



**T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**MAKİNE ÖĞRENMESİ KULLANARAK QAOA PARAMETRELERİNİN
İKİ ŞEHİRLİ GEZGİN SATICI PROBLEMİ İÇİN OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

Burhan ENGİN

Fizik Anabilim Dalı

Fizik Tezli Yüksek Lisans Programı

EYLÜL 2024

T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ KULLANARAK QAOA PARAMETRELERİNİN
İKİ ŞEHİRLİ GEZGİN SATICI PROBLEMİ İÇİN OPTİMİZASYONU

YÜKSEK LİSANS TEZİ

Burhan ENGİN

21435012005

ORCID: 0009-0004-6824-4930

Fizik Anabilim Dalı

Fizik Tezli Yüksek Lisans Programı

Danışman: Prof. Dr. Songül AKBULUT ÖZEN

ORCID: 0000-0001-8025-2141

EYLÜL 2024

BTÜ, Lisansüstü Eğitim Enstitüsü'nün 21435012005 numaralı Yüksek Lisans Öğrencisi Burhan ENGİN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “MAKİNE ÖĞRENMESİ KULLANARAK QAOA PARAMETRELERİNİN İKİ ŞEHİRLİ GEZGİN SATICI PROBLEMİ İÇİN OPTİMİZASYONU” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Songül AKBULUT ÖZEN**
Bursa Teknik Üniversitesi

Jüri Üyeleri : **Dr. Öğr. Üyesi Hüseyin DAĞ**
Bursa Teknik Üniversitesi

Prof. Dr. Necati ÇELİK
Gümüşhane Üniversitesi

Teslim Tarihi :
Savunma Tarihi : 3 Eylül 2024



20.04.2016 tarihli Resmî Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, Bursa Teknik Üniversitesi’nin aboneliği olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

İNTİHAL BEYANI

Bu tezde görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, tez içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde kaynak göstererek belgelediğimi, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Öğrencinin Adı Soyadı: Burhan ENGİN

İmzası:



ÖNSÖZ

Yüksek lisans tezim sırasında bana desteğini hiçbir zaman esirgemeyen aileme,
Her türlü konuda bana bir şekilde çözüm yolu sunup desteğini hiçbir zaman
esirgemeyen danışman hocam Songül AKBULUT ÖZEN'e,

Tez süreci boyunca beni destekleyip motive eden yakın arkadaşlarım Tarık KAPLAN
ve Mert YILDIZ'a, başta Orhan ŞENTÜRK ve Ulaş KESKİN olmak üzere tüm TŞL
grubu üyelerine,

Tezimi tamamlama sürecinde destek olan çok değerli hocalarım Hüseyin DAĞ'a ve
Necati ÇELİK'e,

Bilim İnsanı Destek Programları Başkanlığı (BİDEB) 2211-Yurt İçi Lisansüstü Burs
Programı kapsamında destek olan TÜBİTAK'a teşekkür ederim.

Eylül 2024

Burhan ENGİN

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vi
İÇİNDEKİLER	vii
KISALTMALAR	viii
ÇİZELGE LİSTESİ.....	ix
ŞEKİL LİSTESİ.....	x
ÖZET.....	xi
SUMMARY	xii
1. GİRİŞ	1
1.1 Tezin Amacı	7
1.2 Temel Kavramlar.....	7
1.2.1 Kuantum mekaniksel bazı temel kavramlar	8
1.2.2 Kuantum bilgisayarlar	8
1.2.3 Kübit kavramı	10
1.2.3.1 Çoklu kübitler	12
1.2.4 Kuantum kapıları.....	12
1.2.4.1 Kuantum kapısı (NOT kapısı).....	13
1.2.4.2 Hadamard kapısı.....	14
1.2.4.3 S kapısı.....	15
1.2.4.4 CNOT kapısı	16
1.2.4.5 Z kapısı.....	16
1.2.4.6 RZZ kapısı.....	17
1.2.5 Kuantum algoritmaları	17
1.2.5.1 Gürültülü orta ölçekli kuantum (NISQ) makinesi.....	18
1.2.5.2 Kuantum algoritma yazılım ve kütüphaneleri (qiskit, cirq).....	18
1.2.6 Önemli algoritmalar	18
1.2.6.1 Deutsch algoritması.....	19
1.2.6.2 Deutsch–Jozsa algoritması	21
1.2.6.3 Shor algoritması	22
2. MATERYAL VE YÖNTEM.....	26
2.1 Kuantum Yaklaşık Optimizasyon Algoritması (QAOA).....	26
2.2 Optimizasyon	34
2.3 Test ve Doğrulama	36
3. BULGULAR VE TARTIŞMA.....	38
3.1 Kuantum Bilgisayarı Sonuçları	38
3.2 QAOA Devresinde γ ve β Parametrelerinin Optimizasyonu.....	40
3.3 Yapılan Optimizasyonun Doğruluğunun Test Edilmesi	48
4. SONUÇ VE ÖNERİLER.....	50
KAYNAKÇA	52
ÖZGEÇMİŞ.....	61

KISALTMALAR

BPP	: Box Packaging Problem
COP	: Combinatorial Optimization Problems
GNFS	: General Number Field Sieve
GPR	: Gaussian Process Regression
HL-LHC	: High Luminosity Large Hadron Collider
KP	: Knapsack Problem
LR	: Linear Regression
MIS	: Maximum Independent Set
MSE	: Mean squared error
NISQ	: Noisy Intermediate-Scale Quantum
NN	: Neural Network
PO	: Portfolio Optimization
QAOA	: Quantum Approximate Optimization Algorithm
QEC	: Quantum Error Correction
QECC	: Quantum Error Correction Code
QMC	: Quantum Monte Carlo
QUBO	: Quadratic Unconstrained Binary Optimization
RBF	: Radial Basis Function
RSA	: Rivest–Shamir–Adleman
RSVM	: Support Vector Machine Regression
RTREE	: Regression Tree
SVM	: Support vector machine
TL	: Transfer Learning
TSP	: Travelling Salesman Problem

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 1.1 : Kuantum kapılarının kübitlere uygulanma listesi.	19
Çizelge 1.2 : XOR kapısının doğruluk tablosu.....	20
Çizelge 1.3 : Farklı r değerleri için $2^r \bmod 15$	23
Çizelge 2.1 : 2 Şehir için şehirlerarası mesafeler.	30
Çizelge 3.1 : Parametrelere ve mesafeye bağlı 10 örnek.....	38
Çizelge 3.2 : Parametre modellerinin MSE ve R^2 sonuçları.	43
Çizelge 3.3 : Farklı makine öğrenmesi yöntemlerinin mesafelere göre doğru sonuç olasılığı.	48

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Bloch küresi.....	11
Şekil 1.2 : X kapısı diyagram gösterimi.	13
Şekil 1.3 : Hadamard kapısının diyagram gösterimi.	15
Şekil 1.4 : CNOT kapısının diyagram gösterimi.	16
Şekil 1.5 : Deutsch'un algoritmasını uygulayan kuantum devresi.....	19
Şekil 1.6 : Shor algoritmasının devre diagramı.	25
Şekil 2.1 : Çalışmanın şematik gösterimi.	28
Şekil 2.2 : Şehirler ve aralarındaki mesafenin örnek çizimi.....	30
Şekil 2.3 : Şehir ve mesafelerini hazırlayan kod.	32
Şekil 2.4 : TSP sınıfını oluşturma kodu.....	32
Şekil 2.5 : Kuantum devresi oluşturup çalıştırılan kod.....	33
Şekil 2.6 : 4 kubitli ansatz devresinin γ_0 ve β_0 için uygulanmış gösterimi.....	33
Şekil 3.1 : Simülasyon ile elde edilmiş ölçüm sonuçları.	39
Şekil 3.2 : Kuantum bilgisayar ile elde edilmiş ölçüm sonuçları.....	39
Şekil 3.3 : Veri setinde parametrelerin birbirleriyle ilişki grafiği.....	41
Şekil 3.4 : QAOA parametrelerinin performans sıcaklık haritası.....	42
Şekil 3.5 : Gerçek ve tahmin edilen parametrelerin karşılaştırması.....	43
Şekil 3.6 : SVM modeli için gerçek ve tahmin edilen parametrelerin karşılaştırması.	44
Şekil 3.7 : SVM modeli için parametrelerin MSE (Mavi) ve R^2 (Kırmızı) değerleri.	44
Şekil 3.8 : Random Forest modeli için gerçek ve tahmin edilen parametrelerin karşılaştırması.....	45
Şekil 3.9 : Random Forest modeli için parametrelerin MSE (Mavi) ve R^2 (Kırmızı) değerleri.....	46
Şekil 3.10 : Farklı makine öğrenmesi yöntemlerinin parametre tahminindeki hata karşılaştırması.....	47

MAKİNE ÖĞRENMESİ KULLANARAK QAOA PARAMETRELERİNİN İKİ ŞEHİRLİ GEZGİN SATICI PROBLEMİ İÇİN OPTİMİZASYONU

ÖZET

Bu tez çalışmasında, Kuantum Yaklaşık Optimizasyon Algoritması (QAOA) kullanılarak, iki şehirli Gezgin Satıcı Problemi'nin (TSP) çözümü ve QAOA parametrelerinin makine öğrenmesi ile optimize edilmesi ele alınmıştır. QAOA, kuantum hesaplama yöntemlerinden biri olup, klasik optimizasyon problemlerine etkin çözümler sunmaktadır. Çalışmanın amacı, QAOA yönteminin gamma ve beta parametrelerini optimize ederek, farklı makine öğrenmesi modelleri ile bu parametrelerin etkili bir şekilde tahmin edilmesini sağlamaktır.

İlk aşamada, iki şehirli TSP problemi için QAOA yöntemi ile kuantum devreleri oluşturulmuş ve Aer Simülasyon kullanılarak simülasyonlar gerçekleştirilmiştir. Optimizasyon sürecinde gamma ve beta parametreleri çeşitli başlangıç değerleri ile test edilerek en iyi performansı sağlayan parametreler belirlenmiştir. Parametrelerin optimizasyonu sonucunda elde edilen gamma ve beta değerleri, problem mesafesine (distance) göre örneklenmiş ve sonuç olasılıkları (sol_rates) hesaplanmıştır.

Farklı 5000 parametre ile yapılan hesaplamada optimizasyon için gereken veriler hazırlanmıştır. Elde edilen veriler, StandardScaler kullanılarak ölçeklendirilmiş ve bu verilerle Sinir Ağları Regresyonu, Destek Vektör Makineleri (SVM) ve Random Forest Regresyonu modelleri oluşturulmuştur. Modeller, verilen mesafe değeri için gamma ve beta parametrelerinin doğru sonucun olasılıklarını 0,2'den büyük yapacak şekilde tahmin etmeye yönelik olarak eğitilmiştir. Eğitim ve test aşamalarında, modellerin performansları değerlendirilmiş ve çok düşük R^2 ve çok yüksek Ortalama Kare Hata (MSE) değerleri elde edilmiştir.

Bu düşük performans değerlerine rağmen, çalışmada elde edilen sonuçlar, makine öğrenmesi parametrelerinin optimize edilmesi, model girdilerinin ve model çıktılarının değişkenlerinde yapılacak iyileştirmelerle gelecekte daha başarılı sonuçlar elde edilebileceğine işaret etmektedir. QAOA yöntemi ile elde edilen parametrelerin makine öğrenmesi yöntemleri kullanılarak başarılı bir şekilde tahmin edilebilmesi, klasik optimizasyon problemlerinin kuantum hesaplama yöntemleri ile çözümünde önemli bir adım olarak değerlendirilmektedir.

Bu çalışma, QAOA yönteminin optimizasyon problemlerinde kullanımının yanı sıra, kuantum ve klasik hesaplama yöntemlerinin entegrasyonunun potansiyelini ortaya koymaktadır. Gelecekteki çalışmalar, daha karmaşık TSP problemleri ve diğer optimizasyon problemleri üzerinde benzer yaklaşımlar ile genişletilebilir.

Anahtar kelimeler: Kuantum bilgisayar, Kübit, Kuantum algoritma, Optimizasyon, QAOA, Gezgin Satıcı Problemi (TSP).

OPTIMIZING QAOA PARAMETERS FOR A TWO-CITY TRAVELING SALESMAN PROBLEM USING MACHINE LEARNING

SUMMARY

In this thesis, the solution of the Two-City Traveling Salesman Problem (TSP) using the Quantum Approximate Optimization Algorithm (QAOA) and the optimization of QAOA parameters through machine learning are discussed. QAOA is a quantum computing method that provides effective solutions to classical optimization problems. The aim of the study is to optimize the gamma and beta parameters of the QAOA method and to predict these parameters effectively using different machine learning models.

In the initial phase, quantum circuits were created using the QAOA method for the two-city TSP problem and simulations were performed using Aer Simulator. During the optimization process, various initial values for the gamma and beta parameters were tested, and the parameters that provided the best performance were determined. The optimized gamma and beta values were sampled according to the problem distance and solution probabilities (sol_rates) were calculated.

Data for optimization was prepared by performing calculations with 5000 different parameters. The obtained data were scaled using StandardScaler, and Neural Network Regression, Support Vector Machines (SVM), and Random Forest Regression models were created with this data. The models were trained to predict the gamma and beta parameters to make the solution probabilities greater than 0.2 for a given distance value. During the training and testing phases, the performance of the models was evaluated, and very low R^2 and very high Mean Squared Error (MSE) values were obtained.

Despite these low performance values, the results obtained in this study indicate that more successful results can be achieved in the future by optimizing the machine learning parameters and improving the model inputs and outputs. The successful prediction of the parameters obtained with the QAOA method using machine learning methods is considered an important step in solving classical optimization problems with quantum computing methods.

This study not only demonstrates the use of the QAOA method in optimization problems but also reveals the potential of integrating quantum algorithms with classical computing methods. Future studies can be extended with similar approaches to more complex TSP problems and other optimization problems.

Keywords: Quantum computer, Quantum algorithm, Optimization, Traveling Salesman Problem (TSP), Qubit.

1. GİRİŞ

Kuantum bilgisayarlar, klasik bilgisayarlardan farklı olarak kuantum mekaniğinin prensiplerini kullanarak işlem yapan yeni bir bilgisayar türüdür. Kuantum mekaniği, klasik fizikte gözlemlenemeyen, parçacıkların aynı anda birden fazla durumda bulunabilme (süperpozisyon) ve birbirleriyle etkileşime girebilme (kuantum dolanıklığı) gibi özelliklere sahip olmalarını sağlar. Bu özellikler, kuantum bilgisayarların klasik bilgisayarlara kıyasla önemli ölçüde daha hızlı ve verimli işlem yapabilme potansiyeline sahip olmalarını sağlar [1].

Son yıllarda kuantum bilgisayarları pek çok alanda kullanılmaya başlanmıştır. Bunlar içerisinde Rydberg atomlarına dayalı kuantum bilgi işlem [2], kuantum bilgisayarlarda ilaç tasarımı [3], temel durumun istatistiksel örneklemesini kullanan fermiyonik kuantum Monte Carlo (QMC) yöntemleri ile etkileşen çok elektronlu problemlerin çözümü [4], kuantum malzemelerin kuantum bilgisayarlarla simülasyonu [5] ve kuantum bilgisayarları ile makine öğrenmesi [6] gibi alanlar örnek verilebilir.

Kuantum bilgisayarlarının en büyük problemi var olan kuantum kapılarının klasik kapılara göre çok daha fazla hatalı çalışmasıdır. Bu nedenle, kuantum hesaplama platformlarının çoğu uzun vadede hata düzeltmeli kuantum devreleri gerçekleştirme hedefiyle tasarlanmıştır. Her bir kübitin yedekli olarak kodlanması ve hataları tespit edip düzeltmek için kontrol operatörlerinin tekrar tekrar ölçülmesi gibi yöntemlerle hesaplama süresi uzatılarak kuantum devresi güvenilir bir şekilde çalıştırılabilir [7].

Gelecekte kuantum bilgisayarlarının birçok problemin çözümünde kullanılması öngörülmektedir. Kuantum bilgisayarları kuantum araştırmaları, yüksek enerji fiziği, HL-LHC gibi gelecekteki diğer çarpıştırıcılardan beklenen büyük miktarda veri alanlarında [8], varlık yönetimi, yatırım bankacılığı, bireysel ve kurumsal bankacılık gibi finans alanındaki problemler için ve bunun gibi diğer pek çok alanlarda kullanılmak için geliştirilmeye devam etmektedir.

Kuantum bilgisayarlarının gelişmesini sağlayan en büyük motivasyon kaynağı, Shor Algoritması adı verilen ve Peter Williston Shor tarafından geliştirilen, büyük sayıların çarpanlarına ayrılması için kullanılan kuantum algoritması olmuştur[9].

Kuantum algoritmalar, kuantum bilgisayarların gücünden yararlanarak klasik algoritmalara kıyasla önemli ölçüde daha hızlı ve verimli çözümler sunabilen algoritmalar. Kuantum algoritmalar, süperpozisyon ve kuantum dolanıklığı gibi kuantum mekaniğinin özelliklerini kullanarak klasik algoritmaların çözemediği karmaşık problemleri çözmek için kullanılabilir.

Optimizasyon problemleri, modern dünyanın her köşesinde karşımıza çıkan karmaşık ve zorlayıcı problemlerdir. Mühendislik, finans, lojistik, üretim, sağlık, yapay zekâ ve daha birçok alanda, en iyi çözümü bulmak için optimizasyon teknikleri kullanılmaktadır. Bu problemler, genellikle çok sayıda değişkene ve karmaşık ilişkilere sahip olmasından dolayı klasik algoritmalar tarafından çözümleri uzun zaman alabilir ve hatta bazı problemler için imkânsız hale gelebilir.

Kuantum Yaklaşık Optimizasyon Algoritmaları (QAOA), bu problemlere yeni bir bakış açısı sunarak, kuantum hesaplamanın gücünü kullanarak çözümleri hızlandırma potansiyeline sahiptir. Kuantum bilgisayarlar, klasik bilgisayarlardan farklı olarak, kuantum mekaniğinin prensiplerini kullanarak işlem yapmaktadır. Bu sayede, klasik bilgisayarların çözemediği veya çok uzun sürede çözdüğü problemleri çok daha hızlı bir şekilde çözebilme potansiyeline sahiptir [10–14]. Ancak çoğu problem için kuantum bilgisayarların klasik bilgisayarlara göre bir avantaj sağlayıp sağlamadığı bilinmemektedir [15].

Quantum Approximate Optimization Algorithm (QAOA)'in temel prensibi, bir optimizasyon probleminin çözümünü bir kuantum bit (kübit) sistemi üzerinde kodlamaktır. Kübitler, klasik bitlerin aksine, 0 ve 1 durumlarının süperpozisyonunda bulunabilirler. Bu sayede, QAOA klasik algoritmalara kıyasla çok daha geniş bir arama alanını keşfedebilir ve daha iyi çözümler bulabilir.

Kuantum hesaplama, hesaplamaları daha verimli bir şekilde gerçekleştirmek için kuantum mekaniksel etkilerden (özellikle süperpozisyon, dolanıklık ve kuantum tünelleme) yararlanır. Kuantum hesaplama, geleneksel dijital hesaplamaya kıyasla yürütme süresini ve enerji tüketimini önemli ölçüde azaltma potansiyeline sahiptir. Bu potansiyelin avantajları, nanofabrikasyondaki istikrarlı ilerlemeler ve Moore Yasası

gibi geleneksel donanım ölçeklendirme yasalarının yavaşlaması ile, 2010'larda kuantum hesaplama teknolojisinde ticari ve ulusal güvenlik ilgisine ve yatırımlara yol açmasıdır. Örneğin, geçtiğimiz günlerde Google, kuantum üstünlüğü olarak bilinen ve kuantum bilgisayarların klasik süper bilgisayarlarla tamamlanması zor olan hesaplamaları gerçekleştirebildiğini gösteren önemli bir dönüm noktasına ulaştığını duyurdu [16].

Kuantum hesaplama, gerçek dünya sorunlarına doğrudan uygulanamayan sorunları çözer [17]. Kuantum algoritmaları birçok farklı problemi çözmek için geliştirilmiştir. Bu algoritmaların odaklandıkları problemlerin temelleri farklı olduğu gibi, aynı problem için farklı yaklaşımlar da vardır. Bunların dışında algoritmaların teorik olarak çalışmasını sağlamak için kuantum hata doğrulama (QEC) çalışmaları da yapılmaktadır.

Kuantum hata düzeltmesi konusu ile ilgili 1995'ten beri etkili çalışan kodlar mevcuttur [18]. Hata düzeltmesi üzerine yapılan ilk araştırmalar ağırlıklı olarak QEC'nin yapısı ve işleyişi için kuantum kodlarının geliştirilmesi ile yapılmıştır [19–26]. P. Shor, kuantum bellekte depolanan bilgiler için, dekoherans sürecinin, bellekte depolanan bitlerin her biri üzerinde bağımsız olarak hareket ettiği varsayılarak, etkilerinin nasıl azaltılacağını göstermiştir [27].

Kuantum hata düzeltme kodlarında topolojik kodlar da vardır. Kodun yapısı sayesinde, tek bir kübitteki hata, diğer kübitleri etkilemez ve asıl bilgi korunur. Bu koruma topolojik özellikler sayesinde gerçekleşir. Bilgi, kübitlerin düzenine yerleştirilir ve bu sayede bir kübitteki hata diğer kübitleri etkilemez. Bu konuda çalışmalar devam etmektedir [28–30]. Topolojik bir kodu oluşturmak için topolojik düzenin karakterizasyonu ve tespiti, dolanıklık ve yerel operatörler kullanılarak, farklı topolojik aşamaları ayırt etmenin zorluğu sebebiyle, çok önemli bir işlemdir [31]. Bu işlem için dolanıklık entropisi [32, 33], kuantum Fisher bilgisi [34], kuantum devre karmaşıklığı [31], indirgenemez çoklu parti korelasyonu [35, 36] sıralanabilecek önemli parametrelerdendir. Bu çözümler, topolojik dolanıklık entropisinin belirlenmesi için büyük ölçüde sınırın düzgünlüğüne ihtiyaç duymasından dolayı küçük düzenlemeler gerektirdiği gibi başka problemler de çıkarabilir [37, 38].

Alt sistem kuantum hata düzeltme kodu (QECC) iki temel bileşenden oluşur. Biri bilgi alt sistemi diğeri ölçü alt sistemdir. Bilgi alt sistemi, bilgiyi kübitlerin içerisinde 0 ve

1 olarak saklar. Ölçü alt sistemi ise bilgi sistemindeki hataları tespit etmek için kullanılır. Bunlar faz kübitleri olarak da adlandırılır. Birbirlerine bütünleşmiş bir şekilde çalıştıkları için kübitlerde meydana gelen bir hata ölçü alt sistemde değişikliğe yol açar. Bu değişiklik tespit edilerek hata düzeltilir. Bu hata düzeltme yöntemi farklı tür hataları da düzeltebilecektir. Ancak kodun tasarımı ve uygulaması diğer hata düzeltme yöntemlerine göre daha karmaşık olabilir. Ayrıca ölçü alt sistemi için fazladan kübit kullanılacaktır. Nielsen ve Poulin bu konu ile alakalı 2018’de yaptıkları çalışmada operatör kuantum hata düzeltmeyi genelleştirmeye çalışmışlardır [39]. Daha önce de dekoheranssız alt uzayları [40–43] ve gürültüsüz alt sistemler [26, 44, 45] hata düzeltme kodları ile genelleştirilmiştir. Bunların dışında bazı çalışmalar tüm bu teknikleri bütünleştiren ve birleştiren tek bir çerçeve sunmaktadır [46, 47].

