, a $k(\theta, \theta')$ to kowariancja (lub jądro) funkcji $f(\theta)$. Wartość średnia odzwierciedla posiadane (także w ramach założeń) informacje o funkcji $f(\theta)$, podczas gdy funkcja jądrowa odzwierciedla niepewność posiadanych

¹⁵⁴ D. C. Liu, J. Nocedal, *On the Limited Memory BFGS Method for Large Scale Optimization*, "Mathematical Programming", 1 sierpnia 1989, 45(1), (s. 503-528).

¹⁵⁵ P. I. Frazier, op. cit., s. 7.

¹⁵⁶ S. Tibaldi, et. al., op. cit., s. 3.

informacji¹⁵⁷ (w sposób analogiczny do wariancji w jednowymiarowym rozkładzie normalnym, przy czym należy również uwzględnić kowariancję). Jeżeli wartości w $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ są względnie niskie, oznacza to wysoką pewność co do tego, że przyjęta postać funkcji f jest bliska prawdziwej.

Funkcja jądrowa $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ musi być również podana przez użytkownika. Tibaldi et al. proponuje przyjęcie funkcji jądrowej Matérna postaci:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma^2 \left(1 + \frac{\sqrt{3} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2}{\ell} \right) e^{-\frac{\sqrt{3} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2}{\ell}}, \quad (58)$$

gdzie σ^2 oraz ℓ to hiperparametry charakteryzujące proces Gaussowski. Hiperparametr σ^2 określa wariancję zmiennych losowych, a ℓ to czynnik kontrolujący korelację między zmiennymi. Granicznie, $\ell \rightarrow \infty$, wszystkie punkty $\boldsymbol{\theta}$ są ze sobą skorelowane w tym samym stopniu, a dla $\ell \rightarrow 0$ wszystkie punkty $\boldsymbol{\theta}$ są ze sobą nieskorelowane. W wyniku obliczenia wartości funkcji $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ dla każdej pary punktów ze zbioru (wektora) $\boldsymbol{\theta}$ otrzymamy macierz wariancji-kowariancji \mathbb{K} .¹⁵⁸ Wartości hiperparametrów σ^2 oraz ℓ są wyznaczane przy pomocy zbioru \mathcal{D} .

Hiperparametry jądra są też zwane hiperparametrami *a priori*.¹⁵⁹ Frazier proponuje trzy metody ich wyznaczania. Pierwsza, to wyznaczenie hiperparametrów metodą największej wiarygodności (*maximum likelihood estimator* – MLE), tzn., jeśli przyjmą $\eta = [\sigma^2, \ell]$ za zbiór hiperparametrów, wówczas należy wyznaczyć $\hat{\eta}$ takie, że:

$$\hat{\eta} = \underset{\eta}{\operatorname{argmax}} P(f(x_{1:n})|\eta), \quad (59)$$

gdzie $P(f(x_{1:n})|\eta)$ ma wielowymiarowy rozkład normalny¹⁶⁰.

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 $\hat{\eta}$) i otrzymując wzór postaci:

$$\hat{\eta} = \underset{\eta}{\operatorname{argmax}} P(f(x_{1:n})|\eta) P(\eta) = \underset{\eta}{\operatorname{argmax}} P(\eta|f(x_{1:n})). \quad (60)$$

Przewaga MAP nad MLE ujawnia się, gdy funkcja f zmienia swoje wartości zbyt szybko lub zbyt wolno¹⁶¹.

¹⁵⁷ Ibidem, s. 3.

¹⁵⁸ Ibidem, s. 3–4.

¹⁵⁹ P. I. Frazier, op. cit., s. 6.

¹⁶⁰ Ibidem, s. 5–6.

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:

$$P(f(x) = y|f(x_{1:n})) = \int P(f(x) = y|f(x_{1:n}), \eta) P(\eta|f(x_{1:n})) d\eta, \quad (61)$$

co można przybliżyć wybierając J punktów (np. metodą MCMC)¹⁶²:

$$P(f(x) = y|f(x_{1:n})) \approx \frac{1}{J} \sum_{j=1}^J P(f(x) = y|f(x_{1:n}), \eta = \hat{\eta}_j). \quad (62)$$

Rozkład *a posteriori* jest wyznaczany jako funkcja określona rozkładem *a priori* warunkowa na wartościach θ pochodzącymi ze zbioru \mathcal{D} . Otrzymany rozkład *a posteriori* jest nadal wielowymiarowy i przyjmuje parametry¹⁶³:

$$\mu' = \kappa^T \mathbb{K}^{-1} \mathbf{y}, \quad (63)$$

$$k' = k(\theta, \theta) - \kappa^T \mathbb{K}^{-1} \kappa, \quad (64)$$

gdzie κ to wektor kolumnowy określający kowariancję funkcji k w punkcie θ a wszystkimi pozostałym punktami. Warto zauważyć, że nowa średnia μ' jest liniową kombinacją oszacowań funkcji, tj. elementów wektora \mathbf{y} .

Aby wdrożyć algorytm optymalizacji Bayesowskiej należy przyjąć pewną liczbę iteracji N_{BAYES} . Każda z N_{BAYES} iteracji rozpoczyna się od zaktualizowania rozkładu *a posteriori* f poprzez aktualizację raz przyjętego rozkładu *a priori* danymi ze zbioru \mathcal{D} . Na podstawie zaktualizowanego rozkładu *a posteriori* należy obliczyć funkcję akwizycji oraz znaleźć wektor parametrów θ^* maksymalizujący wartość tej funkcji. Następnie należy obliczyć wartość funkcji dla znalezionego wektora, tj. obliczyć $f(\theta^*)$. Jeżeli $f(\theta^*) > f_n^*$, wówczas należy zaktualizować $f_n^* = f(\theta^*)$ (w przeciwnym przypadku należy nie robić nic). Niezależnie od wyniku, należy dołączyć parę $(\theta^*, f(\theta^*))$ do zbioru \mathcal{D} . Iterację należy zakończyć obliczeniem nowego zestawu hiperparametrów funkcji jądrowej σ i ℓ , po czym rozpocząć kolejną iterację do czasu osiągnięcia iteracji o numerze N_{BAYES} ¹⁶⁴. Znalezionej w ten sposób wektor $2K$ -elementowy (gdzie K to stopień troteryzacji) zawiera zoptymalizowane metodą bayesowską hiperparametry γ i β algorytmu QAOA.

Proces optymalizacji Bayesowskiej można przedstawić w następujących krokach¹⁶⁵:

1. Ustal rozkład *a priori* funkcji celu jako Gaussowski (normalny).

¹⁶¹ Ibidem, s. 6.

¹⁶² Ibidem, s. 6.

¹⁶³ S. Tibaldi, et. al., op. cit., s. 4.

¹⁶⁴ Ibidem, s. 4.

¹⁶⁵ Ibidem, s. 4.

2. Oblicz wartość funkcji celu dla pewnej liczby losowych parametrów θ (traktowanych jak zmienne losowe).
3. Zdefiniuj początkowy zbiór D zawierający punkty $(\theta, f(\theta))$.
4. Oblicz wartości hiperparametrów σ^2 oraz ℓ wybraną metodą.
5. Ustal najniższą (dla minimalizacji) lub najwyższą (dla maksymalizacji) wartość funkcji celu ze zbioru D i oznacz ją f_m .
6. Zaktualizuj rozkład *a posteriori* funkcji celu korzystając ze zbioru D .
7. Oblicz wartość funkcji akwizycji korzystając z zaktualizowanego rozkładu *a posteriori*.
8. Znajdź θ^* maksymalizujący wartość funkcji akwizycji.
9. Oblicz wartość funkcji celu dla θ^* , tzn. $f(\theta^*)$.
10. Jeżeli $f(\theta^*) < f_m$, to ustal $f_m = f(\theta^*)$, w przeciwnym wypadku przejdź do następnego punktu.
11. Dołącz $(\theta^*, f(\theta^*))$ do zbioru D .
12. Oblicz wartości hiperparametrów σ^2 oraz ℓ wybraną metodą.
13. Powtórz kroki 6-12 łącznie N_{BAYES} razy.
14. Zwróć (θ_m, f_m) .

Na koniec tego podrozdziału warto przyjrzeć się kilku założeniom poczynionym przez autora zastosowania tej metody w QAOA. Pierwsze z nich dotyczy uznania rozkładu $f(\theta)$ za wielowymiarowy rozkład normalny. Ponieważ nie znamy postaci $f(\theta)$ 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 problemów¹⁶⁶.

¹⁶⁶ J. Snoek, H. Larochelle, R. P. Adams, *Practical Bayesian Optimization of Machine Learning Algorithms*, w: *Advances in Neural Information Processing Systems*, tom 25, red. F. Pereira, et. al., 2012, wyd. Curran Associates, Inc., <https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html>, s. 3.

Podsumowanie

W powyższym rozdziale pokazano z jakich części składa się algorytm QAOA, tj. z części kwantowej i klasycznej. W ramach części kwantowej zwrócono uwagę na możliwości w zakresie doboru stanu początkowego (WS-QAOA), tj. konstrukcję i parametryzację Hamiltonianów celu i Hamiltonianów mieszających. Przedstawiono zalety oraz wady techniki zwanej troteryzacją. Poruszono kwestię ustalania wartości hiperparametrów β i γ . Omówiono kwestię zagadnienia zachłanności algorytmu oraz jego zdolności do eksploracji przestrzeni rozwiązań. Zapoznano czytelnika z możliwymi ograniczeniami w stosowaniu algorytmu QAOA, takimi jak deficyt osiągalności, uniemożliwiający odnalezienie globalnego optimum optymalizowanej funkcji. Przywołano zarówno publikacje przedstawiające wyprowadzone matematycznie własności algorytmu QAOA jak i te bazujące na wynikach eksperymentów.

ROZDZIAŁ III. Implementacja algorytmu

Poniższy rozdział stanowi opis serii eksperymentów mających na celu sprawdzenie czy zastosowanie algorytmu QAOA z Bayesową hiperoptymalizacją parametrów stanowi opłacalną metodę rozwiązywania problemu marszrutyzacji. 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. Następnie zostaną ukazane wyniki wraz z krótką analizą, a na koniec zostaną przedstawione wnioski i rekomendacje.

III.1 Przygotowanie eksperymentu

III.1.1 Infrastruktura, oprogramowanie i algorytmy

Do obliczeń klasycznych wykorzystano język Python oraz 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 2), 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.1.2). Pakiet 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 biblioteki matplotlib oraz pandas, pakiety scipy oraz numpy pozwoliły na wdrożenie innych, klasycznych optymalizatorów, opisanych dalej.

Do obliczeń kwantowych wykorzystano system *ibm_jakarta* z 7 kubitami, objętości kwantowej równej 16, 2400 CLOPS-ach, bazujący na procesorze kwantowym Falcon r5.1 1H.

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 powyższych¹⁶⁷.

Tabela 2. 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.

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 \hat{H}_C uwzględniającego ograniczenia w postaci zrelaksowanej. NumPyMinimumEigensolver bazuje na procedurze *geev* z 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 –

¹⁶⁷ IBM, *IBM Quantum*, dokumentacja dla środowiska IBM Quantum, <https://quantum-computing.ibm.com/services/resources>.

w rezultacie otrzyma się przekształcenie pozwalające łatwo określić wektory i wartości własne macierzy wejściowej¹⁶⁸.

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 QAOA¹⁶⁹.

