DÜZCE ÜNİVERSİTESİ | LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BLOK ZİNCİRDE YAPAY ZEKA DESTEKLİ YENİ BİR ONAY MEKANİZMASININ GELİŞTİRİLMESİ: OPTİMİZASYON TABANLI ONAY MEKANİZMASI (PoO)

FATİH KÜRŞAD GÜNDÜZ

DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ
(DR) ANABİLİM DALI

DANIŞMAN
DOÇ. DR. SERDAR BİROĞUL

DÜZCE, 2023

T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BLOK ZİNCİRDE YAPAY ZEKA DESTEKLİ YENİ BİR ONAY MEKANİZMASININ GELİŞTİRİLMESİ: OPTİMİZASYON TABANLI ONAY MEKANİZMASI (PoO)

Fatih Kürşad GÜNDÜZ tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Lisansüstü Eğitim Enstitüsü Enstitüsü Elektrik-Elektronik Ve Bilgisayar Mühendisliği (Dr) Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Serdar BİROĞUL

Düzce Üniversitesi

Eş Danışman

Doç. Dr. Utku KÖSE

Süleyman Demirel Üniversitesi

Jüri Üyeleri

Doç. Dr. Serdar BİROĞUL
Düzce Üniversitesi _____

Prof. Dr. Uğur GÜVENÇ
Düzce Üniversitesi _____

Prof. Dr. Yusuf SÖNMEZ
Gazi Üniversitesi _____

Doç. Dr. Ali ÇALHAN
Düzce Üniversitesi _____

Doç. Dr. Osman ÖZKARACA
Muğla Sıtkı Koçman Üniversitesi _____

Tez Savunma Tarihi: 06/12/2023

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

6 Aralık 2023

Fatih Kürşad GÜNDÜZ

TEŞEKKÜR

Doktora öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç. Dr. Serdar BİROĞUL'A en içten dileklerimle teşekkür ederim.

Tez çalışmam boyunca değerli katkılarını esirgemeyen eş danışmanım Doç. Dr. Utku KÖSE'ye de şükranlarımı sunarım.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

6 Aralık 2023

Fatih Kürşad GÜNDÜZ

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

ÇİZELGE LİSTESİ

Sayfa No

KISALTMALAR

BFT	Byzantine Fault Tolerance
GA	Genetic Algoritma
GSP	Gezgin Satıcı Problemi
PoAC	Proof of Activity
PoB	Proof of Burn
PoC	Proof of Capacıty
PoET	Proof of Elapsed Time
PoO	Proof of Optimum
PoS	Proof of Stake
PoW	Proof of Work
SBİ	Saniye Başına İşlem
YZ	Yapay Zeka

SİMGELER

i	Dizi elemanı
n	Dizi elemanı sayısı
n_{best}	Dizinin en iyi çözümü
n_i	Dizi elemanı
p_i	Bir düğüm tarafında çıkarılan blok sayısı

ÖZET

BLOK ZİNCİRDE YAPAY ZEKA DESTEKLİ YENİ BİR ONAY MEKANİZMASININ GELİŞTİRİLMESİ: OPTİMİZASYON TABANLI ONAY MEKANİZMASI (PoO)

Fatih Kürşad GÜNDÜZ

Düzce Üniversitesi
Lisansüstü Eğitim Enstitüsü, Elektrik-Elektronik Ve Bilgisayar Mühendisliği
(Dr)Anabilim Dalı
Doktora Tezi
Danışman: Doç Dr. Serdar BİROĞUL

Aralık 2023, 91 sayfa

Blokzincir sistemleri son dönemin ortaya çıkan popüler teknolojilerdendir. Merkezi olmayan bir sistem olarak blokzincir teknolojisi birçok çözüm sunmuş ancak bu çözümlere bağlı birçok soruna yol açmıştır. En önemli sorunlarından biri de yeni bir konsensüs bloğu oluşturmak için özet(hash) hesaplamalarını çok yoğun yaparken süreye bağlı olarak verimliliğini düşürmesidir. Bu çalışmada, blok oluşturmak için yapılan hesaplamaları optimizasyon algoritmalarına yönlendiren Proof of Work'ten (PoW) kaçınmak için yeni bir model önerilmektedir. Önerilen kanıt mekanizmasına Optimizasyon Kanıtı (Proof of Optimum,PoO) adı verilmektedir. Optimizasyon algoritmalarını çözmek için tasarlanan sisteme problem olarak Gezgin Satıcı Problemi (GSP) tanımlanmıştır. Düğümlerden GSP'yi belirli iterasyonlarla ve popülasyonlarda çözmeleri istenir. Sonuç olarak düğümlerden elde edilen uygunluk, yoğunluk ve zaman değerleriyle bloklar oluşturmaları istenir. PoO ve PoW konsensüs mekanizması sistemde deneysel bir karşılaştırmaya tabi tutulmuştur. Test sonuçları, PoO konsensüs modelinin blok oluşturma süresinin en az şehirli veri seti çözümüne göre 2 sn ile en fazla şehirli veri seti çözümüne göre 60 sn arasında değiştiğini göstermektedir. Bu çalışmada değerlendirilen GSP'deki şehir sayısı değiştirilerek problemlerin zorluk düzeyleri ayarlanabilmektedir. Bu sayede ağda blok oluşturma sorunu her an daha zor veya daha kolay hale getirilebilir. Deneysel analizler sonucunda blokzincirde madenciler arasında blok oluşturma yüzdesi olan merkeziyetsizliğin daha istikrarlı bir değere ulaştığı ve adalet endeksinin ortalama 0,90'ın üzerine çıktığı görülmüştür. Elde edilen değerler PoW ile karşılaştırıldığında blok süresinin daha kararlı olduğu ve blokzincirinin adem-i merkeziyetçiliğinin daha yüksek olduğu gözlemlendi. Bu sayede blokzincir sistemindeki yüksek donanımlı düğümlerin ağa hakim olması engellenmiştir. Böylece düşük donanımlı düğümlerin blok zincirinde blok oluşturma hakkına sahip olması sağlanmıştır.

Anahtar Sözcükler: Blok Zinciri, Optimizasyon, Genetik Algoritma, Gezici Satıcı Problemi, Fikir Birliği Algoritması

ABSTRACT

DEVELOPMENT OF A NEW ARTIFICIAL INTELLIGENCE-SUPPORTED APPROVAL MECHANISM IN BLOCK CHAIN: OPTIMIZATION-BASED APPROVAL MECHANISM (PoO)

Fatih Kürşad GÜNDÜZ
Düzce University
Graduate School, Department of Electrical-Electronics and Computer Engineering
Doctoral Thesis
Supervisor: Assoc. Prof. Dr. Serdar BİROĞUL

December 2023, 91 pages

Blockchain systems are among the popular technologies that have emerged recently. As a decentralized system, blockchain technology has offered many solutions and caused many problems related to these solutions. One of the most important problems is that it performs very intensive hash calculations to create a new consensus block, reducing its efficiency depending on the time. In this study, a new model is proposed to avoid Proof of Work (PoW), which redirects the calculations made to create blocks to optimization algorithms. The proof mechanism proposed in this study is called Proof of Optimization (PoO). The Traveling Salesman Problem (TSP) was introduced into the system designed to solve optimization algorithms. Nodes are asked to solve the TSP at specific iterations and populations. As a result, nodes are asked to create blocks with the obtained fitness, density and time values. PoO and PoW consensus algorithms have been subjected to an experimental comparison in the system. Test results show that the block creation time of the PoO consensus model varies between 2 s for the least urban dataset solution and 60 s for the most urban dataset solution. The difficulty levels of the problems can be adjusted by changing the number of cities in the TSP evaluated in this study. In this way, the problem of creating blocks on the network can be made more difficult or easier at any time. As a result of experimental analysis, it was observed that the decentralization, which is the percentage of block creation among miners in the blockchain, reached a more stable value and the fairness index increased above 0.90 on average. When the obtained values were compared to PoW, it was observed that the block time was more stable and the decentralization of the blockchain was higher. In this way, highly equipped nodes in the blockchain system are prevented from dominating the network. Thus, low-equipped nodes have the right to create blocks in the blockchain.

Keywords: Blockchain; Optimization; Genetic Algorithm; Traveling Salesman Problem; Consensus Algorithm

EXTENDED ABSTRACT

DEVELOPMENT OF A NEW ARTIFICIAL INTELLIGENCE-SUPPORTED APPROVAL MECHANISM IN BLOCK CHAIN: OPTIMIZATION-BASED APPROVAL MECHANISM (PoO)

Fatih Kürşad GÜNDÜZ
Düzce University
Graduate School, Department of Electrical-Electronics and Computer Engineering
Doctoral Thesis
Supervisor: Assoc. Prof. Dr. Serdar BİROĞUL
December 2023, 91 pages

1. INTRODUCTION

Blockchain is a relatively new technology that has gained traction in recent years. Satoshi Nakotomi originally mentioned Bitcoin in his paper "Bitcoin: A Peer-to-Peer Electronic Cash System" in 2008. The following are the fundamental principles and components of blockchain.

The following are the fundamental principles and components of the blockchain:

- Smart Contracts: These contracts operate automatically, are unchangeable, and are visible.
- Shared Ledger: This is the ledger where all blockchain transactions are recorded.
- Miner: A participant who allows new blocks to be added to the blockchain.
- Block: A data structure that contains transactions.
- Block Header: the part of the block containing some information.
- Digest: A cryptographic value that serves as a summary of blocks and transactions.
- Timestamp: shows when block was created.
- Nonce: A random integer used by miners.
- Difficulty: A value showimh how hard it is to make a block.
- Transaction: A transaction in the blockchain.
- Node: A participant of the blockchain network.
- Consensus Algorithms: In a distributed computer setting, consensus methods are used to keep the network stable and build trust among people who aren't familiar with each other. Consensus, privacy, speed, and access are some of the major goals of

these programs. Consensus algorithms have to deal with problems and think about things like security, scale, and how to use energy efficiently. Among these are Proof of Work (PoW), Proof of Stake (PoS), Byzantine Fault Tolerance (BFT), and Proof of Activity (PoAc).

One of the most important things to happen in the last century is blockchain. There are a lot of studies that look at how to combine blockchain and AI. Most of these studies are about how to combine these two technologies, what benefits they can offer, and what problems they might face.

2. MATERIALS AND METHODS

Artificial intelligence technology has garnered significant attention in recent times due to its advancements and potential applications. There are several subfields within this large subject of technology, including machine learning, deep learning, and neural networks. Distributed ledger technology, or blockchain technology, is a kind of data structure that stores digital data in an immutable manner. Particularly in fields like supply chain management, financial transactions, and smart contract automation, this technology offers a lot of promise. Smart contracts are blockchain-based programs that operate autonomously. These contracts start automatically and carry out certain tasks when certain requirements are satisfied. Consensus algorithms facilitate agreement-making among members of a blockchain network. These algorithms are built using a variety of techniques, most notably Byzantine Fault Tolerance (BFT), Proof of Work (PoW), and Proof of Stake (PoS). This thesis presents an analysis of the benefits and possible application areas that may be realized by merging these two technologies, after a survey of the literature on the integration of blockchain and AI. The aforementioned drawbacks of PoW have been addressed by the suggested PoO approval process. Naturally, the framework we provide in the context of this thesis research is built with a wide range of optimization issues in mind and enables comparative analyses utilizing other AI algorithms in the solution method. This effort gives all other researchers the chance to work in this sector and serves as a foundation for disclosing the PoO process.

3. RESULTS AND CONCLUSON

The primary goals of the Proof of Optimum (PoO) consensus method are twofold. First, in order to achieve better transaction throughput, the block time should be controlled more tightly than using the Proof of Work (PoW) consensus mechanism. A second goal is to stop

a computer with a powerful hardware setup from controlling the blockchain, which would increase decentralization.

Giving the consensus nodes the responsibility of solving the tsp (Traveling Salesman Problem) is the primary goal of the data nodes. Iteration, population, crossover, mutation, optimum solution, and reward are among the ideas covered in the tsp job that the data sink provides. The winning consensus drop with the greatest generalization performance receives the prize, which is transferred from the data drop's account to a virtual reservoir account. The time difference is smaller than the amount of time needed to solve an optimization problem, even when the deductions in a distributed system do not have precisely synchronized clocks. A competitively generated block of nodes contains the data of the solved issue. In reality, consensus nodes—also referred to as miners—are the network's labor force. Miners compete to complete the tsp tasks assigned by the data drops in order to earn rewards. To complete tsp tasks, nodes use genetic algorithms. PoO's primary goal is to highlight problem-solving skills rather than requiring large amounts of processing power to verify blockchain transactions. PoO uses the concepts of genetic algorithms to accelerate and improve the energy efficiency of blockchain data processing. Blockchain networks can now operate more quickly and efficiently with less resources thanks to this. PoO furthermore provides an alternative method to PoW and PoS. PoO focuses on the capacity to solve a problem using genetic algorithms, while PoS mining demands possessing a certain quantity of bitcoin instead of the predicted amount of power.

4. RESULTS

In specific iterations and populations, nodes are requested to solve the TSP. As a consequence, nodes are requested to build blocks based on the fitness, density, and time values acquired. In the system, the PoO and PoW consensus algorithms are compared experimentally.

The test results reveal that the PoO consensus model's block construction time ranges from 2 seconds for the least urban dataset solution (ulysses22) to 60 seconds for the most urban dataset solution (gr666). Furthermore, experimental evaluations suggest that decentralisation, which is the proportion of block generation among blockchain miners, has achieved a more stable value, and the fairness index has above 0.90 on average.

PoO has two primary goals. To begin, compared to PoW, the block time is kept under control, resulting in better transaction throughput. Second, preventing a computer with a

powerful hardware configuration from dominating the blockchain. A more decentralized structure is therefore advocated.

1. GİRİŞ

Blokzincir teknolojisi son yılların popüler teknolojisidir. Satoshi Nakotomi tarafından 2008 yılında Bitcoin: A Peer-to-Peer Electronic Cash System isimli makaleyle ilk kez ismi duyulmuştur. Blokzincir üyesi olduğu kullanıcılarının (düğümlerin) her birinin bir kayıt defterine sahip olduğu dağıtık bir sistemdir. Bu sistemde bir işlemin tüm kopyaları kayıt defterlerinde tutulur. Kayıtlardan bir tanesi değiştirilmek istendiğinde diğer kopyaların olduğu defterlerle karşılaştırılma yapılıp sistemde değişikliğe izin verilmemektedir. Sistem bu sayede manipüle edilememektedir. Sistem merkezi bir sistem yerine kullanıcıların güvenliğini paylaştığı dağıtık bir sistem önermektedir. Kullanıcıların her biri sistemin güvenliğinde rol almaktadır [1].

Blokzincir kavramı dağıtık veritabanı teknolojisine benzemektedir. Kısaca tanımlamak gerekirse konsensüs yoluyla işlerin dağıtık olan veritabanlarına eklenmesidir. Dağıtık veri tabanları kullanıcıların(düğüm) adı verilen IP adresi alıp blokzincir ağına bağlanabilen masaüstü notebook ve mobil cihazlardır. Blokzincirde işlemlerin gerçekleşmesi ve veritabanında tutulabilir hale gelmesi için geçerli ve doğrulanmış bir bloğun oluşturulmuş olması gereklidir. Sonraki adımda ise blok diğer node yapılara dağılır. Bloğu alan her node bloğu doğrulayıp işlemleri çalıştırır. İlgili blok node yapısındaki zincirin son halkasına eklenir [2].