Gezgin Satıcı Problemi (TSP), bir gezgin satıcının belirli sayıda şehri bir kez ziyaret edip başlangıç şehrine dönmesi gereken en düşük maliyetli rotayı bulma problemi olarak tanımlanır [48]. Bu problemi çözmek için çeşitli algoritmalar literatürde bulunmaktadır. Bu algoritmalara Karınca kolonisi optimizasyon (ACO) algoritması [49–52], Genetik Algoritma (GA) [53–55] ve Parçacık Sürü Optimizasyonu (PSO) [56–58] algoritması gibi örnekler verilebilir.

Bu tez çalışmasında ele alınan TSP kuantum algoritmalar ile çözümü ilk olarak 2004 yılında R. Martoňák ve diğ. tarafından kuantum tavlama kullanılarak yapılmıştır. Simetrik Gezgin Satıcı Problemi için bir yol-integral Monte Carlo kuantum tavlama yöntemi kullanılmaya çalışılmış ve performansı standart benzetimli tavlama ile karşılaştırılmıştır. Standart TSPLIB'nin 1002 şehir örneği üzerinde test edildiğinde, klasik olandan kesinlikle daha üstün olduğunu göstermişlerdir [59].

Yong Chen ve Pan Zhang 2006 yılında önceki çalışmalara devam ederek yerel optimumlardan kaçışta verimliliği artırmak amacıyla kuantum tavlama için alternatif bir TSP formülasyonu sunmuşlardır. Problem çözümü için üstel bir düşüşü çok yüksek doğrulukla doğruladığını ortaya koydular. TSPLIB95'te bazı standart gezgin satıcı problemlerinin simülasyon çalışmaları, standart benzetimli tavlamadan daha üstün olduğunu göstermişlerdir [60].

Eneko Osaba ve diğ 2021 yılında hibrit kuantum hesaplama ve tabu arama algoritması adında yeni bir çözüm sunmuşlardır. Bu yöntemin işleyişi kuantum erişimlerinin azaltılmasına yöneliktir. Yöntemin kalitesini test etmek için 7 farklı TSP problemi

kullanılmış ve elde edilen sonuçlar kuantum hesaplama kaynaklarına erişimi büyük ölçüde azaltırken, bölümlene problemlerini çözmek için umut verici sonuçlar sunan bir yaklaşım olduğunu öne sürmüşlerdir [61].

Kuantum Yaklaşık Optimizasyon Algoritması (QAOA), kuantum bilgisayarların potansiyelinden yararlanarak optimizasyon problemlerine etkin çözümler sunan yenilikçi bir yöntemdir. QAOA, klasik algoritmaların ötesine geçerek, büyük ve karmaşık veri kümeleri üzerinde daha hızlı ve daha doğru sonuçlar elde etmeyi mümkün kılar. Ancak, QAOA'nın etkinliği, doğru parametrelerin seçilmesine bağlıdır. Gamma ve beta parametrelerinin optimize edilmesi, QAOA'nın performansını doğrudan etkiler ve bu parametrelerin uygun şekilde belirlenmesi, çözüm kalitesini ve hesaplama hızını artırır [62]. İşlem performansı için en önemli etkilerden biri de QAOA'nın parametreleridir. Parametrelerin optimizasyonu ile ilgili çalışmalar son zamanlarda kendini göstermektedir [63–69].

Makine öğrenmesi, büyük veri kümeleri üzerinde karmaşık ilişkileri ve desenleri öğrenme yeteneği sayesinde kuantum algoritmalarında, algoritmanın verimliliğini arttırması için kullanılır.

Piotr Czarnik ve arkadaşları makine öğrenmesi yöntemleri kullanarak kuantum hata düzeltme tekniklerini geliştirmeyi incelemiştirler. Yöntem, büyük ölçüde Clifford kapılarından oluşan ve klasik olarak verimli bir şekilde simüle edilebilen kuantum devreleri aracılığıyla eğitim verilerini $\{X_i^{noisy}, X_i^{exact}\}$ olacak şekilde üretir. Bir IBMQ kuantum bilgisayarında 16 kübit ve 64 kübitlik gürültülü bir simülatörde, temel durum enerji problemi için büyüklük sırasına göre bir hata düzeltme elde edildiği bildirilmiştir [70].

Sumeet Khatri ve arkadaşları kuantum bilgisayarlar için kuantum algoritmalarının derlenmesi konusunda kuantum destekli kuantum derleme (QAQC) adını verdikleri bir hibrit kuantum-klasik algoritma önermişlerdir. Bu algoritmanın daha büyük kuantum algoritmaların derlenmesine izin vereceğini öne sürmüşlerdir [71].

Mitarai ve arkadaşları makine öğrenimi için kuantum devre öğrenimi adını verdikleri hibrit klasik-kuantum bir algoritma önermektedirler. Bir kuantum devresi ile öğrenim yaptıkları yöntemde parametrelerin yinelemeli optimizasyonu yüksek derinlikli devreyi oluşturmalarını sağlar. Makine öğrenimi için düşük derinlikli bir kuantum devresi ve klasik bir bilgisayarı hibritleştirilen bir çerçeve sunmaktadırlar [72].

Alexey Galda ve arkadaşları maksimum kesim (Max-Cut) problemi için QAOA parametrelerini optimize etmeyi amaçlamışlardır. Bu optimizasyonun ve parametrelerin farklı QAOA örnekleri arasında başarılı bir şekilde aktarılabilceğinin, grafiklerin yerel özelliklerine, özellikle de grafiklerin oluşturulduğu alt grafik türlerine dayalı olarak açıklanabileceğini ve tahmin edilebileceğini göstermişlerdir [66].

Montañez-Barrera ve arkadaşları bir problem örneğinin önceden eğitilmiş QAOA parametrelerini farklı kombinatorial optimizasyon problemleri (COP) örneklerinde yeniden kullanmak için bir yöntem olan transfer öğrenmesi (TL) üzerinde çalışmışlardır. Gezgin satıcı problemi (TSP), kutu paketleme problemi (BPP), sırt çantası problemi (KP), maksimum kesim (MaxCut) problemi, maksimum bağımsız küme (MIS) problemi ve portföy optimizasyonunun (PO) küçük örnekleri için optimum QAOA parametrelerini tahmin etmişlerdir. Problemler arasında en iyi aktarılabilir parametreleri üreten problemi BBP olarak raporlanmıştır [65].

Shree Hari Sureshbabu ve arkadaşları ağırlaştırılmış MaxCut (genelleştirilmiş MaxCut) problemi için QAOA parametrelerin optimizasyonunda yerel optimize edicilerin yakınsamasını iyileştirerek iterasyon sayısını ortalama 7,4 kat azaltan sezgisel bir yöntem ortaya koymuşlardır [73]. Buna benzer şekilde belirli bir QAOA derinliğinde, QAOA parametrelerinin tek bir tipik vektörün ağırlaştırılmış MaxCut örneklerine başarıyla aktarılabilceği göstermişlerdir [74].

Mahabubul Alam ve arkadaşları bir veri seti oluşturarak QAOA parametrelerini makine öğrenmesi yöntemleri ile optimize etmeye çalışmışlardır. Gauss Süreci Regresyonu (GPR), Doğrusal Regresyon (LR), Regresyon Ağacı (RTREE) ve Destek Vektör Makinesi Regresyonu (RSVM) olmak üzere 4 farklı makine öğrenmesi yöntemi kullanılarak 330 farklı problem ve her problem için 6 QAOA örneği ile optimum parametreleri oluşturulmuş ve bu veri seti ile eğitilmiştir. Sonuç olarak önerilen yaklaşımın optimizasyon yineleme sayısının ortalama %44,9 azaltılabileceği raporlanmıştır [75].

Sami Khairy ve arkadaşları QAOA parametrelerini takviyeli öğrenme (RL) ve çekirdek yoğunluk tahmini (KDE) yöntemleri ile optimize etmeye çalışmışlardır. Çalışmada veri seti qiskit ile Aer Simülasyonunda yapılmış ve küçük ölçekli problemler için parametreler optimize edilerek elde edilen değerler ile eğitim yapılmıştır. Önerdiğimiz RL ve KDE tabanlı yaklaşımların, yaygın olarak kullanılan

diğer kullanıma hazır optimize edicilerle karşılaştırıldığında verimlilik açısından 30,15 kata kadar arttığını raporlamışlardır [76].

QAOA parametrelerinin optimize edilmesi varyasyon problemlerinin kuantum algoritmalar ile çözülmesinde çok önemli rol oynadığı görülmektedir. Bu tezde Sinir Ağları Regresyon (NN), Destek Vektör Makineleri (SVM) ve Random Forest Regresyon (RF) olmak üzere 3 farklı makine öğrenmesi yöntemleri ile qiskit Aer Simülasyonunda elde edilen veriler üzerinde eğitim yapılarak QAOA parametreleri optimize edilmeye çalışılmıştır. Simülasyon kullanılarak elde edilen veriler klasik algoritmalar ile optimize edilmiş parametreler ile oluşturulmuştur. Öğrenim sonuçları karşılaştırılmış ve yeni problemler için model test edilmiştir. Elde edilen değerlerden büyük hesaplama gerektiren problemler için QAOA parametrelerini oluşturmak için klasik algoritmalarından daha hızlı olan makine öğrenmesi modelleri kullanılabileceği sonucuna varılmıştır. Modellerin doğruluk oranlarının düşüklüğü model eğitim parametreleri değiştirilerek artırılabilir. Nihai olarak bu yöntemin gelecek vadede bir yöntem olduğu anlaşılmıştır.

1.1 Tezin Amacı

Kuantum bilişimine ilgi, son zamanlarda büyük bir artış göstermiştir Ticari olarak kullanılması çok önceden beklense de bu kadar hızlı olması şaşırtıcı olmuştur [77]. Bu tez çalışmasının temel amacı, QAOA'nın performansını artırmak için gama ve beta parametrelerini optimize etmektir. Gama ve beta parametrelerinin TSP problemlerinde en iyi çözümü üretecek şekilde optimize edilmesi ve bu parametrelerin makine öğrenmesi teknikleriyle tahmin edilmesi, farklı makine öğrenmesi modelleri kullanarak, belirli bir mesafe değeri için gama ve beta parametrelerinin doğru sonucunun olasılığının 0.2'den büyük yapacak şekilde tahmin edilmesi ve QAOA'nın performansını artıran metodolojilerin ve sonuçların literatüre kazandırılması çalışmamızı özgün kılmaktadır.

1.2 Temel Kavramlar

Kuantum yaklaşık optimizasyon algoritmalarını anlamak için önce kuantum mekaniğini ve kuantum bilgisayarlarının nasıl çalıştığını anlamak gerekir. Bu bölümde kuantum fiziğinin temel kavramlarına yer verilmiştir.

1.2.1 Kuantum mekaniksel bazı temel kavramlar

Kuantum mekaniği, fiziksel teorilerin geliştirilmesi için matematiksel bir çerçevedir. Kuantum mekaniği kendi başına fiziksel bir sistemin hangi yasalara uyması gerektiğini söylemez, ancak bu tür yasaların geliştirilmesi için matematiksel ve kavramsal bir çerçeve sağlar[1].

Herhangi bir izole fiziksel sistemle ilişkili, sistemin durum uzayı olarak bilinen iç çarpımlı karmaşık bir vektör uzayı (yani bir Hilbert uzayı) vardır. Sistem, durum uzayında bir birim vektör olan durum vektörü tarafından tanımlanır. Kuantum mekaniksel olarak en çok ilgileneceğimiz en basit sistem olarak kübiti ele alalım. Bir kübitin iki boyutlu bir durum uzayı vardır. Durum uzayları $|0\rangle$ ve $|1\rangle$ 'in bu durum uzayı için ortonormal bir temel oluşturduğunu varsayalım. Bu durumda durum uzayındaki keyfi bir durum vektörü

$$|\psi\rangle = a|0\rangle + b|1\rangle, \quad (1.1)$$

yazılabilir. Burada a ve b karmaşık sayılardır. $\langle\psi|\psi\rangle$ 'nin normalize durum olması koşulu $\langle\psi|\psi\rangle = 1$ olmasıdır. Bu nedenle $|a|^2 + |b|^2 = 1$ 'e eşdeğerdir.

Kapalı bir kuantum sisteminin evrimi üniter bir dönüşümle tanımlanır. Yani, sistemin t_1 zamanındaki durumu $|\psi_1\rangle$, sistemin t_2 zamanındaki durumu $|\psi_2\rangle$ ile yalnızca t_1 ve t_2 zamanlarına bağlı olan üniter bir operatör

$$|\psi'\rangle = U|\psi\rangle, \quad (1.2)$$

U ile ilişkilidir. Kuantum mekaniği bize belirli bir kuantum sisteminin durum uzayını veya kuantum durumunu söylemediği gibi, hangi üniter operatörlerin gerçek dünya kuantum dinamiklerini tanımladığını da söylemez. Kuantum mekaniği bize yalnızca herhangi bir kapalı kuantum sisteminin evriminin böyle bir olasılığa sahip olduğunu gösterir [1].

1.2.2 Kuantum bilgisayarlar

Kuantum bilgisayarlar, kuantum mekaniği ilkelerine dayalı olarak çalışan devrim niteliğinde bir bilgi işlem teknolojisidir. Bilgi depolamak ve işlemek için bitleri kullanan klasik bilgisayarların aksine kuantum bilgisayarlar, hem 0 hem de 1 dahil olmak üzere aynı anda birden fazla durumda bulunabilen kübitleri kullanır. Bu benzersiz özellik, kuantum bilgisayarların katlanarak daha hızlı bir şekilde hesaplama

yapmasına ve klasik bilgisayarların zorlandığı karmaşık sorunların üstesinden gelmesine olanak tanır.

Kuantum bilgisayarları yapısında kuantum özelliği gösteren parçacıkları kullanarak bilgi işlemini gerçekleştirirler. Kuantum parçacıklarının en önemli özellikleri süperpozisyon durumunda bulunabilmeleridir. Kuantum bilgisayarlarında, klasik bilgisayarlarda bit adı verilen 0 ve 1 değer alabilen transistör anahtarlama yapıları yerine kuantum bit adı verilen kübitler yer almaktadır. Süperpozisyon durumundaki kuantum parçacıkları kullanılarak bütün işlemler sıralı işlem yerine paralel işlem ile gerçekleşir. Bu nedenle işlem gücü üstel olarak artmaktadır. Klasik bilgisayarlarda işlem süresi üstel olarak artarken kuantum bilgisayarlarında ise lineer bir artış vardır. Sonuç olarak klasik bilgisayarların pratik olmayan bir sürede yapacağı görevi kuantum bilgisayarları çok daha kısa sürede gerçekleştirebilmektedir.

Gelinen noktada kuantum bilgisayarlar hızlarına ek olarak, klasik bilgisayarların erişemeyeceği karmaşık problemleri çözme potansiyeline de sahiptir. Örneğin, kuantum bilgisayarlar daha doğru ve verimli simülasyonlar sağlayarak ilaç keşfi ve protein katlanması gibi alanlarda devrim yaratabilir. Bu gelişmeler, çeşitli bilimsel ve teknolojik alanlarda önemli atılımlara yol açacaktır. Bütün bunlara ek olarak günümüz bilgi teknolojisinin en önemli özelliklerinden birisi olan güvenlik önlemleri, klasik bilgisayarların çözmesinin uzun yıllar sürebileceği karmaşıklıkta şifrelemeler ile alınmaktadır. Örnek olarak, RSA (Rivest–Shamir–Adleman) şifreleme gibi klasik bilgisayarların çözemediği problemleri, kuantum bilgisayarları yeterli sayıda kübit ile çözmeyi mümkün kılar.

Bunların dışında kuantum bilgisayarlarının bazı zorlukları mevcuttur ve hala gelişmektedir. Kullanılan kuantum kapılar %100 doğrulukla çalışmamakla beraber sonuçların üzerinde gürültü oluşturmaktadırlar. Bunun yanı sıra kuantum özelliklerini kontrol edebilmek için çok fazla maliyet harcanması gereklidir. Özellikle kullanılan kapıların hatalarının azaltılmasına yönelik çalışmalar yapılmaktadır [78].

Kuantum bilgisayarlar önemli ölçüde bir potansiyel sunarken, her probleme çözüm üretemeyebilir. Teknoloji ilerlemeye devam ettikçe, araştırmacılar kuantum bilgisayarların etkili bir şekilde çözebileceği, problem yelpazesini genişletmenin yollarını araştırmaktadırlar.

Kuantum bilgisayarlar, çeşitli kuantum sistemleri ile oluşturulabilir. Bunlar tuzaklanmış iyonlar [79, 80], süperiletken kubitler [81, 82], fotonlar [83–87] ve silikon [88, 89] ile yapılmış kuantum bilgisayarlardır. Bunların arasından süperiletken kuantum bilgisayarları, ölçeklenebilirliği ve bir elektronik devre olması sebebiyle ön plana çıkar. Zaten şu anda geliştirilmiş olan bilgisayar sistemleri bazı dönüşümler ve entegrasyonlar ile kuantum bilgisayarları ile çalışabilirler. 1999 yılında Nakamura ve diğ. [90] ilk süperiletken kuantum bilgisayarını geliştirmişlerdir. Bu tarihten sonra gelişimi hızla devam etmiştir. 2014 yılında 5 kubitli bilgisayar geliştirilebilmiştir [91].

Sonuç olarak, kuantum bilgisayarlar bilgi işlem teknolojisinde çığır açan bir ilerlemeyi temsil etmektedir. Kuantum mekaniği ilkelerinden yararlanma yetenekleri, daha hızlı hesaplama, karmaşık sorunları çözme ve veri şifrelemeyi geliştirme için yeni olanaklar sunmaktadır. Bununla birlikte, karmaşıklık, hata hassasiyeti ve sınırlı kullanım alanları gibi zorlukların tam olarak üstesinden gelinmesi gereken bazı durumlar olarak sıralanabilir.

1.2.3 Kübit kavramı

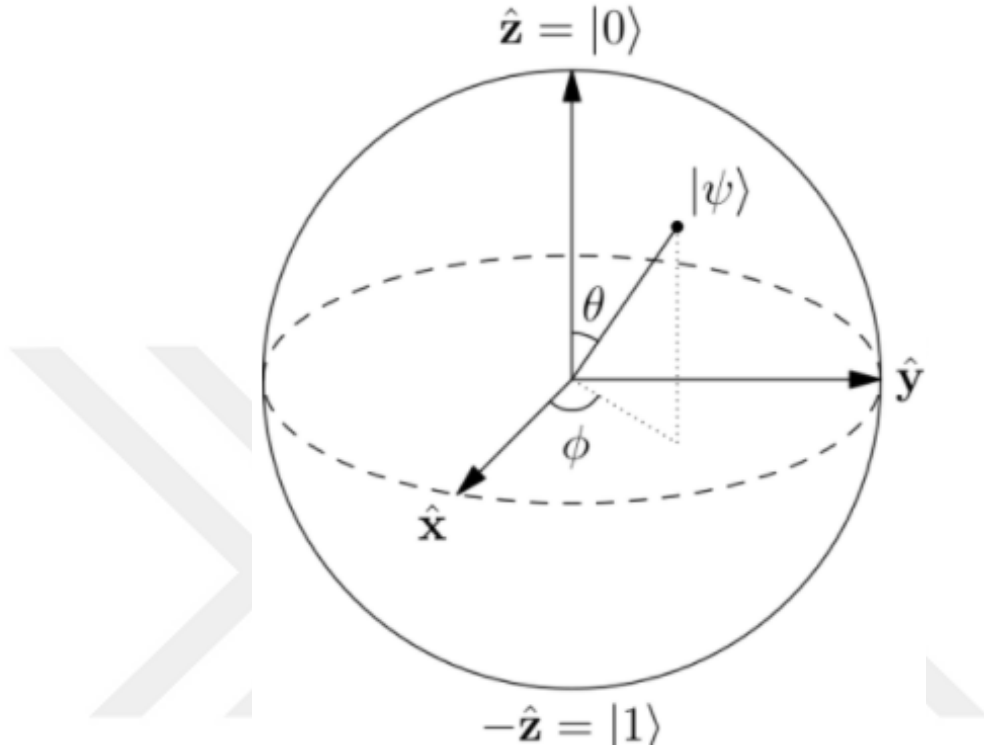
Bir kübit, kuantum bilgisayarlarda temel bilgi birimi olarak görev yapar ve onu bitlerin kullanıldığı klasik bilgisayarlardan ayırır. Bitler yalnızca 0 ya da 1'i temsil edebilirken, kubitler kuantum mekaniği ilkeleri nedeniyle her iki durumda da aynı anda var olma gibi benzersiz bir yeteneğe sahiptir.

Bir kübitin süperpozisyon durumuna alınması tam doğrulukla gerçekleşmez. Bu nedenle hesaplamalarda hata payları oluşabilir. Bu hata paylarını minimize edebilmek için kuantum hata doğrulama gibi bazı yöntemler kullanılır. Ayrıca kubitler çok hassastırlar ve bu nedenle kubitleri oluşturmak çok maliyetli ve zordur.

Temel kuantum fiziği kurallarına baktığımızda bir parçacık süperpozisyon durumundaysa olası bütün durumları aynı anda gerçekleştiriyor demektir. Ancak eğer siz onun durumunu ölçmek isterseniz sonuç sadece bir duruma çökecektir. Elektronlar bir kuantum parçacığdır ve elektronlar kullanılarak kubitler elde edilir. Stern-Gerlach deneyinde bir elektronun iki spin durumu olabildiği görülmüştür. Bunlar

$$|e\rangle = \frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle, \quad (1.3)$$

şeklinde gösterilir. Burada $\frac{1}{\sqrt{2}}$ o durumun gerçekleşme ihtimalini gösterir. $|\uparrow\rangle$ ve $|\downarrow\rangle$ için ise hesaplama biliminde $|0\rangle$ ve $|1\rangle$ durumları kullanılır. Kübitlerin muhtemel durumlarının geometrik gösterimi için Bloch küresi kullanılır. Şekil 1'de ψ durumundaki Bloch küresi gösterilmiştir.



Şekil 1.1 : Bloch küresi.

Şekil 1'de görüldüğü gibi Bloch küresi $|\psi\rangle$ durumunda bulunan bir kübitin teorik durum vektörünün gösterimini verir. Burada θ durum vektörünün z eksenine yaptığı açıyı ve $|0\rangle$ veya $|1\rangle$ olma ihtimalini gösterir. ϕ ise vektörün x ve y eksenleri ile yaptığı açıyı verir ve bu da faz durumunu temsil eder. Kübitlere bazı kuantum kapılar uygulanarak sonuç vektörünün istenilen duruma gelmesi sağlanabilir. Bloch küresi için örnek Şekil 1.1'de de gösterilmiştir.