Kluczowym eksperymentem jest przeprowadzony na własnoręcznie napisanym kodzie implementującym QAOA. Ze względu na niewielki rozmiar opisanych dalej problemów zdecydowano się zastosować troteryzację pojedynczą, stąd też w dalszej części β i $\boldsymbol{\beta}$ oraz γ i $\boldsymbol{\gamma}$ będą znaczeniowo tożsame. Zdefiniowano m.in. funkcje tworzące bramki kwantowe, implementujące je symulacje, (klasyczny) optymalizator Bayesowski do hiperoptymalizacji klasycznej oraz funkcję scalającą 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 N_{BAYES} 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

¹⁶⁸ Qiskit, *NumPyMinimumEigensolver* — *Qiskit 0.43.2 documentation*, <https://qiskit.org/documentation/stubs/qiskit.algorithms.NumPyMinimumEigensolver.html>.

¹⁶⁹ Qiskit, *QAOA documentation* - *Qiskit 0.30.1 documentation*, <https://qiskit.org/documentation/stable/0.30/stubs/qiskit.aqua.algorithms.QAOA.html>.

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 płatne¹⁷⁰.

III.1.2 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 ewentualne oskarżenie o naruszenie twierdzenia nierówności trójkąta dla zestawu F1 nie jest zasadne, ponieważ czas przejazdu (uznawany tutaj za odległość) nie musi odpowiadać długości metrycznej trasy, a ta w rzeczywistości nie zawsze będzie miała postać odcinka. Pozostałe dwa zestawy (T2 oraz F2) stanowią problem oparty o dane podane przez Główny Urząd Statystyczny (GUS) i dotyczą 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 Łodzią¹⁷¹. Macierze odległości dla poszczególnych zestawów przedstawiono w tabeli 3.

Tabela 3. Macierze odległości dla grafów używanych w eksperymentach.

| Numer zestawu | Macierz odległości | Numer zestawu | Macierz odległości |
|---------------|---|---------------|--|
| T1 | $\begin{bmatrix} 0 & 10 & 10 \\ 10 & 0 & 10 \\ 10 & 10 & 0 \end{bmatrix}$ | F1 | $\begin{bmatrix} 0 & 10 & 10 & 10 \\ 10 & 0 & 10 & 10 \\ 10 & 10 & 0 & 10 \\ 10 & 10 & 10 & 0 \end{bmatrix}$ |
| T2 | $\begin{bmatrix} 0 & 267.5 & 93.16 \\ 267.5 & 0 & 202.65 \\ 93.16 & 202.65 & 0 \end{bmatrix}$ | F2 | $\begin{bmatrix} 0 & 267.5 & 93.16 & 213.35 \\ 267.5 & 0 & 202.65 & 143.84 \\ 93.16 & 202.65 & 0 & 171 \\ 213.35 & 143.84 & 171 & 0 \end{bmatrix}$ |

Źródło: opracowanie własne, na podstawie danych 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>.

¹⁷⁰ IBM, *Qiskit Runtime Overview*, <https://quantum-computing.ibm.com/lab/docs/iql/runtime>.

¹⁷¹ Główny Urząd Statystyczny, *Oszacowanie odległości i czasu przejazdu pomiędzy wybranymi gminami w Polsce w 2016 roku*, <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 $x_{i,j}$, $i < j$, gdzie $x_{i,j} = 1$ oznacza, że połączenie (krawędź) jest aktywne, a $x_{i,j} = 0$, 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 tożsamyh 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. Z racji rozmiaru problemu można zauważyć, że nie było potrzebne formułowanie warunku dotyczącego spójności grafu. Tabela 4 zawiera zestawienie liczby wymaganych kubitów dla poszczególnych rozmiarów problemów wraz w procesem redukcji rozmiaru problemu.

Tabela 4. Redukcja rozmiaru problemu VRP.

| | Liczba możliwych rozwiązań | Wymagana liczba kubitów |
|--|----------------------------|-------------------------|
| Obiekt <i>TSP</i> - wersja domyślna | | |
| Przypadek ogólny | $2^{(n^2)}$ | n^2 |
| Dla 3 wierzchołków | 512 | 9 |
| Dla 4 wierzchołków | 65536 | 16 |
| Po usunięciu krawędzi o tożsamyh wierzchołkach | | |
| Przypadek ogólny | $2^{(n(n-1))}$ | $n(n-1)$ |
| Dla 3 wierzchołków | 64 | 6 |
| Dla 4 wierzchołków | 4096 | 12 |
| Po przekształceniu do grafu nieskierowanego | | |
| Przypadek ogólny | $2^{(n(n-1)/2)}$ | $n(n-1)/2$ |
| 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:

$$\widehat{H}_C = A_0 \widehat{H}_0 + \sum_{r=1}^R A_r \widehat{H}_r = A_0 \widehat{H}_0 + A_R \widehat{H}_R \quad (65)$$

gdzie:

\widehat{H}_C – Hamiltonian funkcji celu,
 \widehat{H}_0 – Hamiltonian funkcji kosztu,
 \widehat{H}_r – Hamiltonian r -tej funkcji kary,
 \widehat{H}_R – Hamiltonian wszystkich funkcji kar,
 A_0 – mnożnik funkcji kosztu,
 A_r – mnożnik r -tej funkcji kary,
 A_R – mnożnik łączny wszystkich funkcji kar.

Dla uproszczenia przyjęto $A_0 = 1$. Dobranie wartości A_{r_l} dla ograniczeń lokalnych (oznaczanych r_l) realizujących równania 18 i 19 będzie bazowało na macierzy odległości, przy czym liczba ograniczeń lokalnych r_l jest równa liczbie wierzchołków w grafie, więc r_l -ty warunek lokalny będzie powiązany z i -tym wierzchołkiem. Należy zauważyć, że r_l -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 – zagwarantuje to znalezienie rozwiązania spełniającego warunki ograniczające, maleje jednak szansa na znalezienie rozwiązania optymalnego. Z tego powodu postuluje się ustalanie r_l -tego mnożnika jako wartości nie przewyższającej znacznie wagi (długości) najważniejszej (najdłuższej) krawędzi prowadzącej do wierzchołka, którego dotyczy ograniczenie lokalne, tzn.:

$$A_{r_l} \geq \min(\mathbf{d}_r) \wedge \neg(A_{r_l} \gg \max(\mathbf{d}_r)), \quad (66)$$

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

$$\bigwedge_{i \in V} \sum_{j \neq i} x_{i,j} = 2, \quad (67)$$

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

$$\sum_{r=0}^{n-1} A_r \left(\left(\sum_{j \neq r} x_{r,j} \right) - 2 \right)^2. \quad (68)$$

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

$$A_{r_l} = \max(\mathbf{d}_r), \quad (69)$$

gdzie r oznacza indeks wierzchołka, którego dotyczy ograniczenie, co pokazano w tabeli 5.

Tabela 5. Mnożniki funkcji kary dla lokalnych warunków ograniczających.

| T1 | | | F1 | | | |
|-------|-------|--------|-------|-------|--------|--------|
| A_1 | A_2 | A_3 | A_1 | A_2 | A_3 | A_4 |
| 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| T2 | | | F2 | | | |
| A_1 | A_2 | A_3 | A_1 | A_2 | A_3 | A_4 |
| 267.5 | 267.5 | 202.65 | 267.5 | 267.5 | 202.65 | 213.35 |

Źródło: opracowanie własne.

III.1.3 Postać QUBO problemu

Dla poszczególnych problemów otrzymano następujące Hamiltoniany celu \widehat{H}_C stanowiące postać QUBO:

- T1 z ograniczeniami lokalnymi:

$$15 * I_2 I_1 Z_0 + 15 * I_2 Z_1 I_0 + 15 * Z_2 I_1 I_0 + 5 * I_2 Z_1 Z_0 + 5 * Z_2 I_1 Z_0 + 5 * Z_2 Z_1 I_0;$$

- F1 z ograniczeniami lokalnymi:

$$\begin{aligned} & 5 * Z_5 * \left(\prod_{j=1}^5 I_{5-j} \right) + 5 * \sum_{i=1}^4 \left(\prod_{j=0}^{i-1} I_{5-j} \right) * Z_i * \left(\prod_{j=i+1}^5 I_{5-j} \right) + \\ & + 5 * \left(\prod_{j=0}^4 I_{5-j} \right) * Z_0 + 5 * I_5 I_4 I_3 I_2 Z_1 Z_0 + 5 * I_5 I_4 I_3 Z_2 I_1 Z_0 + 5 * I_5 I_4 I_3 Z_2 Z_1 I_0 + \\ & + 5 * I_5 I_4 Z_3 I_2 I_1 Z_0 + 5 * I_5 I_4 Z_3 I_2 Z_1 I_0 + 5 * I_5 Z_4 I_3 I_2 I_1 Z_0 + 5 * I_5 Z_4 I_3 I_2 Z_1 I_0 + \\ & + 5 * I_5 Z_4 Z_3 I_2 I_1 I_0 + 5 * Z_5 I_4 I_3 I_2 Z_1 I_0 + 5 * Z_5 I_4 I_3 Z_2 I_1 I_0 + 5 * Z_5 I_4 Z_3 I_2 I_1 I_0 + \\ & + 5 * Z_5 Z_4 I_3 I_2 I_1 I_0; \end{aligned}$$

- T2 z ograniczeniami lokalnymi:

$$358.0167 * I_2 I_1 Z_0 + 445.1867 * I_2 Z_1 I_0 + 390.4417 * Z_2 I_1 I_0 + \\ + 133.75 * I_2 Z_1 Z_0 + 133.75 * Z_2 I_1 Z_0 + 101.325 * Z_2 Z_1 I_0;$$

- T1 z ograniczeniami lokalnymi:

$$104 * \left(\prod_{j=0}^4 I_{5-j} \right) * Z_5 + 191.17 * \left(\prod_{j=0}^3 I_{5-j} \right) * Z_1 * I_0 + \\ + 131.075 * \left(\prod_{j=0}^2 I_{5-j} \right) * Z_2 I_1 I_0 + 136.425 * I_5 I_4 Z_3 * \left(\prod_{j=3}^5 I_{5-j} \right) + \\ + 165.83 * I_5 Z_4 * \left(\prod_{j=2}^5 I_{5-j} \right) + 152.25 * Z_5 * \left(\prod_{j=1}^5 I_{5-j} \right) + \\ + 133.75 * I_5 I_4 I_3 I_2 Z_1 Z_0 + 133.75 * I_5 I_4 I_3 Z_2 I_1 Z_0 + 133.75 * I_5 I_4 I_3 Z_2 Z_1 I_0 + \\ + 133.75 * I_5 I_4 Z_3 I_2 I_1 Z_0 + 133.75 * I_5 I_4 Z_3 I_2 Z_1 I_0 + 133.75 * I_5 Z_4 I_3 I_2 I_1 Z_0 + \\ + 101.325 * I_5 Z_4 I_3 Z_2 I_1 I_0 + 101.325 * I_5 Z_4 Z_3 I_2 I_1 I_0 + 101.325 * Z_5 I_4 I_3 I_2 Z_1 I_0 + \\ + 106.675 * Z_5 I_4 I_3 Z_2 I_1 I_0 + 106.675 * Z_5 I_4 Z_3 I_2 I_1 I_0 + 106.675 * Z_5 Z_4 I_3 I_2 I_1 I_0$$

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 wynika 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 A_r , co wynika z dzielenia w równaniu 36. Przygotowane w ten sposób Hamiltoniany pozostają niezmiennie w czasie przetwarzania algorytmu i wraz z γ służą do skonstruowania bramki $U_k^C(\gamma)$, która wraz z bramką $U_k^M(\beta)$ zawiera podstawowe dane wejściowe algorytmu QAOA. Z kolei w postaci nieprzetworzonej, powyższe Hamiltoniany przekazano do algorytmu NumPyMinimumEigensolver.