Blokzincir ağdaki kullanıcıların kendi kendine hem fikir olduğu bir nevi ekosistemle çalışmaktadır. Bu önemli işlevi mutabakat yoluyla işlemlerin doğruluğunu kontrol eden konsensüs mekanizmasıdır. Satoshi Nakamoto'nun blokzincirdeki ana yapısı olan fikir birliği, merkezi olmayan bir mekanizmadır. Bu mekanizma tüm blokların söz konusu mutabakata varmasını sağlar. Böylelikle para birimi, işlemler ve ödemeler merkezi sisteme dayanmadan yapılabilmektedir [1,3].

Blokzincir, kripto para ve akıllı sözleşmeler başta olmak üzere bir çok yeni yapının arka planındaki uygulanan teknolojidir. Bu çalışmada blokzincir ağını, akıllı bir konsensüs algoritması ile yönetilen bir dağıtık defter olarak sunulmuştur. Bu çalışmada ortaya konan yeni onay mekanizmasının adı Proof of Optimum (PoO) yapısıdır. Bu mekanizmanın detayları ilerleyen bölümlerde anlatılmıştır. Proof of Work (PoW) tabanlı bir blokzincir

ağında düğümler bloğu oluşturmak için bir kriptografik bulmaca çözer ve rekabet ederler. En önemli blokzincir projeleri arasında sayılabilecek Bitcoin ve Ethereum PoW tabanlı kriptografik bulmaca problemi dağıtan ve bireysel olarak çözen bir yapıyı kullanır. Bu yapı büyük hesaplama kaynakları gerektirmektedir. Bu yüzden PoW kullanan blok zincir ağları yüksek enerji tüketimine neden olmakta ve enerjinin çoğunu bulmacayı çözmeye harcamaktadır. Bunun yanında güvensiz bir fikir birliği sağlamakla beraber, verimi ciddi anlamda düşmektedir [4].

Blokzincirin popülerliği, hem teknik hem de pratik avantajlardan kaynaklanmaktadır. Teknik olarak, blokzincirin değiştirilemez yapısı, bir kez onaylandığında verilerin değiştirilmesinin ya da silinmesinin neredeyse imkansız olmasını sağlar. Bu durum, taraflar arasında yüksek bir güven seviyesi oluşturur [3].

Dijital devrimin bir parçası olarak blokzincir; bankacılıktan enerji dağıtımına, eğitimden sağlık hizmetlerine kadar birçok sektörde dönüşümcü bir rol oynamaktadır. Örneğin, tedarik zinciri yönetiminde blokzincir, ürünlerin orijinal kaynağından tüketiciye kadar olan yolculuğunu şeffaf bir şekilde izlemeyi mümkün kılar. Bu, sahte ürünlerin önlenmesi, geri çağırma işlemlerinin daha etkili bir şekilde yürütülmesi ve tedarik zincirinin genel verimliliğinin artırılması gibi avantajlar sunar[5].

Ancak, her teknolojik yenilikte olduğu gibi, blokzincirin de kendi zorlukları vardır. Ölçeklenebilirlik, enerji tüketimi, kullanıcı gizliliği gibi konular, blokzincirin yaygın kabulünü ve uygulanmasını engelleyen faktörlerden bazılarıdır [6].

Çalışmanın amacı, PoW'da harcanan hesaplama gücünü PoO'ya aktararak kullanmayı amaçlayan, yeni bir blok zinciri mutabakat algoritması önerilmesidir. Böylelikle ölçeklenebilirlik ve adem-i merkeziyetçilik gibi konulara çözüm sunulmuştur. Aynı zamanda gerçek dünya problemlerinin çözümüne katkıda bulunmaktır.

Blokzincir teknolojisi kısmında blokzincir detaylı bilgiler verilmiştir. Aynı zamanda bu teknolojinin sahip olduğu avantajlar ve dezavantajlardan bahsedilmiştir. Materyal ve yöntem kısmında Yapay zeka (YZ), genetik algoritma, Gezgin Satıcı Problemi(GSP) ve Proof of Optimum(PoO) ilgili teorik bilgiler paylaşılmıştır. Bulgular ve tartışma kısmında ise PoO ile ilgili genel detaylardan bahsedilmiştir. PoO algoritmasının genel yapısı, oluşturulan algoritma yapısı ve simulasyon ortamının detaylarından bahsedilmiştir.

2. BLOKZİNCİR TEKNOLOJİSİ

2.1. AKILLI KONTRATLAR

Akıllı kontratlar dağıtık veri tabanında bir işlem yapıldığı andan itibaren çalıştırılan, arabulucu ve anlaşmazlık olmadan yapılan sözleşmelerdir. Yani, blok zincir teknolojisi üzerinde, otomatik olarak yürütülen ve denetlenen, kendi kendine çalışabilen programcıklardır. Akıllı kontratlar, belirli koşullar yerine getirildiğinde otomatik olarak tetiklenebilir, anlaşmaları veya diğer işlemleri otomatik olarak yürütür. Akıllı kontratlar, güvenilir, şeffaf ve değiştirilemez bir şekilde işlem yapar ve aracıları ortadan kaldırarak işlemleri hızlandırabilir ve maliyetleri düşürürler [7].

2.2. AKILLI KONTRATLARIN TEMEL ÖZELLİKLERİ

Akıllı kontratlar, belirli şartlar yerine getirildiğinde otomatik olarak yürürlüğe giren, özel kodlanmış sözleşmelerdir. Bu otomatik çalışma biçimi, çok çeşitli işlemleri ve eylemleri basit, şeffaf ve güvenli bir şekilde gerçekleştirmek için büyük bir potansiyele sahiptir. Otomasyonun arkasındaki temel fikir, belirli bir olayın veya koşulun meydana gelmesi durumunda belirli bir eylemin gerçekleştirilmesidir. Örneğin, bir kişi bir diğerine belirli bir tarihte belirli bir miktarda kripto para göndermeyi taahhüt ederse, belirtilen tarihte akıllı kontrat otomatik olarak transferi gerçekleştirir [8].

Aynı şekilde, bir e-ticaret işlemi için bir akıllı kontrat oluşturulabilir. Alıcı, ürünü satın alır ve ödemesi akıllı kontratta tutulur. Ürün alıcıya ulaştığında, bir kargo takip numarası veya diğer bir doğrulama aracılığıyla kontrat tetiklenir ve ödeme satıcıya gönderilir. Bu, güvenli, otomatik ve aracısız bir işlemi garantiler[7].

Akıllı kontratların otomasyonu, işlemleri hızlandırabilir, maliyetleri azaltabilir ve taraflar arasında güven oluşturabilir. Ancak, bu otomasyonun doğru bir şekilde çalışması için akıllı kontratın doğru bir şekilde yazılması ve test edilmesi esastır. Hatalı bir kod ya da öngörülmeyen bir durum, istenmeyen sonuçlara yol açabilir. Bu nedenle, bir akıllı kontratın yazılmasında ve uygulanmasında dikkatli olunması gerekmektedir. Akıllı kontratlar, belirli koşullar karşılandığında otomatik olarak işlem yapar [7].

2.2.1. Güvenilirlik

Akıllı kontratlar, blokzincir teknolojisi üzerine inşa edildiğinden, bu teknolojinin

sunduğu güvenilirlik avantajlarından yararlanırlar. Blokzincir, kriptografik olarak güvence altına alınmış, dışarıdan değiştirilemez ve dağıtık bir yapıya sahip olduğundan, bu yapı akıllı kontratların güvenilirliğine doğrudan katkıda bulunur.

Bir akıllı kontrat blokzincir üzerine yüklendiğinde, bu kontratın kodu ya da içeriği sonradan değiştirilemez. Bu, sözleşme şartlarının her iki taraf için de sabit ve değişmez olduğu anlamına gelir. Ayrıca, bu kontratlar şeffaf bir şekilde herkes tarafından incelenebilir. Bu da, sözleşme şartlarına uyulup uyulmadığının herkes tarafından görülebilmesini sağlar.

Akıllı kontratların otomatik çalışma prensibi, insan müdahalesi olmadan işlem yapmalarını sağlar. Bu da, insan kaynaklı hataların ya da kasıtlı müdahalelerin önüne geçer [9].

2.2.2. Şeffaflik

Akıllı kontratlar, genellikle halka açık blokzincirler üzerinde çalışır. Bu, herkesin blokzincir üzerindeki her işlemi ve akıllı kontratı görüntüleyebilmesi anlamına gelir. Akıllı kontratın kodu, çalışma şekli ve işlemleri, genellikle herkes tarafından incelenebilir. Bu da, katılımcıların ya da ilgilenen diğer kişilerin, kontratın ne yaptığını ve nasıl çalıştığını anlamasına yardımcı olur [9].

Bu şeffaflık, akıllı kontratın tarafları arasında güven oluşturur. Taraflar, kontratın koşullarını ve nasıl çalıştığını görebilirler. Böylece, sözleşmenin şartlarının adil ve beklenildiği gibi olduğundan emin olabilirler. Ayrıca, bir anlaşmazlık durumunda, taraflar sözleşmenin ne dediğine ve nasıl çalıştığına dair şeffaf bir kaynağa başvurabilirler.

Ancak şeffaflık, her zaman istenen bir özellik olmayabilir. Özellikle özel ve hassas bilgilerin söz konusu olduğu durumlarda, tüm detayların herkes tarafından görülebilmesi sorun olabilir. Bu nedenle, bazı akıllı kontrat uygulamaları, özel blokzincir ya da bazı bilgilerin gizlenmesine olanak tanıyan yapılar kullanılarak gerçekleştirilir [9].

2.2.3. Küresel Erişim

Akıllı kontratlar, blokzincir sayesinde küresel bir ağ üzerinde çalışmaktadır. Böylece bu kontratların neredeyse her yerden erişilebilir ve kullanılabilir olması sağlanır. Nerde olursa olsun internet erişimi varsa, bir blokzincir ağındaki akıllı kontratlara erişim sağlamak mümkündür [9].

Bu erişim sayesinde, dünyanın farklı yerlerinden insanlar arasında işlem yapmak ve sözleşmeler oluşturmak çok kolaylaşır. Bu durum geleneksel bankacılık sistemlerinin getirdiği maliyetlerden kaçınılmasına yardımcı olur [10].

Ayrıca, akıllı kontratların doğası sayesinde, bu tür işlemler hızlı bir şekilde gerçekleştirilmektedir. Şartlar yerine getirildiğinde, kontrat otomatik olarak yürürlüğe girer ve ilgili işlemler çalışır. Bu da zaman dilimi farklılıkları ve diğer birtakım sorunlar nedeniyle oluşabilecek gecikmeler engellenebilir [4].

2.2.4. Ortak Kayit Defteri

Ortak kayıt defteri birden fazla taraf arasında paylaşılan ve sürekli güncellenen bir veri kaydıdır. Böylesi bir defter, blokzincir ağındaki çok sayıda düğüm tarafından kullanılır. Ancak, Ortak kayıt defterinin birtakım zorlukları ve sınırlamaları bulunmaktadır. Özellikle ölçeklendirme, gizlilik ve enerji tüketimi gibi konularda bazı teknik aksaklıklar mevcuttur. Yeni bir teknoloji olması nedeniyle de, yasal çerçevelerin eksikliği veya belirsizliği, bu teknolojinin geniş çapta benimsenmesini zorlaşmaktadır [5].

2.2.5. Ortak Kayit Defterinin Temel Özellikleri

2.2.5.1. Dağıtılmış Yapı

Birçok bağımsız bilgisayarın bir ağ üzerinde bir araya gelerek tek bir bütünsel sistem gibi çalıştığı bir yapıdır. Bu bilgisayarlar, genellikle düğüm olarak adlandırılır ve birbirleriyle sürekli iletişim halindedirler. Amaç, dağıtık bilgisayarların koordineli bir şekilde çalışarak, bireysel yeteneklerinin ötesinde bir performans ve işlevsellik sağlamasıdır [5,11].

Bu yapı, genellikle merkezi olmayan bir sistem olarak karşımıza çıkar. Merkezi bir otorite veya kontrol noktası olmaksızın, tüm düğümler eşit yetkiye sahip olabilir ve kendi başlarına kararlar alabilirler. Bu dağıtılmış doğa, sistemin genel dayanıklılığını artırır, çünkü tek bir noktada arıza, tüm sistemin çökmesine neden olmaz[11].

Dağıtılmış sistemler, ölçeklenebilirlik, yüksek erişilebilirlik ve dayanıklılık gibi avantajlara sahiptir. Ancak, bu tür sistemlerin tasarımı, kurulumu ve yönetimi karmaşıktır. Ayrıca bu sistemlerde koordinasyon, veri tutarsızlığı ve güvenlik gibi bazı zorluklarla da karşılaşılabilir. Ancak, doğru şekilde uygulandığında, dağıtılmış yapılar, geleneksel merkezi sistemlere göre birçok avantaj sunar [11].

2.2.5.2. Yetkilendirme

Blokzincir yetkilendirme, blokzincir teknolojisi kullanılarak gerçekleştirilen bir kimlik doğrulama ve erişim kontrol sürecidir. Bu süreç, kullanıcıların, cihazların veya sistemlerin belirli bir blokzincir ağı veya uygulamasına güvenli bir şekilde erişimini ve etkileşimini yönetmeye yardımcı olur. Blokzincir yetkilendirme, genellikle şifreleme, dijital imzalar ve akıllı kontratlar gibi teknikler kullanılarak gerçekleştirilir [6].

2.2.5.3. Konsensüs

Konsensüs, bir ağdaki veya sistemindeki tüm katılımcıların, bir konuda ortak bir anlaşmaya varması sürecidir. Kripto para ve blokzincirde, konsensüs algoritmasının amacı ağdaki tüm düğümlerin bir bloğun veya işlemin geçerliliği konusunda anlaşmaya varmasını sağlar. Bu durum merkezi olmayan sistemlerde güvenlik ve bütünlüğü sağlamak için önemlidir [3].

2.2.6. Blokzincir Uygulama Alanlari

2.2.6.1. Finans ve Bankacılık

Blokzincir, finansal işlemlerin hızlandırılmasına ve maliyetlerin azaltılmasını sağlar. Kripto paralar, blokzincirin bu sektördeki en çok bilinen uygulamalarından biridir. Ama bunun yanısıra uluslararası para transferleri ve tedarik zinciri finansmanı gibi alanlarda da kullanımları mevcuttur. Bazı bankaların blokzinciri kullanarak kendi dijital paralarını oluşturma çalışmaları mevcuttur [3,5,10].

2.2.6.2. Tedarik Zinciri ve Lojistik

Ürünlerin kaynağını ve tedarik zinciri boyunca nasıl hareket ettiğini takip etmek için blokzincir kullanılması yaygın bir uygulama olarak kullanılabilir. Böylece, sahtecilik önlenir, ürün kalitesini ve güvenliğini doğrulamaya yardımcı olunur. Basitce, bir tüketici, bir ürünün gerçek olup olmadığını veya sürdürülebilir bir kaynaktan gelip gelmediğini kontrol edebilir [10].

2.2.6.3. Sağlık Sektörü

Hasta kayıtlarının güvenli ve özel bir şekilde saklanması, paylaşılması ve doğrulanması için blokzincir kullanılabilir. Üstelik klinik deneyler, ilaç izlenebilirliği ve epidemiyolojik araştırmalar için de kullanılmaktadır [3,10].