Denklem 1.3'te $|\uparrow\rangle$ ve $|\downarrow\rangle$ yerine $|0\rangle$ ve $|1\rangle$ yazarsak katsayılarını da α ve β olarak belirtirsek $|\alpha|^2 + |\beta|^2 = 1$ olur. Daha sonra bu ifadeyi;

$$|\psi\rangle = e^{i\gamma} \left(\cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle \right) \quad (1.4)$$

şeklinde düzenlenebilir. Burada $e^{i\gamma}$ ifadesi ölçülebilir bir değer olmadığından yok sayılabilir [1]. Sonuç olarak elde ettiğimiz denklem;

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right)|1\rangle \quad (1.5)$$

şeklini alır. Bloch küresi sadece tek bir kübitin gösterilmesine olanak tanır. Çoklu kübit gösterimi ile alakalı çalışmalar[92] olsa da ortak kabul görmüş bir gösterim bulunmamaktadır.

Kübitlerin kuantum durumları işlemleri daha iyi anlayabilmek ve yorumlamak için genellikle matris gösterimleriyle tanımlanır. Bir kübitin durumlarını ifade eden matrisler

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (1.6)$$

şeklinde gösterilmektedir.

Bir kübitin ne kadar veri saklayabileceği konusunda, Bloch küresine ve Denklem 1,5'e bakıldığında teorik olarak sonsuz olması gerekmektedir. Çünkü θ sonsuz değer alabilir. Ancak ölçüm yaptığımızda sadece bir değere çöktüğünü hatırlarsak kübit ile sadece 1 veya 0 verilerini işleyebiliriz. Burada çoklu kübit sistemleri ile daha fazla veri işlememiz sağlanmaktadır.

1.2.3.1 Çoklu kübitler

Çoklu kübitler ile 2'lik sistemdeki veri bilimini kullanarak işlem yapabiliriz. Örneğin 5 sayısının 2'lik sistemde karşılığı 101'dir. Bu şekilde 5 sayısını kübitlere işlenecek olursa 3 kübit gerekir ve durumları 1 veya 0 durumuna getirerek $|101\rangle$ durum vektörü şeklinde ifade edilebilecektir. Örnek olarak 2 kübit olan sistem için kuantum durumu

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \quad (1.7)$$

şeklinde gösterilir. Burada katsayıların karesi $|\alpha_{ij}|^2$ ölçüm yapıldığında o durumun gelme olasılığını ifade eder.

Çoklu kübitlerle işlem yapılırken CNOT kapısı gibi kullanılan bazı kuantum kapılar iki kübitin kombinasyonu ile CNOT kapısı gibi bazı kuantum kapılar ise üç kübitin kombinasyonu ile çalışmaktadır.

1.2.4 Kuantum kapıları

Kuantum kapıları kübitleri manipüle ederek sonuç vektörünü belirli faz durumlarına ve açılara döndürür. Bu şekilde bir dizi kuantum kapısı ile istenen problemin çözümü

için devre kurulur. Kurulan devrenin sonunda ölçüm yapılır ve ölçüm sonucunda çıkan değere göre problem çözülmüş olur.

Kuantum devreleri kurmak için bazı temel kapılar vardır. Bunların farklı kombinasyonları ile yeni kapılar da oluşturulabilir. Bazı temel kuantum kapıları aşağıda anlatılmıştır.

1.2.4.1 Kuantum kapısı (NOT kapısı)

X kapısı, DEĞİL kapısı veya invertör olarak da bilinen temel bir mantıksal kapıdır. Tek bir girişi ve bir çıkışı vardır. X kapısının çalışma prensibi oldukça basittir: Giriş sinyali ne olursa olsun, çıkış sinyali tersidir. Burada kuantum bilgisayarı için kullanılan X kapısı $|0\rangle$ durumuna uygulandığında $|1\rangle$ durumuna dönüştürür. X kapısının matris gösterimi

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (1.8)$$

şeklindedir. Eğer kuantum durumu $\alpha|0\rangle + \beta|1\rangle$ ise matris gösterimi

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (1.9)$$

şeklinde olur. X kapısı uygulandığında ise

$$X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \quad (1.10)$$

şeklinde olur. Burada görüldüğü gibi $|0\rangle$ ve $|1\rangle$ olma ihtimalleri X kapısı uygulandığında yer değiştirmiştir.

Kuantum kapılarının birbirleriyle etkileşimlerinde daha karmaşık devreler kurulacağı için diyagram gösterimleri ile belirtilerek kolay ayrışması sağlanmaktadır. X kapısının kuantum devre diyagram gösterimi Şekil 2'de gösterilmiştir.



Şekil 1.2 : X kapısı diyagram gösterimi.

1.2.4.2 Hadamard kapısı

Hadamard kapısı, kuantum bilgisayarlarda kullanılan temel bir kuantum mantık kapısıdır. Hadamard kapısı en temel anlamda kubitleri süperpozisyon durumuna getiren kapıdır. Hadamard kapısı tek bir kubit üzerinde işlem yapar. Hadamard kapısından sonra kubit, hem $|0\rangle$ hem de $|1\rangle$ durumunda eşit olasılıkla bulunma durumuna getirilir.

Hadamard kapısının matris gösterimi

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.11)$$

şeklindedir. Hadamard kapısı $|0\rangle$ ve $|1\rangle$ durumlarında

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad (1.12)$$

farklı sonuçlar üretir. Denklem 1.12'de görüldüğü gibi Hadamard $|0\rangle$ durumuna etki ettirildiğinde sonuç içerisinde $|1\rangle$ de bulunur. Bu da süperpozisyon durumunu gösterir. Eğer kuantum durumu $\alpha|0\rangle + \beta|1\rangle$ ise Hadamard kapısı uygulandığında matris gösterimi

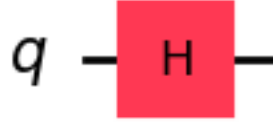
$$\alpha H|0\rangle + \beta H|1\rangle = \alpha \frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad (1.13)$$

olur. Hadamard kapısının bir diğer özelliği ise üniter olmasıdır yani hadamard kapısından geçmiş bir kubit tekrar geçerse ilk durumuna geri döner. Denklem 2.12'de Hadamard kapısından geçirilen $|0\rangle$ kubitini tekrar Hadamard kapısından geçirirsek durum vektörü

$$H \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 + 1 = 2 \\ 1 - 1 = 0 \end{pmatrix} = \frac{1}{2} 2|0\rangle = |0\rangle \quad (1.14)$$

şeklini alır. Denklem 1.14'te görüldüğü gibi $|0\rangle$ ile başlayan kubit 2 kez hadamarddan geçirilince sonuç tekrar $|0\rangle$ olur.

Hadamard kapısının diyagram gösterimi Şekil 3'te verilmiştir.



Şekil 1.3 : Hadamard kapısının diyagram gösterimi.

1.2.4.3 S kapısı

S kapısının görevi kübitin faz durumunu değiştirmektir. Bu nedenle bir diğer adı Faz Kapısı'dır. Bir başka özelliği ise, kuantum hesaplamada özel bir kapı seti olan belirli bir kübit sayısına etki eden, Clifford kapısı olmasıdır [93]. Kübit durum vektörünü Z yönünde $\frac{\pi}{2}$ kadar döndürür.

S kapısının matris gösterimi

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad (1.15)$$

şeklinde verilir. Bu matristen görüneceği gibi $|0\rangle$ durumu ile etki ettirildiğinde durumunu değiştirmeyecektir. Sadece $|1\rangle$ durumunda faz dönüşümü gerçekleşir. S kapısı uygulandığında durum vektörü

$$S|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, S|1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ i \end{pmatrix} = i|1\rangle \quad (1.16)$$

şeklini alır. Denklem 1.16'da görüldüğü gibi S faz dönüşümü $\frac{\pi}{2}$ kadar döndürülmesi sadece $|1\rangle$ durumu için geçerlidir. Burada i şeklinde bahsedilen

$$e^{\frac{i\pi}{2}} = \cos\left(\frac{\pi}{2}\right) + i\sin\left(\frac{\pi}{2}\right) = 0 + i = i \quad (1.17)$$

euler dönüşümü ile $\pi/2$ olduğu belli olmaktadır. Aslında S kapısı P kapısının $\frac{\pi}{2}$ değer almış halidir. P kapısının matris gösterimi

$$P(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix} \quad (1.18)$$

şeklindedir. Burada bazı özel açılar Z, S ve T kapıları

$$P(\lambda = \pi) = Z, \quad (1.19)$$

$$P\left(\lambda = \frac{\pi}{2}\right) = S, \quad (1.20)$$

$$P\left(\lambda = \frac{\pi}{4}\right) = T, \quad (1.21)$$

şeklinde ifade edilir.

1.2.4.4 CNOT kapısı

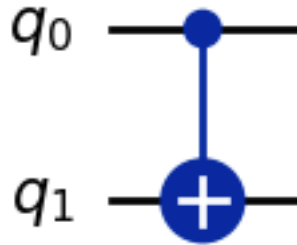
CNOT kapısı X kapısı ile aynı işlemi yapar ancak ikinci bir kübiti kontrol ederek yapar. Eğer ikinci kübitin durumu $|1\rangle$ ise X durumunu uygular, $|0\rangle$ ise durumunu değiştirmez.

CNOT kapısının matematiksel gösterimleri

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad (1.22)$$

$$CX_{q_0, q_1} = I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1| \quad (1.23)$$

şeklindedir. Burada I birim operatördür, $|0\rangle\langle 0|$ ve $|1\rangle\langle 1|$ operatörü kübitin durumunu kontrol eder. Eğer kontrol kübiti $|0\rangle$ ise I operatörünü uygular, eğer $|1\rangle$ ise X operatörünü uygular. CNOT kapısının diyagram gösterimi ise Şekil 4'te verilmiştir.



Şekil 1.4 : CNOT kapısının diyagram gösterimi.

1.2.4.5 Z kapısı

Z kapısı bir kübitin $|1\rangle$ durumuna -1 faz faktörü ekleyen kapıdır. $|0\rangle$ durumundaki bir kübite uygulandığında durumunda veya fazında herhangi bir değişiklik yapmaz. Pauli-Z matrisi ile

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (1.24)$$

temsil edilir. Z kapısı aynı zamanda Bloch küresi üzerinde z eksenini etrafında π kadar döndürmek anlamına gelir. Ayrıca RZ operatörünün π parametresi almış hali $-iZ$ 'ye eşittir.

$$RZ(\pi) = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = -iZ \quad (1.25)$$

1.2.4.6 RZZ kapısı

RZZ kapısı iki kübite etkileşimli şekilde Z kapısının uygulanması demektir. Bu kapı iki kubit arasındaki ZZ etkileşimini belirli bir açıyla döndürerek uygular. RZZ kapısı

$$R_{ZZ}(\theta) = e^{-i\frac{\theta}{2}Z \otimes Z} \quad (1.26)$$

şeklinde ifade edilir. Burada θ döndürme açısını temsil eder. Matris gösterimi

$$R_{ZZ} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\theta}{2}} \end{pmatrix} \quad (1.27)$$

şeklindedir.

1.2.5 Kuantum algoritmaları

Kuantum algoritmaları, klasik bilgisayarların uzun sürede çözmediği veya çözemediği problemleri çözmeye yarayan kuantum devreleri kurmanın yöntemidir. Kuantum bilgisayarlarında kubitleri kapılardan geçirerek manipüle edebiliyoruz. Ancak bunun belirli bir sonuç vermesi gerekiyor. Zira kuantum hesaplamada sürdürülen işlemlerin sonucunu gözlemleyemiyoruz. Yani bir devre kurduğumuzda kapılardan geçen kubitin sonuç denklemini göremiyoruz. Bunun için ölçüm yapmamız gerekiyor. Ancak kubit ölçüm yapıldığında da mümkün olabilecek kuantum durumlardan yalnızca bir tanesine çöküyor. Bu nedenle kuantum algoritmaları ile devreler kurup ölçüm sonucunda doğru sonuca çökebilme ihtimali artırılıyor. Birden fazla kez aynı işlemi yaptırarak sonuçlar toplanıyor ve bu sonuçlar içerisinde en çok gelen sonuç cevabımız olarak kabul ediliyor. Tarihte ilk kuantum algoritması David Deutsch'un 1985 yılında geliştirdiği algoritmadır [94]. Bu algoritma direk bir problemi çözmez ancak problemlerin nasıl çözülebileceğine dair bir öncü kaynak olarak kabul edilebilir.

1.2.5.1 Gürültülü orta ölçekli kuantum (NISQ) makinesi

Gürültülü orta ölçekli kuantum (NISQ) makinesi, kullanımda olan bilgisayarların gelecekte yerine geçmesi beklenen bilgisayarları temsil eder. Burada bahsedilen gürültü gelecekte olması planlanan kusursuz kuantum bilgisayarları gibi değildir. NISQ makineleri hatalar karşısında güvenilir değildirler. Kübitler öngörülemeyen bir şekilde durum değiştirebilir ve bu da kesin olmayan hesaplamalara neden olur. NISQ makineleri kuantum dolanıklık (iki veya daha fazla kuantum parçacığının, birbiriyle bağlantılı olacak şekilde bir durum paylaşması) ve süperpozisyon ilkelerine dayanır. Bazı makalelerde şu anda NISQ çağına girildiğinden bahsedilmektedir [95, 96]. Klasik ve kuantum bilgisayarlarının birlikte hesaplama yaptığı kuantum optimizasyon algoritmaları, NISQ çağında önemli bir kuantum avantajı için umut vaat etmektedir [62, 97–99]. Bunun en önemli örneği QAOA'dır [62]. QAOA'lar kimyasal simülasyonlar [100, 101], makine öğrenmesi [102] ve çeşitli mühendislik uygulamalarında [103, 104] kullanılmıştır.

1.2.5.2 Kuantum algoritma yazılım ve kütüphaneleri (qiskit, cirq)

Kuantum bilgisayarlarını kullanabilmek için devreleri kurabilmemiz ve programlayabilmemiz gerekir. Bunun için de en etkili yol klasik bilgisayar sistemlerini kullanarak yapmaktır. Bu nedenle birçok firma ve geliştirici kuantum bilgisayarlarında kullanılmak veya klasik bilgisayarlarda kuantum algoritmalarını test edebilmek için simülasyon oluşturabilecek programlama dilleri veya programlama dilleri için kütüphaneler geliştirdiler. Bunlardan bazıları sıralanmıştır; Quipper [105], Scaffold/Scaffold [106], LIQUi) [107], QWire [108], qiskit [109], cirq [110].

1.2.6 Önemli algoritmalar

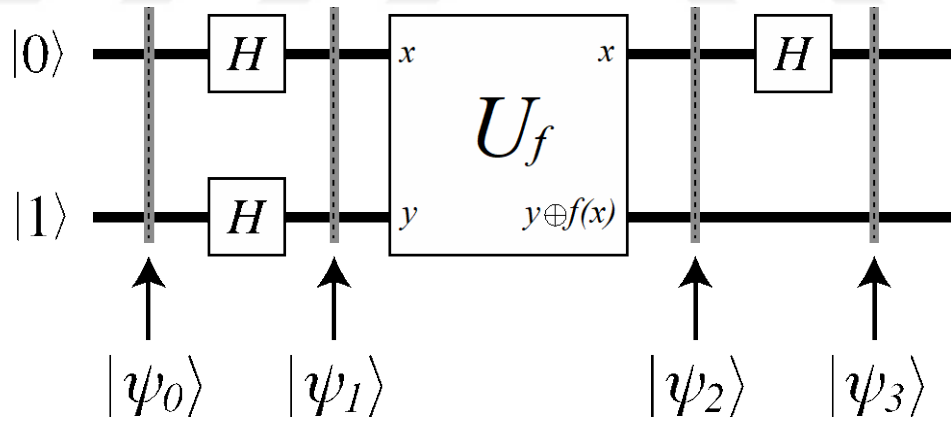
Kuantum bilgisayarlarını çalıştırıp mantıklı sonuçlar alabilmek için geliştirilen algoritmaların en temeli 1.2.6.1'de bahsedildiği gibi Deutsch'un algoritmasıdır. Bu algoritmadan sonra kübitleri anlamlandırabilmenin mantığı kavranmış ve asal çarpanlara ayırma, arama ve optimizasyon problemleri gibi problemler üzerine yeni algoritmalar üretilmiştir. Çizelge 1.1'de kuantum kapılarının kübitlere uygulanma listesi bulunmaktadır.

Çizelge 1.1 : Kuantum kapılarının kübitlere uygulanma listesi.

Kuantum Kapı	Matris Gösterimi	$ 0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$ 1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
I	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

1.2.6.1 Deutsch algoritması

Deutsch algoritması, kuantum bilgisayarların klasik bilgisayarlara kıyasla nasıl daha hızlı çalışabileceğini gösteren ilk kuantum algoritmasıdır [111]. Algoritmanın amacı, bir fonksiyonun sabit mi yoksa dengeli mi olduğunu belirlemektir. Burada "sabit" bir fonksiyon, tüm girişler için aynı çıktıyı üretir, "dengeli" bir fonksiyon ise girişlerin yarısı için 0, diğer yarısı için 1 çıktısı üretir. Şekil 1.5'te algoritmayı uygulayan kuantum devresi aşamalarıyla birlikte gösterilmektedir.



Şekil 1.5 : Deutsch'un algoritmasını uygulayan kuantum devresi.

Şekil 1.5'de gösterilen U_f operatörü içerisinde \oplus simgesi ile gösterilen işlem XOR işlemini gösterir. XOR işlemi, iki bitlik girişin farklı olduğu durumlarda sonucu 1, aynı olduğu durumlarda sonucu 0 olarak verir. Çizelge 1.2'de XOR işleminin doğruluk tablosu gösterilmektedir. Bu doğruluk tablosu XOR işleminin giriş ve çıkış değerlerini göstermektedir.

Çizelge 1.2 : XOR kapısının doğruluk tablosu.

y	$f(x)$	$y \oplus f(x)$
0	0	0
0	1	1
1	0	1
1	1	0

Deutsch algoritmasının aşamalarına baktığımız zaman önce başlangıç durumu $|0\rangle$ ve $|1\rangle$ olan 2 kübit seçilir. Başlangıç durumu

$$|\psi_0\rangle = |01\rangle, \quad (1.28)$$

şeklinde olur. Daha sonra bu iki kübit hadamard kapısından geçirilerek

$$|\psi_1\rangle = \left[\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \right] \left[\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \right], \quad (1.29)$$

şekilde süperpozisyon durumuna getirilir. Burada U_f operatörünü ψ_1 durumuna uyguladığımızda iki ihtimal ortaya çıkmaktadır. Bunlar

$$|\psi_2\rangle = \begin{cases} \pm \left[\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \right] \left[\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \right], & f(0) = f(1) \\ \pm \left[\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \right] \left[\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \right], & f(0) \neq f(1), \end{cases} \quad (1.30)$$

şeklinindedir. Son olarak ilk kübiti tekrar Hadamard kapısından geçirdiğimizde

$$|\psi_3\rangle = \begin{cases} \pm |0\rangle \left[\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \right], & f(0) = f(1) \\ \pm |1\rangle \left[\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \right], & f(0) \neq f(1), \end{cases} \quad (1.31)$$

durumu elde edilir. Burada $f(0) \oplus f(1)$ işleminin sonucunun eğer $f(0) = f(1)$ ise 0, değil ise 1 olduğunu düşünürsek $|\psi_3\rangle$ durumunu tekrar düzenlediğimizde

$$|\psi_3\rangle = \pm |f(0) \oplus f(1)\rangle \left[\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \right], \quad (1.32)$$

şeklini alır. Denklem 1.32'de görüldüğü gibi ilk kübit için ölçüm yapıldığında sonuç 0 ise f fonksiyonu sabittir, eğer 1 ise dengelidir.

1.2.6.2 Deutsch–Jozsa algoritması

Deutsch algoritmasının genelleştirilmiş hali olan Deutsch–Jozsa algoritması 1992 yılında Deutsch ve Jozsa tarafından geliştirilmiştir [112] ve 1998 yılında Cleve, Ekert, Macchiavello, ve Mosca tarafından iyileştirilmiştir [113]. Bu algoritmanın fiziksel olarak gerçekleştirilmesi Benjamin Perez-Garcia ve diğ. tarafından yapılmıştır [114]. Deutsch-Jozsa algoritması, n bitlik bir fonksiyonun sabit mi (her giriş değeri için aynı sonucu verir) yoksa dengeli mi (giriş değerlerinin yarısı için 0, diğer yarısı için 1 sonucunu verir) olduğunu belirlemek için kullanılır. Bu algoritma, klasik bir bilgisayarın ortalama olarak $2^{n-1} + 1$ kez fonksiyon çağırması gereken problemi sadece bir kez çağırarak çözer.

Deutsch-Jozsa algoritması temel adımları aşağıda verilmiştir. İlk adımda başlangıç durumu belirlenir. Başlangıç durumunda n tane kübit (kontrol kübiti) $|0\rangle$ ve bir tane hedef kübiti $|1\rangle$ durumuna getirilir. Durum vektörü

$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle \quad (1.33)$$

$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle, \quad (1.34)$$

şeklinde olur. Başlangıç durumdaki bütün kübitler hadamard kapısından geçirilerek süperpozisyon durumuna getirilir. Hadamard kapısından geçirildikten sonra durum vektörü

$$|\psi_1\rangle = \sum_{x \in \{0,1\}^n} \frac{|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right], \quad (1.35)$$

şeklini alır. Burada $|0\rangle$ durumundaki kübit hadamard kapısından geçtikten sonra $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$ şeklini alır. Bir sonraki aşamada $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$ şekilde bir fonksiyon kullanılarak istenilen doğru cevabın faz değerinin döndürülmesi sağlanacaktır.

$$|\psi_2\rangle = \sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)} |x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \quad (1.36)$$

Denklem 1.23’de verilen $|\psi_2\rangle$ durumunda $f(x)$ sabit ($f(0) = f(1)$) olduğunda her $|x\rangle$ durumu için aynı olacaktır. Bu durumda

$$(-1)^{f(x)} |x\rangle = \begin{cases} +|x\rangle, & f(x) = 0 \\ -|x\rangle, & f(x) = 1 \end{cases} \quad (1.37)$$

elde edilecektir. Ancak $|\psi_2\rangle$ durumunda $f(x)$ dengeli ($f(0) \neq f(1)$) olduğunda her $|x\rangle$ durumu için farklı olacaktır. Bu durumda

$$\sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle = \begin{cases} +|0\rangle - |1\rangle, & f(0) = 0, f(1) = 1 \\ -|0\rangle + |1\rangle, & f(0) = 1, f(1) = 0 \end{cases} \quad (1.38)$$

elde edilecektir. Denklem 1.14'te gösterildiği gibi Hadamard kapısının bir kübite uygulandığı matris gösterimi, lineer formda yazılır ise $H|x\rangle = \sum_z (-1)^{xz} |z\rangle / \sqrt{2}$ elde edilir. Bu formülü genelleştirildiği zaman

$$H^{\otimes n} |x_1, \dots, x_n\rangle = \frac{\sum_{z_1, \dots, z_n} (-1)^{x_1 z_1 + \dots + x_n z_n} |z_1, \dots, z_n\rangle}{\sqrt{2^n}} \quad (1.39)$$

elde edilir. $(-1)^{xz}$, x ve z arasındaki çarpımın sonucuna göre $+1$ veya -1 değerini alır. $|\psi_2\rangle$ durumunda Hadamard kapısı tüm kontrol kubitlerine uygulanır ise

$$|\psi_3\rangle = \sum_z \sum_x \frac{(-1)^{xz+f(x)} |z\rangle}{2^n} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \quad (1.40)$$

son durum şeklinde tanımlanır. Burada xz çarpımı bit çarpımıdır. Bu durumda $|z\rangle$ durumu (kontrol kubitlerinin tamamı) ölçüldüğünde sonucun hepsi $|0\rangle$ durumunda ise $f(x)$ fonksiyonu sabit fonksiyondur. Eğer hepsi $|0\rangle$ değilse $f(x)$ fonksiyonu dengelidir. Böylece Deutsch-Jozsa algoritması klasik bilgisayarların $2^{n-1} + 1$ döngüde çözebileceği problemi bir döngüde çözmüş olur.