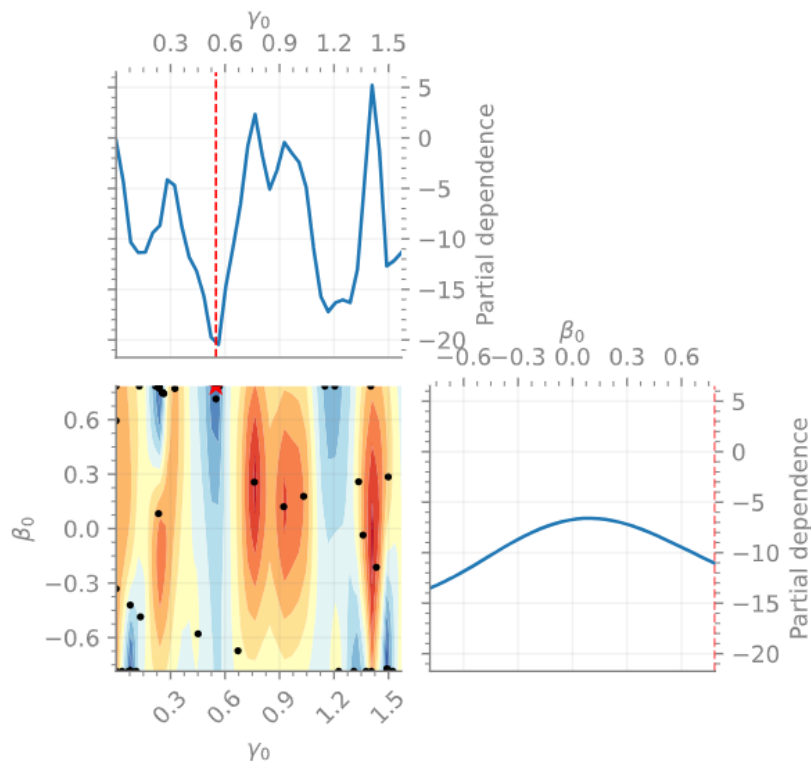
III.2 Wyniki

III.2.1 Obliczenia klasyczne – hiperoptymalizacja

Na rysunkach 6-9 przedstawiono przebieg procesu hiperoptymalizacji parametrów β i γ . Algorytm szukał takiego zestawu hiperparametrów, aby zminimalizować funkcję celu, tj. minimalizować wartość oczekiwaną długości trasy po uwzględnieniu kar za niespełnienie

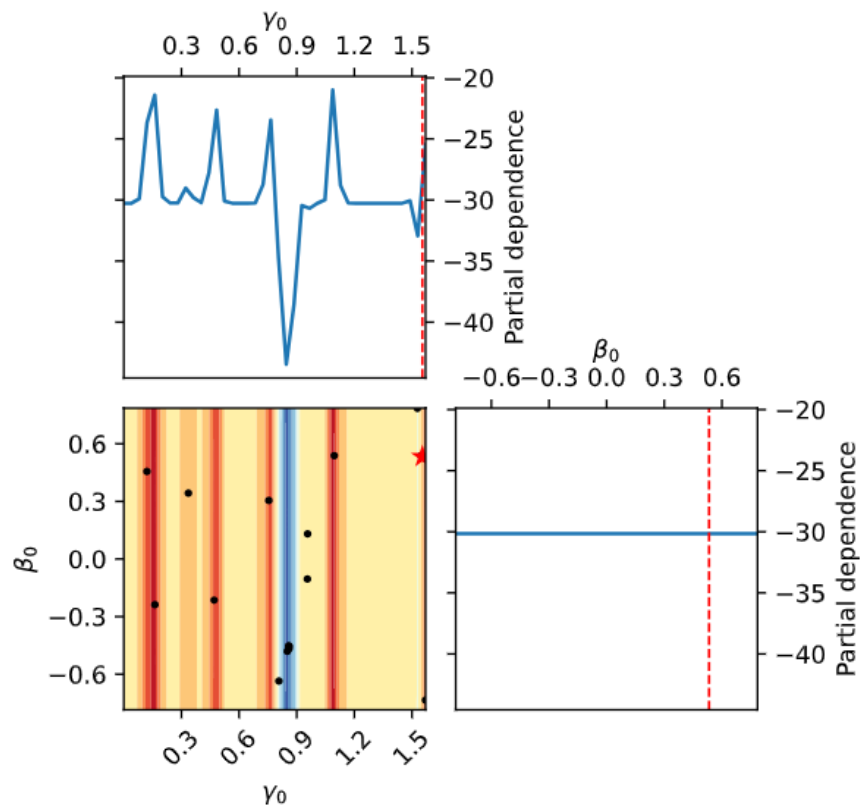
warunków ograniczających. Na każdym z wykresów na rysunkach 6-9 widoczna jest znacznie większa zmienność wartości funkcji celu ze względu na wartości parametru γ ; potwierdzają to wykresy zależności częściowej (PDP – Partial Dependence Plot), na których widać dużą nieregularność i skomplikowanie wykresu dla γ , podczas gdy dla β 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 γ niż dobór wartości parametru β , tj. należy się bardziej skupić na hiperoptymalizacji parametrów związanych z bramką implementującą Hamiltonian celu $U_k^C(\gamma)$ niż na hiperparametrach związanych z bramką implementującą Hamiltonian mieszający $U_k^M(\beta)$.

Rysunek 6. Hiperoptymalizacja parametrów metodą Bayesowską dla problemu T1.



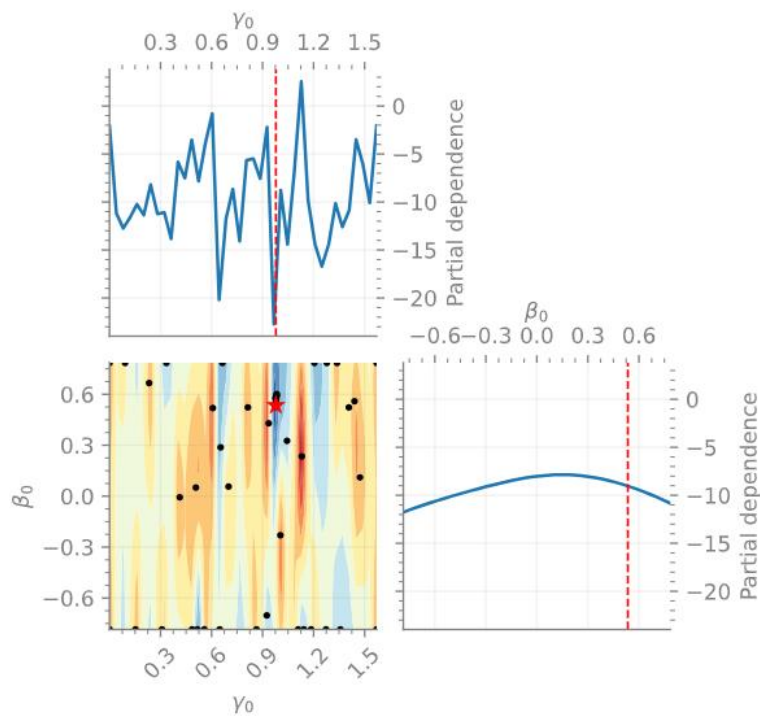
Źródło: opracowanie własne.

Rysunek 7. Hiperoptimalizacja parametrów metodą Bayesowską dla problemu T2.



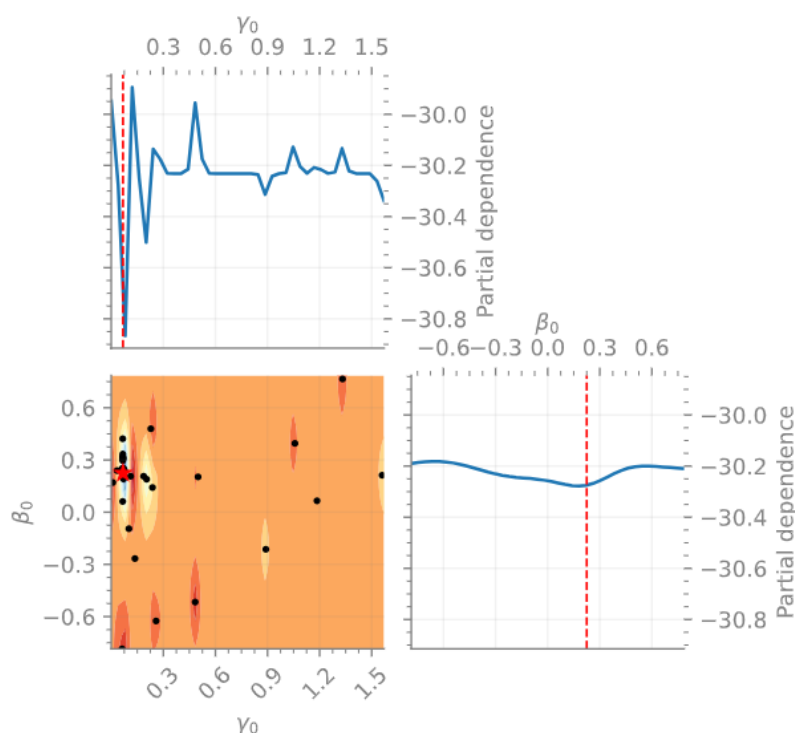
Źródło: opracowanie własne.

Rysunek 8. Hiperoptimalizacja parametrów metodą Bayesowską dla problemu F1.



Źródło: opracowanie własne.

Rysunek 9. Hiperoptymalizacja parametrów metodą Bayesowską dla problemu F2.



Źródło: opracowanie własne.

III.2.2 Obliczenia klasyczne – czas obliczeń

W tabeli 6 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 Bayesowskich 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 6. Wyniki optymalizacji metodami klasycznymi.

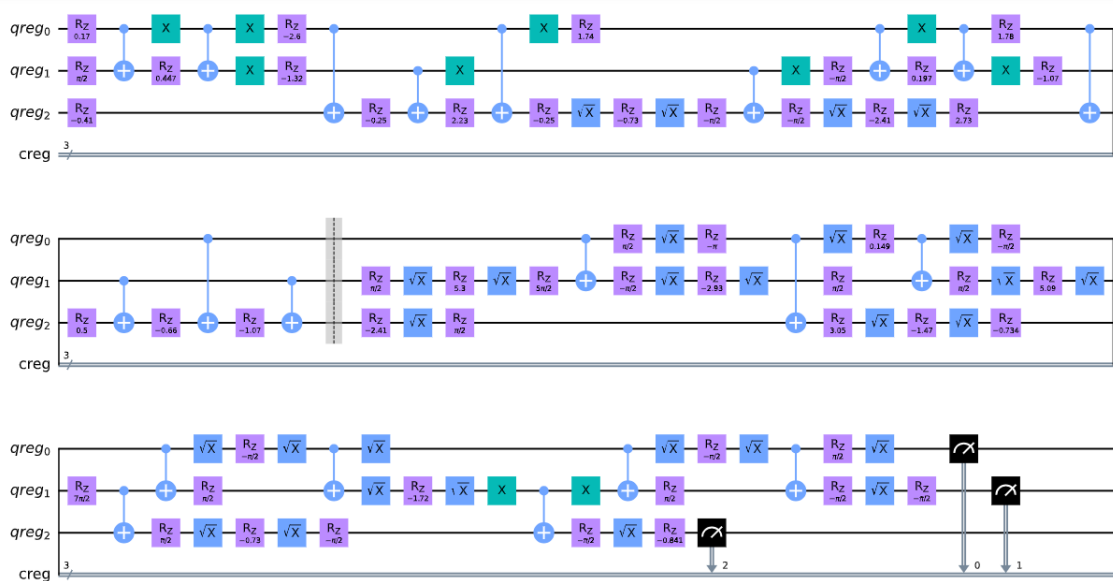
| Algorytm (klasyczny) | Średni czas obliczeń [ms] | Rezultat |
|--------------------------------------|---------------------------|---|
| T1 z ograniczeniami lokalnymi | | |
| Przeszukiwanie wyczerpujące | 0.003 | Sukces |
| NumPy Eigensolver | 14.8 | Sukces |
| Qiskit QAOA | 159 | 32 iteracje, p-ństwo 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.