2.2.6.4. Emlak Sektörü

Blokzincir, emlak işlemlerini basit, şeffaf ve güvenilir hale getirebilir. Akıllı kontratlar sayesinde, emlak satın alma işlemleri daha hızlı ve kolay bir şekilde gerçekleştirilebilir. Ayrıca, tapu kayıtlarının blokzincir üzerinde saklanması, bu kayıtların değiştirilmez ve doğrulanabilir olmasını sağlar [2,10].

2.2.6.5. Oylama Sistemleri

Blokzincir, seçimlerin ve referandumların güvenli, şeffaf ve manipülasyona karşı dirençli bir şekilde gerçekleştirilmesine olanak tanır. Bu teknoloji, oyların doğruluğunu ve bütünlüğünü garanti ederek demokratik süreçleri güçlendirebilir.

Yukarıda kısaca belirtilen bu uygulama alanları, blokzincirin potansiyelini ve sektörler arası uygulanabilirliğini göstermektedir. Teknolojinin olgunlaşması ve adaptasyonun artmasıyla birlikte, bu ve diğer sektörlerdeki uygulama alanlarının sayısının artması beklenmektedir [10].

2.3. BLOKZİNCİR KAVRAMLARI

Blokzincir, sistemine dahil olan bilgisayarlardaki verilerin diğer bilgisayarlarında görebildiği merkezi olmayan dağıtık bir sistemdir. Blokzinciri verilerin değiştirilmesine izin vermeyen bir önceki veri özetini bir sonraki veride tutan bir yapısı vardır. Bu özellikle blokzincir sisteminde katılımcıların veritabanı kayıtlarını değiştirilmesine izin verilmemektedir [10].

2.3.1. Düğüm

Blokzincirde düğüm demek blokzincir ağına katılan ve blokzincirin kopyasını tutan herhangi bir cihaz demektir. Ağın temel bileşenleri olan düğümler, blokzincirin çalışmasında önemli bir role sahiptir. Blokzincirin dağıtık bir yapıda olmasını sağlarlar. Sistemde işlemler ve blokların doğrulanmasında önemli bir rolleri vardır. İşlemleri ve blokları ağ üzerindeki diğer düğümlerle paylaşırlar. Bu sayede bir merkezi otorite ya da tek bir veritabanına bağlı kalmadan veri bütünlüğünü ve doğruluğunu koruma sağlanmaktadır. Düğümler arasındaki bu ağ, blokzincirin merkezsiz, şeffaf ve değiştirilemez olmasını sağlar. Bu, blokzincir teknolojisinin temel avantajlarından biridir [4]. Düğüm türleri aşağıda listelenmiştir.

2.3.1.1. Tam Düğüm

Tam düğüm, blokzincir ağında çalışan bir cihazdır ve blokzincirin tüm geçmişini saklayabilme özelliği bulunur. Bu özelliği sayesinde, ağ üzerinde gerçekleşen her işlemi ve bloğu bağımsız olarak doğrulayabilir. Bu özellik, tam düğümün blokzincirin bütünlüğü ve güvenliği için kritik bir öneme sahip olmasını sağlar. Blokzincire gönderilen her işlem ve blok, tam düğüm tarafından belirlenen bazı kurallara uygun olup olmadığına bakılarak doğrulanabilir. Eğer bir işlem veya blok bu kurallara uymazsa tam düğüm tarafından reddedilecektir[2].

2.3.1.2. Hafif düğüm

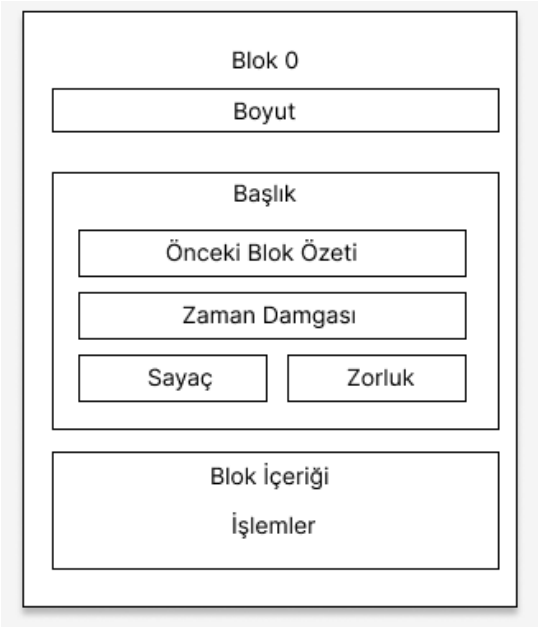
Hafif düğüm, blokzincir ağında çalışan bir cihazdır. Ancak tam düğümler gibi blokzincirin tüm geçmişini saklamaz. Sadece belirli bilgilere erişir ve bu bilgileri doğrulamak için tam düğümlere güvenir. Hafif düğümler, özellikle mobil cihazlar için vardır ve genellikle sadece blok başlıklarını saklar. Bu başlıklar sayesinde işlemlerin geçerliliğini hızlı bir şekilde kontrol edebilir. Bu, hafif düğümün daha az depolama alanı ve bant genişliği kullanmasını sağlamaktadır [9].

2.3.2. Madenci

Madenciler blokzincir sistemindeki işlemleri doğrulayan ve bu doğrulama işlemi karşılığında ücret olarak kriptopara alan düğümler olarak anılırlar. Madenciler doğrulama işlemlerini yapabilmek için karmaşık matematik problemlerini çözmek zorundadır. Problemin çözülmesiyle blokzinciri ağına yeni bir blok eklenir. Bu işlem sayesinde sistemde devamlılık ve güvenilirlik sağlar [5,10].

2.3.3. Blok

Blokzincir üzerinde yapılmış işlemlerin kayıtlarının ve onaylarının tutulduğu bir veri formatıdır. Her bir blok birbirine zincir şeklinde bağlıdır. Zincir bağı sayesinde blokzincir manipüle edilemez. Bloklar üzerinden işlemler herkes tarafından görüntülenebilir. Şekil 2.1'de örnek blok yapısı verilmiştir [4]. Blok zincirdeki ilk blok genesis blok olarak adlandırılır ve genesisten başlayarak birbiri özetlerini önceki blok özetlerinde tutarak zincir oluşturulur [5].



Şekil 2.1. Blok yapısı

2.3.4. Blok Başlığı

Blok başlığı, blokzincirdeki bir bloğun tanımlayıcı özelliğidir. Her blok içerisinde, o bloğun kendine özgü bilgilerini bulunduran bir başlık vardır. Bu başlıkta, bir önceki bloğun kriptografik özetini içeren bir özet değeri vardır. Bu da blokların sıralı bir şekilde birbirine bağlamaktadır.

Aynı zamanda bloğun ne zaman oluşturulduğunu gösteren bir zaman damgası da bulunmaktadır. Madencilik süreciyle ilgili, başlıkta bir zorluk değeri ve bir nonce değeri de içerir. Zorluk hedefi, yeni bir bloğun kabul edilebilmesi için gereken özet değerinin karmaşıklığını gösterirken, nonce değeri madencilerin bu zorluk hedefine ulaşmak için kullandığı rastgele bir numaradır [3].

2.3.5. Özet

Kısaca bir blokta yer alan tüm işlemlerin kriptografik bir özetidir. Bu özet, bloktaki işlemlerin değişip değişmediğini hızlı ve etkili bir şekilde kontrol edebilmek için kullanılır. Bunun içinde genellikle Merkle Ağacı adında bir veri yapısı kullanılır. Merkle Ağacı, işlemlerin hash değerlerini alarak ağaç yapısında bir özet oluşturur. İşlem verilerinin her biri özetlenir ve bu özet değerleri çiftler halinde tekrar özetlenerek üst seviyelere doğru ilerlenir. En üstte tek bir özet değeri kalana kadar bu işlem devam eder. Bu son özet, Merkle ağacı köküdür ve blok başlığında tutulmaktadır. Tüm işlemlerin doğruluğunu ve bütünlüğünü hızlı bir şekilde doğrulamak için kullanılır. Eğer bir bloktaki

tek bir işlem bile değiştirilirse, bu değişiklik tüm Merkle Ağacını ve dolayısıyla Merkle Ağacı Kökü'nü etkiler. Bu, blokzincirin değiştirilemez ve güvenilir bir yapıda olmasına katkıda sağlayacaktır [2].

2.3.6. Zaman Damgası

Zaman damgası, bir olayın ya da verinin ne zaman gerçekleştiğini belirten bir bilgidir. Blokzincirde, zaman damgası genellikle her bloğa eklenir ve ilgili bloğun ne zaman oluşturulduğunu belirtir. Blokzincirin kronolojik sırasını doğrulamak genel olarak bir önemli veridir.[1].

Blokzincirdeki zaman damgası, genellikle bloğun madencilik süreci tamamlandığında eklenir. Bu, ağı kullanan tüm katılımcıların, belirli bir işlemin ya da bloğun ne zaman gerçekleştiğine dair ortak bir anlayışa sahip olmalarını sağlar. Zaman damgasının kullanılmasının bir diğer avantajı da blokzincirdeki işlemlerin sırasını koruma ve değişikliklerin ya da çakışmaların tespit edilmesine yardımcı olmasıdır. Eğer iki madenci aynı anda iki farklı bloğu aynı pozisyona eklemeye çalışırsa, bu blokların zaman damgaları sayesinde hangi bloğun önce oluşturulduğunu belirleyebilir ve bu bilgiye göre hangi bloğun kabul edileceğine karar verilebilir [4].

2.3.7. Nonce

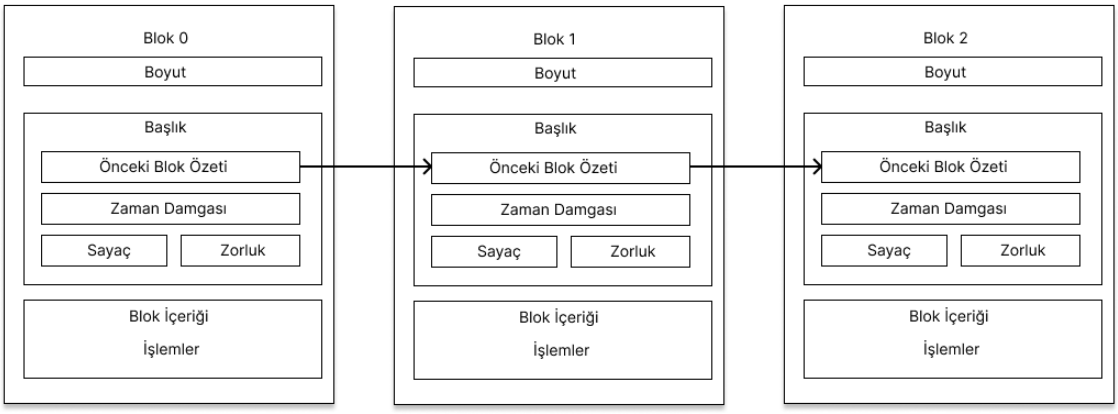
Nonce verisi, genellikle kriptografik işlemler ve özellikle blokzincir madenciliğinde karşımıza çıkan bir kavramdır. "Number used once" yani "bir kez kullanılan sayı" anlamına gelmektedir. Madencilerin yeni bir blok oluşturabilmesi için belirli bir zorluk hedefine ulaşması gerekir ve belirli bir bloğun özet değerinin belirli bir değerin altında tutulur. Madenciler, blok başlığındaki bilgileri ve bu nonce değerini özetini ve elde edilen sonucu belirlenen zorluk hedefine uygun mu değil mi diye kontrol ederler. Eğer değilse, nonce değerini değiştirir ve tekrar deneyerek doğru özet değerini bulmaya çalışırlar [4].

Nonce'un amacı, madencilik sürecinin zorlaştırılmasıdır. Bu sayede ağın güvenliğini artırılır. Madencilerin bu zorluk hedefine ulaşması için gereken süre, blokzincirin protokolüne göre belirlenen süreye uyması için nonce değeri sürekli olarak değiştirilebilir. Bu durum, madencinin doğru nonce değerini bulana kadar devam eder. İşte bu durum PoW olarak adlandırılır [5].

2.3.8. Zorluk

Zorluk değeri, blokzincir madenciliğinde bir bloğun ağa eklenmeden önce karşılaması gereken bir koşulu ifade etmektedir. Özellikle PoW tabanlı kripto paralarda, zorluk hedefi madenciliğin ne kadar zor ya da kolay olacağını tanımlar. Kısaca, madenciler yeni bir blok oluşturmak için belirli bir matematiksel bulmacayı çözmeye çalışırlar. Bu bulmacanın çözümü sonucunda, bloğun içeriği ve diğer bazı bilgilere dayalı olarak bir özet değeri oluşturulur [1,2].

Zorluk hedefi, Örneğin PoW; için genellikle belirli bir sayıda sıfır ile başlayan özet değerlerini üretme olasılığını belirlemek için kullanılır. Zorluk hedefi 10 sıfırla başlayan bir özet değeri için, madencinin bu hedefi karşılaması gereken birçok farklı kombinasyonu demektir. Blokzincir ağı belirli bir zaman aralığı içinde yeni blokların eklenme hızını sabit tutmayı amaçlar. Eğer madenciler blokları çok hızlı oluşturuyorsa, zorluk hedefi arttırılır, böylece madenciliğin zorluğu artar ve blok eklenme hızı düşer. Tersi bir durumda, yani blokların eklenmesi çok yavaşsa, zorluk hedefi azaltılır ve madencilik daha kolaylaştırılır. [2,3].



Şekil 2.2. Örnek blokzincir yapısı

2.3.9. Transaction

Blokzincirde bir işlem, belirli bir veri transferini temsil etmektedir. En yaygın kullanımı kripto paraların transferidir. Bir kişi bir miktar kripto parayı başka bir kişiye gönderirken bu bir işlem olarak kaydedilir. İşlem, gönderenin özel anahtarıyla dijital olarak imzalanarak, işlemin gerçekten o kişi tarafından yapıldığını ve işlem içeriğinin sonradan değiştirilmediğini garanti eder. İşlemler girişler ve çıkışlardan oluşur. Giriş, işlemin kaynaklandığı önceki işlemleri temsil ederken, çıkış ise yeni sahiplik durumunu belirtir. Blokzincir ağındaki düğümler, yeni bir işlemi aldıklarında bu işlemi doğrularlar. Eğer

işlem geçerli ise, madenciler bu işlemi yeni bir bloğa eklerler. İşlem bir bloğa eklendiğinde ve ağın geri kalanı tarafından kabul edildiğinde, bu işlem onaylanmış kabul edilir. Onaylandıktan sonra, işlemi geri almak veya değiştirmek neredeyse imkansızdır. Bu da blokzincirin güvenilirliğini sağlar [7].

2.3.9.1. İşlem Özeti

İşlem özeti, blokzincirdeki bir işlemin benzersiz bir tanımlayıcısıdır. Bir işlem oluşturulduğunda, işlem bilgisi bir hash fonksiyonundan geçirilir ve bu fonksiyon, belirli bir uzunlukta bir dizi karakter katarı üretir. Buna işlem özeti denir. Bu özet, işlemin içeriğinin değiştirilmesine karşı izin vermez. Yani işlemdeki herhangi bir bilginin değişmesi, hash değerinin tamamen farklı olmasına neden olur. İşte bu özellik sayesinde, işlemlerin bütünlüğü ve değiştirilemezliği sağlanır [3,4].