1.2.6.3 Shor algoritması

Shor algoritması, kuantum bilgisayarlar üzerinde çalışan bir algoritmadır ve klasik bilgisayarlara kıyasla çok daha verimli bir şekilde büyük sayıların asal çarpanlarına ayrılmasını sağlar. Bu algoritma, Peter Shor tarafından 1994 yılında geliştirilmiştir ve kuantum bilgisayarların potansiyel gücünü göstermek için önemli bir örnektir [115]. Algoritma, RSA gibi modern şifreleme sistemlerinin güvenliğine ciddi bir tehdit oluşturmaktadır çünkü bu sistemler büyük sayıların asal çarpanlarına ayrılmasının zorluğuna dayanır. Shor algoritması N tamsayısını çarpanlara ayırmak için polinomsal zamanda $O((\log N)^2 (\log \log N) (\log \log \log N))$ hızı ile çalışır. Bu hız, bilinen en hızlı asal çarpanlara ayırma yöntemi olan GNFS (Genel sayı alanı eleği) algoritmasının $O\left(\exp\left(\left(\frac{64}{9}b\right)^{\frac{1}{3}} (\log b)^{\frac{2}{3}}\right)\right)$ hızına kıyasla çok daha etkilidir [116, 117].

Shor algoritması, temelde kuantum paralelliği kullanarak bir modüler fonksiyonun periyodunu verimli bir şekilde bulur. Bu periyot ise sayıların asal çarpanlarının bulunmasında kullanılır.

İlk adımda $1 < a < N$ olacak şekilde rastgele bir a tamsayısı seçilir. Öklid algoritması ile a ve N arasındaki en büyük ortak bölen bulunur. Eğer $EBOB(N, a) \neq 1$ ise bu değer N 'in bir asal çarpanıdır ve algoritma sonlanır. Eğer $EBOB(N, a) = 1$ ise $\text{mod } N$ 'de a 'nın 0 olmayan r kuvveti $a^r \equiv 1(\text{mod } N)$ sağlayacak şekilde bulunur. Burada periyodu bulmak için, $\forall a \in \mathbb{Z}, a^0 = 1$ olacağından bir sonraki 1 değerini veren r değerini bulmak gerekir. Eğer r tek sayı veya $a^{\frac{r}{2}} \equiv -1(\text{mod } N)$ ise yeni bir a değeri seçilerek işlemler tekrarlanır. Bulunan r değeri

$$a^r \equiv 1 \text{ mod } N \Rightarrow a^r - 1 \equiv 0 \text{ mod } N \Rightarrow (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) \equiv 0 \text{ mod } N \quad (1.41)$$

formülü ile $EBOB(a^{\frac{r}{2}} - 1, N)$ ve $EBOB(a^{\frac{r}{2}} + 1, N)$ değerlerinin 1'e eşit olmayan sonuçlarında N tamsayısının asal çarpanı bulunmuş olur. Eğer her iki değer de 1'e eşit ise yeni bir a değeri ile işlemler tekrarlanır.

Bu işlemleri bir örnek ile anlatacak olursak, $N = 15$ değerini ve $a < N$ olacak şekilde $a = 2$ değerini seçersek, $EBOB(2, 15) = 1$ olur. Bu nedenle periyodu bulmak için $a^r \equiv 1(\text{mod } 15)$ olacak şekilde r değeri bulunur.

Çizelge 1.3 : Farklı r değerleri için $2^r \text{ mod } 15$.

r	$a^r \equiv (\text{mod } 15)$	
1	$2^1(\text{mod } 15)$	2
2	$2^2(\text{mod } 15)$	4
3	$2^3(\text{mod } 15)$	8
4	$2^4(\text{mod } 15)$	1

Çizelge 1.3'de görüldüğü gibi $r = 4$ 'te tekrarlanıyor. Bulunan $r = 4$ değeri çift sayıdır ve $2^2 \not\equiv -1(\text{mod } N)$ olduğundan tekrarlamaya gerek yoktur. Denklem 1.26'da yerine koyarsak

$$EBOB\left(2^{\frac{4}{2}} - 1, 15\right) = 3, EBOB\left(2^{\frac{4}{2}} + 1, 15\right) = 5 \quad (1.42)$$

sonuçlarını elde ederiz. Burada 3 ve 5 tamsayıları 15 tamsayısının asal çarpanlarıdır. Bu sonuçtan anlaşıldığı gibi bir modüler fonksiyonun periyodunu bulmak çarpanlara ayırmanın önemli bir adımıdır. Kuantum bilgisayarlarını kullanarak bu periyodu hesaplayabiliriz.

Shor algoritmasının kuantum kısmı için öncelikle kubitler hadamard kapısından geçirilerek süperpozisyon durumuna getirilir. Bu durumda kuantum durumu

$$|\psi_0\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle \quad (1.43)$$

şeklini alır. Burada Q , N^2 'den büyük en küçük 2^n değeridir. n sayısı ise asal çarpanları bulunması istenen tamsayının kubitlere işlenmesi için gereken kubit sayısıdır. Örneğin, $N = 15$ için $15^2 = 225$ olacağından bu değerden büyük en küçük 2^n değeri $n = 8$, $2^8 = 256$ 'dır. Bu da 8 kubit kullanılacağı anlamına gelir. Daha sonra periyodun bulunması için gereken $f(x) = a^x \bmod N$ fonksiyonu n adet ikinci kubit destesine işlenir ve kuantum durumu

$$|\psi_1\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle |a^x \bmod N\rangle \quad (1.44)$$

şeklini alır. Bu durumda $a^x \bmod N$ değerinin elde etmek için Şekil 1.6'da da görüldüğü gibi kontrollü-U operatörü uygulanır. İlk kubit grubunun durumuna $|x\rangle$ ikinci kubit grubunun durumunu $|\psi\rangle$ olarak belirlersek

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (U^x |\psi\rangle) |x\rangle = |\psi\rangle \otimes \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i x \theta} |x\rangle \quad (1.45)$$

şeklinde kontrollü-U operatörü her $|x\rangle$ durumu için uygulanmış olur. Burada $e^{2\pi i x \theta}$ kısmındaki $\theta = \frac{y}{2^n}$, $y \in \{0, \dots, 2^n - 1\}$ şeklindedir. Bu nedenle durum vektörü

$$|\psi\rangle \otimes \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i \frac{xy}{2^n}} |x\rangle \quad (1.46)$$

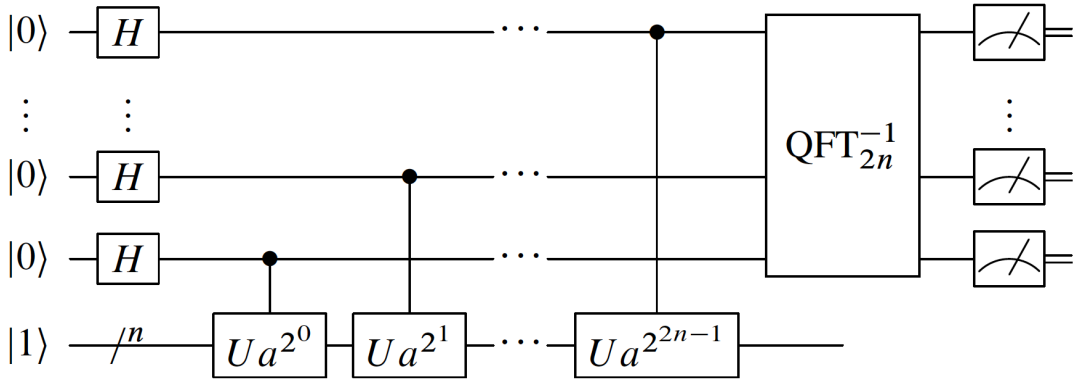
şeklini alır. Bu aşamadan sonra elde edilen Fourier uzayındaki kuantum durumu, QFT^{-1} (Ters Kuantum Fourier Dönüşümü) ile orijinal baz uzayına geri dönüştürülür. QFT^{-1} uygulandığında

$$|\psi_2\rangle = |\psi\rangle |y\rangle \quad (1.47)$$

elde edilir. İlk kubit grubu $|y\rangle$ ölçüldüğünde elde edilen sonuç elde edilen sonuçtan periyot hesaplanır. Bu değer sürekli kesirler algoritması ile $\frac{y}{2^n} \approx \frac{k}{r}$ yaklaşımı yapılarak hesaplanır. Burada y ölçüm sonucu ve r periyottur. Periyot bulunduktan sonra klasik yöntem ile asal çarpanlar bulunur. Bunun için önce $a^{r/2} \bmod N \neq -1$ eşitsizliği doğrulanır. Eğer sonuç -1 'e eşit değilse

$$EBOB\left(a^{\frac{r}{2}} - 1, N\right) = p, EBOB\left(a^{\frac{r}{2}} + 1, N\right) = q \quad (1.48)$$

işlemleri yapılarak p ve q asal sayıları bulunur. Bu sayılar ise $N = p \times q$ şeklinde N tamsayısının asal çarpanlarıdır. Eğer $p \times q$ doğru sonucu vermiyorsa sadece p veya sadece q sayısının N tamsayısının çarpanı olup olmadığı kontrol edilir. Şekil 1.6'da Shor algoritmasının diyagram gösterimi verilmektedir.



Şekil 1.6 : Shor algoritmasının devre diagramı.

2. MATERYAL VE YÖNTEM

Uygulanacak metot Şekil 2.1’de belirtildiği gibi 3 temel aşamadan oluşmaktadır. QAOA uygulaması, parametrelerinin optimize edilmesi ve optimizasyonun doğrulanması kısımlarından oluşmaktadır. Bu çalışma, qiskit kütüphanesi kullanılarak, QAOA’nun TSP problemi için IBM kuantum bilgisayarlarında ve Aer simülasyonunun kullanılmasından oluşmaktadır.

2.1 Kuantum Yaklaşık Optimizasyon Algoritması (QAOA)

QAOA, karmaşık optimizasyon problemlerini çözmek için kuantum bilgisayarların gücünden yararlanan bir algoritmadır. NP-zor problemler için oldukça faydalı olan bu algoritma, kuantum bitleri ve kuantum kapıları kullanarak olası çözümleri değerlendirmeye dayanmaktadır.

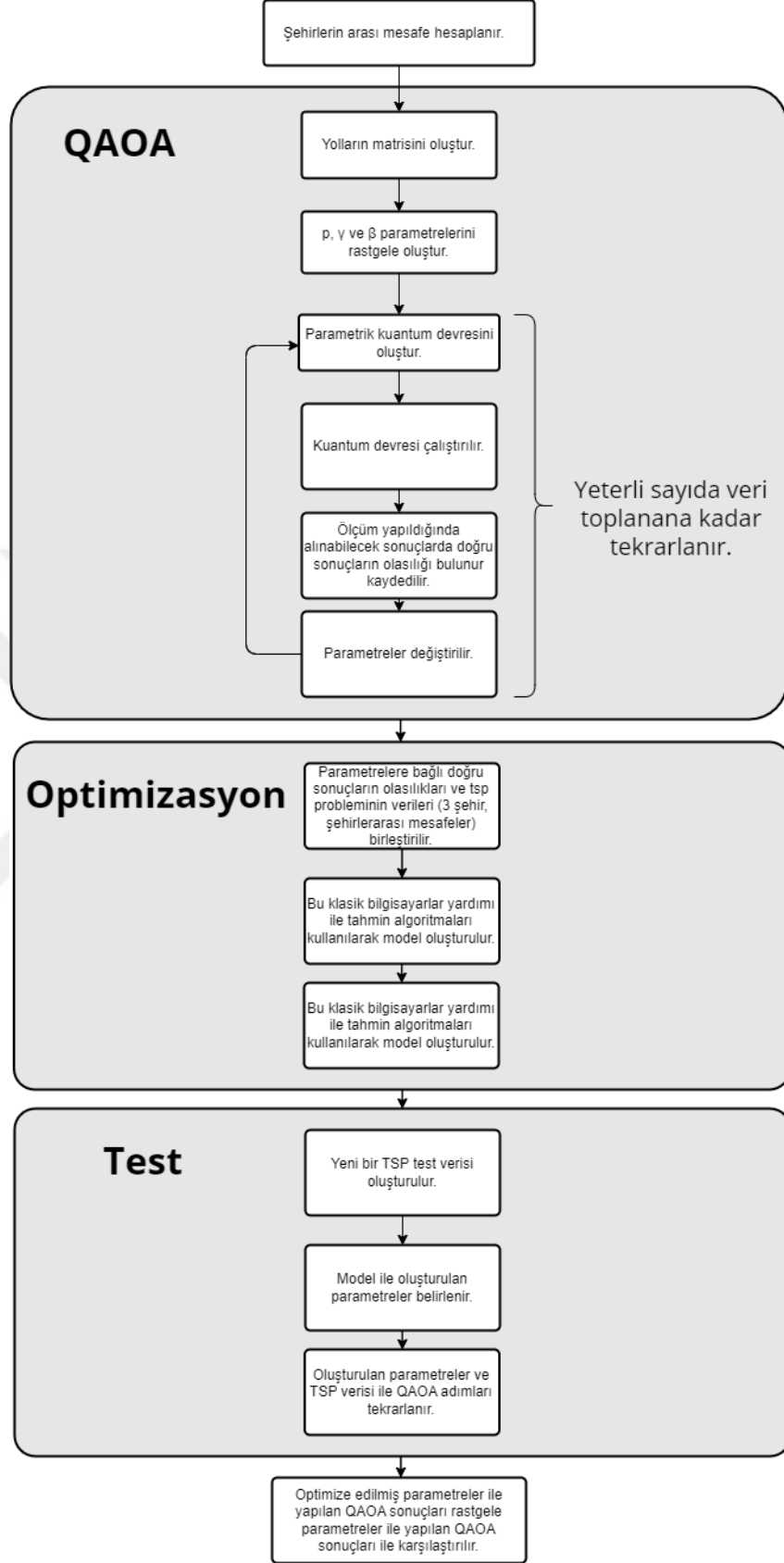
Gezgin satıcı problemini kuantum bilgisayar ile çözmek için gereken kuantum yaklaşık optimizasyon algoritması, problem tanımı ve parametrelili kuantum devresi olmak üzere iki temel aşamadan oluşmaktadır. Kuantum Yaklaşık Optimizasyon Algoritmasını (QAOA) kullanarak Seyahat Satıcısı Problemine (TSP) çözüm bulmak ve farklı parametre setlerini inceleyerek ve optimize ederek TSP probleminin çözümünün ne kadar iyileştirilebileceğini araştırmak amaçlı bu tez çalışması kapsamında, problem tanımı için önce şehirler ve şehirler arası mesafeler belirlenmelidir. Daha sonra belirlenen bu değerler kubitler ile tanımlanarak hesaplamalar gerçekleştirilir.

Bu aşamalarda, objektif fonksiyon tanımlanır ve parametreler optimize edilerek en iyi çözüm bulunmaya çalışılır. Algoritma, belirlenen tekrarlamaya sayısına ulaşana veya belirli bir kriter karşılanana kadar işlemleri tekrarlar. Bu tekrarlamaya işlemi ne kadar çok olursa işlem sonucu o kadar doğru sonuç vermektedir. Tekrarlamaya işlemi ifade eden parametre p simgesi ile ve QAOA operatörlerinin uygulama sırasını kontrol eden parametre γ ile gösterilir. Bu parametrelerin en verimli olanını tespit etmek için tekrarlı testler yaparak optimum sonucu bulmak gerekmektedir. Ancak bu her problem için fazladan uğraş anlamına gelmektedir. Daha önce Montanez-Barrera ve diğ. 2024

yılında gerçekleştirdikleri çalışmada aktarılmış öğrenme (TL) metodu ile önceden eğitilmiş parametreleri kullanarak problemleri çözmeye çalışmışlardır [118]. Bu tez çalışmasında da benzer bir metot uygulanacaktır. Ancak parametrelerin optimizasyonunun başlangıç problemine bağlı olması sağlanarak daha etkili parametreler elde edilmeye çalışılmıştır.

QAOA'nın TSP problemi için klasik algoritmalarından çok daha hızlı olması da bir avantaj sağlamaktadır. QAOA'nın problemi çözmek için gerekli adım sayısı şehir sayısına ve derinliğine bağlı olarak polinomal ($O(poly(n, p))$) şekilde artarken şu anda klasik algoritmaların en hızlısı olan Bellman-Held-Karp Algoritması ancak $O(n^2 \times 2^n)$ adımda çözebilmektedir [119].





Şekil 2.1 : Çalışmanın şematik gösterimi.

QAOA için önce problem belirlendi ve kübitlere tanımlandı. Bu problemin kübitlere tanımlanması için, herhangi bir optimizasyon problemini ikili değişkenler kullanarak temsil etmenin bir yolu olan Quadratic Unconstrained Binary Optimization (QUBO) kullanılmıştır. Bunun için önce problem matrisi oluşturulup ikili sisteme uygun hale

$$C(x) = \sum_{i=1}^n \sum_{j=1}^n w_{i,j} \left(\sum_{p=1}^n x_{i,p} x_{j,(p+1)} \right) \quad (2.1)$$

şeklinde dönüştürüldü. Burada $x_{i,j}$ i 'nci şehrin j pozisyonunda ziyaret edilip edilmediğini belirten ikili değişken ve $w_{i,j}$ Denklem 2.6'da örnek olarak verildiği gibi problem matrisidir. Daha sonra ikili sistemde oluşturulan QUBO maliyet fonksiyonu

$$Z_i |x_i\rangle = (-1)^{x_i} |x_i\rangle = (1 - 2x_i) |x_i\rangle \Rightarrow \frac{1 - Z_i}{2} |x_i\rangle = x_i \quad (2.2)$$

dönüşümünün yardımı ile maliyet operatörü (H_C) elde edilir. Denklem 2.2'de elde edilen sonuç Denklem 2.1'de x_i yerine yazılırsa

$$H_C = \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n w_{i,j} \left(\sum_{p=1}^n (1 - Z_{i,p} - Z_{j,(p+1)} + Z_{i,p} Z_{j,(p+1)}) \right) \quad (2.3)$$

elde edilir. Maliyet hamiltoniyeninden sonra karıştırıcı hamiltoniyen (H_M) oluşturuldu. Karıştırıcı hamiltoniyen operatörünün amacı X ekseninde de döndürerek tüm enerji durumlarını gezmemizi sağlamaktır. Bu nedenle

$$H_M = \sum_{i,p} X_{i,p} \quad (2.4)$$

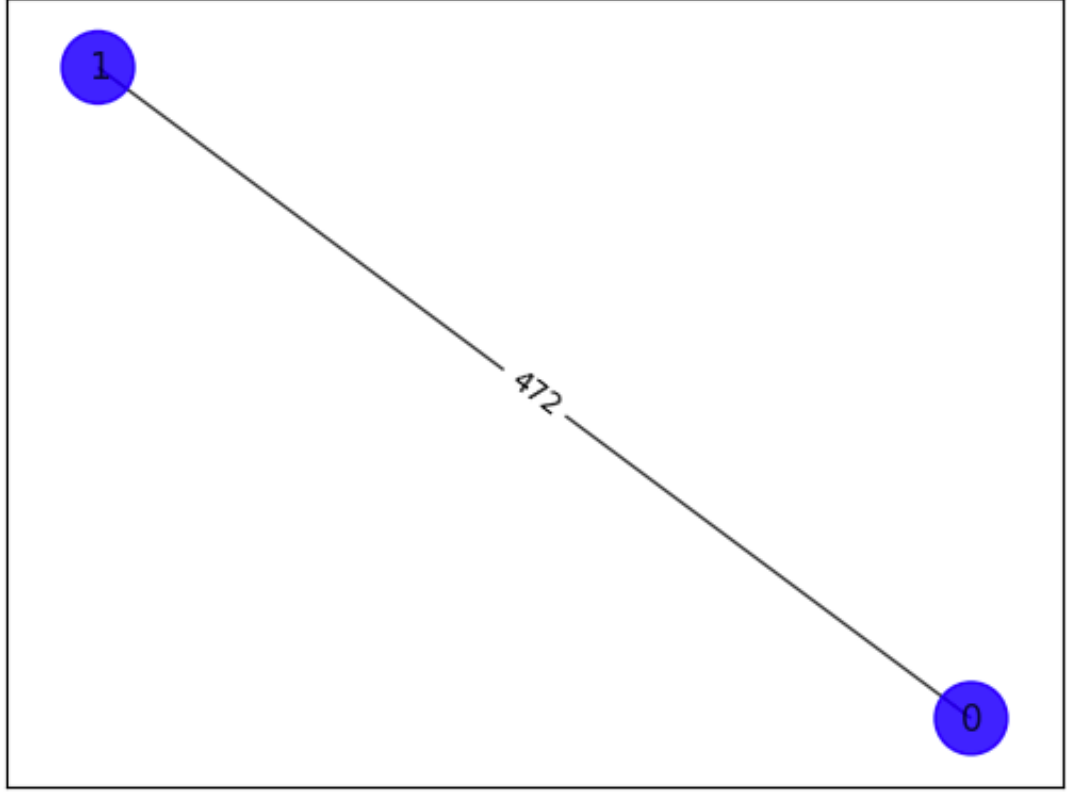
şeklinde oluşturulur. QAOA'nın bir sonraki adımı için H_C ve H_M operatörleri

$$U_C = e^{-i\gamma H_C}, \quad (2.5)$$

$$U_M = e^{-i\beta H_M} \quad (2.6)$$

şeklinde parametrik üniter operatörlere dönüştürülmüştür.

Problem tanımı için gezgin satıcı probleminde önce şehir sayısı ve şehirler arası mesafeler belirlenecektir. Burada 2 şehir ele alınmıştır. Mesafeler Çizelge 2.1'de ve harita Şekil 2.2'da belirtilmiştir.



Şekil 2.2 : Şehirler ve aralarındaki mesafenin örnek çizimi.

Çizelge 2.1 : 2 Şehir için şehirlerarası mesafeler.

Şehirler	1	2
1	0	472
2	472	0

n adet şehir için oluşturulacak matris

$$\begin{bmatrix} \phi_{11} & \cdots & \phi_{1n} \\ \vdots & \ddots & \vdots \\ \phi_{n1} & \cdots & \phi_{nn} \end{bmatrix} \quad (2,7)$$

şeklinde oluşturulur. Bu nedenle problemi çözmek için şehirler arası mesafeler matrisi

$$\begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} = \begin{bmatrix} 0 & 472 \\ 472 & 0 \end{bmatrix} \quad (2,8)$$

şeklindedir.