III.2.3 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 $U_k^C(\gamma)$ oraz $U_k^M(\beta)$ musi się odbywać w oddzielnym środowisku. 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 $U_k^C(\gamma)$ oraz $U_k^M(\beta)$, 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, jednak im wyższy poziom optymalizacji, tym wyższe ryzyko popełnienia błędów numerycznych. W ramach 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. Na rysunku 10 przedstawiono przykładowy obwód kwantowy po transpilacji dla problemu T1 (szara linia (bariera) oddziela zdekomponowaną bramkę $U_k^C(\gamma)$ od zdekomponowanej bramki $U_k^M(\beta)$).

Rysunek 10. Obwód kwantowy dla problemu T1 po transpilacji.



Źródło: opracowanie własne.

Ze względu na ograniczony dostęp do infrastruktury kwantowej, zastosowano technikę ciepłego wystartowania (WS-QAOA) dla problemów z 4 wierzchołkami. Amplitudy odpowiadające stanom zawierającym 4 aktywne krawędzie podwojono, a pozostałe zmniejszono odpowiednio tak, aby stan kwantowy był poprawnie zdefiniowany. Analogicznie zmieniono postać macierzy U_k^M pochodzącej od Hamiltonianu mieszającego, do macierzy U_k^{M,c^*} zgodnie z równaniem 51.

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 $U_k^C(\gamma)$ i $U_k^M(\beta)$ w języku Python, w bibliotece qiskit.
3. Zdefiniowanie obwodu kwantowego z wykorzystaniem biblioteki qiskit, zawierającego pojedynczą iterację części kwantowej algorytmu QAOA (czyli zastosowanie bramek $U_k^C(\gamma)$ i $U_k^M(\beta)$ lub $U_k^{M,c^*}(\beta)$ 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 hiperoptrymalizacji parametrów metodą Bayesowską z $N_{BAYES} = 1$.
10. Powtórzenie kroków 2-9.
11. Powtórzenie kroków 2-7 (ostatnia iteracja bez części klasycznej algorytmu).
12. Analiza końcowych wyników algorytmu QAOA.

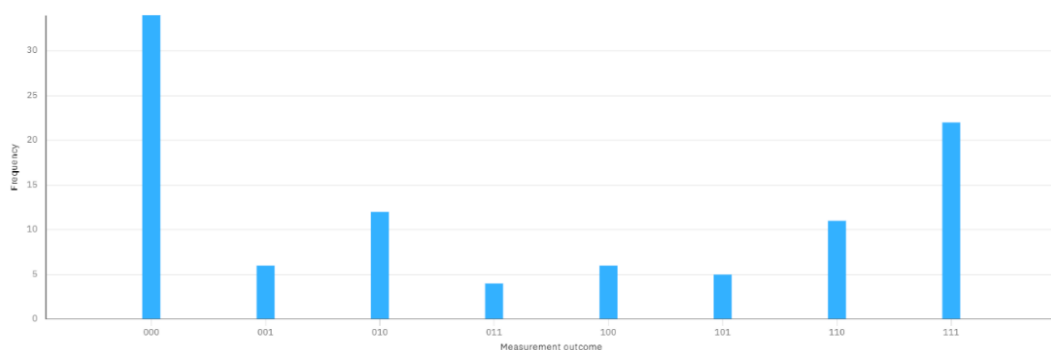
W tabeli 7 przedstawiono czasy przetwarzania poszczególnych etapów wraz z liczbą bramek, tzn. operacji wymaganych do ukończenia jednej iteracji części kwantowej algorytmu. Na rysunkach 11-16 przedstawiono histogramy otrzymanych wyników po dokonaniu pomiaru w ostatniej iteracji algorytmu QAOA. Wyniki są przedstawione w postaci izomorficznego przekształcenia wartości zmiennych binarnych – pierwsza cyfra oznacza wartość zmiennej $x_{0,1}$, kolejna $x_{0,2}$, itd.

Tabela 7. Czasy przetwarzania algorytmu QAOA w sekundach.

| Iteracja | Etap | T1 | T2 | F1 | F2 | F1-WS | F2-WS |
|---------------------------------------|----------------------------------|-------|-------|--------|--------|--------|--------|
| 1 | Przygotowanie bramek (2-4) | 0.23 | 0.24 | 0.41 | 0.45 | 0.44 | 0.47 |
| | Część kwantowa (6) | 2.749 | 2.433 | 3.789 | 3.989 | 3.626 | 3.994 |
| | Klasyczna hiperoptymalizacja (9) | 0.89 | 0.91 | 1.31 | 1.45 | 1.23 | 1.51 |
| 2 | Przygotowanie bramek (2-4) | 0.24 | 0.24 | 0.45 | 0.44 | 0.43 | 0.49 |
| | Część kwantowa (6) | 2.348 | 2.701 | 3.821 | 4.023 | 3.901 | 4.510 |
| | Klasyczna hiperoptymalizacja (9) | 0.91 | 0.90 | 1.67 | 1.58 | 1.71 | 1.57 |
| 3 | Przygotowanie bramek (2-4) | 0.21 | 0.25 | 0.44 | 0.52 | 0.47 | 0.55 |
| | Część kwantowa (6) | 2.801 | 2.656 | 3.860 | 4.111 | 3.591 | 4.003 |
| Czas razem | | 10.33 | 10.83 | 15.75 | 16.56 | 15.40 | 17.10 |
| Średnia liczba bramek po transpilacji | | 142 | 135 | 12 198 | 12 120 | 21 451 | 21 461 |

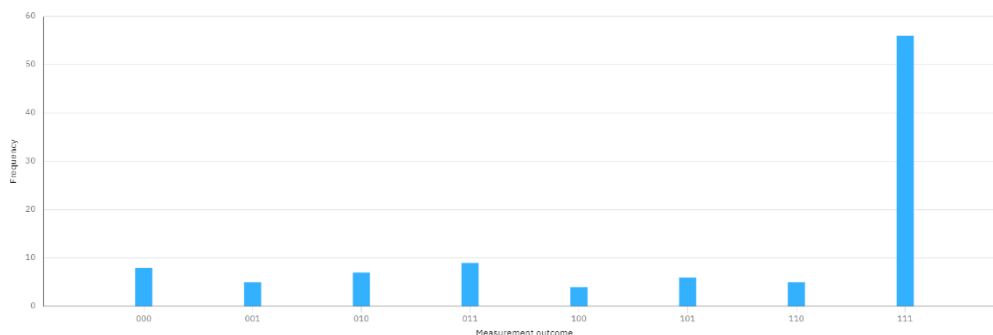
Źródło: opracowanie własne.

Rysunek 11. Histogram liczebności otrzymanych wyników końcowych dla problemu T1.



Źródło: wykres wygenerowany automatycznie w środowisku IBM Quantum na podstawie danych własnych.

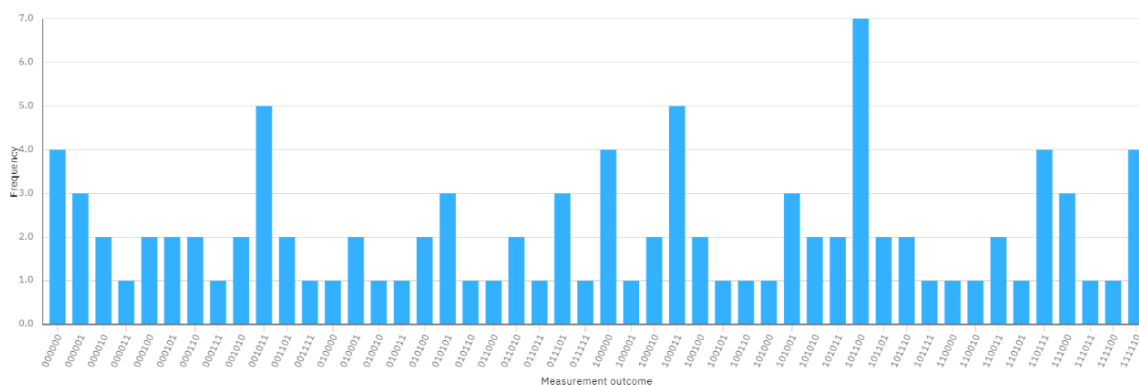
Rysunek 12. Histogram liczebności otrzymanych wyników końcowych dla problemu T2.



Źródło: wykres wygenerowany automatycznie w środowisku IBM Quantum na podstawie danych własnych.

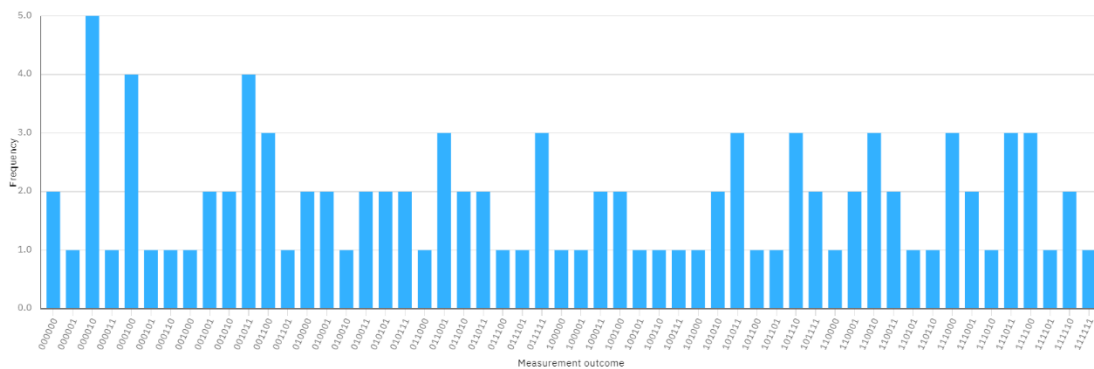
Można zauważyć, że dla problemów o identycznych odległościach między wierzchołkami w kwantowej wersji QAOA większość prób optymalizacji zakończyło się porażką. Dla problemu T1 jedyną poprawną odpowiedzią (i równocześnie najlepszą) jest „111”, podczas gdy podobną dominującą odpowiedzią było „000”, czyli brak jakichkolwiek krawędzi. Warto także zwrócić uwagę, że dla instancji T2 problemu takiego nie zauważono. Przyczyn tego można się doszukiwać w problemach z klasyczną częścią optymalizacji – w tej instancji problemu aktywowanie każdej kolejnej krawędzi zmniejsza wartość funkcji kosztu, więc wystarczy zastosować dowolny algorytm zachłanny. Należy także zwrócić uwagę, że przy deaktywacji jednej krawędzi w rozwiązaniu dla problemu T2, co najmniej jedna kara powiązana jednym wierzchołkiem (za niespełnienie ograniczeń lokalnych) będzie większa niż długość krawędzi, za której brak została naliczona kara – pomimo tej samej metody doboru wartości mnożników funkcji kary, dla jednego problemu należy uznać ją za bardziej, a dla innego – za mniej surową.