2.3.9.2. İşlem Boyutu

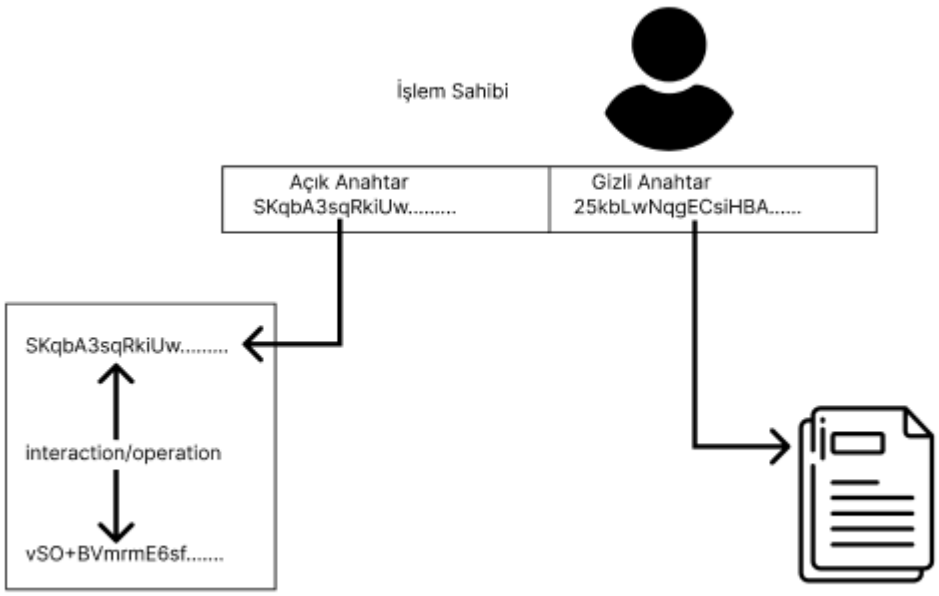
İşlem boyutu bir işlemin ne kadar boyutu olduğunu gösteren bir metriktir. İşlemde yer alan girişlerin, çıkışların ve diğer başlık gibi verilerin toplamının byte cinsinden ölçülmesiyle belirlenmektedir. İşlem ücretlerinin hesaplanmasında ve bloğun toplam boyutunu sınırlandırmada işlem boyutu önemli bir rol oynar. İşlemin boyutu, bu girişlerin ve çıkışların sayısına, bu giriş ve çıkışlarda saklanan veriye bağlı olarak değişir [1].

2.3.9.3. İşlem Giriş

İşlem girişi özellikle kripto para işlemlerinde, bir işlem gerçekleştirildiğinde bu içeriğin nereden geldiğini belirtmek için işlem girişleri kullanılır. Bu veri, çift harcamanın önüne geçmek ve işlemlerin doğruluğunu sağlamak için kullanılır. Bu sayede bir kullanıcının aynı fonları birden fazla kez harcamasının önüne geçilir[2].

2.3.9.4. İşlem Çıkış

İşlem çıkışı, blokzincirde bir işlemin parçası olan ve fonların kime ve ne kadar gönderildiğini belirten bileşendir. Bir işlemde, belirli bir miktar kripto parayı belirli bir alıcıya göndermek için bir ya da daha fazla işlem çıkışı oluşturulabilir. Bu şekilde, işlem çıkışları, fonların nereye gittiğini belirtir ve işlemin tamamlanmasını sağlar. Harcanmamış işlem çıkışları, gelecekteki işlemlerde girdi olarak kullanılabilir **[**7]. Aşağıda şekil 2.3'te örnek işlem yapısı görülmektedir.



Şekil 2.3. Örnek işlem yapısı

2.3.10. Blokzinciri Türleri

Blokzinciri kavramı 3 farklı kategoriye ayrılmaktadır.

2.3.10.1. Herkese açık blokzincir

Bu tür blokzinciler, katılımın ve erişimin herhangi bir kısıtlama olmaksızın gerçekleştiği blokzincir türleridir. Bu, şeffaflığı ve güvenliği artırırken, öte yandan madencilik adı verilen işlem doğrulama sürecini de zorlaştırabilir. Bu tür blokzincirde herkes bu blokzincirlere katılabilir ve işlem gerçekleştirebilir.

Bu sayede işlemler, herhangi bir merkezi otoritenin kontrolü altında olmadan gerçekleştirilir ve doğrulanır. İşlem doğrulama süreci, madencilik adı verilen bir süreçle gerçekleştirilir. Madenciler, karmaşık bulmacalar çözerek yeni bloklar oluşturur. Bu blokları blokzincirine eklerler. Bu işlem karşılığında, madenci düğümlere belirli miktarda kripto para birimi ödülü verilmektedir. Bu süreç, ağın güvenliğini artırır ve kötü niyetli faaliyetlerin engeller [2].

Herkese açık blokzincirlerin genellikle açık kaynaklı kodlara sahiptir. Bu sayede yazılımcılar, bu blokzincirlerin kodunu açıp inceleyebileceği, değişiklikler önerebileceği ve yeni sürümlere katkıda bulunabileceği anlamına gelir. Bu durum, topluluk tarafından sürekli bir denetim ve geliştirme sürecini beraberinde getirir. Bu herkese açık ve katılımcı

yapı, blokzincirin evrimini hızlandırmaktadır [11].

2.3.10.2. Konsorsiyum blokzincir

Konsorsiyum blokzincirler özel blokzincirlerdir. Belirli bir grup veya organizasyon tarafından kontrol edilirler. Bu tür blokzincirler, herkese açık blokzincirlerin tamamen merkezi olmayan yapısından farklıdır ve sadece belirli üyelerin veya kurumların izinli olarak katılabildiği yapıya sahiptir. Bu durum, sadece belirli katılımcıların doğrulama sürecine katılacağı anlamına gelir [2].

Bu tür blokzincirlerin en büyük avantajı, işlem hızı ve ölçeklenebilirliktir. Herkese açık blokzincirlerinden farkı, bu tür blokzincir ağlarında işlemlerin doğrulanması için genellikle tüm ağın katılımı gerekir. Ancak konsorsiyum blokzincirlerinde, sadece belirli doğrulayıcılar işlem onaylarını yaparlar. Böylece daha hızlı işlem süreleri elde edilir. Özellikle finans, tedarik zinciri yönetimi ve kurumsal işlemler gibi sektörlerde popülerdir. Bu konularda, şeffaflık ve güvenliği korurken aynı zamanda hızlı işlem sürelerine ve gizliliğe de ihtiyaç duyulur. Konsorsiyum blokzincirler, bu ihtiyaçları dengeli bir şekilde karşılayabilirler[2].

2.3.10.3. Özel blokzincir

Bu blokzincir türü sadece yetkili kullanıcıların bu sistemi kullanabildiği bir blokzincir türüdür. Sadece yetkili kullanıcılar bu sistemi kullanabilir. Daha merkezi bir yapıya sahiptirler. Çünkü sadece belli kullanıcı zümresine ait olanlar işlem yapabilmektedir [2].

Özel blokzincirler, belirli organizasyonlar tarafından kontrol edilirler. Sadece belirli bireylerin veya kurumların ulaşabildiği blokzincirlerdir. Genellikle ticari işletmeler veya kurumlar için iç kullanıma yöneliktir. Bu blokzincirlerin ana özelliği, katılımın özel izinle sağlanmasıdır. Bu durum, sadece yetkilendirilmiş kullanıcıların ağa ulaşabileceği ve işlemler yapabileceği anlamına gelir [2].

En büyük avantajlarından biri güvenlik ve gizliliktir. Dışarıdan bir saldırganın ağa zarar vermesi veya hassas verilere erişmesi çok daha zordur. Ayrıca, işlemlerin doğrulanması ve kaydedilmesi daha hızlıdır çünkü tüm ağın katılımını gerektirmeyecektir. Bu sayede işletmelerin özel blokzincirleri kendi iç süreçleri için özelleştirmelerine olanak tanır.

Bu tür blokzincirlerin bir diğer avantajı ise özelleştirilebilirlik ve esnekliktir. Bir işletme veya organizasyon, özel bir blokzinciri oluşturur, belirli iş süreçleri ve ihtiyaçları için tam olarak nasıl bir yapı istediklerini belirleyebilir. Bu durum, özellikle tedarik zinciri, finans

ve diğer özel sektör uygulamaları için çok uygundur. Ancak bu ortaya konan esneklik, aynı zamanda bu blokzincirlerin genel blokzincirlerin sunduğu tam merkezsizlik ve şeffaflık kuralına uymadığı anlamına da gelir [2]. Çizelge 2.1'de Türlerine göre blokzincir tablosu ve özellikleri verilmiştir.

Çizelge 2.1. Türlerine göre blokzincir tablosu ve özellikler

Parametreler	**Genel Blokzincir**	**Özel Blokzincir**	**Konsorsiyum Blokzincir**
İşlem Hızı (TPS)	7-30	100 ve üzeri	100 ve üzeri
Ölçeklenebilirlik	Zorlukla Duyarlı	İyi	Orta
Güvenlik	Yüksek	Yüksek	Yüksek
Erişim	Herkese Açık	Sınırlı	Sınırlı
İşbirliği	Tam Merkezsiz	Sınırlı	Kısıtlı
Veri Gizliliği	Genellikle Açık	Yüksek	Orta-Yüksek
İşlem Maliyeti	Değişken	Daha Yüksek	Daha Düşük
İşlem Onay Süresi	10-60 dakika	1 saniye-10dakika	1 saniye – 10 dakika
Konsensus	PoW, PoS, DPOS	Özel Algoritmalar	Özel Algoritmalar
Okuma İzni	Herkese Açık	Sınırlı	Sınırlı
Değişmezlik	Yüksek	Yüksek	Yüksek
Verim	Değişken	Yüksek	Orta
Merkezilik	Tam Merkezsiz	Merkezi	Kısıtlı Merkeziyet
Konsensus Süreci	Katılımcılar Arasında Oylama	Merkezi Yetkililer	Sınırlı Katılımcılar Arasında Oylama

2.4. KONSENSÜS ALGORİTMALARI

Blokzincir değişmezlik ve şeffaflığı esas alan bir sistemdir. Bunu yapmak için mevcut bir merkezi otorite yoktur. Ancak blokzincirdeki her işlemin tamamen güvenli ve doğrulanmış olduğu kabul edilmektedir. Bu durum konsensüs algoritmasının varlığı ile sağlanır. Konsensüs algoritması, blokzincir ağının tüm eşlerinin dağıtılmış defterin mevcut durumu hakkında ortak bir anlaşmaya vardığı bir uzlaşmadır. Böylece, konsensüs algoritmaları blokzincir ağında düğümler arası güvenilirlik elde eder ve dağıtılmış bir bilgi işlem ortamında bilinmeyen düğümler arasında güven ortamı oluşturur. Gerçekte blokzincir ağına eklenen her yeni bloğun blokzincirdeki tüm düğümler tarafından uzlaşılan tek versiyonu olmasını sağlar [12].

Konsensüs algoritmaları, ağdaki tüm düğümlerin, ağdaki bir değer konusunda uzlaşmaya varmalarını sağlayan algoritmalardır. Bu algoritmalar, özellikle dağıtılmış sistemler ve blokzincir teknolojisi gibi alanlarda çok önemlidir. Ağdaki tüm düğümlerin ortak bir değeri kabul etmelerini ve böylece blokzincir ağını genel durumunu koordine etmelerini sağlar [13].

Konsensüs algoritmaları, dağıtılmış sistemlerde ve blokzincir ağlarında güvenliği ve bütünlüğü sağlamak için çok önemli bir role sahiptir. Bu algoritmalar, ağdaki tüm düğümlerin, ağın durumu veya değeri hakkında uzlaşmaya varmasını sağlar. Ağın koordinasyonunu ve güvenliği bu sayede artar. [14].

2.4.1. Konsensüs Algoritmalarının Temel Fonksiyonları

2.4.1.1. Anlaşma Sağlama

Konsensüs algoritması, blokzincir ağlarında katılımcıların genel bir anlaşmaya varmasını sağlamak için kullanılmaktadır. Blokzincir ağlarında herhangi bir merkezi otorite veya aracı kurum olmadığı için , tüm katılımcıların bir şekilde anlaşması gerekir. Bu, özellikle blokzincirde bir işlemin veya bloğun doğru ve geçerli olup olmadığını belirlemek için önemlidir. Konsensüs algoritması, tüm düğümlerin bu anlaşmayı sağlamasında yardımcı olmaktadır[15].

Konsensüs algoritması, kötü niyetli düğümlerin sistemi manipüle etmeye istemeleri durumunda bile ağın doğru bir şekilde çalışmasını sağlar. Bu, özellikle açık, izinsiz

blokzincirlerde(Bitcoin,Ethereum) kritik önemdedir. Çünkü herkes bu ağlara katılabilir ve potansiyel olarak kötü niyetli eylemler yapabilir. Kısaca konsensüs algoritması, ağdaki tüm düğümlerin, ağın durumu hakkında uzlaşıya varmasını sağlar. Bu, ağın güvenliğini ve bütünlüğünü korumak için kritik bir işlevdir [15].

2.4.1.2. Güvenlik

Konsensüs algoritmaları, blokzincirde işlemlerin ve veri transferlerinin güvenliğini sağlamaktadır. Bu algoritmalar, kötü niyetli düğümlerin ağı manipüle etmesini veya ağa zarar vermesini engeller. Herhangi bir konsensüs algoritmasının güvenliği, algoritmanın tasarımına, uygulamasına ve ağdaki katılımcıların davranışlarına bağlı olarak değişir [16].

2.4.1.3. Performans

Performans, blokzincirin kullanım amacına göre kritik bir faktördür. Örneğin, bir finansal uygulama için hızlı işlem süreleri esastır. Fakat merkezi olmayan bir depolama sistemi için enerji verimliliği daha önemli olabilir. Bu nedenle, bir blokzincir projesi geliştirirken veya bir teknoloji seçerken, konsensüs algoritmasının performansını göz önünde tutmak gerekir. Konsensüs algoritmaları, ağın genel performansını ve işlem kapasitesini artırır. Bu da, ağın ölçeklenebilirliğini ve işlem hızına olumlu yönde etkiler. [14,15,16].

2.4.2. Konsensüs Algoritmalarının Zorlukları ve Sorunları

2.4.2.1. Ölçeklenebilirlik

Blokzincirin ölçeklenebilirliği, ağın büyüyen işlem hacmini ne kadar etkili bir şekilde işleyebildiğini belirtir. Bu kapasite, konsensüs algoritmasının seçimine duyarlıdır. Çünkü her algoritma farklı işlem hızlarına, güvenlik protokollerine ve özelliklere sahiptir. PoW, ilk blokzincir projelerinde kullanılan en yaygın algoritmadır. Ancak, ölçeklenebilirlik konusunda sınırlamaları bulunmaktadır. Örneğin, Bitcoin, saniyede sadece birkaç işlemi işleyebilir. Bu sınırlı kapasite, yüksek işlem talebi sırasında ağ tıkanıklığına neden olabilir [15]. Bununla birlikte PoS ve DPoS gibi diğer konsensüs algoritmaları, daha hızlı işlem onay süreleri sağlarlar. Böylece daha iyi ölçeklenebilirlik sağlama potansiyeline sahiptir. Özellikle DPoS, sınırlı sayıda seçilmiş düğümler kullanarak yüksek işlem hızlarına ulaşırlar. Ölçeklenebilirlik kısaca, bir blokzincir ağının işlem hacmini ve hızını artırma yeteneğini ifade eder. Bu durum, blokzincir teknolojisinin en büyük sıkıntılarından biridir. Çünkü mevcut blokzincir ağları, saniyede sadece sınırlı sayıda işlemi işleyebilir.