Denklem 2.3'deki matris 2x2 matristir. Her bir yol için bir kübit ayırmamız gerektiği için 4 kübite ihtiyacımız vardır. Kübitler için önce başlangıç durumu

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \quad (2,9)$$

şeklinde oluşturulur. Burada n toplam kübit sayısını gösterir. Bizim örneğimiz için bu değer 4'tür. Denklem 2.9 bütün kübitlerin hadamard kapısından geçirilerek

süperpozisyon durumunda olduğunu ve mümkün olan durumların tamamını kapsadığını gösterir. Bu durumda sistem taban durumundadır. Başlangıç durumundan sonra RZZ, X ve CX gibi kapılar uygulanarak β ve γ parametrelerinin olduğu operatörler uygulanmıştır. Parametrelili kuantum durumu

$$|\beta, \gamma\rangle = e^{-i\beta_P H_M} e^{-i\gamma_P H_P} \dots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P} |S\rangle \quad (2,10)$$

şeklindedir. Burada p algoritmanın derinliğini yani operatörlerin kaç kez tekrarlı uygulanacağını gösterir. Elde edilen kuantum durumunu oluşturmak için devre kurulmuştur. Bu devrede önemli olan durum farklı sonuçlar ve karşılaştırma için β ve γ parametreleri değiştirilerek devre tekrar kurulmuştur.

Bulunan sonucun doğruluğunu test etmek için bütün ihtimaller belirlenmiştir. 4 kubitli bir sistemde 2^4 adet farklı sonuç vardır. Ancak bu sonuçların hepsi bizim problemimizin çözümü değildir. Problemin muhtemel çözümleri

$$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = |0101\rangle \Rightarrow [2, 2] \quad (2,11)$$

gibi çözümler olmalıdır. Burada 2 rakamı, 2. şehri temsil etmektedir. $|0101\rangle$ gibi çözümler problem için bir çözüm olmadığı gibi gerçek bir çözüm de değildir. Çünkü bir şehirden yola çıkıp bütün şehirleri dolaştıktan sonra tekrar aynı şehre gelmenin en kısa yoludur. Bu nedenle $\phi_{ii} = 1$ veya hem $\phi_{12} = 1$ hem de $\phi_{22} = 1$ olamaz. Çünkü yola 1 şehirden çıkan gezginin hem 2 ye hem de 1'e aynı anda gitmesi mümkün değildir. Buna göre kuantum algoritmasının doğru sonucu $|1001\rangle$ veya $|0110\rangle$ olmalıdır.

Bu işlemleri yapabilmek için IBM online kuantum bilgisayarları kullanıldı. Testler, Python ile qiskit kütüphanesi kullanılarak `ibm_osaka` adlı kuantum bilgisayarda gerçekleştirildi.

```
n = 2
cities = {
  0 : (41.01, 28.96),
  1 : (39.92, 32.86)
}

distances = {
  0 : {0: 0, 1: 472},
  1 : {0: 472, 1:0}
}
```

Şekil 2.3 : Şehir ve mesafelerini hazırlayan kod.

Şekil 2.3'te verilen koda 2 adet şehir belirlenip bunların aralarındaki mesafeler kaydedilmiştir.

Şekil 2.4'te verilen kod içerisinde görüldüğü üzere bir graph oluşturularak içerisinde şehir ve mesafe bilgileri konulmuştur. Daha sonrasında TSP içerisinde bu graph eklenmiştir. Mesafeler matrisi böylece elde edilmiştir.

```
G = nx.Graph()

for i in range(n):
  for j in range(n):
    if i!=j:
      G.add_edge(i, j, weight = list(list(distances.values())[i].values())[j])
tsp = Tsp(graph= G)
adj_matrix = nx.to_numpy_array(tsp.graph)

colors = ["b" for node in tsp.graph.nodes]
pos = list(cities.values())
```

Şekil 2.4 : TSP sınıfını oluşturma kodu.

Bu TSP problemini bir kuantum devresine dönüştürmek için birtakım dönüşümler gerekmektedir. Problem önce kuadratik programlamaya sonrasında ise QUBO problemine dönüştürülmektedir. Bu şekilde Z operatörünün lineer kombinasyonu şeklinde devre kurulmuş olur.

```

qp = tsp.to_quadratic_program()
qp2qubo = QuadraticProgramToQubo()
qubo = qp2qubo.convert(qp)
qubitOp, offset = qubo.to_ising()

ansatz = QAOAAnsatz(qubitOp, reps=2)

#IBM kuantum bilgisayarına bağlanıp,
#devre bilgisayara göre dönüştürülüp çalıştırılır.

pm = generate_preset_pass_manager(backend=backend, optimization_level=3)

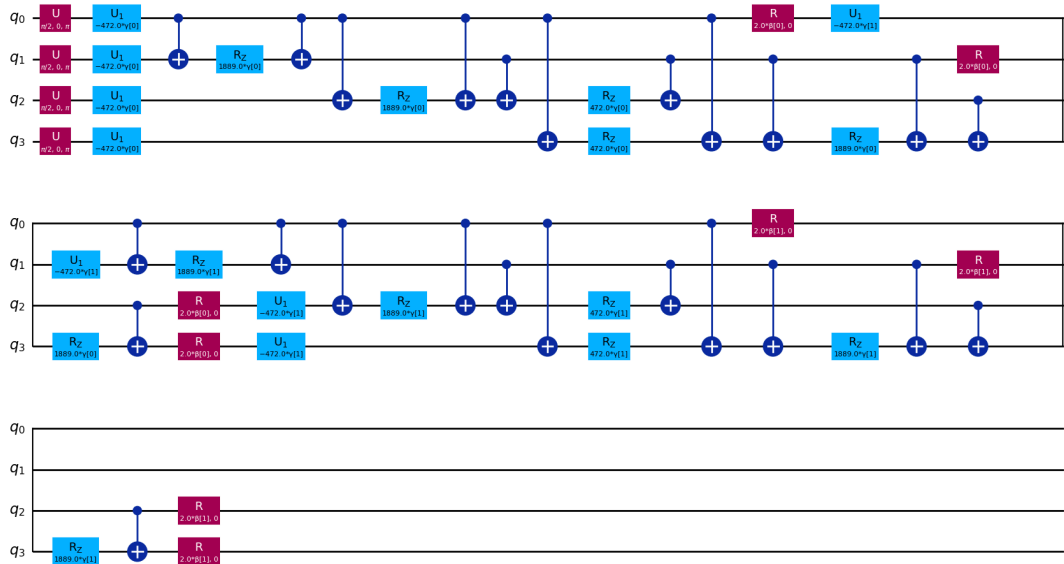
ansatz_isa = pm.run(ansatz)

hamiltonian_isa = qubitOp.apply_layout(ansatz_isa.layout)
...
...
x0 = 2 * np.pi * np.random.rand(ansatz_isa.num_parameters)

```

Şekil 2.5 : Kuantum devresi oluşturup çalıştırılan kod.

Şekil 2.5'te elde edilen ansatz adlı değişken (QAOA Ansatz devresi) Şekil 2.6'da gösterilmiştir. x_0 değişkeni γ ve β değişkenlerini oluşturur. p ise ansatz oluşturulurken reps parametresi ile 2 olarak seçilmiştir. Burada görüldüğü gibi γ_0 'dan başlayarak β_p 'ye kadar sıralı operatörler kullanılmıştır.



Şekil 2.6 : 4 kubitli ansatz devresinin γ_0 ve β_0 için uygulanmış gösterimi.

Kurulan bu devre bir kuantum bilgisayar simülasyonu olan Aer simülasyonunda çalıştırılmıştır. Çalıştırılan devre bize ikilik sistemde tüm sonuçların gelme olasılığını verir. Burada her ihtimalin maliyeti

$$C(x) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n W_{j,k} \cdot x_{i,j} \cdot x_{i+1,k} \quad (2.12)$$

formülü ile hesaplanmıştır. Burada i turdaki pozisyonu, j ve k şehirleri temsil eder. Enerjinin minimum olması TSP'nin çözümü olması demektir. Burada klasik bilgisayarlar yardımı ile parametreler değiştirilerek minimum enerjiyi veren parametreler bulunmuş ve kaydedilmiştir.

2.2 Optimizasyon

Parametreleri optimize etmek için öncelikle simülasyonun gerçek kuantum bilgisayarına olan benzerliğinin belirlenmesi gerekmektedir. Bunun için bir TSP problemi belirlenmiştir. Problem maliyet enerjisini minimum yapan parametreler hesaplanmış ve bu parametreler ile hem simülasyon hem de gerçek kuantum bilgisayarlarında, problem için kurulan devre için ölçüm yapılmıştır. Yapılan ölçümlerle doğru cevapların sonuçları karşılaştırılarak ve simülasyonun hata payı hesaplanmıştır.

Veri seti, simülasyon üzerinde 5000 farklı parametre ile 2 şehirli TSP problemi için hesaplanarak oluşturulmuştur. Bunun için önce her bir problemin kuantum devresi kurulup bu devrenin maliyet enerjisi rasgele parametreler ile ölçülmüştür. Ölçüm sonucunda minimum maliyet enerjisini veren parametreler ile devre kuantum durumu ölçülmüş ve sonuçlar şehir sayısı, şehirlerarası mesafe, doğru sonucun olasılığı, γ_0 , γ_1 , β_0 ve β_1 olarak kaydedilmiştir.

Elde edilen veri seti ile önce her bir değer, bir diğer değere göre grafiği çizilerek bir benzetim yapılmaya çalışılmıştır. Bunun için önce veriler bir tabloda tutulmuş sonrasında bir grafik matris haline getirilerek birbirleriyle olan ilişkileri yorumlanmıştır. Bu adım optimize etmek için ön çalışma adımdır. Bu tür grafikler verilerin birbirleriyle olan ilişkilerini anlamak ve veri setinde olası desenleri tespit etmek için kullanılır.

Belirli mesafe deęerleri için doęru sonucun olasılıęı deęerinin 0.2'den büyük olup olmadığını ve bu durumu saęlayan γ ve β parametrelerini tahmin etmek amacıyla makine öğrenmesi yöntemleri kullanılmıştır. Modeller, verilen mesafe deęeri için gamma ve beta parametrelerinin doęru sonucun olasılıklarını 0,2'den büyük yapacak şekilde tahmin etmeye yönelik olarak eğitilmiştir. 0.2 deęerinin önemi 4 kübitli bir sistemde bu deęerden büyük olasılıęı olan sonucun genel olarak dięer sonuçlar arasında en yüksek oran olmasıdır. Bu nedenle eğitim yapılırken doęru sonuç olarak bu deęerden büyük veriler kabul edilmiştir. Kullanılan veri seti, farklı mesafeler ve bu mesafeler için optimizasyon sonucu ile elde edilen γ ve β parametrelerini içermektedir. Veri seti ayrıca, her bir optimizasyon çalışmasının sonucunda elde edilen doęru sonucun olasılıęı deęerlerini de barındırmaktadır.

İlk olarak, doęru sonucun olasılıęı deęerinin 0,2'den büyük olup olmadığını belirlemek için ikili bir hedef deęişken oluşturulmuştur. Doęru sonucun olasılık deęeri 0.2'den büyük olan durumlar pozitif (1), dięerleri negatif (0) olarak işaretlenmiştir. Veriler daha sonra eğitim ve test setlerine ayrılmış ve Standard Scaler kullanılarak ölçeklendirilmiştir. Sinir Ağları Regresyonu, Destek Vektör Makineleri ve Random Forest Regresyon yöntemleri uygulanmış ve sonuçlar karşılaştırılmıştır.

Sinir Ağları Regresyonu, derin öğrenme yöntemlerinden biridir ve karmaşık ilişkileri öğrenme yeteneęi ile bilinir. Bu çalışmada, dört parametreyi tahmin etmek için ayrı ayrı dört model eğitilmiştir. Sinir Ağları Regresyonu, belirli bir mesafe deęeri için parametreleri tahmin etmede ortalama olarak en iyi performansı göstermiştir. Ancak, bazı durumlarda dięer yöntemlere göre daha yüksek hata payı gözlemlenmiştir. Bu yöntemin avantajı, verilerdeki karmaşık ilişkileri yakalayabilmesidir, ancak modelin eğitimi daha uzun sürmektedir.

Destek Vektör Makineleri (SVM), sınıflandırma ve regresyon analizinde kullanılan güçlü ve esnek bir makine öğrenmesi algoritmasıdır. Gezgin Satıcı Problemi (TSP) için Kuantum Yaklaşık Optimizasyon Algoritması (QAOA) ile elde edilen gama ve beta parametrelerinin optimizasyonu için SVM kullanılmıştır. Veri seti, farklı mesafeler ve bu mesafeler için QAOA optimizasyonu sonucunda elde edilen parametrelerle oluşturulmuş, eğitim ve test setlerine ayrılarak standardize edilmiştir. SVM, veri noktalarını geniş aralıklara ayıran bir hiper düzlemlerle çalışır ve Radial Basis Function (RBF) çekirdeęi ile doğrusal olmayan ilişkileri modelleyebilir. Her bir gama ve beta parametresi için ayrı ayrı eğitilen SVM modellerinin performansı, MSE ve R^2 skoru ile deęerlendirilmiştir.

Random Forest Regresyon, bir topluluk öğrenme yöntemi olup, birden fazla karar ağacını birleştirerek tahminler yapar. Bu çalışmada, dört parametreyi ($\gamma_0, \beta_0, \gamma_1, \beta_1$) tahmin etmek için ayrı ayrı dört Random Forest modeli eğitilmiştir. Random Forest yöntemi, genellikle düşük hata payları ile en tutarlı sonuçları vermiştir. Bu yöntemin avantajı, aşırı öğrenmeye karşı dirençli olması ve genellikle yüksek doğrulukla çalışmasıdır.

Optimizasyon için elde edilen sonuçlar bir tabloda toplanmıştır. Bu tablo şehir sayısı, şehirlerarası mesafe, tekrarlama sayısı (p), γ ve β parametreleri ve doğru sonucun olasılık değerinden oluşmaktadır. Bu tablo makine öğrenmesi yöntemleri için eğitim verilerini oluşturmaktadır. Eğitimin girdi verileri için sadece problem ve doğru sonucun olasılığı verilmiştir. Tablodaki şehirlerarası mesafe değeri problemi `sol_rates` adlı sütun doğru sonucun olasılığını oluşturmaktadır. Eğitimin çıktısı olarak parametrelerin tahmin edilmesi istenmektedir. Verilerin hepsi $p = 2$ derinlikle yani parametre sayısı olarak $2 \times p$ şeklinde 4 olacak şekilde oluşturulduğundan dolayı 4 parametre tahmini yapılmaktadır.

2.3 Test ve Doğrulama

Bu bölümde modelin doğruluğunu test edebilmek için veri seti oluşturulmuştur. Veri seti, modelin performansını bağımsız olarak değerlendirmek için kullanılmıştır. Bu adım, modelin genelleme yeteneğini test etmek için kritik öneme sahiptir.

Bunun için önce yeni mesafeler ile TSP problemleri oluşturulmuştur. Bu problemlerin çözümü için gerekli parametreler her bir problem için devre kurularak simülasyon üzerinden hamiltoniyen enerjisi minimize edilerek hesaplanmıştır. Daha sonrasında bu parametreler, oluşturulan model kullanılarak hesaplanmıştır. Simülasyon ile arasındaki uyuma bakılarak modelin doğruluğu hesaplanmıştır. Modellerin hata oranları hesaplanırken simülasyon üzerinde klasik yöntemler ile hesaplanan minimum enerjiyi veren parametreler ile elde edilen doğru sonucun olasılığı ve modelin test mesafesi için tahmin ettiği parametreler ile elde edilen doğru sonucun olasılığı karşılaştırılmıştır. Bu hesap için

$$Hata Oranı = \frac{DSO_{Klasik Algoritma} - DSO_{Model}}{DSO_{Klasik Algoritma}} \quad (2.13)$$

formülü kullanılır. Burada DSO doğru sonucun olasılığıdır.

Ayrıca modelin üzerinde test edilen 10 farklı TSP problemi kuantum bilgisayarlarında da hesaplanarak aralarındaki hata payı hesaplanmıştır. Kuantum bilgisayarında hesaplanan TSP çözümlerinde modelin verimliliği hesaplanmıştır. Burada sonucun doğru olup olmadığını kontrol etmek için doğru sonuç için olasılığın artmış olması beklenmektedir. Eğer bütün olasılıklar eşit olsaydı

$$2^4 = 16 \Rightarrow \frac{1}{16} = 0,0625 \quad (1.14)$$

şeklinde olurdu. Burada 4 kübitli bir sistem için olasılık hesaplanmıştır. Eğer bir kuantum durumu için 0,0625'ten daha yüksek olasılıklı bir sonuç elde edilmiş ise o durumun genlik değerinin arttığını gösterir. Bu da algoritmanın verdiği sonuç olarak kabul edilmiştir. Model ile tahmin edilen parametreler kullanılarak kuantum bilgisayarında hesaplanan TSP çözümü için doğru sonuç olasılığı hesaplanmıştır ve bu hesaba göre değerlendirilmiştir.

3. BULGULAR VE TARTIŞMA

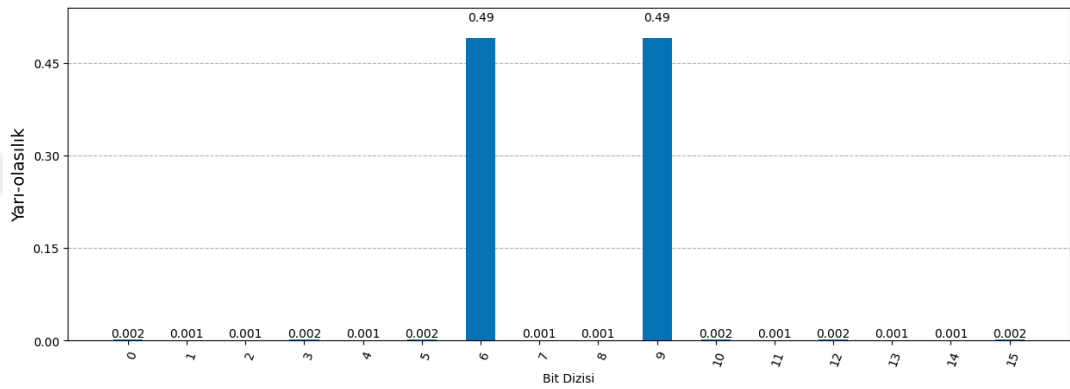
3.1 Kuantum Bilgisayarı Sonuçları

Kuantum bilgisayarında çalıştırılan Şekil 2.6'daki devre önce tahmin edici ve sonrasında örnekleme sistemine gönderilerek çalışması için planlanmıştır. Tahmin edici sistem devrenin verilen parametrelere göre maliyet fonksiyonunun enerjisini hesaplamaktadır. Örnekleme sisteminde ise verilen parametreler ile kurulan devrenin sonuna ölçüm kapıları eklenerek ölçüm yapılmaktadır. Tahmin edici sisteminin gerekli parametreleri bulması için kuantum bilgisayarının çok uzun süre çalışması gerektiğinden, klasik sistemdeki simülasyon ile kalibre edilerek optimizasyon için kuantum devresi, simülasyonda çalıştırılmıştır. Simülasyon için devre 5000 farklı parametre gurubu ile çalıştırılmıştır. Elde edilen sonuçların her biri örnekleme sisteminde de çalıştırılarak doğru sonuçların gelme olasılıkları belirlenmiştir. Çizelge 3.1'de Parametrelere ve mesafeye bağlı $|0110\rangle$ gelme olasılığı için 10 örneğin değerleri verilmiştir.

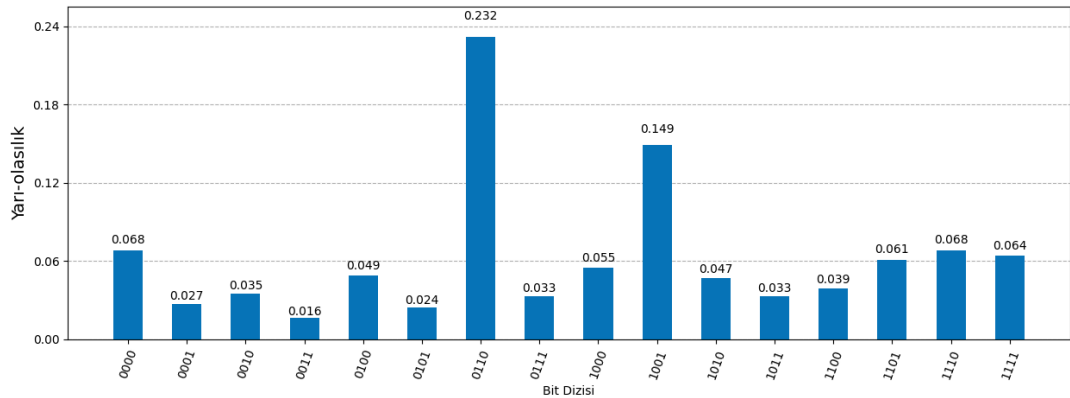
Çizelge 3.1 : Parametrelere ve mesafeye bağlı 10 örnek.

γ_0	β_0	γ_1	β_1	Mesafe	Doğru Sonucun Olasılığı
4,195412	4,214437	4,028968	1,463868	77,196326	0,490063
6,711467	5,477465	1,267793	-0,649812	198,277458	0,489196
5,106420	3,950699	1,688308	1,758576	91,182397	0,478532
3,650413	0,444960	6,065420	5,286768	532,425175	0,476378
2,604382	4,345102	7,121881	1,687281	322,013454	0,475937
2,561147	1,192865	4,250240	5,288210	5,890773	0,475161
2,066530	6,869915	5,718585	2,020136	331,252737	0,472356
3,650372	0,494990	2,377271	4,810387	622,417711	0,471267
3,737453	5,060314	4,403739	4,172626	0,064157	0,469576
7,324643	4,276696	1,370298	0,211077	45,769103	0,469539

Elde edilen verilere göre en yüksek oranda doğru sonuca karşılık gelen parametreler kuantum bilgisayarına giriş parametreleri olarak verilmiştir. Burada yaklaşık 0,49'a denk gelen sonuç oranı 2 şehirli TSP probleminin çözümü olan 6 ve 9 için %50 ihtimale çok yakın bir değerdir. Bu nedenle parametreleri doğru olarak kabul edebiliriz. Şekil 3.1'de simülasyon ile elde edilmiş ölçüm sonuçları Şekil 3.2'de ise aynı parametreler kullanılarak kuantum bilgisayarında ölçülmüş sonuçlar gösterilmektedir. Kuantum bilgisayarında yapılan ölçümlerde tekrar ölçüm (shot) 10000 olarak belirlenmiştir.



Şekil 3.1 : Simülasyon ile elde edilmiş ölçüm sonuçları.



Şekil 3.2 : Kuantum bilgisayar ile elde edilmiş ölçüm sonuçları.

Simülasyon ve gerçek kuantum bilgisayar deneylerindeki farklılıklar, genellikle kuantum dekoheransı, gürültü ve sistem hatalarından kaynaklanmaktadır. İlk grafik, Aer simülasyonu ile elde edilen sonuçları göstermektedir ve belirli durumlar (6 ve 9) için yüksek başarı oranlarına işaret etmektedir. İkinci grafikte ise gerçek kuantum bilgisayar üzerinde aynı parametrelerle yapılan deneylerde bu olasılıkların daha düşük olduğu görülmektedir. Bu durum, simülasyonun ideal koşulları yansıttığını, ancak gerçek kuantum bilgisayarın gürültü ve hata etkilerine maruz kaldığını göstermektedir. Mevcut hata ve gürültü başka kuantum bilgisayarlarında farklı sonuçlar verebilir.