Rysunek 13. Histogram liczebności otrzymanych wyników końcowych dla problemu F1.



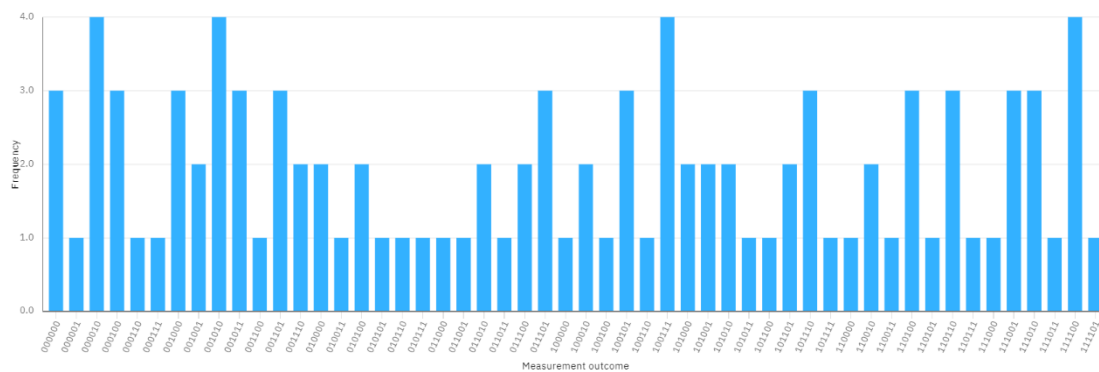
Źródło: wykres wygenerowany automatycznie w środowisku IBM Quantum na podstawie danych własnych.

Rysunek 14. Histogram liczebności otrzymanych wyników końcowych dla problemu F2.



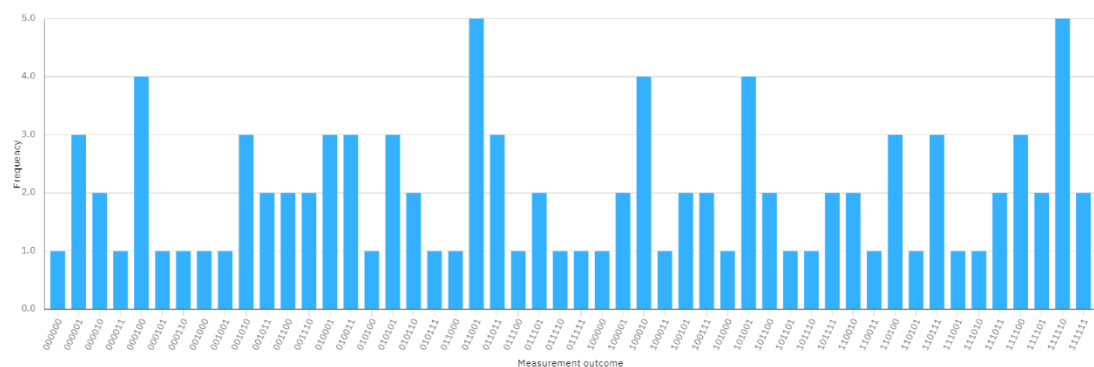
Źródło: wykres wygenerowany automatycznie w środowisku IBM Quantum na podstawie danych własnych.

Rysunek 15. Histogram liczebności otrzymanych wyników końcowych dla problemu F1-WS.



Źródło: wykres wygenerowany automatycznie w środowisku IBM Quantum na podstawie danych własnych.

Rysunek 16. Histogram liczebności otrzymanych wyników końcowych dla problemu F2-WS.



Źródło: wykres wygenerowany automatycznie w środowisku IBM Quantum na podstawie danych własnych.

Optymalizacja dla problemów F1 i F2 zakończyła się niepowodzeniem. Nie tylko nie udało się wskazać wyraźnie, które rozwiązanie jest najlepsze, ale nawet nie udało się znaleźć

rozwiązać poprawnych - rozkład wyników pomiarów można uznać za zbliżony do jednostajnego. Nie daje się zauważyć żaden znaczący wzorec. Taki wynik może być skutkiem zbyt niskimi wartościami mnożników funkcji kar. Znaczenie może mieć również szum kwantowy wynikający z infrastruktury, jednak w świetle wyników dla problemów T1 i T2 twierdzenie to nie znajduje potwierdzenia w danych. Również zastosowanie dodatkowego środka w postaci techniki ciepłego wystartowania nie pomogło w poprawie wyników. Wobec mnogości czynników wpływających na wynik (w tym parametryzacji QAOA) zdolność omawianego algorytmu do znajdowania

III.3 Wnioski i dalsze prace

W pierwszej kolejności można zauważyć, że dla symulacji klasycznych hiperoptymalizacja metodami Bayesowskimi nie przyniosła lepszych rezultatów niż pozostałe metody. Odznacza się ona bardzo długim czasem przetwarzania. Ogólnie dłuższy czas przetwarzania dla algorytmów QAOA należy uznać za wynik stosowania algorytmu kwantowego na maszynie klasycznej. Niezależnie od tego, należy zauważyć, że część klasyczna algorytmu QAOA jest (zgodnie z założeniami) przetwarzana na maszynie klasycznej, stąd też można porównać szybkość działania optymalizatorów numerycznych, z optymalizatorem Bayesowskim – porównanie to wypada na niekorzyść optymalizatora Bayesowskiego.

Oddzielnie należy zwrócić uwagę na czas przetwarzania na procesorze kwantowym. Należy zwrócić uwagę na wielokrotnie większą liczbę operacji wymaganych dla problemu z czterema wierzchołkami – wymagane jest około 90-krotnie więcej bramek. Mimo to, czas przetwarzania wzrósł względnie niewiele (mniej niż 2-krotnie). Można to tłumaczyć procesami wymaganymi do uruchomienia procesu, jak resetowanie stanu kubitów, czy też kończenie tego procesu, np. przez pomiar. Również koszt przygotowania kodu w języku QASM 2.0 (w wyniku transpilacji) nie urósł znacznie. Stanowi to bardzo dobrą informację dla skalowalności algorytmu QAOA.

Niezadowolające wyniki testów na procesorze kwantowym (z wyjątkiem problemu T2) potwierdzają, że istotnym jest dobór parametrów β oraz γ , może to także stanowić punkt wyjścia do dyskusji o wielkości mnożników funkcji kary – kwestia ta nie była do tej pory podnoszona w literaturze dotyczącej algorytmu QAOA. Zaproponowana metoda chociaż zdaje się sprawdzać w innych przypadkach, tak dla problemów, gdzie odległości między

wierzchołkami (czyli wagi krawędzi) są podobne, wielkość kary za niespełnienie ograniczeń jest za mała w porównaniu do kosztu związanego z aktywacją krawędzi.

Na uwagę zasługuje również brak metody doboru wspomnianych parametrów. Krokiem naprzód jest porównanie krajobrazu funkcji celu w zależności od β i γ – zmienność funkcji w zależności od β jest mniejsza niż w zależności od γ , należy więc więcej uwagi poświęcić eksploracji wartości parametru γ .

Na podstawie wykonanych eksperymentów do kluczowych wniosków pomocnych w przygotowaniu algorytmu optymalizacyjnego z wykorzystaniem QAOA w rozwiązywaniu problemu VRP można zaliczyć:

- opisana w rozdziale III.1.2 metoda doboru mnożników funkcji kary nie daje satysfakcjonujących wyników dla problemów, w których wagi krawędzi są takie same lub zbliżone – działa natomiast dla problemów o zróżnicowanych wagach;
- przy doborze wektorów parametrów β i γ , należy więcej uwagi poświęcić wektorowi γ , ponieważ dla różnych wektorów β wartość funkcji celu pozostaje podobne lub nawet niezmienna;
- niewielkie zmiany wektora γ powodują duże zmiany w wartości funkcji celu – należy zastosować metody odporne na nieregularny kształt funkcji;
- dla niewielkich problemów wykorzystanie metod Bayesowskich w klasycznej części algorytmu QAOA nie jest dobrym rozwiązaniem – czas obliczeń jest znaczny, a wymóg wielokrotnego powtórzenia eksperymentu powoduje, że w podobnym (lub szybszym) czasie działanie zakończy algorytm przeszukiwania wyczerpującego, który jako algorytm deterministyczny na pewno znajdzie najlepsze rozwiązanie.

Na uwagę zasługują także dalsze możliwości rozwoju algorytmu QAOA. Po pierwsze, należy zwrócić uwagę na fakt, że problem marszrutyzacji jest znacznie bardziej złożonym problemem niż problem max-cut na którym w większości publikacji sprawdza się działania powyższego algorytmu. Równocześnie przykłady w rozdziale I.3 wskazują na to, że to właśnie na tym problemie warto się skupić, z racji znalezionych zastosowań problemu VRP. Dla przykładu, znaleziony krajobraz funkcji celu dla parametrów γ jest znacznie bardziej skomplikowany niż mapy ciepła prezentowane w publikacjach związanych z problemem max-cut. Warto zatem podjąć się dodatkowych wysiłków mających na celu porównanie przechodniości wniosków wyciąganych dla różnych klas problemów. Warto dodać, że przez zastosowanie troteryzacji na poziomie równym jeden, istnieje ryzyko

zaistnienia opisywanego w rozdziale II.2.1. deficytu osiągalności, brakuje jednak danych do potwierdzenia lub odrzucenia tej hipotezy.

Kolejnym punktem zasługującym na uwagę jest zmniejszanie rozmiaru problemu VRP. Redukcję przedstawioną w tabeli 4 można uznać za znaczną – każde działanie zmniejszające rozmiar problemu bez utraty informacji, należy uznać za pożądane, niezależnie od stosowanego algorytmu. W przypadku algorytmu VRP jest to szczególnie istotne ze względu na przyrost liczby krawędzi w tempie kwadratowym (wielomianowym) względem liczby wierzchołków i przyrost liczby możliwych rozwiązań w tempie wykładniczym względem liczby krawędzi. W omawianym eksperymencie był to warunek konieczny rozpoczęcia obliczeń na obecnej infrastrukturze kwantowej. Warto zatem dalej poszukiwać możliwości zmniejszenia rozmiaru problemu także dla bardziej skomplikowanych rodzajów problemu VRP przedstawionych na rysunku 2.

Następnym elementem jest wybór optymalizatora klasycznego – na chwilę obecną brakuje metod doboru metody hiperoptymalizacji parametrów algorytmu QAOA. W powyższej pracy udało się jedynie wykazać nieefektywność hiperoptymalizacji Bayesowskiej dla małych problemów, nie ma jednak podstaw do uogólnienia tego wniosku do bardziej ogólnych problemów.

W powyższej pracy udało się dokonać postępu w kwestii doboru mnożników funkcji kary dla problemu VRP. Przedstawiono instancje o dwóch typach struktur, należy jednak pamiętać, że powyższa praca nie wyczerpuje puli możliwych kształtów grafów. Kwestią otwartą pozostaje także sposób formułowania ogólnej postaci funkcji kary, omawiana w rozdziale I.2.2. – szczególnie brakuje literatury na temat formułowania ograniczeń globalnych dotyczących spójności grafu. Ponownie, należy uznać tę kwestię jako ściśle zależną od dokładnej struktury instancji problemu.