Bu durum, blokzincirin geniş çapta benimsenmesi ve kullanılması için bir engel oluşturmaktadır [14].

2.4.2.2. Enerji Tüketimi

Enerji tüketimi sorunu, özellikle son yıllarda sürdürülebilirlik endişeleri nedeniyle önemli bir sorun olmuştur. Konsensüs algoritması seçimi, bir blokzincirin ne kadar enerji tükettiği üzerinde doğrudan etkiye sahiptir. PoW, enerji tüketimi açısından en sorunlu konsensüs mekanizmasıdır. PoW, madencilerin zor matematiksel problemleri çözmek için rekabet ettiği bir mekanizmadır. Büyük blokzincir ağlarında özellikle bitcoinde, devasa miktarda hesaplama gücü ve dolayısıyla enerji tüketimini yapılmaktadır. Bununla karşılaştırıldığında, PoS ve türevleri, enerji açısından çok daha verimli olabilmektedir. PoS'ta, madencilik yarışı yerine, doğrulayıcılar, sahip oldukları parayla orantılı bir olasılıkla rastgele seçilir. Bu durum, çok daha az hesaplama gücü gerektirir. Dolayısıyla enerji konusunda PoW'a göre çok daha verimlidir. Özetle, konsensüs algoritmasının enerji tüketimi üzerinde büyük bir etkisi vardır. PoW, enerji yoğunluğu nedeniyle eleştirilirken, PoS gibi alternatifler daha sürdürülebilir bir yaklaşım sunmaktadır [16].

2.4.2.3. Güvenlik

Blokzincir, işlemleri ve verileri güvence altına almak için kriptografi kullanmaktadır. Buna rağmen, çeşitli güvenlik sorunlarıyla karşı karşıyadır. En basitinden, %51 saldırısı, bir madenci veya madenci grubunun ağın çoğunluğunu kontrol etmesi durumunda gerçekleşebilir. Bu saye de onlaa çifte harcama yapabilirler ve geçerli işlemleri geri alabilirler. Buna ek olarak, akıllı kontratların güvenlik açıkları bulunmaktadır. Böylece kötü niyetli aktörlerin fonları çalmasına veya sistemi manipüle etmesine yol açmaktadır. Diğer güvenlik sorunları arasında, özel anahtarların çalınması, yazılım hataları ve ağ güvenliği zafiyetleri bulunmaktadır. Bu sorunlar, blokzincir teknolojisinin güvenliğini ve bütünlüğünü tehlikeye atabilir ve kullanıcılar için önemli riskler oluşturmaktadır. Bu nedenle, blokzincir uygulamalarının güvenlik önlemleri ve protokollerine özel dikkat gösterilmesi esastır [16].

2.4.2.4. Adem-i Merkeziyetçilik

Adem-i merkeziyetçilik, blokzincirin temel prensiplerinden birisidir. Verilerin bir merkez yerine ağın tüm düğümleri arasında dağıtıldığı bir yapıdır. Bu yapı ile güvenlik ve şeffaflık sağlanır. Tek bir noktadan hata veya manipülasyon riskini azaltmaktadır. Her bir işlem, ağa bağlı her düğüm tarafından doğrulanır ve onaylandıktan sonra blokzincire

eklenir. Bu da her işlemin değiştirilemez ve kalıcı bir kaydının oluşmasını sağlar. Merkezi otoritelerin veya aracıların gerekli olmaması, kullanıcılara daha fazla bağımsızlık sunmaktadır. Ağın adem-i merkeziyetçi yapısı, hız ve ölçeklenebilirlik gibi bazı teknik sorunları da beraberinde getirebilir [15].

2.4.3. Konsensüs Algoritmaları

2.4.3.1. Proof of Work (PoW)

Satoshi Nakamoto tarafından geliştirilen PoW blokzincir platformları tarafından yayınlanan kripto güvenilirliği ve merkezi olmayan bitcoin için tasarlanan konsensüs mekanizmasıdır. Bitcoin tarafından kullanılmaktadır. PoW düğümlerin rastgele bir matematiksel özet problemi çözmesi gerekir. İşlem başarılı olduğunda zincire yeni işlemler ve veriler eklenmektedir. İşlem gücü ve katılımcı sayısı arttıkça algoritma daha güvenli olmaktadır. PoW yüksek oranda kabul görmüş bir konsensüs algoritması olmasına yine de devasa boyutlarda enerji tüketmektedir [1,17].

2.4.3.2. Proof of Stake (PoS)

PoS, blokzincirdeki düğümlerin PoW'dan farklı olarak kim ne kadar yatırım yapmışsa blokzincire o kadar blok ekleyebilmektedir. PoS, PoW'dan daha az enerji tüketmektedir ve %51 saldırılarına PoW'dan daha iyidir. PoS'de çatallanma durumu söz konusudur. Çatallanmada aynı anda 2 zincir var olur ve 2 zincirde de madencilik yapılır. Bu çatallanma saldırı olarak yapılırsa madenci bu saldırıyı destekleyebilir. Buna Nothing at Stake(tehlikede olan hiçbir şey yok) denir. PoS her zaman tek başına yeterli değildir. Bu nedenle genellikle kullanılan projeler PoS temelli çok gelişmiş algoritmalar kullanmaktadır [18].

2.4.3.3. Byzantine Fault Tolerance (BFT)

BFT, ağdaki kötü niyetli düğümlerim onaylarını geçersiz kılan bir konsensüs mekanizmasıdır. Bu mekanizma Bizanslı generallerin kullandığı bir modelden esinlenerek oluşturulmuş bir yöntemdir. Bu yöntemde imparatordan gelen emirlerin doğrulanması için basit ve etkili bir metot kullanılır. Emirler birden fazla ulakla gönderilir. Emirler geldiğinde ulaklar bu emri paylaşıyordu. Emirler doğrulanmışsa emir doğru olduğu kabul ediliyordu. Bu metot kritik sistemlerde başarı ile uygulanmaktadır [19].

2.4.3.4. Proof of Activity (PoAc)

PoAc protokolü, PoW ve PoS‘tan oluşturulan melez bir mekanizmadır. Bu mekanizma başlangıçta PoW algoritmasını kullanır, boş blok zincirlerinde çalışır ve %51 saldırısını engeller. İlk önce matematiksel problemleri çözer ve blokzincire ekleme yapar. Algoritma daha sonra blokzincirin kabul edilebilir statüsüne sahip bloklar için PoS algoritmasını etkinleştirilir. PoAc'ın güvenlik depolama ve ağ iletişimi açısından etkili olduğu kanıtlanmıştır. Daha az veri kullanması ve daha fazla güvenlik gerektirmesi sebebiyle YZ uygulamaları için kullanışlıdır [20].

2.4.3.5. Proof of Burn (PoB)

PoB algoritması PoW'un enerji tüketimine çözüm olarak ortaya çıkmıştır. Düğümlerin, yeni bloklar oluşturmaları için oluşturdukları jetonları(coin) yakmalarına izin verilmiştir. PoB, kullanıcıları başlangıçta yatırım yapmalarına ve ağdaki hisselerini yaratmalarına izin vererek, yetkili düğümler olmaları sağlanır. Bu aynı zamanda PoW'un enerji tüketimine bir çözümdür. Ek olarak, jeton yakma stratejisi blokzincirindeki jeton sayısını azaltır yani jeton enflasyonunu engeller. Bu nedenle, jeton değeri artar[21].

2.4.3.6. Proof of Elapsed Time (PoET)

PoET algoritması düğümleri, yeni blokoluşturma sürecine dahil etmek yerine, yeni bloklar yaratabilen bir lider düğüm belirler. Minimum son kullanma tarihi olan düğüm lider olarak seçilir. Lider düğüm yeni blokları oluşturur ve imzasını tüm düğümlere gönderir. PoET protokolü sürekli rastgele lider seçim algoritmasını yürütür ve sürekli yeni liderler bulur. Ayrıca, aynı düğümlerin lider olarak seçildiğinden veya minimum ayar değerinin sık sık belirli değerlere atanmasından dolayı kötü niyetli kullanıcıların bulunmasını sağlar [22].

2.4.3.7. Proof of Capacıty (PoC)

PoW mekanizması elde olan bilgi işlem gücünü hesaplamada kullanmaktadır. Özet algoritmaları blokların kilidini açabilmek için kullanılmaktadır. Alan ispatı olarak bilinen PoC blokzincirdeki katılımcıların depolama alanlarını kullanarak madencilik yapılır. PoW'daki özet fonksiyonları kullanmak yerine olası tüm basamakları sabit sürücüde depolar ve eşleşen bloğu bulur [23].

2.4.3.8. Avalanche

Avalanche, yüksek ölçeklenebilirlik, merkeziyetsizlik ve enerji verimliliği gibi

üstünlükler sunar. Yüksek işlem hızlarına ve düşük gecikme süresi sağlar. Aynı zamanda merkezi bir otoriteye veya düğüm kümesine dayanmadan uzlaşı sağlamaktadır. Bununla birlikte, bu mekanizmanın güvenlik riskleri ve mevcut altyapı uyum sorunları gibi dezavantajları da bulunmaktadır. Yeni ve gelişmekte olan bir algoritma olduğundan, daha fazla test ve iyileştirme sürecine ihtiyaç duyabilir [24].

2.4.3.9. HoneyBadgerBFT

HoneyBadgerBFT algoritması, ağdaki güvenli iletişimi sağlamak için işbirliği ve kriptografik teknikler kullanır. HoneyBadgerBFT, hızlı onaylama, yüksek güvenlik seviyeleri ve dayanıklılık gibi avantajlar sunar. Ağdaki hatalı veya düşmanca davranan düğümlere karşı da etkili bir koruma sağlar. Yüksek güvenlik gerektiren uygulamalar için daha uygun bir çözümdür. Yine de, diğer konsensüs algoritmalarına göre daha karmaşık olabilir ve daha fazla hesaplama kaynağı gerektirebilir [19]. Konsensüs algoritmalarını karşılaştırmalı olarak Çizelge 2.2'de verilmiştir.

Çizelge 2.2. Konsensüs algoritmaları

Özellik	**Yetkilendirme**	**Enerji Verimi**	**Verim**	**Kullanın**
PoW	Yok	Yok	Düşük	Bitcoin
PoS	Yok	Kısmen	Düşük	Peercoin
PBFT	İzinli	Var	Yüksek	Hyperledger
DPOS	Yok	Kısmen	Yüksek	Bitshares
PoB	Yok	Var	Düşük	Slime
PoET	İzinli	Kısmen	Düşük	Intel
PoC	Yok	Var	Yüksek	OANDA
Avalanche	İzinli	Var	Yüksek	Defi
HoneyBadgerBFT	İzinli	var	Yüksek	HoneyBadgerBFT

Tüm ele alınan konsensüs algoritmalarının bazı sorunlarının bertaraf edilebilmesi için PoO adında yeni bir konsensüs algoritması fikri bu tez çalışması altında ortaya konulmuştur. PoO amacı, PoW'da harcanan hesaplama gücünü PoO'ya aktararak kullanmayı amaçlayan, önerilen bir blok zinciri mutabakat algoritmasıdır.

2.4.4. Önerilen mekanizma Proof of Optimum (PoO)

Madencilerin hesaplama açısından matematiksel bulmacaları çözmek için yarıştığı PoW'un aksine PoO, bu hesaplama gücünü pratik problem çözme için kullanmaya

odaklanır. PoO algoritması madencilere optimizasyon problemleri sunar ve madenciler de bu problemlere çözüm bulmak rekabet etmeye başlarlar. Başarılı bir çözüm bulan madenci ödüllendirilir. PoO, optimizasyon problemlerini mutabakat mekanizmasına entegre ederek 3 hedefe ulaşmayı amaçlamaktadır:

Verimli Blok Zinciri Doğrulaması: PoO, optimizasyon problemlerini çözmek için hesaplama gücünden yararlanır ve bu da blok zinciri işlemlerinin verimli bir şekilde doğrulanmasına ve yeni blokların oluşturulmasına katkıda bulunur.

Adem-i merkeziyetçilik: Düğümlerin yani katılımcıların ağa dahil olup, blok oluşturulmasında daha fazla katkıda bulunmalarını sağlamak.

Gerçek Dünya Problem Çözümü: PoO, lojistik, finans, mühendislik ve daha fazlası dahil olmak üzere çeşitli alanlarda uygulamaları olabilecek gerçek dünya optimizasyon problemlerini çözmeyi amaçlamaktadır. Bu da madenciler tarafından gerçekleştirilen hesaplama çalışmalarını pratik zorlukların çözümünde değerli kılmaktadır.

Özetle, PoO'nun birincil amacı, PoW'da harcanan hesaplama kaynaklarını optimizasyon problemlerini çözmek için yeniden kullanmak, böylece blok zinciri doğrulamasını daha verimli hale getirmek ve gerçek dünya problemlerinin çözümüne katkıda bulunmaktır. Adem-i merkeziyetçiliğin yani katılımcıların blok oluşturmada ağda daha fazla rol alması noktasında faydalı bir proof mekanizması olarak tez çalışmasının ana unsurunu oluşturmaktadır.

2.5. LİTERATÜR TARAMASI

Salah vd. (2019) çalışmalarında blokzincirin son zamanların en gözde konularından biri olduğunda bahsetmiştir ve bir teknoloji olarak benimsenmeye başladığını anlatmıştır. Bu çalışmada blokzincir ile YZ'nin birleştirilmesinde elde edilen yeni YZ konsepti dağıtık YZ'nin ortaya çıktığından bahsedilmiştir. Blokzincirin temel özellikleri ile YZ bu özelliklerden nasıl yararlanabileceğine dair bir bakış sunulmaktadır. Blokzincir ve YZ entegrasyonu tartışılmış ve dağıtık ve güvenilir bir ekonomi için yeni bir ekosistem geliştirilmesine yardımcı olabileceğini belirtilmiştir. YZ blokzincir ile uygulanma alanları tespit edilmiş ve nasıl kullanılabileceği anlatılmıştır [24].

Wang (2019) çalışmasında blokzincir ve YZ ortak nasıl kullanılabileceğine 3 başlıkta anlatmıştır. İlk önce iki bağımsız teknoloji kabul edilen blokzincir makine öğrenmesi

arasında bir bağlantı kurmuştur. İkincisi makine öğrenmesi ve blokzincir kullanarak birleşik bir çerçeve önermiştir. Bu çerçevede hem otomasyon, hem güvenilirlik sorunlarını üzerine çözümler üretilmiştir. Üçüncüsü tek bir iş parçacığından makine uygulaması ile çalışan birden fazla makinede birden fazla iş parçacığına çeviren blokzincir yaklaşımı kullanıldığı anlatmıştır [25].

Ferrer (2016) çalışmasında, blokzinciri teknolojisinin sürü robotik alanındaki yeni ortaya çıkan 4 soruna nasıl çözümler üretebileceği anlatmıştır. Güvenlik, karar verme, davranış farklılaştırma ve savaş robotik sistemleri için iş modelleri ve senaryolar üreterek tanımlanmıştır. Son olarak, bu iki teknolojinin kombinasyonundan kaynaklanan sınırlamalar ve olası sorunlar açıklanmıştır [26].

Montes ve Goertzele (2019), çalışmalarında YZ gelişimi için alternatif bir yol önermiş ve dağıtık YZ teknolojileri için demokratikleştirilmiş Pazar teknolojisi üzerinde çalışmışlardır. Ben Goertzel tarafından geliştirilen bir havza projesi olan ve Singularity Net kullanılarak geliştirilen bir sistemin özelliklerini ve avantajlarını anlatmışlardır. Ayrıca YZ arasında önemli ölçüde koordineli çalışan bir altyapı oluşmuştur. YZ'nin yapay genel zeka adında yeni bir konsepte evrilmesi tartışılmıştır [27].