Şekil 3.1 ve Şekil 3.2'deki fark bize kuantum bilgisayarının yüksek hatalar içerebileceğini göstermektedir. Buna rağmen problemin sonucu $|0110\rangle$ ve $|1001\rangle$ için sırasıyla yaklaşık %23 ve %15 olasılıkla doğru bir şekilde belirlenmiştir.

Şekil 3.1 ve Şekil 3.2'de görüldüğü gibi kübit durumu 6 sayısına karşılık gelen $|0110\rangle$ ve 9 sayısına karşılık gelen $|1001\rangle$ olmak üzere toplam 2 durum en yüksek olasılığa sahiptir. Bu durumlar izlenecek rotayı belirtmektedir. Burada şehir sayısına göre toplam kübit sayısı

$$n = 2, n_q = 4, \frac{n_q}{n} = \frac{4}{2} = 2 \quad (4.1)$$

$$n_1 = |01\rangle, n_2 = |10\rangle \quad (4.2)$$

şeklinde bölünür ve sonuca göre kübitler parçalara ayrılır. Burada n_1 1. şehri n_2 2. şehri ifade etmektedir. Kübitlerin 1 durumlarının önemi sıralamayı ifade etmektedir. 1. şehir için $|01\rangle$ ifadesi 2. sırada yer aldığını göstermektedir. Bu şekilde sıralandığı zaman;

$$|01\rangle \Rightarrow 2, |10\rangle \Rightarrow 1, [2,1] \quad (4.3)$$

şeklinde sıralanmış olur. Burada görüldüğü gibi kuantum algoritması doğru çalışmış ve doğru parametreler bulunmuştur.

3.2 QAOA Devresinde γ ve β Parametrelerinin Optimizasyonu

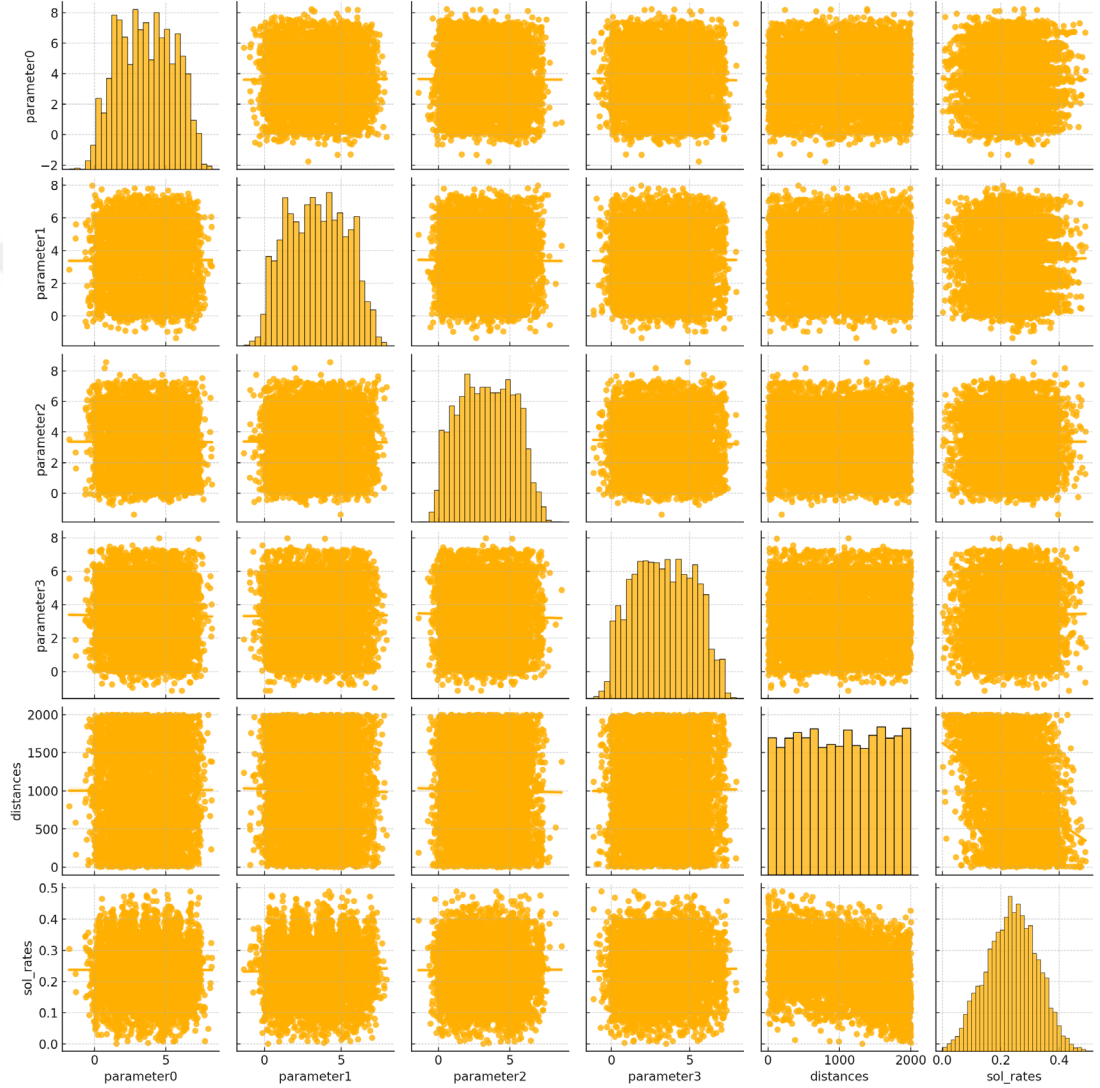
Bu çalışmada, farklı makine öğrenimi ve derin öğrenme yöntemleri kullanılarak verilen mesafe değeri için doğru sonuç olasılığını 0,2'den büyük yapacak parametrelerin tahmin edilmesi hedeflenmiştir. Sinir Ağları Regresyonu, Destek Vektör Makineleri (SVM) Regresyonu ve Random Forest Regresyonu olmak üzere üç farklı yöntem uygulanmış ve sonuçları karşılaştırılmıştır.

Şekil 3.3 veri seti parametrelerinin birbirleriyle aralarındaki ilişkiyi göstermektedir. Her bir parametrenin histogramları, verilerin çoğunlukla 0 ile 6 arasında yoğunlaştığını göstermektedir. Serpilme diyagramında, parametreler arasındaki dağılım genellikle geniş ve belirgin bir desen göstermemektedir. Bu durum parametreler arasında belirgin bir doğrusal ilişki olmadığını göstermektedir.

Mesafeler (distances) histogramı, mesafelerin geniş bir aralıkta (0'dan 2000'e kadar) dağıldığını göstermektedir. Serpilme diyagramı, mesafeler ile diğer parametreler

arasında belirgin bir ilişki olmadığını göstermektedir. Veriler geniş bir aralığa yayılmış ve belirgin bir desen göstermemektedir.

Sol_rates histogramı, verilerin büyük çoğunluğunun 0,0 ile 0,4 arasında yoğunlaştığını göstermektedir. Serpilme diyagramı, sol_rates ile diğer değişkenler arasında belirgin bir ilişki olmadığını göstermektedir. Veriler geniş bir aralığa yayılmış ve belirgin bir desen göstermemektedir.



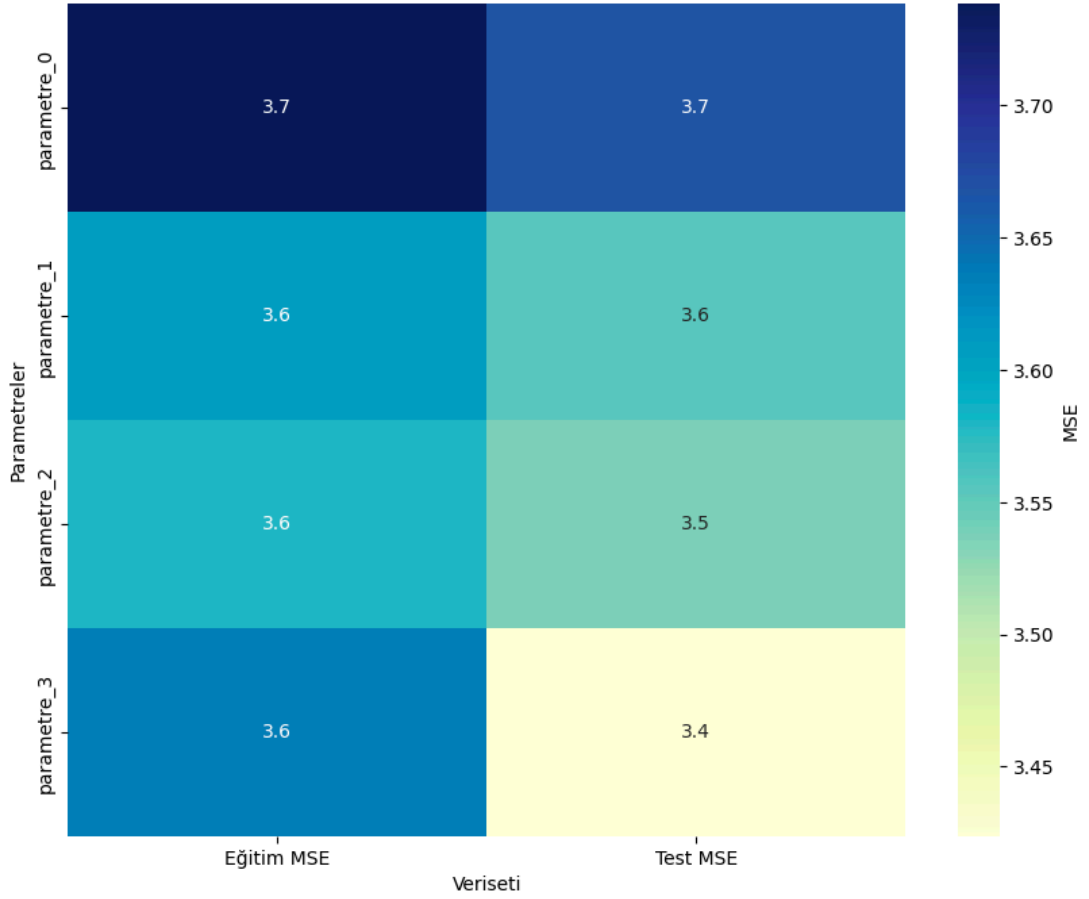
Şekil 3.3 : Veri setinde parametrelerin birbirleriyle ilişki grafiği.

Şekil 3.3 değişkenler arasındaki ilişkileri görselleştirmekte ve genel olarak belirgin bir doğrusal veya güçlü korelasyon olmadığını göstermektedir. Parametreler arasındaki ilişkiler zayıf ve veriler geniş bir aralığa yayılmış durumdadır. Bu, verilerin karmaşık ve doğrusal olmayan ilişkiler içerdiğini düşündürülebilir. Bu nedenle, doğrusal modellerin (örneğin, doğrusal regresyon) bu veri setinde iyi performans göstermesi beklenemez. Daha karmaşık modeller (örneğin, non-linear modeller, topluluk

öğrenmesi yöntemleri veya derin öğrenme teknikleri) verilerin altında yatan desenleri yakalamakta daha başarılı olabilir.

Aşağıdaki grafiklerde belirtilen parametre_0, parametre_1, parametre_2 ve parametre_3 değerleri sırasıyla γ_0 , β_0 , γ_1 ve β_1 değerlerini ifade etmektedirler.

Sinir Ağları Regresyonu, belirli bir mesafe değeri için parametreleri tahmin etmede ortalama olarak en iyi performansı göstermiştir. Ancak, mesafeler için diğer yöntemlere göre daha yüksek hata payı gözlemlenmiştir. Bu yöntemin avantajı, verilerdeki karmaşık ilişkileri yakalayabilmesidir, ancak modelin eğitimi daha uzun sürmektedir. Şekil 3.4'te QAOA parametrelerinin performans sıcaklık haritası verilmiştir.



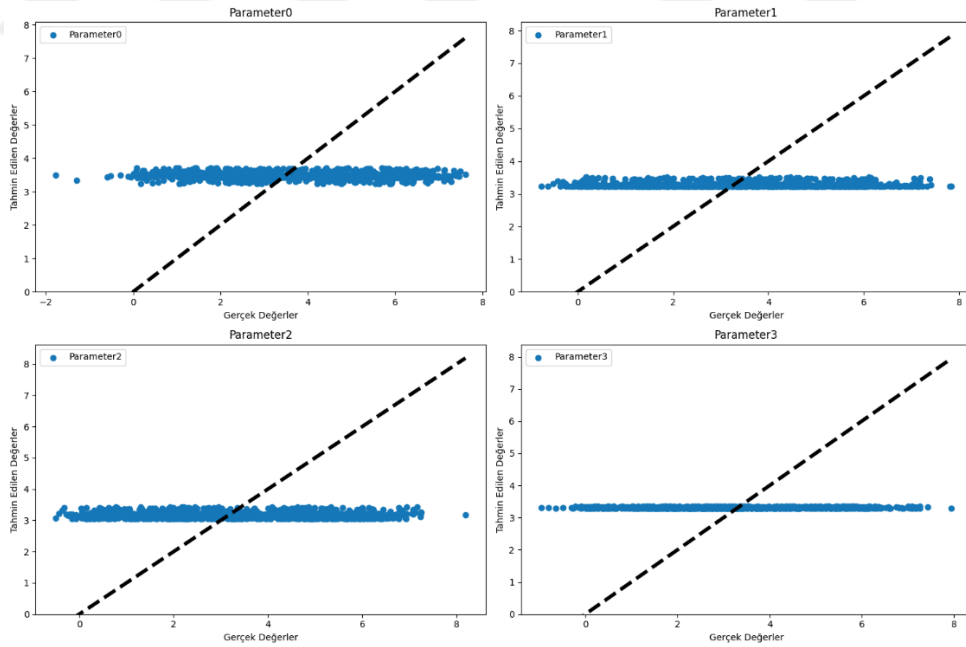
Şekil 3.4 : QAOA parametrelerinin performans sıcaklık haritası.

Şekil 3.4'te her parametrenin ayrı ayrı eğitim ve test için ortalama kare hatası (MSE) değerlerini göstermektedir. Bu değerin düşük olması modelin iyi çalıştığını göstermektedir. Ancak eğitilen modelin 3,4 -3,7 aralığı ile yüksek bir MSE değeri oluşturduğu gözlemlenmiştir. Şekil 3.5'te verilen grafiklerde kırmızı çizgi üzerinde mavi noktaların toplanması modelin verimliliğinin yüksekliğini gösterir. Ancak

modelin belirli aralıklarda kalarak doğru bir tahmin yapamadığı görülmektedir. Şekil 3.4 ve Şekil 3.5'te verilen ve Çizelge 3.2'de ortalama kare hatası (MSE) ve R^2 değerlerinin belirtildiği verilere göre Sinir Ağları modeli bizim sistemimizde ideal sonuçları vermediği gözlemlenmiştir. R^2 değerinin 1'e yakın olması modelin doğruluğunu göstereceği için elde edilen veriler en fazla yaklaşık olarak 0,12 olması bu modelin verimliliğinin düşük olduğunu göstermektedir.

Çizelge 3.2 : Parametre modellerinin MSE ve R^2 sonuçları.

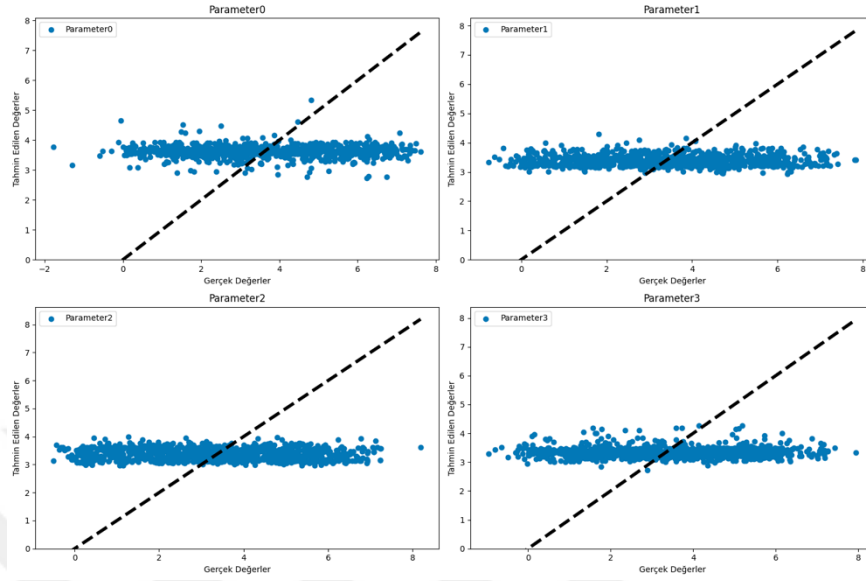
Model	MSE/ R^2	Değer
γ_0	MSE	3.666584561172882
γ_0	R^2	-0.015929479994736706
β_0	MSE	3.5556293065465407
β_0	R^2	-0.004067435959450583
γ_1	MSE	3.537608944205181
γ_1	R^2	-0.01220796306967098
β_1	MSE	3.4237463098830503
β_1	R^2	-0.007615868928915193



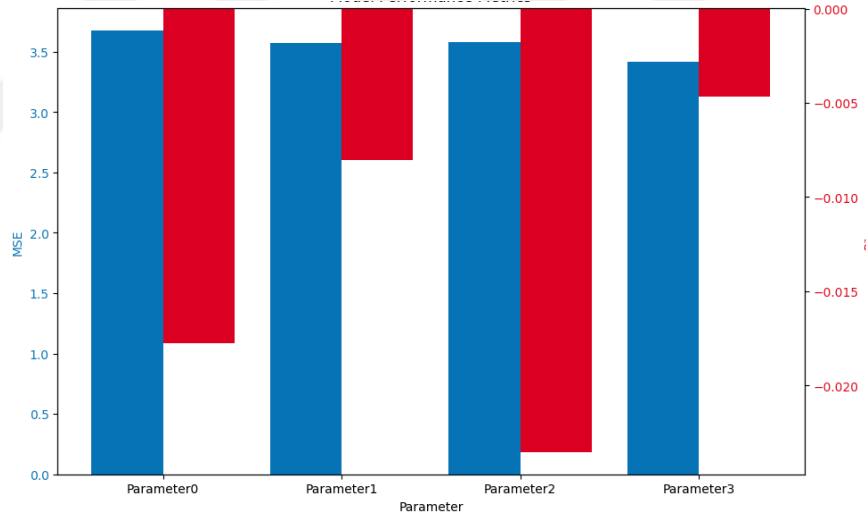
Şekil 3.5 : Gerçek ve tahmin edilen parametrelerin karşılaştırması.

Şekil 3.6 ve Şekil 3.7'de Destek Vektör Makineleri (SVM) Regresyonu ile eğitilmiş parametreler modelidir. Şekil 3.6'da verilen grafikte görüldüğü gibi modeldeki parametre tahminleri gerçek tahminler ile tam uyuşmamaktadır. 2 ile 4 arasında

değişen tahmin değerleri mevcuttur. Serpilme diyagramında noktalar geniş bir yayılım gösterir ve kırık çizgi etrafında yoğunlaşmamaktadır bu da modelin tahminlerinin çok da doğru olmadığını göstermektedir.

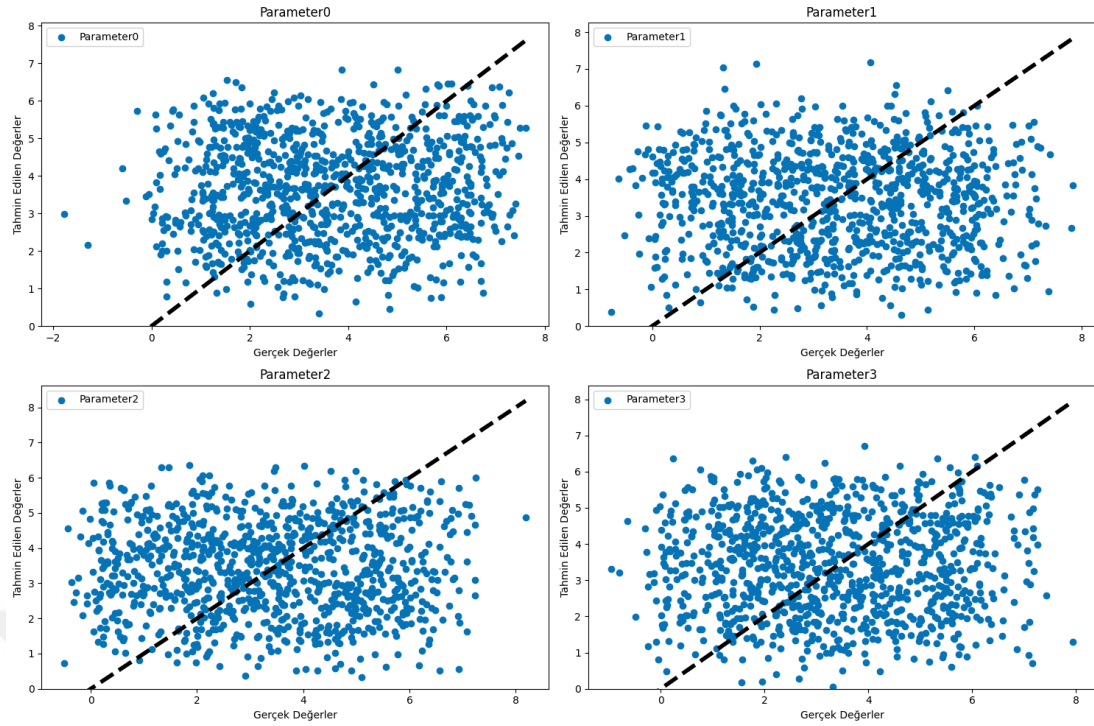


Şekil 3.6 : SVM modeli için gerçek ve tahmin edilen parametrelerin karşılaştırması.



Şekil 3.7 : SVM modeli için parametrelerin MSE (Mavi) ve R^2 (Kırmızı) değerleri.