Ograniczenia w dostępie do infrastruktury i oprogramowania znacznie wpłynęły na możliwość analizowania większych problemów, a także na łączenie obliczeń klasycznych z kwantowymi. Do czynników, które będą sprzyjać rozwojowi badań nad algorytmem QAOA należy zaliczyć dostępność procesorów kwantowych o większej liczbie kubitów, aby móc analizować bardziej złożone problemy. Przydatnym, zarówno dla badań jak ich wdrożeń praktycznych, jest rozwój i powstawanie oprogramowania tworzącego środowiska hybrydowe, tj. łączącego obliczenia na maszynach klasycznych z obliczeniami na procesorach kwantowych – w chwili obecnej łączenie tych obliczeń jest utrudnione, ponieważ wymagają one manualnego kopiowania wyników obliczeń pomiędzy maszynami.

Podsumowanie

W powyższym rozdziale przedstawiono cztery problemy optymalizacyjne klasy VRP, a także przetestowano algorytmy, które miały za zadanie wskazać najlepsze rozwiązanie, były to: wyszukiwanie wyczerpujące, gotowy program NumPyMinimumEigensolver, klasyczna symulacja algorytmu QAOA z wykorzystaniem optymalizatora COBYLA, klasyczna symulacja algorytmu QAOA z wykorzystaniem hiperoptymalizacji Bayesowskiej oraz algorytm QAOA z hiperoptymalizacją Bayesowską uruchomiony w kontrolowanym przez badacza środowisku hybrydowym. Omówiono charakterystykę używanej infrastruktury kwantowej i związane z nią ograniczenia. Przedstawiono także autorską technikę doboru mnożników w funkcjach kar, a także pokazano, w jaki sposób można zmniejszyć rozmiar problemu, a tym samym ilość wymaganych mocy obliczeniowych. Najszybszy okazał się być algorytm wyszukiwania wyczerpującego, co nie było zaskoczeniem ze względu na niewielką liczbę wierzchołków w grafach. Wątpliwości wzbudził czas obliczeń przeprowadzanych przez optymalizator Bayesowski, a także niesatysfakcjonujące wyniki dla algorytmu QAOA wykorzystującego infrastrukturę kwantową. Wyciągnięto wnioski i zaproponowano rekomendacje w zakresie przygotowania problemu przed przekazaniem go do algorytmu, a także wskazano kierunki rozwoju dalszych badań rozwoju algorytmu QAOA, ze szczególnym uwzględnieniem jego parametryzacji.

ZAKOŃCZENIE

Chociaż zastosowanie w powyższej pracy algorytmu QAOA do rozwiązywania problemu marszrutyzacji nie zakończyło się pełnym sukcesem, to eksperymenty pozwoliły na wyciągnięcie wielu ciekawych wniosków, wskazując tym samym istotne kierunki przeprowadzania dalszych badań. Ograniczenia infrastrukturalne spowodowały, że możliwe było analizowanie jedynie niewielkich instancji problemu, ale też wymusiły odmienne podejście, co umożliwiło dyskusję nad wprowadzaniem usprawnień w stosowaniu omawianego algorytmu.

Sformułowanie problemu odbiegające od tego proponowanego przez literaturę i gotowe biblioteki programistyczne pokazało, że sposób zapisu założeń i ograniczeń dla zadania optymalizacyjnego może przynieść wymierne korzyści w postaci nawet 3-krotnego zmniejszenia ilości wymaganych mocy obliczeniowych. Z drugiej strony, tak duże oszczędności nie będą możliwe dla innych typów problemu marszrutyzacji (np. problemu VRP z oknami czasowymi), dlatego należy zastosować indywidualne podejście do zadania.

Nowością jest także osobne podejście do każdego z ograniczeń lokalnych przy formułowaniu mnożników funkcji kary. Zaproponowano autorską metodę formułowania wag, która jednak przy określonym układzie wierzchołków nie będzie satysfakcjonująca, w szczególności, kiedy odległości między wierzchołkami są takie same lub zbliżone. Stanowi to jednak postęp w konstruowaniu danych wejściowych dla algorytmu QAOA i może stanowić interesujący obiekt dalszych badań mających służyć rozwojowi tej metody optymalizacji. Sprzyjać temu będzie rozwój infrastruktury kwantowej, co pozwoli na rozpatrywanie zachowania się algorytmu dla bardziej złożonych problemów omawianej klasy.

Fiaskiem zakończyło się zastosowanie metod Bayesowskich w klasycznej części algorytmu QAOA – nie udało się odnaleźć minimum funkcji celu. Należy jednak zwrócić uwagę, że otrzymywane wyniki były znacznie lepsze dla symulacji klasycznych a niżeli obliczeń kwantowych – może to być wynikiem różnej liczby iteracji w poszczególnych etapach algorytmu – w szczególności dla obliczeń kwantowych, gdzie dokonano odstępstw od metodyk proponowanych w literaturze, było to jednak niezbędne ze względu na ograniczenia w dostępie do procesora kwantowego. Podobnie jak w poprzednim przypadku, rozwój infrastruktury kwantowej, ale też oprogramowania łączącego obliczenia klasyczne z kwantowymi będą stanowiły kluczowy element w testowaniu możliwości metod

Bayesowskich jako wsparcia algorytmu QAOA. Udało się jedynie potwierdzić wnioski płynące z literatury, dotyczące zmienności wartości funkcji celu w zależności od parametrów β i γ , tzn. funkcja celu charakteryzuje się wysoką zmiennością ze względu na wartość parametru γ , niezależnie od skomplikowania Hamiltonianu celu; z kolei dla parametru β zmienność ta jest mniejsza.

Pokazane w rozdziale II potencjalne usprawnienia zostały wykorzystane w rozdziale III w celu poprawy efektywności algorytmu QAOA. Okazało się jednak, że część z nich, jak ciepłe wystartowanie nie doprowadziły do poprawy jakości otrzymywanych rozwiązań. Może to wynikać z problemów w innych częściach konstrukcji algorytmu (np. w działaniu optymalizatora klasycznego), ale też z tego, że owe usprawnienia były projektowane z myślą o problemie max-cut. Równocześnie, należy zwrócić uwagę na brak literatury poruszającej zastosowanie tych technik dla bardziej skomplikowanych problemów optymalizacji kombinatorycznej, do których zalicza się problem VRP. Analogiczny wniosek można wysnuć dla zróżnicowanych sformułowań warunków ograniczających, których kodowanie będzie się odbywać w inny sposób w zależności od dokładnego typu instancji problemu VRP, a tym samym zużywać w różnym stopniu moce obliczeniowe.

Powyższa praca uzupełniła nieliczną literaturę dotyczącą zastosowań algorytmu QAOA w rozwiązywaniu problemu marszrutyzacji. Pozwoliła ona na osiągnięcie postępów, które wraz z rozwojem i upowszechnieniem się technologii kwantowych będą oferowały wymierne korzyści dla podmiotów chcących wykorzystać algorytm będący przedmiotem tej pracy. Z drugiej strony, liczne ograniczenia wymusiły dostosowanie założeń przeprowadzanych eksperymentów, co przełożyło się na liczne niepowodzenia, ale też pozwoliło na podniesienie wielu kwestii mogących stanowić przyszłe pytania badawcze, takich jak:

- W jaki sposób należy dobierać mnożniki funkcji kar dla lokalnych warunków ograniczających?
- Czy metodyki doboru funkcji kary i jej mnożników będą różne dla problemów VRP o różnych strukturach odległości między wierzchołkami?
- Jakie znaczenie dla efektywności algorytmu QAOA ma dobór klasycznego optymalizatora klasycznego?
- Jakie czynniki należy wziąć pod uwagę planując wybór klasycznego optymalizatora dla algorytmu QAOA?

- Jaki wpływ na szybkość przetwarzania algorytmu QAOA mają jego poszczególne części (ze szczególnym uwzględnieniem stopnia skomplikowania Hamiltonianu)?
- Jak można zoptymalizować proces transpilacji tak, aby otrzymywane obwody były możliwie jak najszybciej obliczane przy możliwie niewielkim narażeniu na zaszumienie?
- Jak stopień troteryzacji wpływa na zdolność algorytmu QAOA do rozwiązywania problemu marszrutyzacji?

Przedstawione wyniki nie pozwoliły na potwierdzenie tezy postawionej we wstępie o przydatności algorytmu QAOA w rozwiązywaniu problemu marszrutyzacji. Nie oznacza to równocześnie, że nie jest możliwe efektywne wdrożenie tego algorytmu dla problemów tej klasy – omówione perspektywy rozwoju i źródła przewagi technologii kwantowej będą stanowiły motor napędowy wzrostu możliwości algorytmu QAOA, pozwalając tym samym na dalsze badania w zakresie optymalizacji z wykorzystaniem tej metody. Odpowiedzi na postawione wyżej pytania oraz rekomendacje udzielone na ich podstawie mogą doprowadzić nie tylko do przeliczalnych korzyści, ale też stanowić interesujący przedmiot badań.

BIBLIOGRAFIA

1. Akshay V., et. al., *Reachability Deficits in Quantum Approximate Optimization*, "Physical Review Letters", 5 marca 2020, 124(9), 090504, <https://doi.org/10.1103/PhysRevLett.124.090504>, [29-10-2022];
2. Amazon Web Services, Inc., *Quantum Cloud Service - Quantum Computing Service - Amazon Braket - AWS*, <https://aws.amazon.com/braket/>, [19-03-2023];
3. Applegate D., et. al., *Implementing the Dantzig-Fulkerson-Johnson Algorithm for Large Traveling Salesman Problems*, "Mathematical Programming", lipiec 2003, 97(1), s. 91–153, <https://doi.org/10.1007/s10107-003-0440-4>, [12-09-2022];
4. Arute F., et. al., *Quantum Supremacy Using a Programmable Superconducting Processor*, "Nature", 23 października 2019, 574(7779), s. 505–510, <https://doi.org/10.1038/s41586-019-1666-5>, [29-10-2022];
5. Baldwin Ch. H., et. al., *Re-examining the quantum volume test: Ideal distributions, compiler optimizations, confidence intervals, and scalable resource estimations*, "Quantum", 9 maja 2022, 6(9), s. 707, <https://doi.org/10.22331/q-2022-05-09-707>, [29-10-2022];
6. Bank Światowy: *World Development Indicators / DataBank*, <https://databank.worldbank.org/reports.aspx?source=2&series=NY.GDP.MKTP.CD&country=>, [05-02-2023];
7. Barak B., et. al., *Beating the random assignment on constraint satisfaction problems of bounded degree*, 11 sierpnia 2015, <http://arxiv.org/abs/1505.03424>, [29-10-2022];
8. Bektas T., *The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures*, "Omega", czerwiec 2006, 34(3), s. 63-211, <https://doi.org/10.1016/j.omega.2004.10.004>, [12-09-2022];
9. Benioff P., *The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines*, "Journal of Statistical Physics", czerwiec 1980, nr 22, s.563–591, <https://doi.org/10.1007/BF01011339>, [08-10-2022];
10. Bennett Ch. H., et. al., *Strengths and Weaknesses of Quantum Computing*, "SIAM Journal on Computing", 5 października 1997, 26(5), s. 1510-1523, <https://doi.org/10.1137/S0097539796300933>, [08-10-2022];