Özyılmaz vd. (2018) çalışmalarında iot, makine öğrenmesi ve blokzincir harmanlanmış merkezi olmayan veri pazarından bahsetmişlerdir. Bu pazarda YZ etmenlerinin etkileşime girebilmesi ve son pazarlık yapabilmesi hedeflenmiştir [28].

Harris ve Waggoner (2018) çalışmalarında, makine öğrenmesi kavramının son zamanlarda YZ alanında büyük ilerlemeler sağladığına, fakat bunların merkezi olma eğiliminde olduklarını anlatmıştır. Çalışmalardaki büyük veri kümelerinin genellikle tescilli olduğunu ve eğitmek için çaba harcamadan eski hale geldiğini anlatmıştır. Bu sorunu çözebilmek adına katılımcıların işbirliğine dayalı olarak bir veri kümesi oluşturmaları ve sürekli güncellenen bir modeli barındırmak için akıllı sözleşmeleri kullanmaları için bir çerçeve önermişlerdir. Bu model, çıkarım için kullanmanın ücretsiz olabileceği bir blokzincir üzerinde herkese açık olarak paylaşılmayı hedeflemiştir. İdeal öğrenme sorunları, bir modelin kişisel asistanlar, oyun oynama, öneri sistemleri vb. benzer girdiler için defalarca kullanıldığı senaryoları içermektedir. Modelin bazı test setlerine göre doğruluğunu korumak için hem finansal hem de finansal olmayanları önermişlerdir [29].

Dai vd. (2018) çalışmalarında esnek ve güvenli kaynak paylaşımını sağlamak için YZ ve

blokzincir kablosuz ağlara entegre ederek yeni nesil kablosuz ağlar için güvenli ve akıllı bir mimari önermiştir. Sistem kullanımını en üst düzeye çıkarmak için bir blokzinciri güçlendirilmiş içerik önbellek sorunu önerilmiş ve derin takviye öğrenimini kullanarak yeni bir önbellek düzeni geliştirilmiştir. Bulunan sonuçlarla önerilen bu mimarini faydaları anlatılmıştır [30].

Teerapittayanon ve Kung (2019) çalışmalarında belirli bir sınıflandırma problemi için makine öğrenme modellerinin doğruluğunu arttırmada akran işbirliğini teşvik eden merkezi olmayan bir YZ modeli ağı olan DaiMoN'i ortaya koymuştur. DaiMoN akranların gelişmiş doğrulukta modeller sunabileceği ve diğer akranların doğruluk gelişimini doğrulayabileceği özerk bir ağdır. Sistem, modeli kimin eğittiğini ve doğruluğunu, geliştirildiği zamanı, ne kadar geliştirildiği ve yeni güncellenen modeli nerede bulacağı da dahil olmak üzere, kritik bilgilerin kaydını tutmak için yalnızca ek bir merkezi olmayan defter tutma yapısını sağlamıştır. Bu model doğruluk değerlendirmesini gizli test etiketleriyle etkinleştirmek için DaiMoN, bu makalede önerilen, etiketler için yeni bir öğrenilebilir Mesafe Gömme (DEL) işlevi kullanır. Her test veri kümesine özgü olan DEL, test etiketi vektörünü düşük boyutlu bir alana yerleştirerek, yaklaşık olarak veri kümesinin test etiketi vektörü ile sınıflandırıcı tarafından çıkarılan bir etiket vektörü arasındaki mesafeyi koruyarak karıştırır. Bu nedenle, gerçek test etiketlerine erişmelerini sağlamadan, paydaşları tarafından PoI sağlar. DEL altında akranların model doğruluğunu doğru bir şekilde değerlendirebildiğine dair analiz ve deneysel kanıtlar sunulmuştur. Ayrıca gömme işlevini tersine çevirmenin zor olduğunu ve dolayısıyla DEL'in hile yapmak için test etiketlerini kurtarmayı amaçlayan saldırılara karşı dirençli olduğunu da savunmuşlardır [31].

Singh vd. (2019) çalışmalarında, mevcut en yeni teknikler ve uygulamalar ile IoT için blokzincir ve YZ yı birleştirmenin etkin bir yolunu sağlayan YZ ile bir blokzincir etkin iot mimarisi önermiştir. Önerilen mimariyi değerlendirilip, iki bölüme nicel analiz ve nitel analiz olmak üzere ayırılmıştır. Nitel değerlendirmede, YZ ve blokzincirin IoT uygulamalarında YZ odaklı blokzincir ve blokzincir odaklı YZ ile nasıl kullanılacağını açıklanmaktadır. Nicel analizde, cihazdaki mevcut araştırmaları karşılaştırmak için BlockIoTIntelligence mimarisinin performans değerlendirmesini sunulmuştur. Değerlendirme sonuçları, önerilen mimarinin mevcut IoT mimarilerine göre performans gösterdiğini ve mevcut zorlukları azalttığını anlatılmıştır [32].

Chen vd. (2018) çalışmalarında, YZ teknolojisine dayanan fikir birliği algoritması

önermiştir. Fikir birliğine ulaşmak için özel bir sinir ağı ve süper düğümleri ile rastgele düğümleri elde eden dinamik bir eşik tasarlamıştır. Elde edilen sonuçlar uygulamanın, karmaşık işlem ve tekelden kaçınarak PoW, PoS ve DPoS algoritmalarınn avantajlarının birleştirdiğini göstermektedir. Ayrıca, güvenlik ve işlem onaylama hızı açısından son teknoloji ürünü konsensüs çerçevelerine olumlu bir yaklaşım sağlamaktadır [33].

Qiu vd. (2020) vd. makalelerinde, ağ güvenliğinin karşılaştığı zorlukları ve blokzincir kullanarak ağ güvenliğine yönelik fırsatları tanıtmış ve blokzincir bağlamında ağ güvenliği zorluklarına bir çözüm önerilmiştir [34].

Aditya Raj Singh vd. (2022) blokzincir ve YZ, bu makalede bahsedildiği gibi hemen hemen her sektörde gelişmeyle sonuçlanan ilerlemelere ve büyümeye yol açmıştır. YZ ve blokzincir fikirlerinin harika bir hızla yayıldığı aktarılmıştır [35].

Garg (2017), çalışmasında YZ sistemlerindeki son gelişmeler, makine öğrenimi alanındaki başarımı ve blokzinciri akıllı kontratlarla birleştirildiğinde çok yeni çalışma alanları ortaya çıktığını anlatmıştır. Bununla beraber YZ söz konusu olduğunda akıllı sözleşme, bir YZ sistemi oluşturmak için kullanılabilir veya yardımcı olabilir şeklinde değerlendirilmiştir. Oylama yoluyla bir ikilem sırasında fikir birliğine varılabilmektedir [36].

Khan ve Mangde vd. (2022) çalışmalarında, blokzincirin YZ yönelik riski azaltabilmesi ve YZ blokzincirin performansını iyileştirebilmesi nedeniyle iki teknolojinin birbirini tamamladığını belirtmiştir [37].

Muthukrishnan ve Duraisamy (2020) çalışmalarında, blokzincir teknolojisinin dijital bilgileri nasıl dağıttığı ve kopyalanmasını nasıl engellediği ile yeni bir İnternet türünün nasıl oluşturduğu incelemiştir. Aynı zamanda, blokzincirin merkezi olmayan ve dağıtılmış bir yapıda nasıl çalıştığına dair temel prensipler ele alınmıştır. YZ kavramı, insan benzeri yeteneklere sahip makinelerin tasarımı ve uygulamaları ile birlikte detaylı olarak ele alınmıştır. Özellikle makine öğrenimi, yapay sinir ağları ve derin öğrenme kavramlarına değinilmiştir. Çalışmanın sonunda, blokzincir ve YZ'nin birleşiminin iş dünyasında yarattığı fırsatlar, avantajlar ve zorluklar tartışılmıştır [38].

Imteaj vd. (2021) çalışmalarında, son dönemin popüler teknolojileri olan YZ ve blokzincirin birleşimine odaklanmıştır. Blokzincirin güvenilir ve merkezi olmayan yapısının, üçüncü bir tarafa ihtiyaç duymadan nasıl etkileşim sağladığı ve veri kaydını nasıl güvende tuttuğu ele alınmıştır. YZ çevresel gözlemlerle nasıl akıllı kararlar aldığı üzerinde durulmuştur. Çalışma bu iki teknolojinin entegrasyonunun önemini,

uygulamalarını, karşılaştığı zorlukları ve potansiyel araştırma konularını kapsamaktadır. [39].

Masurkar vd.(2023), makalelerinde, herhangi bir merkezi sunucuda model havuzu oluşturma veya düzenleme gerektirmeyecek şekilde merkezi olmayan birleşik öğrenmenin ötesine geçen, blokzincirine dayalı yeni bir merkezi olmayan YZ paradigması önermişlerdir [40].

Wang vd. (2021), makalelerinde, güvenli veri paylaşımı, veri gizliliğinin korunması ve güvenilir YZ kararlarının ve merkezi olmayan YZ'nin desteklenmesi dahil olmak üzere blokzincirin YZ'ya bu dört açıdan nasıl fayda sağlayabileceğine dair kapsamlı bir inceleme yapmışlardır [41].

Xing ve Marwala (2018) çalışmalarında, YZ ve blokzincir teknolojilerinin hızla nasıl yayıldığına dikkat çekilmiştir. Her iki teknolojinin de kendi içerisinde karmaşık teknolojik özelliklere ve iş dünyası üzerinde çok boyutlu etkilere sahip olduğu belirtilmiştir. Özellikle blokzincirin merkezi olmayan bir yapıda olduğuna dair yaygın bir yanılgı olduğunu, fakat bu sistemlerin temelinde hâlâ belirli bir geliştirici grubunun olduğu vurgulanmıştır. Akıllı kontratların, insan programcılar tarafından oluşturulan ve blokzincir üzerinde çalışan kodlar ve verilerden oluştuğu, bu nedenle hatalardan muaf olmadığı belirtilmiştir. Çalışma, YZ'nin bu tür hataları önlemek için nasıl kullanılabileceğini ve blokzincir 2.0'ın hedeflerine ulaşmak için YZ teknikleriyle nasıl geliştirilebileceğini incelemiştir. YZ ve blokzincirin birleşiminin sayısız olasılık yaratacağı sonucuna varılmıştır [42].

Deng (2019) çalışmasında AIBC (Artificial Intelligence BlockCloud) anlatmıştır. AIBC YZ ve blokzincir teknolojilerini kullanarak bilgi işlem ve depolama kaynaklarının düşük maliyetle paylaşılmasını sağlayan merkezi olmayan bir ekosistemdir. Çalışmada, AIBC'nin yapısını ve nasıl çalıştığını detaylı olarak incelenmiştir [43].

Zheng ve Dai (2022), çalışmalarında blokzincir teknolojisinin sunduğu avantajlara ve karşılaştığı zorluklara odaklanmışlardır. YZ ile blokzincirin entegrasyonunun, operasyonel bakım, akıllı sözleşmelerin kalitesi ve kötü niyetli davranış tespiti gibi konularda çözüm sunabileceği belirtilmiştir. Makale, "blokzincir zekası" olarak adlandırılan bu entegrasyonun potansiyelini ve faydalarını vurgulamakta, ayrıca konunun anlaşılmasını derinleştirmek için bir örnek olay incelemesi sunmaktadır [44].

Yarali(2022), çalışmasında blokzincir ve YZ'nin teknolojik inovasyonda nasıl öne çıktığını ele almıştır. Blokzincirin, YZ'nin potansiyel risklerini azaltabileceği ve iş modellerini dönüştürebileceği vurgulanmıştır. Çeşitli sektörlerde, özellikle imalat,

medya, telekomünikasyon, perakende, kamu hizmetleri, sağlık ve finansal hizmetlerde blokzincirin benimsendiği belirtilmiştir. Ayrıca, artırılmış gerçeklik (AR) ve sanal gerçeklik teknolojilerinin nasıl bir entegrasyon ve uygulama potansiyeline sahip olduğu tartışılmıştır [45].

Panda ve Jena (2021), çalışmalarında blokzincir ve YZ'nin son dönemdeki önemli teknolojik ilerlemelerini incelemişlerdir. Blokzincirin, işlemleri değiştirilemez bir şekilde kaydetme kapasitesine sahip olduğu ve güvenilir bir üçüncü tarafa ihtiyaç duymadan farklı tarafların etkileşimde bulunmasına olanak tanıyan akıllı sözleşmelerle nasıl kullanılabileceği üzerinde durulmuştur. Öte yandan, YZ'nin makineleri insanlar gibi düşünme ve karar verme yetenekleriyle donatma potansiyeli tartışılmıştır. Makale, YZ'nin güçlendirmek için blokzincir teknolojisinin nasıl kullanılabileceğini ayrıntılı bir şekilde araştırıp, YZ'yi daha güvenli ve verimli hale getirmek için blokzincirin nasıl kullanabileceğine dair literatürü özetlemişlerdir [46].

Wang vd. (2021), çalışmalarında YZ uygulamalarını desteklemek için blokzincirin nasıl kullanılabileceğini ele almıştır. Özellikle, model eğitimi için güvenli veri paylaşımı, veri gizliliğinin korunması, güvenilir YZ kararı ve merkezi olmayan YZ gibi konularda blokzincirin YZ nasıl fayda sağlayabileceğine odaklanılmıştır. 2018 ve 2021 yılları arasında yayımlanan 27 İngilizce makale üzerinde yapılan analiz, birçok araştırma zorluğunu ve fırsatını ortaya koymuştur [47].

Tagde (2021) çalışmasında, sağlık sektöründe blokzincir ve YZ teknolojilerinin yenilikçi uygulamaları ele alınmıştır. Sağlık endeksleriyle ilgili veriler, Web of Sciences ve çeşitli resmi kuruluşlardan yapılan Google anketleri üzerinden toplanmıştır. İki teknolojinin nasıl entegre edilerek sağlıkta genelleştirilebilir bir analitik teknolojinin hayata geçirilmesine katkıda bulunabileceği tartışılmıştır. Blokzincirin, e-Sağlıkta güvenilir YZ modelleri oluşturma potansiyelini ve tıp profesyonellerinin hastaların medikal kayıtlarına erişimini nasıl kolaylaştırabileceği vurgulanmıştır. YZ, büyük veri miktarları ve önerilen algoritmalarla karar verme yeteneği sunarken, blokzincir kriptografik kayıtların depolanmasını sağlamaktadır. Bu teknolojilerin entegrasyonu, medikal sistemin verimliliğini artırabilir, maliyetleri düşürebilir ve sağlık hizmetlerini demokratikleştirebilir şeklinde sonuç oluşturmuşlardır [48].

Gupta vd. (2021), çalışmalarında blokzincir ve YZ işbirliğine vurgu yapmıştır. blokzincir, güvenli bağlantılar oluşturarak iletişimi ve işlem izleme yöntemlerini devrimleştirmeyi hedeflerken, YZ karmaşık görevleri mükemmel bir şekilde öğrenmek için büyük miktarda bilgi işlem gücü kullanır. Bu iki farklı teknoloji, birleştirildiğinde daha fazla

fırsat yaratabilir. Her ne kadar her iki teknolojinin farklı altyapıları ve temel prensipleri olsa da birbirlerine çok şey sunabilirler. YZ'nin blokzincir yeteneklerini artırabileceği ve blokzincirin YZ'nin kararlarını daha anlaşılır hale getirebileceği belirtilmiştir. YZ'nin blokzincirde gereksinim duyulan büyük miktarda işlem gücünü azaltabileceği ve bu kombinasyonun, karmaşık öğrenme ve karar verme problemlerini çözmek için gereken büyük veri setlerini işleyebileceği vurgulanmıştır [49].