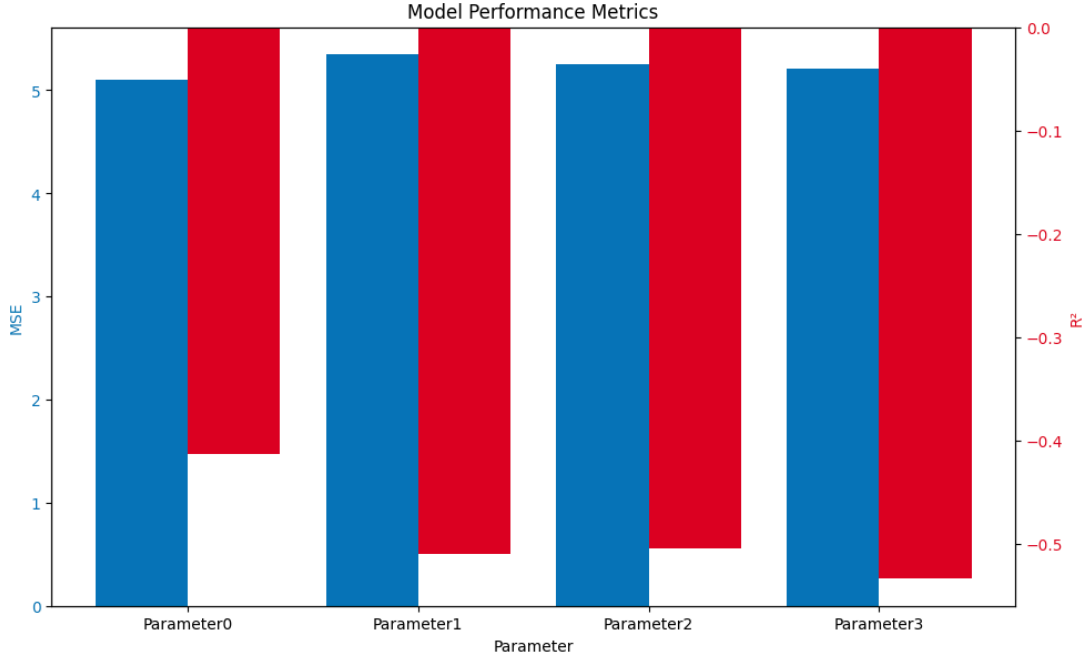
MSE değerleri genellikle yüksektir, bu modelin tahminlerinde önemli hatalar olduğunu gösterir. R^2 değerleri ise 0'a yakın veya negatiftir. Bu da modelin veri varyansını yeterince iyi açıklayamadığını gösterir.



Şekil 3.8 : Random Forest modeli için gerçek ve tahmin edilen parametrelerin karşılaştırması.

Şekil 3.8’de Serpilme diyagramı, gerçek ve tahmin edilen değerlerin karşılaştırılmasını sağlar. Grafikteki noktalar gerçek değerleri, kesikli çizgi ise mükemmel tahmini gösterir. İdeal durumda tüm noktalar kesikli çizgi üzerinde yer almalıdır. Noktalar genel olarak kesikli çizgi etrafında toplanmış, ancak yayılımın geniş olduğu gözlemlenmiştir.

Şekil 3.9’deki bar grafikleri, MSE ve R^2 metriklerini karşılaştırır. Düşük MSE (Mean Squared Error), modelin daha iyi olduğunu göstermektedir. R^2 değeri, modelin veri varyansını ne kadar iyi açıkladığını göstermektedir.

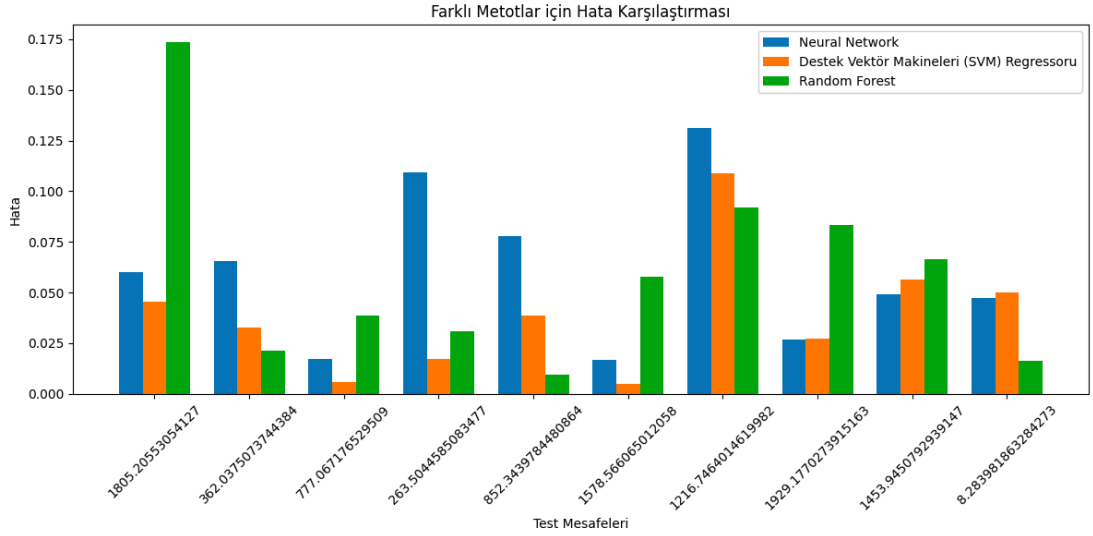


Şekil 3.9 : Random Forest modeli için parametrelerin MSE (Mavi) ve R^2 (Kırmızı) değerleri.

MSE değerleri genellikle yüksektir, bu da modelin tahminlerinde önemli hatalar olduğunu gösterir. R^2 değerleri ise 0'a yakın veya negatiftir. Bu da modelin veri varyansını yeterince iyi açıklayamadığını göstermektedir.

Verilen modellerin performans metriklerine ve Serpilme diyagramına dayanarak, tüm modellerin parametre tahminlerinde düşük performans gösterdiği söylenebilir. Serpilme diyagramında, tahmin edilen ve gerçek değerler arasında büyük sapmalar görülmekte ve performans metrikleri (MSE ve R^2) model performansının idealin oldukça altında olduğunu göstermektedir.

Şekil 3.10 üç farklı modelin (Sınır Ağları, Destek Vektör Makineleri (SVM) Regresyonu ve Random Forest) test setleri üzerindeki hata oranlarını karşılaştırmaktadır. Hata oranı, her bir test mesafesi için modellenen ve gerçek değerler arasındaki farkı temsil etmektedir.



Şekil 3.10 : Farklı makine öğrenmesi yöntemlerinin parametre tahminindeki hata karşılaştırması.

Her üç model de test mesafeleri üzerinde farklı hata oranları göstermektedir. Hata oranları, farklı test mesafeleri için geniş bir aralıkta değişmektedir. Bu, modellerin her test mesafesinde tutarlı performans göstermediğini gösterir.

Sinir Ağları, bazı test mesafelerinde (örneğin, 263.5044585083477, 1578.566065012058) en düşük hata oranlarına sahipken, bazılarında (örneğin, 1929.1770273915163) hata oranları daha yüksektir. Genel olarak, Sinir Ağları modeli, test mesafelerine göre değişken bir performans sergilemektedir.

Random Forest, bazı test mesafelerinde (örneğin, 1805.20553054127, 362.0375073744384) en düşük hata oranlarına sahipken bazılarında (örneğin, 852.3439784480864) daha yüksektir. Buradan hareketle Random Forest modeli, test mesafelerine göre değişken bir performans sergilemekte olduğu anlaşılmaktadır.

SVM modeli, bazı test mesafelerinde (örneğin, 777.067176529509, 1216.7464014619982) en düşük hata oranlarına sahipken bazı test mesafelerinde (örneğin, 1805.20553054127) daha yüksektir. Genel olarak, SVM modeli Sinir Ağları modeli ve Random Forest modeli ile karşılaştırıldığında belirli test mesafelerinde diğer modellere göre daha düşük hata oranları göstermektedir.

Sinir Ağları ve SVM modelleri genellikle daha düşük hata oranlarına sahiptir, ancak her iki model de bazı test mesafelerinde yüksek hata oranları göstermektedir. Random Forest modeli genellikle daha yüksek hata oranlarına sahiptir ve performansı daha

değişkendir. Genel olarak, SVM modeli, diğer iki modele kıyasla daha düşük ve tutarlı hata oranları ile daha iyi performans göstermektedir.

Bu hata karşılaştırması, her bir modelin farklı test mesafeleri üzerindeki performansını değerlendirmek için yararlıdır. Ancak, her modelin performansı test mesafelerine göre değişmektedir. Çizelge 3.3'te farklı makine öğrenmesi yöntemlerinin mesafelere göre doğru sonuç olasılığı verilmiştir.

Çizelge 3.3 : Farklı makine öğrenmesi yöntemlerinin mesafelere göre doğru sonuç olasılığı.

Mesafe	Sinir Ağları	Destek Vektör Makineleri (SVM) Regresyonu	Random Forest	Kuantum Bilgisayarı
1805,2055	0,0260	0,0455	0,2596	0,0862
362,0375	0,0173	0,0613	0,0618	0,0829
777,0672	0,0712	0,0262	0,0497	0,0883
263,5045	0,2203	0,2570	0,1421	0,1112
852,3440	0,1428	0,1973	0,0556	0,065
1578,5661	0,0892	0,0009	0,0483	0,1059
1216,7464	0,0008	0,3124	0,0402	0,1321
1929,1770	0,0781	0,0443	0,1348	0,0515
1453,9451	0,1335	0,0679	0,0180	0,0843
8,2840	0,1480	0,0778	0,0845	0,1008

3.3 Yapılan Optimizasyonun Doğruluğunun Test Edilmesi

Elde edilen sonuçlar, makine öğrenmesi modellerinin istenen performansa ulaşamadığını açıkça göstermektedir. Bu değerlendirme, modellerin MSE ve R^2 metrikleri ile yapılan performans analizine dayanmaktadır. MSE, modelin tahmin ettiği değerler ile gerçek değerler arasındaki farkın karesinin ortalamasını ifade ederken, R^2 değeri, gerçek değer ile modelin tahminleri arasındaki uyumun bir ölçüsüdür. Yüksek MSE değerleri, modelin tahminlerinde büyük hatalar yaptığını ve gerçek sonuçlardan sapmaların fazla olduğunu işaret ederken, düşük R^2 değerleri modelin açıklayıcılığının yetersiz kaldığını ve veri üzerindeki değişkenliği iyi bir şekilde yakalayamadığını göstermektedir.

Bu sonuçlar, modelin veriye uygun bir şekilde eğitilemediğini ve öngörülen performansa ulaşamadığını doğrulamaktadır. Eğitim sürecinde kullanılan veri seti, modelin öğrenme kapasitesi ya da seçilen model parametreleri bu sonuçlara neden olabilmektedir. Özellikle, modelin karmaşıklığı ile verinin karakteristikleri arasındaki uyumsuzluk, bu tür performans düşüklüklerine yol açabilmektedir. Ayrıca, modelin fazla ya da yetersiz öğrenme yapması durumu da bu performans metriklerinde olumsuz etkiler yaratabilmektedir.

Sonuç olarak, modelin eğitimi sürecinde bir dizi faktörün yeniden gözden geçirilmesi gerekmektedir. Verinin daha iyi analiz edilmesi, modelin karmaşıklığının optimize edilmesi ve eğitim sürecindeki hiper parametrelerin iyileştirilmesi gibi adımlar, modelin performansını artırmak için değerlendirilebilir. Bu bağlamda, farklı model türlerinin denenmesi veya mevcut modelin iyileştirilmesi, ilerleyen süreçlerde daha başarılı sonuçlar elde edilmesine katkı sağlayabilir.

4. SONUÇ VE ÖNERİLER

Bu tezde, Gezgin Satıcısı Problemi'ne (TSP) çözüm bulmak için Kuantum Yaklaşık Optimizasyon Algoritması (QAOA) kullanılmıştır. Farklı parametre setlerini inceleyerek ve optimize ederek TSP probleminin çözümünün ne kadar iyileştirilebileceğini araştırılmıştır. Öncelikle TSP problemini kuantum bilgisayarlarında çalıştırabilmek için problem kübitlere aktararak şehirler arası mesafeler, bu kübitleri belirli ölçülerle döndürülecek şekilde işlenmiştir. Kuantum devresi kurulup deney tamamlanmıştır. Kuantum bilgisayarı önce simülasyon ile kalibre edilerek simülasyonda yapılan çalışmaların doğruluğu sağlanmıştır.

Sinir Ağları, Destek Vektör Makineleri (SVM) ve Random Forest yöntemleri kullanılarak modeller eğitilmiştir. Modellerin performansı, ortalama kare hatası ve R^2 değeri gibi metriklerle değerlendirilmiştir. Modellerin performansları Serpilme diyagramı ve bar grafikleri ile görselleştirilmiştir. Hata oranları ve sonuç değerleri, simülasyon sonuçları ve kuantum ölçümleri ile karşılaştırılmıştır.

Random Forest ve SVM modelleri genel olarak daha düşük hata oranları ve daha tutarlı performans göstermiştir. Sinir Ağları modeli genellikle daha yüksek hata oranları ile düşük performans sergilemiştir. Simülasyon sonuçları ve kuantum ölçümleri arasında bazı mesafelerde belirgin farklar gözlemlenmiştir. Bazı mesafelerde simülasyon sonuçları kuantum ölçümlerine göre daha yüksek veya düşük sonuçlar vermiştir. Modeller, test problemlerinde doğru sonucun olasılığı bakımında kuantum bilgisayarı ile karşılaştırıldıklarında, en yakın sonucu yaklaşık %3,9 ile SVM modelinin oluşturduğu gözlemlenmiştir.

Makine öğrenmesi modellerinin performans değerlendirmesi, istenen tahmin doğruluğuna ulaşamadığını ortaya koymuştur. Yüksek MSE ve düşük R^2 değerleri, modellerin hedeflenen sonuçları üretmede yetersiz kaldığını açıkça göstermektedir. Bu sonuçlar, mevcut verinin özellikleri ile modelin yapısı arasında bir uyumsuzluk olduğunu ve modelin eğitim sürecinde yeterli genelleme yeteneği kazanamadığını işaret etmektedir. Modelin veriye uygun bir şekilde öğrenememesi hem modelin

karmaşıklığı hem de eğitim sürecinde kullanılan parametrelerin yeniden ele alınması gerektiğini göstermektedir.

Gelecekteki çalışmalarda, bu modellerin performansını artırmak için veri setinin daha dikkatli bir şekilde incelenmesi, model yapılandırmasının optimize edilmesi ve alternatif modelleme tekniklerinin denenmesi önem arz etmektedir. Bu sonuçlar, yalnızca mevcut modelin eksikliklerini vurgulamakla kalmamakta, aynı zamanda modelin iyileştirilmesine yönelik adımların atılması gerektiğini de ortaya koymaktadır. Bu doğrultuda veriye uygun modellerin geliştirilmesi, elde edilen sonuçların doğruluğunu ve güvenilirliğini artırmak için kritik bir önem taşımaktadır.

Gelecekte yapılması planlanan çalışmalarda kullanılacak optimizasyon yöntemi değiştirilerek farklı yöntemler ile daha yaklaşık sonuçlar alınabilecektir. Bu sayede problem belirlendikten sonra doğru parametreleri bulmaya vakit harcanmamış olacaktır. Farklı ve daha büyük veri setleri ile modellerin eğitilmesi ve genelleştirilebilirliğinin artırılması, QAOA ve benzeri kuantum algoritmalarının performansını artırmak için yeni yöntemlerin geliştirilmesi, kuantum bilgisayarlarının donanım ve yazılım düzeyinde iyileştirilmesi, kuantum hesaplama ve makine öğrenmesi yöntemlerinin birleştirilerek yeni ve etkili çözümler geliştirilmesi ve farklı disiplinlerdeki problemlere kuantum ve makine öğrenmesi yöntemlerinin uygulanması gibi çözümler bu çalışmada alınan sonuçları arttırabilir. Sonuç olarak, bu tez çalışma, kuantum algoritmalarının optimizasyon problemlerindeki etkinliğini artırmak için önemli bir adım niteliğindedir. Elde edilen bulgular ve öneriler, gelecekteki çalışmalar için değerli bir rehber olacaktır.

Bu tez çalışması, TSP probleminin QAOA ile çözümü için parametrelerin makine öğrenmesi ile doğru bir şekilde tahmin edilemediğini göstermiştir. Sonuçlar, QAOA parametrelerinin doğru şekilde eğitilememesinin veri setinden, problemin derinliğinden ve makine öğrenmesi yönteminden kaynaklandığını göstermektedir.

KAYNAKÇA

- [1] Nielsen, M. A. & Chuang, I. L. (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge.: Cambridge University Press.
- [2] Wu, X., Liang, X., Tian, Y., Yang, F., Chen, C., Liu, Y.-C., Tey, M. K., & You, L. (2020). A concise review of Rydberg atom based quantum computation and quantum simulation, *Chinese Physics B*, 30 (2) 020305.
- [3] Santagati, R., Aspuru-Guzik, A., Babbush, R., Degroote, M., González, L., Kyoseva, E., Moll, N., Oppel, M., Parrish, R. M., Rubin, N. C., Streif, M., Tautermann, C. S., Weiss, H., Wiebe, N., & Utschig-Utschig, C. (2024). Drug design on quantum computers, *Nature Physics*, 20(4), 549-557.
- [4] Huggins, W. J., O’Gorman, B. A., Rubin, N. C., Reichman, D. R., Babbush, R., & Lee, J. (2022). Unbiasing fermionic quantum Monte Carlo with a quantum computer, *Nature*, 603 (7901), 416–420.
- [5] Bassman, L., Urbanek, M., Metcalf, M., Carter, J., Kemper, A. F., & De Jong, W. A. (2021). Simulating quantum materials with digital quantum computers, *Quantum Science and Technology*, 6 (4), 043002.
- [6] Schuld, M. & Petruccione, F. (2021). *Machine Learning with Quantum Computers*. Switzerland, AG.: Springer Cham.
- [7] Bravyi, S., Dial, O., Gambetta, J. M., Gil, D., & Nazario, Z. (2022). The future of quantum computing with superconducting qubits, *Journal of Applied Physics*, 132 (16), 160902.
- [8] Gray, H. M. & Terashi, K. (2022). Quantum Computing Applications in Future Colliders, *Frontiers in Physics*, 10, 864823.
- [9] Ladd, T. D., Jelezko, F., Laflamme, R., Nakamura, Y., Monroe, C., & O’Brien, J. L. (2010). Quantum computers, *Nature*, 464, 45-53.
- [10] Amaro, D., Modica, C., Rosenkranz, M., Fiorentini, M., Benedetti, M., & Lubasch, M. (2022). Filtering variational quantum algorithms for combinatorial optimization, *Quantum Science and Technology*, 7 (1), 015021.
- [11] Blekos, K., Brand, D., Ceschini, A., Chou, C. H., Li, R. H., Pandya, K., & Summer, A. (2024). A review on Quantum Approximate Optimization Algorithm and its variants, *Physics Reports*, 1068 , 1–66.
- [12] Kokail, C., Maier, C., Van Bijnen, R., Brydges, T., Joshi, M. K., Jurcevic, P., Muschik, C. A., Silvi, P., Blatt, R., Roos, C. F., & Zoller, P. (2019). Self-Verifying Variational Quantum Simulation of the Lattice Schwinger Model, *Nature*, 569 (7756), 355–360.
- [13] Bauer, B., Bravyi, S., Motta, M., & Chan, G. K. L. (2020). Quantum algorithms for quantum chemistry and quantum materials science, *Chemical reviews*, 120 (22), 12685–12717.

- [14] **Delgado, A., Arrazola, J. M., Jahangiri, S., Niu, Z., Izaac, J., Roberts, C., & Killoran, N.** (2021). Variational quantum algorithm for molecular geometry optimization, *Physical Review A*, 104 (5), 052402.
- [15] **Hadfield, S., Wang, Z., Rieffel, E. G., O’Gorman, B., Venturelli, D., & Biswas, R.** (2018). Quantum approximate optimization with hard and soft constraints, *In Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, 15–21.
- [16] **Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M. P., Hartmann, M. J., Ho, A., Hoffmann, M., Huang, T., Humble, T. S., Isakov, S. V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P. V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandrà, S., McClean, J. R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M. Y., Ostby, E., Petukhov, A., Platt, J. C., Quintana, C., Rieffel, E. G., Roushan, P., Rubin, N. C., Sank, D., Satzinger, K. J., Smelyanskiy, V., Sung, K. J., Trevithick, M. D., Vainsencher, A., Villalonga, B., White, T., Yao, Z. J., Yeh, P., Zalcman, A., Neven, H., & Martinis, J. M.** (2019). Quantum supremacy using a programmable superconducting processor, *Nature* 2019 574:7779, 574 (7779), 505–510.
- [17] **J., A., Adedoyin, A., Ambrosiano, J., Anisimov, P., Casper, W., Chennupati, G., Coffrin, C., Djidjev, H., Gunter, D., Karra, S., Lemons, N., Lin, S., Malyzhenkov, A., Mascarenas, D., Mniszewski, S., Nadiga, B., O’Malley, D., Oyen, D., Pakin, S., Prasad, L., Roberts, R., Romero, P., Santhi, N., Sinitsyn, N., Swart, P. J., Wendelberger, J. G., Yoon, B., Zamora, R., Zhu, W., Eidenbenz, S., Bärtschi, A., Coles, P. J., Vuffray, M., & Lokhov, A. Y.** (2018). Quantum Algorithm Implementations for Beginners, *ACM Transactions on Quantum Computing*, 3 (4), 1-92.
- [18] **Calderbank, A. R., & Shor, P. W.** (1996). Good quantum error-correcting codes exist, *Physical review A, Atomic, molecular, and optical physics*, 54 (2), 1098–1105.
- [19] **Steane, A. M.** (1996). Simple quantum error-correcting codes, *Physical review A, Atomic, molecular, and optical physics*, 54 (6), 4741–4751.
- [20] **Steane, A. M.** (1996). Multiple Particle Interference and Quantum Error Correction, *Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*, 452 (1954), 2551–2577.
- [21] **Gottesman, D.** (1996). Class of quantum error-correcting codes saturating the quantum Hamming bound., *Physical review A, Atomic, molecular, and optical physics*, 54 (3), 1862–1868.
- [22] **Knill Raymond, E. L.** (1997). Theory of quantum error-correcting codes, *Physical Review A*, 55 (2), 900–911.
- [23] **Plenio, M. B., Vedral, V., & Knight, P. L.** (1997). Quantum error correction in the presence of spontaneous emission., *Physical Review A*, 55 (1), 67–71.