11. Bhasin A., *NEC, D-Wave, and the Australian Department of Defence Collaborate on Quantum Computing Initiative*, 20 kwietnia 2021, <https://www.dwavesys.com/company/newsroom/press-release/nec-d-wave-and-the-australian-department-of-defence-collaborate-on-quantum-computing-initiative/>, [12-07-2023];
12. Classiq, *Quantum Computing's use cases*, 21 lipca 2022, <https://www.classiq.io/insights/quantum-computing-use-cases?fbclid=IwAR3E12Ooc3OEMQyIVrO2jrl1DIav4wkgkJTVHiYW9uPUgc73JB3KwA2JxrU>, [29-03-2023];
13. Crooks G. E., *Gates, States, and Circuits*, 1 marca 2023, https://threeplusone.com/pubs/on_gates.pdf, [17-02-2023];
14. Dantzig G., Fulkerson R., Johnson S., *Solution of a Large-Scale Traveling-Salesman Problem*, "Journal of the Operations Research Society of America", 1 listopada 1954, 2(4), s. 393-410, <https://doi.org/10.1287/opre.2.4.393>, [08-10-2022];
15. Dantzig G. B., Ramser J. H., *The Truck Dispatching Problem*, "Management Science", październik 1959, 1(6), s. 80–81, <https://doi.org/10.1287/mnsc.6.1.80>, [08-10-2022];
16. Das A., Chakrabarti B., *Quantum Annealing and Related Optimization Method*, 2005, wyd. Springer, <https://link.springer.com/book/10.1007/11526216>, [13-10-2022];
17. Date P., Arthur D., Pusey-Nazzaro L., *QUBO Formulations for Training Machine Learning Models*, "Scientific Reports", 11 maja 2021, 11(1), 10029, <https://doi.org/10.1038/s41598-021-89461-4>, [29-10-2022];
18. Dirac P. A. M., *A New Notation for Quantum Mechanics*, "Mathematical Proceedings of the Cambridge Philosophical Society", 35(3), lipiec 1939, s. 416–418, <https://doi.org/10.1017/S0305004100021162>, [08-10-2022];
19. Drexl M., *Rich Vehicle Routing in Theory and Practice*, "Logistics Research", 1 sierpnia 2012, 5(1), s. 47-63, <https://doi.org/10.1007/s12159-012-0080-2>, [29-10-2022];
20. Egger D. J., Marecek J., Woerner S., *Warm-starting quantum optimization*, "Quantum", 17 czerwca 2021, 5(479), <https://doi.org/10.22331/q-2021-06-17-479>, [29-10-2022];
21. Farhi E., Goldstone J., Gutmann S., *A Quantum Approximate Optimization Algorithm*, 14 listopada 2014, <http://arxiv.org/abs/1411.4028>, [08-10-2022];

22. Farhi E., Goldstone J., Gutmann S., *A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem*, 25 czerwca 2015, <http://arxiv.org/abs/1412.6062>, [08-10-2022];
23. Farhi E., Harrow A. W., *Quantum Supremacy through the Quantum Approximate Optimization Algorithm*, 20 października 2019, <https://doi.org/10.48550/arXiv.1602.07674>, [08-10-2022];
24. Farhi E., et. al., *Quantum Algorithms for Fixed Qubit Architectures*, 17 marca 2017, <http://arxiv.org/abs/1703.06199>, [29-10-2022];
25. Feynman R. P., *Simulating Physics with Computers*, "International Journal of Theoretical Physics", czerwiec 1982, nr 21, s.467–488, <https://doi.org/10.1007/BF02650179>, [08-10-2022];
26. Flood M. M., *The Traveling-Salesman Problem*, "Operations Research", luty 1956, 4(1), s. 61-75, <http://www.jstor.org/stable/167517>, [29-10-2022];
27. Fortune Business Insights, *Quantum Computing Market Size, Share. Growth Forecast 2029*, <https://www.fortunebusinessinsights.com/quantum-computing-market-104855>, [05-02-2023];
28. Frazier P. I., *A Tutorial on Bayesian Optimization*, 8 lipca 2018, <https://doi.org/10.48550/arXiv.1807.02811>, [11-11-2022];
29. Glover F., Kochenberger G., Du Y., *A Tutorial on Formulating and Using QUBO Models*, 4 listopada 2019, <https://doi.org/10.48550/arXiv.1811.11538>, <https://arxiv.org/abs/1811.11538>, [29-10-2022];
30. Główny Urząd Statystyczny, *Oszacowanie odległości i czasu przejazdu pomiędzy wybranymi gminami w Polsce w 2016 roku*, 8 lipca 2019, <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>, [01-07-2023];
31. Google, *Google Quantum AI*, <https://quantumai.google/cirq>, [19-03-2023];
32. Grover L. K., *A fast quantum mechanical algorithm for database search*, 19 listopada 1996, <https://doi.org/10.48550/arXiv.quant-ph/9605043>, [08-10-2022];
33. Ha J., *Some Optimal Inapproximability Results*, "Journal of the ACM", 1 stycznia 2001, 48(4), s. 798-859, <http://www.cs.umd.edu/~gasarch/BLOGPAPERS/max3sat1.pdf>, [05-02-2023];
34. Hastings M. B., *Classical and Quantum Bounded Depth Approximation Algorithms*, 1 sierpnia 2019, s. 16. <https://doi.org/10.48550/arXiv.1905.07047>, [11-11-2022];

35. Hidary J. D., *A Brief History of Quantum Computing*, w: *Quantum Computing: An Applied Approach* (s. 11-16), red. J. D. Hidary, 2019, wyd. Springer Cham, https://doi.org/10.1007/978-3-030-23922-0_2, [09-10-2022];
36. IEEE Spectrum, *IBM Unveils 433-Qubit Osprey Chip*, <https://spectrum.ieee.org/ibm-quantum-computer-osprey>, [05-02-2023];
37. IBM, *IBM Quantum*, dokumentacja dla środowiska IBM Quantum, <https://quantum-computing.ibm.com/services/resources>, [01-07-2023];
38. IBM, *IBM Quantum Computing. Roadmap*, 1 października 2015, <https://www.ibm.com/quantum/www.ibm.com/quantum/roadmap>, [05-02-2023];
39. IBM, *Qiskit Runtime Overview*, <https://quantum-computing.ibm.com/lab/docs/iql/runtime>, [01-07-2023];
40. Jones D. R., Schonlau M., Welch M. J., *Efficient Global Optimization of Expensive Black-Box Functions*, "Journal of Global Optimization", 1 grudnia 1998, 13(4), s. 452-492, <https://doi.org/10.1023/A:1008306431147>, [14-12-2022];
41. Joyce D., *Cauchy's Inequality*, wyd. Clark University, 2015, <http://math.clarku.edu/~ma130/cauchy.pdf>, [27-02-2023];
42. Kadowaki T., Nishimori H., *Quantum Annealing in the Transverse Ising Model*, "Physical Review E", 25 kwietnia 1998, 58(5), s. 5355-5363, <https://doi.org/10.1103/PhysRevE.58.5355>, [12-11-2022];
43. Katzgraber H. G., et. al., *Seeking Quantum Speedup Through Spin Glasses: The Good, the Bad, and the Ugly*, "Physical Review X", 1 września 2015, 5(3), 031026, <https://doi.org/10.1103/PhysRevX.5.031026>, [11-11-2022];
44. Kalai G., Rinott Y., Shoham T., *Google's Quantum Supremacy Claim: Data, Documentation, and Discussion*, 25 stycznia 2023, <http://arxiv.org/abs/2210.12753>, [12-07-2023];
45. Kara I., Laporte G., Bektas T., *A Note on the Lifted Miller–Tucker–Zemlin Subtour Elimination Constraints for the Capacitated Vehicle Routing Problem*, "European Journal of Operational Research", listopad 2004, 158(3), s. 793-795, [https://doi.org/10.1016/S0377-2217\(03\)00377-1](https://doi.org/10.1016/S0377-2217(03)00377-1), [29-10-2022];
46. Karp R. M., *Reducibility among Combinatorial Problems*, w: *Complexity of Computer Computations. The IBM Research Symposia Series*, s. 85-103, red. R. E. Miller, J. W. Thatcher, J. D. Bohlinger, wyd. Springer / Plenum Press, Nowy Jork, 1972, [08-10-2022];

47. Komisja Europejska, *The EU'S Quantum Technologies Flagship | Shaping Europe's Digital Future*, <https://digital-strategy.ec.europa.eu/en/library/eus-quantum-technologies-flagship>, doi: 10.2759/538327, [26-01-2023];
48. Komisja Europejska, *EU Deploys First Quantum Technology in 6 Sites across Europe*, 4 października 2022, https://ec.europa.eu/commission/presscorner/detail/en/IP_22_5914, [19-03-2023];
49. Kulkarni R. V., Bhawe P. R., *Integer Programming Formulations of Vehicle Routing Problems*, "European Journal of Operational Research", kwiecień 1985, 20(1), s. 58-67, [https://doi.org/10.1016/0377-2217\(85\)90284-X](https://doi.org/10.1016/0377-2217(85)90284-X), [13-10-2022];
50. Lavrijsen W., et. al., *Classical Optimizers for Noisy Intermediate-Scale Quantum Devices*, "2020 IEEE International Conference on Quantum Computing and Engineering (QCE)", 2020, s. 267–277, <https://doi.org/10.1109/QCE49297.2020.00041>, [11-03-2023];
51. Lenstra J. K., Kan A. H. G. R., *Complexity of Vehicle Routing and Scheduling Problems*, "Networks", 1 czerwca 1981, 11(2), s. 221–227, <https://doi.org/10.1002/net.3230110211>, [29-10-2022];
52. Liu D. C., Nocedal J., *On the Limited Memory BFGS Method for Large Scale Optimization*, "Mathematical Programming", 1 sierpnia 1989, 45(1), s. 503-528, <https://doi.org/10.1007/BF01589116>, [11-11-2022];
53. Lucas A., *Ising formulations of many NP problems*, "Frontiers in Physics", 2014, nr 2, <https://www.frontiersin.org/articles/10.3389/fphy.2014.00005>, [29-09-2022];
54. Machin M., *Air cargo optimization solution ready for freighters in D-Wave quantum computers*, 12 maja 2022, https://quantum-south.com/airlinescargo/air-cargo-optimization-d-wave-quantum-computers/?fbclid=IwAR2NHkPhqHVRhsBcOgWJxNGgY2F7Gm8wdzzeEZpKdwaaN_lpWj3w2quY4qk, [05-07-2023];
55. MathWorks, *Quantum Computing with Matlab*, <https://www.mathworks.com/products/quantum-computing.html>, [19-03-2023];
56. 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#/, [29-04-2023];
57. McKinsey & Company, *Quantum Technology Monitor*, kwiecień 2023, <https://www.mckinsey.com/~media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/quantum%20technology%20sees%20record%20investments%20>