Murty ve Shri (2021) çalışmalarında, blokzincir ve YZ sağlık sektöründe bireysel olarak değerlerini kanıtlamıştır. Blokzincir, veri güvenliği ve geçerliliği ile bilinirken, YZ, algoritmalar aracılığıyla karar verme yeteneği sunmaktadır. YZ, makinelere çeşitli algoritmaları kullanarak sonuçları tahmin etmeleri için eğitim vermektedir. Özellikle sağlık sektörü gibi veriye çok ihtiyacın olduğu uygulamalar için blokzincir, bu algoritmalar için yüksek kaliteli veri sağlar. İlgili çalışmada, blokzincir ve YZ'nin entegrasyonunun sağlık sektöründe veri analizi, kalite güvenceli akıllı sözleşmelerin oluşturulması ve kişiselleştirilmiş tedavi planlarının kolaylaştırılmasında nasıl yardımcı olabileceği üzerinde durulmuştur [50].

Hussain ve Turjman (2021) çalışmalarında, blokzincir ve YZ üzerine yapılan çalışmalar ve teknolojik yenilikler inkâr edilemez bir hızla yayıldığını ve çalışma fırsatlarının arttığını anlatmıştır. Her iki çalışma alanı da yenilikçi doğaları ve çok boyutlu iş imkanları ile farklıdır. Blokzincir, dijital para çağında dijital olarak açığa çıkarılan güvenli ve merkezi olmayan bir şekilde kişisel kayıtları, bilgileri ve kayıtları değiştirmek için otomatik ödemeleri robotize edebildiğini belirtmiştir. İlgili çalışmada, YZ'nin blokzincirdeki uygulamaları hakkında kapsamlı bir genel bakış sunulmuştur. YZ'nin araştırma alanına özel olarak blokzincir uygulamalarının ve platformlarının yükselişini gözden geçirilmiştir. Ayrıca, bu iki teknolojinin dijital ekonomiye olan etkisi sınıflandırılmıştır. Bu teknolojilerin sağlanması sırasında tanımlanan zorlukları ve sorunları da incelenmiştir. YZ ve blokzincirin entegrasyonunun, doğru şekilde dikkate alındığında bilim adamları ve otoritelere %90'a kadar bir doğruluk sağladığı bulunmuştur [51].

Muheidat ve Tawalbeh (2021) çalışmasında, YZ ve blokzincir teknolojilerinin birleşimi, endüstri ve günlük uygulamalarda büyümekte olup, merkezi sistemlerdeki veri erişimi darboğazlarına çözüm sunmuşlardır. Blokzincir merkezi olmayan yapısı, veri güvenliği ve paylaşımını sağlarken, YZ bu verileri analiz ederek içgörüler sunabilmektedir. Bu entegrasyon, bankacılık ve sağlık gibi sektörlerde büyümeyi teşvik ederken, artan dijital hizmet kullanımı ile birlikte güvenlik risklerini de beraberinde getirmektedir. Ancak,

YZ’nin blokzincir ile birleştirilmesi, bu tehditlere karşı dayanıklı bir savunma oluşturabilir olduğu bu çalışmada vurgulanmıştır. Bu çalışma da ayrıca bu iki teknolojinin siber güvenlikte nasıl bir yakınsama oluşturduğunu ve siber-fiziksel sistemlerin güvenliğini nasıl artırdığını detaylı olarak ele alınmıştır [52].

Saigal (2020) çalışmasında, blokzincir teknolojisinin, merkezi olmayan bir sistem üzerine kurulmuş olup, işlemleri şeffaf, hızlı ve güvenli bir şekilde gerçekleştirdiğini anlatmıştır. Bitcoin'in ortaya çıkışıyla popüler olan blokzincir, sadece finansal sektörle sınırlı kalmayıp, iş dünyasında güven sağlama potansiyeli ile diğer alanlarda da kullanılmaya başlanmıştır. YZ ise makinelerin büyük veri miktarlarından öğrenmelerini ve etkili kararlar alabilmelerini sağlar. Blokzincirin veri kaydı ve doğrulama odaklı yapısı, YZ ‘nin veri akışındaki desenleri tanıma kapasitesiyle birleştiğinde, endüstrilerde devrim yaratabilir olduğu ifade edilmiştir. Bu çalışma da, blokzincirin güçlü mimarisinin YZ’nin farklı yönlerini nasıl iyileştirebileceğini, özellikle sağlık sektörü bağlamında incelenmiştir [53].

Harris (2020) çalışmasında, makine öğrenimi ile YZ alanında büyük ilerlemelerin sağlandığı ifade edilmiştir. Ancak bu sonuçların merkeziyetçi olabileceğinden söz edilmiştir. Gerekli büyük veri kümeleri genellikle mülkiyet altındadır ve modeller hızla güncelliğini yitirebilir. Microsoft Research'ün blokzincir üzerindeki İşbirlikçi ve Merkezsiz YZ teklifi, katılımcıların bir veri kümesini birlikte oluşturmasını ve bir kamu blokzincirinde sürekli güncellenen bir modeli paylaşmasını sağlamıştır. Bu çalışmada, modellerin doğruluğunu koruması ve doğru veri sunan katılımcıların kar elde etme şansını artırmak için Öz-Değerlendirme teşvik mekanizmasını kullanmanın en iyi uygulamalarını önerilmiştir. Üç model (Perceptron, Naive Bayes ve En Yakın Centroid Sınıflandırıcısı) ve üç veri seti üzerinde simulasyonlar analiz edilmiştir. Modellerin bir kamu blok zincirinde akıllı sözleşmelerde barındırıldığında doğruluğu, kullanıcı dengeleri ve işlem maliyetleri karşılaştırılmıştır. Ethereum için açık kaynaklı bir uygulama ve Python'da yazılmış simulasyonlar sağlanmıştır [54].

Gulati vd. (2020) çalışmasında, blokzincir ve YZ, son dönemin en popüler teknolojik trendleri arasında yer aldığını ifade etmiştir. Yapılabilecek çalışmaların çok fazla potansiyeli olduğu ve birçok yeniliğe kapı araladığı belirtilmiştir [55].

Kshetri (2019) çalışmasında, YZ ve blokzincirin, ekonomi ve toplum üzerinde güçlü etkiler yarattığı anlatılmıştır. YZ ve blokzincir, sektörlerin ve pazarların performansını dramatik şekilde etkileyebilecek güçlü tamamlayıcı yeteneklere sahip olduğunu belirtmiştir. Her iki teknoloji de birbirinin performansını ve işlevselliğini potansiyel

olarak geliştirebilir olarak değerlendirilmesi yapılmıştır [56].

Senthilkumar (2020) çalışmasında, blokzincir ve YZ, günümüz dünyasında ön plana çıkan iki devrim niteliğinde teknoloji olduğu ifade edilmiştir. Bu iki teknoloji, B2B çevresini destekleyerek daha üstün sonuçların elde edilmesine yardımcı oldukları ifade edilmiştir [57].

Sgantzos ve Grigg (2019) çalışmasında, YZ ve blokzincirin birleşiminin nasıl dönüştürücü bir teknolojik örnek teşkil edebileceği derin öğrenme dünyasından örneklerle açıklamışlardır. Blokzincirin yüksek derecede güvenli bir depolama ortamı olarak sunulması, veri bütünlüğünün korunmasında teknolojik bir devrim anlamına geldiği bu çalışmada da belirtilmiştir. Blokzincirin değiştirilemez yapısı, derin öğrenme için yüksek kaliteli, kalıcı ve büyüyen veri kümeleri oluşturmak için verimli bir ortam sağladığı gösterilmiştir. YZ veblokzincirin birleşimi, IoT, kimlik, finansal piyasalar, sivil yönetim, akıllı şehirler, küçük topluluklar, tedarik zincirleri, kişiye özgü tıp gibi alanlarda etkili olabildiği ve birçok kişiye fayda sağladığı ifade edilmiştir [58].

Parker ve Bach (2019), çalışmasından blokzincir iot ve YZ ile ilgili birkaç konuya değinerek bir literatür taraması yapmıştır. Çalışma YZ, blokzincir ve iot birleşimin etkilerini araştırmaktadır [59].

Dillenberger (2019) vd. çalışmalarında, finansal ödemelerin tedarik zincirlerindeki ürün hareketleri, kimlik doğrulama bilgileri ve diğer birçok varlık hakkında bilgi içeren blokzincir kayıtlarını incelemiştir. Bu veriler üzerinde yapılan analizler, köken tarihçeleri, öngörücü planlama, sahtekarlık tespiti ve düzenleyici uyumluluk gibi bilgiler sağlanmıştır. Yazarlar, kullanıcı dostu yapılandırılabilir panolar, öngörücü modeller, köken tarihçeleri ve uyumluluk kontrolü sunan blokzincire bağlı analiz motorlarını tanımlamıştır. Ayrıca, blokzincir verilerinin dış veri kaynaklarıyla nasıl birleştirilebileceğini, coğrafi olarak dağınık veriler üzerinde YZ modeli oluşturmayı mümkün kılan güvenli ve özel analizleri ve güvenilir YZ için köken ve soy takibi sağlayan bir model oluşturma geçmişi yaratmayı da ele almışlardır [60].

Marwala ve Xing (2018) makalesinde, YZ (AI) ve blokzincir kavramlarının olağanüstü bir hızla yayıldığı vurgularlar. Birçok geliştirilebilecek yönünün olduğundan bahsederler [61].

Babu (2020) vd. çalışmasında da güvenlikle ilgili verileri ve çoklu platformlarda son kullanıcılar ile güvenlik uygulamaları/cihazları arasında toplamak, iletmek, paylaşmak ve işlemek için blokzincir açık uygulamasını arka planda kullanarak ve YZ ve davranış analiziyle entegre edilmiş akıllı sözleşmeleri kullanarak veri kararlarını uygulamak üzere

tasarlanmış sistemler ve yöntemler sunmuştur [62].

Saritha (2021) vd. çalışmasında, YZ, blokzincir ve Nesnelerin İnterneti IoT kavramlarının kamuya ve şirketlere benzersiz fırsatlar sunduğunu vurgulamıştır [63].

Swan(2018), çalışmasında, blokzincir 'in dağıtılmış defterlerini kamusal ve özel blokzincirler, kurumsal blokzincir uygulamaları ve özellikle derin öğrenme blokzincir olarak adlandırılan sonraki nesil YZ sistemlerindeki rolleri bağlamında tartışmaktadır [64].

Goel vd. (2021) çalışmalarında blokzincir, çeşitli ticaretler üzerinde güvenli mekanik etkiler sunan yeniçağ bir teknoloji olduğunu vurgular. Bu sistemin, maliyetin azaltılması ve zamanın tasarrufu için herhangi bir tek noktada başarısızlığın ortadan kaldırılmasına yardımcı olduğunu vurgulamıştır. Şimdiye kadar, blokzincir ile ilgili çalışmaların çoğu, şifrelenmiş para formuyla ilgili gelişen teknolojilere sınırlı olduğu değerlendirilmiştir. Farklı akıllı cihazların ilerlemesi ve YZ ve makine öğrenimi alanı ile blokzincir teknolojisinin disiplinlerarası işbirliği, gelecekteki araştırmalar için oldukça faydalı olacağı fikri üzerine odaklanılmıştır [65].

Dai (2020) vd. çalışmalarında, blokzincir zekasının mevcut blokzincir sistemlerini daha da geliştirmek için büyük bir potansiyele sahip olduğunu kabul ederken, bu yeni alanda açık araştırma yönleri bulunduğunu aktarmıştır. Çalışmada, blokzincir üzerinde federatif öğrenme, blokzincir kolektif zeka bahşetme ve blokzincir otomasyonunu artırmak için YZ gibi çeşitli yönlerden zekasının açık meseleleri tartışılmıştır [66].

3. MATERYAL VE YÖNTEM

3.1. YAPAY ZEKA (YZ)

Yapay zeka (YZ), bilgisayar biliminin en dinamik ve hızla gelişen dallarından biri olarak kabul edilir. Son on yılda, bu alanda gerçekleşen teknolojik ilerlemeler sayesinde, sadece akademik araştırmaların veya bilim kurgu filmlerinin konusu olmaktan çıkıp günlük yaşantımızın vazgeçilmez bir parçası haline geldi. Bu hızla ilerleyen teknolojinin yarattığı etki, Endüstri Devrimi'nin buharlı makinenin icadıyla kıyaslanabilir derecede devrim niteliğindedir.

Birçok kişi, YZ denildiğinde, kendi kendine hareket eden robotları veya bilgisayarların insan gibi düşünme yeteneğine sahip olmasını hayal eder. Ancak YZ'nin gerçekte ne olduğunu ve günlük yaşamımıza nasıl entegre olduğunu anlamak için bu kavramı daha geniş bir perspektifle değerlendirmek gerekir. YZ, basitçe, makinelere ve yazılımlara insan benzeri düşünme ve problem çözme yetenekleri kazandırma bilimidir. Bu, bir bilgisayarın karmaşık bir matematiksel problemi çözmesinden, bir oyunu oynamasına, hatta bir resmi tanımasına kadar geniş bir yelpazede uygulamalara sahip olabilir [67].

Günümüzde YZ, birçok alanda etkisini göstermektedir. Örneğin, otomotiv endüstrisinde, otonom araçların gelişiminde kritik bir rol oynamaktadır. Sağlık sektöründe, hastalıkların teşhisinde ve tedavisinde doktorlara yardımcı olacak algoritmaların geliştirilmesinde önemli bir yere sahiptir. Finans sektöründe, yatırım stratejilerini optimize etmek için kullanılan algoritmalara kadar, YZ'nin etkisi her yerdedir.

Ancak, YZ'nin bu hızla ilerlemesi, beraberinde birçok etik ve toplumsal sorunu da getirmektedir. Örneğin, bir algoritma tarafından alınan bir kararın yanlış olması durumunda sorumluluğun kimde olacağı veya YZ teknolojilerinin istihdam üzerindeki etkisi gibi konular, önümüzdeki yıllarda tartışılmaya devam edecektir.

Sonuç olarak YZ, hem fırsatlar hem de zorluklar sunan bir teknolojidir. Ancak şüphesiz ki, bu teknolojinin potansiyelini en iyi şekilde kullanabilmek ve olası risklerini minimize edebilmek için toplumsal bir bilinç ve eğitim gerekmektedir. Bu nedenle, YZ konusundaki bilgi ve farkındalığın artırılması, geleceğin teknolojik ve toplumsal zorluklarına hazırlıklı olabilmek için kritik bir öneme sahiptir [67].

YZ kullanım alanları aşağıdaki şekilde sınıflandırılır.

1. Problem çözümü
2. Oyunların modellenmesi
3. Bilgi modellenmesi
4. Doğal dil işleme
5. Örüntü tanıma
6. Uzman sistemler
7. Robotik

3.2. YAPAY ZEKANIN KULLANIM AMACI

YZ çalışmalarının amacı insan zekasında esinlenilerek, insan zekası gerektiren görevleri makinelere yaptırmaktır. Genel olarak bu amaç 3 ana başlık altında toplanabilir.