- [24] **Bennett, C. H., DiVincenzo, D. P., Smolin, J. A., & Wootters, W. K.** (1996). Mixed State Entanglement and Quantum Error Correction, *Physical review A, Atomic, molecular, and optical physics*, 54 (5), 3824–3851.
- [25] **Cleve Daniel, R. G.** (1997). Efficient computations of encodings for quantum error correction, *Physical Review A*, 56 (1), 76–82.
- [26] **Knill, E., Laflamme, R., & Viola, L.** (2000). Theory of quantum error correction for general noise, *Physical review letters*, 84 (11), 2525–2528.
- [27] **Shor, P. W.** (1995). Scheme for reducing decoherence in quantum computer memory, *Physical review A, Atomic, molecular, and optical physics*, 52 (4), R2493–R2496.
- [28] **Mong, R. S. K., Clarke, D. J., Alicea, J., Lindner, N. H., Fendley, P., Nayak, C., Oreg, Y., Stern, A., Berg, E., Shtengel, K., & Fisher, M. P. A.** (2013). Universal topological quantum computation from a superconductor/Abelian quantum Hall heterostructure, *Physical Review X*, 4 (1), 011036.
- [29] **Sau, J. D., Lutchyn, R. M., Tewari, S., & Sarma, S. Das** (2009). A generic new platform for topological quantum computation using semiconductor heterostructures, *Physical Review Letters*, 104 (4), 040502.
- [30] **Alicea, J., Oreg, Y., Refael, G., von Oppen, F., & Fisher, M. P. A.** (2010). Non-Abelian statistics and topological quantum information processing in 1D wire networks, *Nature Physics*, 7 (5), 412–417.
- [31] **Chen, X., Gu, Z.-C., & Wen, X.-G.** (2010). Local unitary transformation, long-range quantum entanglement, wave function renormalization, and topological order, *Physical Review B - Condensed Matter and Materials Physics*, 82 (15), 155138.
- [32] **Levin, M. & Wen, X.-G.** (2005). Detecting topological order in a ground state wave function, *Physical Review Letters*, 96 (11), 110405.
- [33] **Kitaev, A. & Preskill, J.** (2005). Topological entanglement entropy, *Physical Review Letters*, 96 (11), 110404.
- [34] **Zhang, Y.-R., Zeng, Y., Fan, H., You, J. Q., & Nori, F.** (2017). Characterization of topological states via dual multipartite entanglement, *Physical Review Letters*, 120 (25), 250501.
- [35] **Zhou, D. L.** (2008). Irreducible multi-particle correlations in states without maximal rank, *Physical Review Letters*, 101 (18), 180505.
- [36] **Liu, Y., Zeng, B., & Zhou, D. L.** (2016). Irreducible many-body correlations in topologically ordered systems, *New Journal of Physics*, 18 (2), 023024.
- [37] **Papanikolaou, S., Raman, K. S., & Fradkin, E.** (2007). Topological phases and topological entropy of two-dimensional systems with finite correlation length, *Physical Review B - Condensed Matter and Materials Physics*, 76 (22), 224421.
- [38] **Williamson, D. J., Dua, A., & Cheng, M.** (2018). Spurious topological entanglement entropy from subsystem symmetries, *Physical Review Letters*, 122 (14), 140506.
- [39] **Nielsen, M. A. & Poulin, D.** (2005). Algebraic and information-theoretic conditions for operator quantum error-correction, *Physical Review A - Atomic, Molecular, and Optical Physics*, 75 (6), 064304.
- [40] **Duan, L. M., & Guo, G. C.** (1997). Preserving Coherence in Quantum Computation by Pairing Quantum Bits, *Physical Review Letters*, 79 (10), 1953–1956.

- [41] **Zanardi Mario, P. R.** (1997). Noiseless Quantum Codes, *Physical Review Letters*, 79 (17), 3306–3309.
- [42] **Lidar, D. A., Chuang, I. L., & Whaley, K. B.** (1998). Decoherence-Free Subspaces for Quantum Computation, *Physical Review Letters*, 81 (12), 2594–2597.
- [43] **Palma, G. M., Suominen, K. A., & Ekert, A.** (1996). Quantum Computers and Dissipation, *Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*, 452 (1946), 567–584.
- [44] **Zanardi, P.** (2000). Stabilizing quantum information, *Physical Review A*, 63 (1), 012301.
- [45] **Kempe, J., Bacon, D., Lidar, D. A., & Whaley, K. B.** (2001). Theory of decoherence-free fault-tolerant universal quantum computation, *Physical Review A*, 63 (4), 042307.
- [46] **Kribs, D., Laflamme, R., & Poulin, D.** (2005). Unified and generalized approach to quantum error correction, *Physical review letters*, 94 (18), 180501.
- [47] **Kribs, D. W., Laflamme, R., Poulin, D., & Lesosky, M.** (2006). Operator quantum error correction, *Quantum Information and Computation*, 6 (4 & 5), 382–399.
- [48] **Hoffman, K. L., Padberg, M., & Rinaldi, G.** (2001). *Traveling salesman problem*. In Springer eBooks (pp. 1573–1578). Springer, Boston, MA 1573–1578.
- [49] **Bifan, L., Lipo, W., & Wu, S.** (2008). Ant colony optimization for the traveling salesman problem based on ants with memory, *Proceedings - 4th International Conference on Natural Computation, ICNC 2008*, 7, 496–501.
- [50] **Li, L., Ju, S., & Zhang, Y.** (2008). Improved ant colony optimization for the traveling salesman problem, *Proceedings - International Conference on Intelligent Computation Technology and Automation, ICICTA 2008*, 1, 76–80.
- [51] **Hlaing, Z. C. S. S. & Khine, M. A.** (2011). Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm, *International Journal of Information and Education Technology*, 404–409.
- [52] **Hingrajya, K. H., Kumar Gupta, R., Singh Chandel, G., & Proudtyogiki Vishwavidyalaya, G.** (2012). An Ant Colony Optimization Algorithm for Solving Travelling Salesman Problem, *International Journal of Scientific and Research Publications*, 2 (8), (ISSN: 2250-3153).
- [53] **Dwivedi, V., Chauhan, T., Saxena, S., & Agrawal, P.** Travelling Salesman Problem using Genetic Algorithm, *IJCA proceedings on development of reliable information systems, techniques and related issues*, 1(1), 25.
- [54] **Moon, C., Kim, J., Choi, G., & Seo, Y.** (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints, *European Journal of Operational Research*, 140 (3), 606–617.
- [55] **Gupta, S. & Panwar, P.** (2013). Solving Travelling Salesman Problem Using Genetic Algorithm, *International Journal of Advanced Research in Computer Science and Software Engineering*, 3 (6), 2277.
- [56] **Yan, X., Zhang, C., Luo, W., Li, W., Chen, W., & Liu, H.** (2012). Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm, *International Journal of Computer Science*, 9 (6), 264-271.

- [57] **Fan, H.** (2010). Discrete Particle Swarm Optimization for TSP based on Neighborhood, *Journal of Computational Information Systems*, 6(10), 3407-3414.
- [58] **Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C., & Wang, Q. X.** (2007). Particle swarm optimization-based algorithms for TSP and generalized TSP, *Information Processing Letters*, 103 (5), 169–176.
- [59] **Martonak, R., Santoro, G. E., & Tosatti, E.** (2004). Quantum annealing of the Traveling Salesman Problem, *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 70 (5), 057701.
- [60] **Chen, Y. & Zhang, P.** (2006). Optimized annealing of traveling salesman problem from the n th-nearest-neighbor distribution, *Physica A: Statistical Mechanics and its Applications*, 371 (2), 627–632.
- [61] **Osaba, E., Villar-Rodriguez, E., Oregi, I., & Moreno-Fernandez-de-Leceta, A.** (2021). Hybrid Quantum Computing - Tabu Search Algorithm for Partitioning Problems: Preliminary Study on the Traveling Salesman Problem, *2021 IEEE Congress on Evolutionary Computation, CEC 2021 - Proceedings*, 351–358.
- [62] **Farhi, E., Goldstone, J., & Gutmann, S.** (2014). A Quantum Approximate Optimization Algorithm, *arXiv preprint arXiv:1411.4028*.
- [63] **Falla, J., Langfitt, Q., Alexeev, Y., & Safro, I.** (2024). Graph representation learning for parameter transferability in quantum approximate optimization algorithm, *Quantum Machine Intelligence*, 6 (2), 1–16.
- [64] **Alexeev, Y., Khairy, S., Shaydulin, R., Cincio, L., & Balaprakash, P.** (2020). Reinforcement Learning for Finding QAOA Parameters, *Bulletin of the American Physical Society*, 65 (1).
- [65] **Montanez-Barrera, J. A., Willsch, D., & Michielsen, K.** (2024). Transfer learning of optimal QAOA parameters in combinatorial optimization, *arXiv preprint arXiv:2402.05549*.
- [66] **Galda, A., Liu, X., Lykov, D., Alexeev, Y., & Safro, I.** (2021). Transferability of optimal QAOA parameters between random graphs, *Proceedings - 2021 IEEE International Conference on Quantum Computing and Engineering, QCE 2021*, 171–180.
- [67] **Hadfield, S., Wang, Z., Rieffel, E. G., O’Gorman, B., Venturelli, D., & Biswas, R.** (2018). Quantum approximate optimization with hard and soft constraints, *In Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, 15–21.
- [68] **Hao, T., He, Z., Shaydulin, R., Larson, J., & Pistoia, M.** (2024). End-to-End Protocol for High-Quality QAOA Parameters with Few Shots, *arXiv preprint arXiv:2408.00557*.
- [69] **Sakai, R., Matsuyama, H., Tam, W.-H., Yamashiro, Y., & Fujii, K.** (2024). Linearly simplified QAOA parameters and transferability, *arXiv preprint arXiv:2405.00655*.
- [70] **Czarnik, P., Arrasmith, A., Coles, P. J., & Cincio, L.** Error mitigation with Clifford quantum-circuit data, *Quantum*, 5, 592.
- [71] **Khatri, S., Larose, R., Poremba, A., Cincio, L., Sornborger, A. T., & Coles, P. J.** (2019). Quantum-assisted quantum compiling, *Quantum*, 3, 140.
- [72] **Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K.** (2018). Quantum circuit learning, *Physical Review A*, 98 (3), 032309.

- [73] Sureshababu, S. H., Herman, D., Shaydulin, R., Basso, J., Chakrabarti, S., Sun, Y., & Pistoia, M. (2024). Parameter Setting in Quantum Approximate Optimization of Weighted Problems, *Quantum*, 8, 1231.
- [74] Shaydulin, R., Lotshaw, P. C., Larson, J., Ostrowski, J., & Humble, T. S. (2023). Parameter Transfer for Quantum Approximate Optimization of Weighted MaxCut, *ACM Transactions on Quantum Computing*, 4 (3), 1-15.
- [75] Alam, M., Ash-Saki, A., & Ghosh, S. (2020). Accelerating Quantum Approximate Optimization Algorithm using Machine Learning, *In 2020 Design, Automation & Test in Europe Conference & Exhibition*, (pp. 686-689). IEEE.
- [76] Khairy, S., Shaydulin, R., Cincio, L., Alexeev, Y., & Balaprakash, P. (2020). Learning to Optimize Variational Quantum Circuits to Solve Combinatorial Problems, *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (03), 2367–2375.
- [77] Preskill, J. (1998). Quantum computing: pro and con, *Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*, 454 (1969), 469–486.
- [78] Nation, P. D., Kang, H., Sundaresan, N., & Gambetta, J. M. (2021). Scalable Mitigation of Measurement Errors on Quantum Computers, *PRX Quantum*, 2 (4), 040326.
- [79] Blatt, R. & Roos, C. F. (2012). Quantum simulations with trapped ions, *Nature Physics* 2012 8:4, 8 (4), 277–284.
- [80] Leibfried, D., Blatt, R., Monroe, C., & Wineland, D. (2003). Quantum dynamics of single trapped ions, *Reviews of Modern Physics*, 75 (1), 281.
- [81] Krantz, P., Kjaergaard, M., Yan, F., Orlando, T. P., Gustavsson, S., & Oliver, W. D. (2019). A quantum engineer’s guide to superconducting qubits, *Applied Physics Reviews*, 6 (2), 021318.
- [82] Kjaergaard, M., Schwartz, M. E., Braumüller, J., Krantz, P., Wang, J. I. J., Gustavsson, S., & Oliver, W. D. (2020). Superconducting Qubits: Current State of Play, *Annual Review of Condensed Matter Physics*, 11 (1), 369–395.
- [83] Wang, X., Luo, Y., Huang, H., Chen, M., Su, Z., Liu, C., Chen, C., Li, W., Fang, Y., Jiang, X., Zhang, J., Li, L., Liu, N., Lu, C., & Pan, J. (2018). 18-Qubit Entanglement with Six Photons’ Three Degrees of Freedom, *Physical review letters*, 120 (26), 260502.
- [84] Wang, X., Chen, L., Li, W., Huang, H., Liu, C., Chen, C., Luo, Y., Su, Z., Wu, D., Li, Z., Lu, H., Hu, Y., Jiang, X., Peng, C., Li, L., Liu, N., Chen, Y., Lu, C., & Pan, J. (2016). Experimental Ten-Photon Entanglement, *Physical review letters*, 117 (21), 210502.
- [85] Huang, H., Wang, X., Rohde, P. P., Luo, Y., Zhao, Y., Liu, C., Li, L., Liu, N., Lu, C., & Pan, J. (2018). Demonstration of topological data analysis on a quantum processor, *Optica*, 5 (2), 193–198.
- [86] Huang, H., Zhao, Q., Ma, X., Liu, C., Su, Z., Wang, X., Li, L., Liu, N., Sanders, B. C., Lu, C., & Pan, J. (2017). Experimental Blind Quantum Computing for a Classical Client, *Physical review letters*, 119 (5), 050503.
- [87] Wang, H., Qin, J., Ding, X., Chen, M., Chen, S., You, X., He, Y., Jiang, X., You, L., Wang, Z., Schneider, C., Renema, J. J., Höfling, S., Lu, C.,

- & Pan, J.** (2019). Boson Sampling with 20 Input Photons and a 60-Mode Interferometer in a 10^{14} -Dimensional Hilbert Space, *Physical review letters*, 123 (25), 250503.
- [88] **Kane, B. E.** (1998). A silicon-based nuclear spin quantum computer, *Nature*, 393 (6681), 133–137.
- [89] **He, Y., Gorman, S. K., Keith, D., Kranz, L., Keizer, J. G., & Simmons, M. Y.** (2019). A two-qubit gate between phosphorus donor electrons in silicon, *Nature*, 571 (7765), 371–375.
- [90] **Nakamura, Y., Pashkin, Y. A., & Tsai, J. S.** (1999). Coherent control of macroscopic quantum states in a single-Cooper-pair box, *Nature*, 398 (6730), 786–788.
- [91] **Barends, R., Kelly, J., Megrant, A., Veitia, A., Sank, D., Jeffrey, E., White, T. C., Mutus, J., Fowler, A. G., Campbell, B., Chen, Y., Chen, Z., Chiaro, B., Dunsworth, A., Neill, C., O’Malley, P., Roushan, P., Vainsencher, A., Wenner, J., Korotkov, A. N., Cleland, A. N., & Martinis, J. M.** (2014). Superconducting quantum circuits at the surface code threshold for fault tolerance, *Nature* 2014 508:7497, 508 (7497), 500–503.
- [92] **Wie, C.-R.** (2014). Bloch sphere model for two-qubit pure states, *arXiv preprint arXiv:1403.8069*.
- [93] **Grier, D. & Schaeffer, L.** The Classification of Clifford Gates over Qubits, *Quantum*, 6, 734.
- [94] **Deutsch, D.** (1985). Quantum theory, the Church–Turing principle and the universal quantum computer, *Proceedings of the Royal Society of London A Mathematical and Physical Sciences*, 400 (1818), 97–117.
- [95] **Preskill, J.** (2018). Quantum Computing in the NISQ era and beyond, *Quantum*, 2, 79.
- [96] **Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M. P., Hartmann, M. J., Ho, A., Hoffmann, M., Huang, T., Humble, T. S., Isakov, S. V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P. V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandrà, S., McClean, J. R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M. Y., Ostby, E., Petukhov, A., Platt, J. C., Quintana, C., Rieffel, E. G., Roushan, P., Rubin, N. C., Sank, D., Satzinger, K. J., Smelyanskiy, V., Sung, K. J., Trevithick, M. D., Vainsencher, A., Villalonga, B., White, T., Yao, Z. J., Yeh, P., Zalcman, A., Neven, H., & Martinis, J. M.** (2019). Quantum supremacy using a programmable superconducting processor, *Nature*, 574 (7779), 505–510.
- [97] **Wang, Z., Wei, S., Long, G. L., & Hanzo, L.** (2022). Variational quantum attacks threaten advanced encryption standard based symmetric cryptography, *Science China Information Sciences*, 65 (10), 1–12.
- [98] **Peruzzo, A., McClean, J., Shadbolt, P., Yung, M. H., Zhou, X. Q., Love, P. J., Aspuru-Guzik, A., & O’Brien, J. L.** (2014). A variational

- eigenvalue solver on a photonic quantum processor, *Nature Communications* 2014 5:1, 5 (1), 1–7.
- [99] **Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., & Coles, P. J.** (2021). Variational quantum algorithms, *Nature Reviews Physics*, 3 (9), 625–644.
- [100] **Wei, S., Li, H., & Long, G.** (2019). A Full Quantum Eigensolver for Quantum Chemistry Simulations, Research.
- [101] **McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S., & Yuan, X.** (2018). Quantum computational chemistry, *Reviews of Modern Physics*, 92 (1), 015003.
- [102] **Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S.** (2016). Quantum Machine Learning, *Nature*, 549 (7671), 195–202.
- [103] **Wang, Z., Hadfield, S., Jiang, Z., & Rieffel, E. G.** (2017). Quantum Approximate Optimization Algorithm for MaxCut: A Fermionic View, *Physical Review A*, 97 (2), 022304.
- [104] **Harrigan, M. P., Sung, K. J., Neeley, M., Satzinger, K. J., Arute, F., Arya, K., Atalaya, J., Bardin, J. C., Barends, R., Boixo, S., Broughton, M., Buckley, B. B., Buell, D. A., Burkett, B., Bushnell, N., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Demura, S., Dunsworth, A., Eppens, D., Fowler, A., Foxen, B., Gidney, C., Giustina, M., Graff, R., Habegger, S., Ho, A., Hong, S., Huang, T., Ioffe, L. B., Isakov, S. V., Jeffrey, E., Jiang, Z., Jones, C., Kafri, D., Kechedzhi, K., Kelly, J., Kim, S., Klimov, P. V., Korotkov, A. N., Kostritsa, F., Landhuis, D., Laptev, P., Lindmark, M., Leib, M., Martin, O., Martinis, J. M., McClean, J. R., McEwen, M., Megrant, A., Mi, X., Mohseni, M., Mruczkiewicz, W., Mutus, J., Naaman, O., Neill, C., Neukart, F., Niu, M. Y., O’Brien, T. E., O’Gorman, B., Ostby, E., Petukhov, A., Putterman, H., Quintana, C., Roushan, P., Rubin, N. C., Sank, D., Skolik, A., Smelyanskiy, V., Strain, D., Streif, M., Szalay, M., Vainsencher, A., White, T., Yao, Z. J., Yeh, P., Zalcman, A., Zhou, L., Neven, H., Bacon, D., Lucero, E., Farhi, E., & Babbush, R.** (2020). Quantum Approximate Optimization of Non-Planar Graph Problems on a Planar Superconducting Processor, *Nature Physics*, 17 (3), 332–336.
- [105] **Green, A. S., Lumsdaine, P. L., Ross, N. J., Selinger, P., & Valiron, B.** (2013). Quipper: A Scalable Quantum Programming Language, *ACM SIGPLAN Notices*, 48 (6), 333–342.
- [106] **JavadiAbhari, A., Patil, S., Kudrow, D., Heckey, J., Lvov, A., Chong, F. T., & Martonosi, M.** (2015). ScaffCC: Scalable compilation and analysis of quantum programs, *Parallel Computing*, 45 (NA), 2–17.
- [107] **Wecker, D. & Svore, K. M.** (2014). LIQUi|>: A Software Design Architecture and Domain-Specific Language for Quantum Computing, *arXiv preprint arXiv:1402.4467*.
- [108] **Rand, R., Paykin, J., & Zdanczewicz, S.** (2018). Qwire practice: Formal verification of quantum circuits in coq, *Electronic Proceedings in Theoretical Computer Science, EPTCS*, 266 , 119–132.
- [109] **Cross, A., Cross, & Andrew** (2018). The IBM Q experience and QISKit open-source quantum computing software, *APS*, 2018 , L58.003.

- [110] **Ho, A., & Bacon, D.** (2018). Announcing Cirq: an open source framework for NISQ algorithms, *Google AI Blog*, 18.
- [111] **Deutsch, D.** (1985). Quantum theory, the Church–Turing principle and the universal quantum computer, *Proceedings of the Royal Society of London A Mathematical and Physical Sciences*, 400 (1818), 97–117.
- [112] **Deutsch, D. & Jozsa, R.** (1992). Rapid solution of problems by quantum computation, *Proceedings of the Royal Society of London Series A, Mathematical and physical sciences*, 439 (1907), 553–558.
- [113] **Cleve, R., Ekert, A., Macchiavello, C., & Mosca, M.** (1997). Quantum Algorithms Revisited, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 454 (1969), 339–354.
- [114] **Perez-Garcia, B., McLaren, M., Goyal, S. K., Hernandez-Aranda, R. I., Forbes, A., & Konrad, T.** (2015). Quantum computation with classical light: implementation of the Deutsch-Jozsa Algorithm, *Physics Letters, Section A: General, Atomic and Solid State Physics*, 379 (28–29), 1675–1680.
- [115] **Shor, P. W.** (1994). Algorithms for quantum computation: Discrete logarithms and factoring, *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 124–134.
- [116] **Hanoymak, T. & Chehrazi, A.** (2019). Fundamental Structure of Shor’s Quantum Algorithm for Factoring Integers, *Turk J Math Comput Sci*, 11 (2), 78–83.
- [117] **Pomerance, C.** A Tale of Two Sieves, *Notices of the American Mathematical Society*, 43(12), 1473-1485.
- [118] **Montanez-Barrera, J. A., Willsch, D., & Michielsen, K.** (2024). Transfer learning of optimal QAOA parameters in combinatorial optimization, *arXiv preprint arXiv:2402.05549*.
- [119] **Nguyen, Q. N.** (2020). Travelling Salesman Problem and Bellman-Held-Karp Algorithm.

ÖZGEÇMİŞ

Ad-Soyad : Burhan ENGİN

Doğum Tarihi ve Yeri :

E-posta :

ÖĞRENİM DURUMU:

- **Lisans** : 2021, Yıldız Teknik Üniversitesi, Fen-Edebiyat Fakültesi, Fizik Bölümü

MESLEKİ DENEYİM VE ÖDÜLLER:

- SYS Yazılım (2022) – Yazılım Geliştirici
- Bursa Teknik Üniversitesi (2022 -) – Araştırma Görevlisi

DİĞER ESERLER, SUNUMLAR VE PATENTLER:

Dahil olunan projeler;

- Bilimsel Araştırmalarda X-Işınları Analiz Teknikleri-IV (biXat-2022). Çevrim içi, 27 30 Ekim, 2022. TÜBİTAK 2237-A, (Yerel düzenleme kurulu/yardımcı personel). Tamamlandı.
- Tek-Kristal Sezyum Kurşun Bromür Yarı-İletken Malzemenin Oda Sıcaklığında 662 keV Enerjili Gama Işınlarına Karşı Dedeksiyon Özelliklerinin Araştırılması, Proje No: 221N007, BTÜ-BAP-NAP, 2022 (21.06.2022-20.12.2023). Araştırmacı, Tamamlandı.
- 6 Şubat 2023 Kahramanmaraş ve 20 Şubat 2023 Hatay Depremleri Sonrası Kapalı Alan Radon Konsantrasyonlarının Belirlenmesi ve Deprem-Radon İlişkisinin Araştırılması. TÜBİTAK (1002), Proje No: 123F072, 15-Mayıs-15 Kasım 2023. . (15.05.2023-15.11.2023), Bursiyer, Tamamlandı.
- Depremlerin kapalı ortam radon konsantrasyonuna etkisinin araştırılması: Kahramanmaraş örneği, BTÜ-BAP-YL, Proje No: 230Y011, 2023 (21.07.2023-20.07.2024), Araştırmacı, Tamamlandı.

Bildiriler;

- [1] Engin, B., Özen, M., Özen S. A. ve Çelik, N. (2024). "Study of the radiation detection efficiency and resolution properties of a perovskite CsPbBr₃ single-crystal by means of EGS4", *International Conference On Natural Sciences And Technologies (ICONAT-2024)*, Antalya, Türkiye: 30 May-1 Jun 2024.