- progress%20on%20talent%20gap/quantum-technology-monitor-april-2023.pdf, [05-07-2023];
58. Microsoft, *Introduction to Q# & Quantum Development Kit - Azure Quantum*, <https://learn.microsoft.com/en-us/azure/quantum/overview-what-is-qsharp-and-qdk>, [04-02-2023];
 59. Miller C. E., Tucker A. W., Zemlin R. A., *Integer Programming Formulation of Traveling Salesman Problems*, "Journal of the ACM", październik 1960, 7(4), s. 326-329. <https://doi.org/10.1145/321043.321046>, [08-10-2022];
 60. Miller K. et. al., *An Improved Volumetric Metric for Quantum Computers via more Representative Quantum Circuit Shapes*, 14 lipca 2022, <http://arxiv.org/abs/2207.02315>, [12-09-2022];
 61. Moll N., et. al., *Quantum optimization using variational algorithms on near-term quantum devices*, "Quantum Science and Technology", lipiec 2018, 3(3), 030503, <https://doi.org/10.1088/2058-9565/aab822>, [09-10-2022];
 62. Morales M. E. S., Tlyachev T., Biamonte J., *Variational learning of Grover's quantum search algorithm*, "Physical Review A", 27 grudnia 2018, 98(6), 062333, <https://doi.org/10.1103/PhysRevA.98.062333>, [11-11-2022];
 63. Morvan A., et. al., *Phase transition in Random Circuit Sampling*, 21 kwietnia 2023, <http://arxiv.org/abs/2304.11119>, [05-07-2023];
 64. Nagy M., *Quantum computation and quantum information*, "IJPEDS", 1 lutego 2006, 21(5), <https://doi.org/10.1080/17445760500355678>, [29-10-2022];
 65. Nielsen M. A., Chuang I. L., *Quantum Computation and Quantum Information: 10th Anniversary Edition*, wyd. Cambridge University Press, 9 grudnia 2010, doi:10.1017/CBO9780511976667, [29-09-2022];
 66. Obst J., *Parameter Initialization for Warm-Starting QAOA*, 2022, <https://doi.org/10.18419/opus-12366>, [11-11-2022];
 67. Orloff C. S., *A Fundamental Problem in Vehicle Routing*, "Networks", 1974, 4(1), s. 35–64, <https://doi.org/10.1002/net.3230040105>, [08-10-2022];
 68. Otterbach J. S., et. al., *Unsupervised Machine Learning on a Hybrid Quantum Computer*, 15 grudnia 2017, <https://doi.org/10.48550/arXiv.1712.05771>, <https://arxiv.org/abs/1712.05771>, [01-12-2022];
 69. Pan F., Chen K., Zhang P., *Solving the sampling problem of the Sycamore quantum circuits*, "Physical Review Letters", 22 sierpnia 2022, 129(9), 090502, <https://doi.org/10.1103/PhysRevLett.129.090502>, [05-07-2023];

70. Pires F., *IBM Introduces CLOPS Performance Standard for Quantum Computing*, 3 listopada 2021, <https://www.tomshardware.com/news/ibm-introduces-clops-performance-standard-for-quantum-computing>, [19-03-2023];
71. Pratt V. R., *An $n \log n$ Algorithm to Distribute n Records Optimally in a Sequential Access File*, w: *Complexity of Computer Computations. The IBM Research Symposia Series*, s. 111-118, red. Raymond E. Miller, James W. Thatcher, Jean D. Bohlinger, wyd. Springer / Plenum Press, Nowy Jork, 1972, https://doi.org/10.1007/978-1-4684-2001-2_11, [29-04-2023];
72. Prescient & Strategic Intelligence, *Quantum Computing Market Size & Share Forecast Report 2030*, <https://www.psmarketresearch.com/market-analysis/quantum-computing-market>, [05-02-2023];
73. Qiskit, *NumpyMinimumEigensolver, Qiskit 0.43.2 documentation*, dokumentacja oprogramowania, https://qiskit.org/documentation/stubs/qiskit.algorithms.minimum_eigensolvers.NumPyMinimumEigensolver.html, [01-07-2023];
74. Qiskit, *QAOA documentation - Qiskit 0.30.1 documentation*, dokumentacja oprogramowania, <https://qiskit.org/documentation/stable/0.30/stubs/qiskit.aqua.algorithms.QAOA.html>, [01-07-2023];
75. Qiskit, <https://qiskit.org/>, [29-03-2023];
76. Raff S., *Routing and Scheduling of Vehicles and Crews: The State of the Art*, "Computers & Operations Research", 1983, 10(2), s. 83–86, [https://doi.org/10.1016/0305-0548\(83\)90030-8](https://doi.org/10.1016/0305-0548(83)90030-8), [08-10-2022];
77. Ramirez H. et.al., *QAOA Bench-Marking*, <https://medium.com/mit-6-s089-intro-to-quantum-computing/qaoa-bench-marking-7dfdd8a31e54>, [12-10-2022];
78. Reddy A., et. al., *Comment on the Quantum Supremacy Claim by Google*, 29 sierpnia 2021, <http://arxiv.org/abs/2108.13862>, [05-07-2023];
79. Rigetti Computing, *PyQuil: Quantum programming in Python*, 18 marca 2023, <https://github.com/rigetti/pyquil>, [05-07-2023];
80. Salehi Ö., Glos A., Miszczak J. A., *Unconstrained Binary Models of the Travelling Salesman Problem Variants for Quantum Optimization*, "Quantum Information Processing", 22 stycznia 2022, 21(2), <https://doi.org/10.1007/s11128-021-03405-5>, [29-09-2022];
81. Shaydulin R., Wild S. M., *Exploiting Symmetry Reduces the Cost of Training QAOA*, "IEEE Transactions on Quantum Engineering", 17 marca 2021, 2(5), <https://doi.org/10.1109/TQE.2021.3066275>, [11-11-2022];

82. Snoek J., Larochelle H., Adams R. P., *Practical Bayesian Optimization of Machine Learning Algorithms*, w: *Advances in Neural Information Processing Systems*, tom 25, red. F. Pereira, et. al., 2012, wyd. Curran Associates, Inc., <https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html>, [01-12-2023];
83. Streif M., Leib M., *Comparison of QAOA with Quantum and Simulated Annealing*, 7 stycznia 2019, <https://doi.org/10.48550/arXiv.1901.01903>, [29-10-2022];
84. Streif M., Leib M., *Training the quantum approximate optimization algorithm without access to a quantum processing unit*, "Quantum Science and Technology", 12 maja 2020, 5(3), 034008, <https://doi.org/10.1088/2058-9565/ab8c2b>, [29-10-2022];
85. Susskind L., Friedeman A., *Mechanika kwantowa. Teoretyczne minimum*, wyd. Prószyński Media Sp. z o.o., Warszawa, 2016, ISBN: 9788380692367;
86. Tibaldi S., et. al., *Bayesian Optimization for QAOA*, 30 września 2022, <https://doi.org/10.48550/arXiv.2209.03824>, [11-11-2022];
87. Verdon G., Broughton M., Biamonte J., *A quantum algorithm to train neural networks using low-depth circuits*, 14 grudnia 2017, <https://doi.org/10.48550/arXiv.1712.05304>, <https://arxiv.org/abs/1712.05304>, [01-12-2022];
88. Villalonga B., et. al., *Establishing the Quantum Supremacy Frontier with a 281 Pflop/s Simulation*, "Quantum Science and Technology", 27 kwietnia 2020, 5(3), 034003, <https://doi.org/10.1088/2058-9565/ab7eeb>, [29-04-2023];
89. Walter R., *Functional Analysis*, wyd. McGraw-Hill Book Company, 1991, ISBN: 9788301158026;
90. Wecker D., Hastings M. B., Troyer M., *Towards Practical Quantum Variational Algorithms*, "Physical Review A", 2 października 2015, 92(4), 042303, <https://doi.org/10.1103/PhysRevA.92.042303>, [08-10-2022];
91. Wong T. G., *Introduction to Classical and Quantum Computing*, 1 maja 2022, <http://www.thomaswong.net/>, [29-09-2022];
92. Xanadu, <https://www.xanadu.ai/>, [19-03-2023];
93. Zhu D., et. al., *Training of quantum circuits on a hybrid quantum computer*, "Science Advances", 18 października 2019, 5(10), <https://doi.org/10.1126/sciadv.aaw9918>, [01-12-2022];
94. -, *The Nobel Prize in Physics 2022*, <https://www.nobelprize.org/prizes/physics/2022/summary>, [31-01-2023];

95. -, *Rolls-Royce and Classiq Collaborate on Quantum Algorithm Design for Computational Fluid Dynamics*, 18 października 2022, <https://www.businesswire.com/news/home/20221018005044/en/Rolls-Royce-and-Classiq-Collaborate-on-Quantum-Algorithm-Design-for-Computational-Fluid-Dynamics>, [18-10-2022];

SPIS TABEL

| | |
|--|----|
| Tabela 1. Podstawowe operacje kwantowe i ich oznaczenia (cz. 1/2). | 14 |
| Tabela 2. Wersje oprogramowania użyte w czasie eksperymentów. | 54 |
| Tabela 3. Macierze odległości dla grafów używanych w eksperymentach. | 56 |
| Tabela 4. Redukcja rozmiaru problemu VRP. | 57 |
| Tabela 5. Mnożniki funkcji kary dla lokalnych warunków ograniczających. | 59 |
| Tabela 6. Wyniki optymalizacji metodami klasycznymi. | 64 |
| Tabela 7. Czasy przetwarzania algorytmu QAOA w sekundach. | 67 |

SPIS RYSUNKÓW

| | |
|--|----|
| Rysunek 1. Wizualizacja sfery Blocha. | 10 |
| Rysunek 2. Odmiany problemu VRP. | 19 |
| Rysunek 3. Przykład rozwiązania problemu VRP, dla którego nie jest spełniony warunek spójności grafu. | 21 |
| Rysunek 4: Schemat algorytmu QAOA. | 35 |
| Rysunek 5. Wartość rozwiązania po jednej iteracji WS-QAOA z parametrem ϵ | 45 |
| Rysunek 6. Hiperoptymalizacja parametrów metodą Bayesowską dla problemu T1. | 61 |
| Rysunek 7. Hiperoptymalizacja parametrów metodą Bayesowską dla problemu T2. | 62 |
| Rysunek 8. Hiperoptymalizacja parametrów metodą Bayesowską dla problemu F1. | 62 |
| Rysunek 9. Hiperoptymalizacja parametrów metodą Bayesowską dla problemu F2. | 63 |
| Rysunek 10. Obwód kwantowy dla problemu T1 po transpilacji. | 65 |

| | |
|---|----|
| Rysunek 11. Histogram liczebności otrzymanych wyników końcowych dla problemu T1.... | 67 |
| Rysunek 12. Histogram liczebności otrzymanych wyników końcowych dla problemu T2.... | 68 |
| Rysunek 13. Histogram liczebności otrzymanych wyników końcowych dla problemu F1. | 68 |
| Rysunek 14. Histogram liczebności otrzymanych wyników końcowych dla problemu F2. | 69 |
| Rysunek 15. Histogram liczebności otrzymanych wyników końcowych dla problemu F1-WS. | 69 |
| Rysunek 16. Histogram liczebności otrzymanych wyników końcowych dla problemu F2-WS. | 69 |

STRESZCZENIE

W pracy przedstawiono wykorzystanie kwantowego algorytmu QAOA w rozwiązywaniu problemu marszrutyzacji (VRP). W pierwszej części wprowadzono podstawowe pojęcia z zakresu informatyki kwantowej oraz pokazano matematyczną reprezentację problemu VRP. Opisano stan rozwoju technologii związanych z obliczeniami kwantowymi. Następnie omówiono podstawowe pojęcia związane z algorytmem QAOA, takie jak troteryzacja, konstrukcja Hamiltonianów, *warm-starting*, poświęcono także oddzielny podrozdział hiperoptymalizacji parametrów z wykorzystaniem metod Bayesowskich. Przygotowany algorytm uruchomiono zarówno jako symulację na klasycznym komputerze, jak i z wykorzystaniem procesora kwantowego Falcon r5.11H. Dokonano porównania z innymi metodami optymalizacji. Praca przedstawia rzadko poruszane w literaturze zastosowanie algorytmu QAOA dla problemu VRP, a także prezentuje sposób na zmniejszenie wymiarowości problemu oraz własną metodę doboru mnożników w funkcji kary. Wyniki przeprowadzonych testów nie potwierdziły przydatności omawianego algorytmu, ale pozwoliły na wskazanie licznych usprawnień.