1. Makinaları daha akıllı hale getirmek
2. Zekanın ne olduğunu anlamak
3. Makinaları daha faydalı hale getirmek

Çizelge 3.1'de YZ'nin iki boyutta ortaya konan sekiz tanımı yapılmıştır. Soldaki tanımlar başarıyı insan performansına uygunluk açısından ölçerken Rasyonellik Sağdakiler, rasyonellik adı verilen ideal bir performans ölçüsüne göre ölçülür. Sistem, bildiklerini göz önünde bulundurarak "doğru olanı" yapıyorsa rasyoneldir [68].

Çizelge 3.1. Yapay zeka

İnsanca Düşünmek	**Akılcı Düşünme**
" ." "İnsan düşüncesiyle ilişkilendirdiğimiz faaliyetlerin, karar verme, problem çözme, öğrenme gibi faaliyetlerin otomasyonu"	"Hesaplamalı modellerin kullanımı yoluyla zihinsel yetilerin incelenmesi." "Algılamayı, akıl yürütmeyi ve eylemde bulunmayı mümkün kılan hesaplamaların incelenmesi."

İnsanca Davranmak	Akılcı Davranmak
"İnsanlar tarafından yapıldığında zeka gerektiren işlevleri yerine getiren makineler yaratma sanatıdır." "Şu anda insanların daha iyi olduğu işlerin bilgisayarlara nasıl yaptırılacağının araştırılması."	"Hesaplamalı Zeka, akıllı ajanların tasarımı üzerine yapılan bir çalışmadır." "YZ yapay nesnelerdeki akıllı davranışlarla ilgilenir."

3.3. YAPAY ZEKÂ ALT DALLARI

YZ, makinelerin insan benzeri görevleri yerine getirmesi için programlanabilen geniş bir teknoloji alanını kapsar. YZ'nin alt kategorilerinden bazıları makine öğrenimi, sezgisel algoritmalar ve derin öğrenmedir.

3.3.1. Makine öğrenmesi

Makine Öğrenimi (Machine Learning, ML), YZ'nin alt kategorilerinden biri olarak bilgisayarların, özel olarak programlanmadan veriler üzerinden öğrenmelerine olanak tanıyan bir bilim dalıdır [69].

3.3.1.1. Temel Prensipler

Makine öğrenimi alanında bir modelin, veri üzerindeki deneyimlerinden öğrenmesini ifade eder. Bu, modelin veriye maruz kaldıkça performansını geliştirmesi ve daha doğru tahminlerde bulunabilmesi anlamına gelir. Deneyim ile öğrenme, özellikle güçlendirilmiş öğrenme gibi bazı makine öğrenimi yaklaşımlarında merkezi bir role sahiptir.

Güçlendirilmiş öğrenmede, bir ajan, belirli bir görevi gerçekleştirmek için en iyi eylemleri seçmeye çalışırken bir ödül mekanizması tarafından yönlendirilir. Ajan, deneyimlerinden öğrenir, yani eylemlerinin sonuçlarına (ödüllerine veya cezalarına) dayanarak stratejilerini ayarlar. Bu yaklaşım, ajanın zamanla daha yüksek ödüller elde edecek şekilde davranışını optimize etmesini sağlar [70].

Özetle, deneyim ile öğrenme, makine öğreniminin temel taşlarından biridir. Bir modelin deneyimlerinden öğrenmesi, onun belirli bir görevde daha başarılı olmasına olanak tanır.

Bu öğrenme süreci, modelin eğitim verisine maruz kaldıkça ve gerçek dünya senaryolarıyla karşılaştıkça devam eder. Bu, modelin sürekli olarak gelişen ve değişen verilere adapte olmasını sağlar.

Tahmin ile Öğrenme: Tahmin öğrenme, makine öğrenimi içinde önemli bir konsepttir. Temel olarak, bir makine öğrenimi modelinin, eğitim verisi üzerindeki örneklerden öğrendiği bilgiyi kullanarak henüz görmediği verilere ilişkin tahminlerde bulunmasını ifade eder. Bu, regresyon ve sınıflandırma gibi birçok makine öğrenimi algoritmasının ana amacıdır [71].

Regresyon: Regresyon, sürekli bir çıktı değişkenini tahmin etmeye yönelik bir tahminleme yöntemidir. Örneğin, bir evin özelliklerine (oda sayısı, konumu, büyüklüğü vb.) dayanarak fiyatını tahmin etmek istenirse, regresyon yöntemi kullanılabilir.

Sınıflandırma: Sınıflandırma, belirli kategorilere veya sınıflara ayrılmış olan çıktı değişkenini tahmin etmeye yöneliktir. Örneğin, bir e-postanın spam olup olmadığını tahmin etmek istenirse, sınıflandırma yöntemi kullanılabilir.

Tahminleme sürecinde, model öncelikle eğitim verisi üzerinde eğitilir. Eğitim süreci boyunca, modelin tahminleri gerçek sonuçlarla karşılaştırılır ve modelin performansı değerlendirilir. Modelin hataları minimize edecek şekilde ağırlıkları veya parametreleri ayarlanır. Bu süreç, model istenen doğruluk seviyesine ulaşana kadar veya belirlenen bir iterasyon sayısına kadar devam eder. Tahminleme, makine öğreniminin en temel görevlerinden biridir ve birçok pratik uygulamada kullanılır. Stok fiyatları tahmini, hastalık teşhisi, müşteri davranışları analizi ve daha birçok alan buna örnek verilebilir. Bu tür tahminleme görevleri, ML modelinin doğru ve güvenilir tahminler yapabilmesi için genellikle büyük miktarda etiketli veri gerektirir [72].

3.3.2. Sezgisel Algoritmalar

Sezgisel algoritmalar, karmaşık problemlerin çözülmesi için genellikle pratik ve yaklaşık çözümler üretmeyi amaçlayan algoritmalardır. Bu tür algoritmaların temel amacı, problemi tam anlamıyla çözmek yerine yeterince iyi bir çözümü hızlı bir şekilde bulmaktır. Sezgisel algoritmalar, özellikle NP-zor (non-deterministic polynomial-time hard) problemler gibi tam bir çözümün hesaplanmasının pratikte mümkün olmadığı durumlar için kullanışlıdır [73]. Birçok sezgizel algoritmalar mevcuttur ve bunlarla ilgili birçok çalışma ve uygulama bulunmaktadır. Aşağıda bu algoritmalardan bir kısmından bahsedilmiştir.

Tırmanış Algoritması[74]

Taklitçi Soğuma[75]

Tabu Arama[76]

Parçacık Sürü Algoritması[77]

Karınca Kolonisi Optimizasyonu[78]

Benzetimli Sürü Algoritması

Ateş Böceği Algoritması (Firefly Algorithm)

Yapay Arı Kolonisi Algoritması

Gri Kurt Sürü Optimizasyonu [79]

PoO onay mekanizmasında Genetik Algoritma (GA) kullanılmasının temel nedeni, optimizasyon problemlerinin etkin bir şekilde çözülmesine yönelik GA'nın güçlü ve esnek yapısının sağladığı avantajlardır. GA, biyolojik evrimin temel prensiplerini taklit ederek tasarlanmış bir YZ yöntemidir ve genetik süreçleri matematiksel optimizasyon problemlerine uygular. GA'nın geniş uygulama alanı, esnekliği, paralel işleme yeteneği ve evrimsel doğası sayesinde optimizasyon problemlerini çözmek için etkili bir araçtır. Bu nedenle PoO, GA'yı kullanarak çeşitli problemleri başarılı bir şekilde ele alabilir ve blok zincir teknolojisinin verimliliğini artırabilir.

3.4. GENETİK ALGORİTMA

Genetik algoritma (GA), evrimsel mekanizmalardan esinlenen bir optimizasyon algoritmasıdır. Çok boyutlu bir uzayda bit uygunluk fonksiyonuna göre optimize yapan ve her adımda en iyi sonucu üreten bireyin hayatta kalması presibine dayanan bir çözüm arama yöntemidir. Çözüm kümesine popülasyon denir. Popülasyonu da genler oluşturur [43].

Biyolojik evrim teorisinden esinlenerek ortaya konulmuş olan Genetik algoritma çözüm alanını rastgele bicimde tarayarak en iyi çözümü arayan bir yöntemdir. Çözüme ulaşabilmek için önce karar değişkeni uzayında rastgele noktalar topluluğu alınır. Bu noktalar bazı eşleşmeler yapılarak toplumun bazı üyeleri yok edilir yerine yenileri oluşturulur. Bu yeni üyelerin çözüme katılması ile daha sağlıklı çözümler elde edilir. Böylece amaca birçok koldan yaklaşılır [80].

Bir genetik algoritma, genellikle bir popülasyonu rastgele oluşturulan aday çözümlerle başlatır. Bu çözümler, genetik algoritmanın her iterasyonunda (nesil) değerlendirilir ve bir uygunluk fonksiyonu kullanılarak derecelendirilir. Uygunluk fonksiyonu, bir çözümün ne kadar iyi olduğunu ölçer. Daha uygun çözümler, bir sonraki nesil için ebeveyn olarak seçilme olasılığı daha yüksek olan çözümlerdir. Ebeveynler, çaprazlama ve mutasyon operatörleri kullanılarak bir sonraki nesil için yeni çözümler (çocuklar) üretir. Bu süreç, belirli bir sayıda nesil boyunca veya diğer durdurma kriterleri karşılandığında tekrar edilir [80].

Genetik algoritmalar, geniş bir problem yelpazesinde etkili bir şekilde kullanılmıştır. Bunlar optimizasyon problemleri, makine öğrenimi, finans vb daha pek çok alandır. Bununla birlikte, genetik algoritmaların performansı ve etkinliği, kullanılan parametreler ve operatörler gibi faktörlere bağlıdır ve bu nedenle dikkatlice seçilmelidir [81].

3.4.1. Genetik algoritma adımları

Basit GA'nın temel adımları şu şekildedir.

1. Topluluk içerisindeki bireylerin miktarı sabit bir rakamla ifade edilmemiş olup, gerçekleştirilen çalışmalar neticesinde, genel olarak popülasyon büyüklüğünün 30 ila 100 arasında olmasının uygun olduğu tavsiye edilmektedir.
2. Bir kromozomun ne denli iyi olduğunun tespit edilmesini sağlayan fonksiyona uygunluk fonksiyonu adı verilir. Bu fonksiyon, dizilimlerin uygunluk seviyelerini tespit etme işlemi olan uygunluk skorlamasını gerçekleştirir. Bu fonksiyon, genetik algoritmanın (GA) temelini oluşturur ve problemin kendine has özelliklerine göre tasarlanan biricik bileşendir. Uygunluk fonksiyonu, kromozomlardaki bilgileri, problemin değişkenlerine dönüştürerek bu bilgilerin anlaşılır hale gelmesini sağlar. Belirlenen bu parametreler çerçevesinde uygunluk değeri hesaplanarak kromozomun ne kadar uygun olduğu belirlenir. GA'nın etkinliği, bu fonksiyonun doğruluk ve hassasiyetine oldukça bağlıdır.
3. Kromozomların çiftleştirilmesi, onların uygunluk puanlarına dayanarak gerçekleştirilir. Bu çiftleştirmeyi yapabilmek için çeşitli yöntemler mevcuttur, bunlardan bazıları rulet çarkı yöntemi (roulette wheel selection) ve turnuva yöntemi (tournament selection) gibi eşleme stratejileridir. En sık tercih edilen yöntemlerden biri olan rulet çarkı yönteminin çalışma prensibi şöyle açıklanabilir;

a. Her bir bireyin uygunluk puanları bir çizelgede saklanır.

b. Bu puanlar bir araya getirilerek toplam değer hesaplanır.

c. Her bireyin değeri, toplam değere bölünür ve böylece [0, 1] aralığında oranlar elde edilir. Bu oranlar, her bireyin seçilme şansını ifade eder ve bu oranlar bir liste veya tablo içerisinde kaydedilir.

d. Bireylerin seçilme ihtimallerinin kaydedildiği tablodaki oranlar birikimli olarak toplanır ve rastgele seçilen bir sayıya kadar bu toplamda ilerlenir. Rastgele seçilen bu sayıya erişildiğinde veya bu sayı aşıldığında, toplama en son eklenmiş olan kromozomun seçildiği kabul edilir. Rulet tekerleği yönteminde, çözümlerin uygunluk değerlerinin pozitif olması şarttır, zira negatif bir uygunluk değeri, o çözümün seçilme olasılığını ortadan kaldıracaktır. Eğer bir topluluğun büyük bir kısmının uygunluk değeri negatifse, bu durum yeni nesillerin belirli bir noktada tıkanıp kalmasına yol açabilir.

4. Çaprazlama ve mutasyon işlemleri, genetik algoritmanın işleyişinde merkezi rol oynar. Çaprazlama işlemi, genellikle iki kromozomun seçilen bölümlerinin yerlerinin değiştirilmesi sürecidir. Mutasyon ise, belirli bir kromozomun içindeki genlerin rastgele değiştirilmesidir. Eğer mutasyon olasılığı çok düşükse, popülasyonda önemli özellikler kaybolabilir ve bu da en uygun çözümün bulunmasını engelleyebilir. Diğer taraftan, çok yüksek bir mutasyon olasılığı, mevcut çözümlerin bozulmasına yol açabilir. Bu nedenle, genellikle mutasyon oranı %0.1 ile %15 arasında bir değer olarak ayarlanır. Çaprazlama oranı ise, genetik çeşitliliği korumak ve sağlamak amacıyla %60 ile %90 arasında bir değer aralığında seçilir.

5. Bazı problemlerde, genetik değişimlerin sonucu olarak, bireylerin gen dizileri aynı bilgiyi içermeli ve başlangıç neslindeki gen sayıları ile aynı kalması beklenir. Bu durum, özellikle genlerin bir sıralı yapıda önemli bilgiler taşıdığı problemlerde geçerlidir, örneğin, bir diziye dayalı optimizasyon problemleri veya belli bir yapısal sıralama gerektiren durumlar. Çaprazlama ve mutasyon işlemlerinin uygulanmasının ardından, gen dizisinin bozulmaması ve ilk nesil ile tutarlılığını koruması için, problem türüne özel olarak bir tamir mekanizması devreye sokulabilir. Tamir operatörü, genetik algoritmanın aradığı çözüm alanını

daraltmadan, genetik çeşitliliği korumak ve dizi bütünlüğünü sağlamak için tasarlanmış bir yöntemdir. Eğer bu tamir işlemi uygulanmazsa, genetik algoritma çözüm uzayından sapabilir ve bu durum algoritmanın etkin bir çözüm bulma kapasitesini ciddi şekilde azaltabilir veya imkansız hale getirebilir.

6. Eski nesillerden elde edilen diziler çıkarılır ve yerlerine yeni nesilden sabit büyüklükteki bir popülasyon oluşturmak için yeni diziler eklenir
7. Mevcut çözüm havuzunda bulunan en iyi performansa sahip birey, bir sonraki nesle doğrudan taşınarak korunur; bu sürece elitizm adı verilir
8. Yeni oluşturulan popülasyondaki tüm dizilimler için uygunluk değerleri yeniden hesaplanır ve bu sayede yeni popülasyonun genel performansı değerlendirilir.
9. Algoritma, önceden tanımlanan iterasyon sayısına ulaşana veya belirli bir durdurma kriteri karşılanana kadar tekrar tekrar çalıştırılır.
10. Genetik algoritmanın yürütme sürecinin sonunda, popülasyondaki en yüksek uygunluk değerine sahip kromozom, en iyi çözüm olarak seçilir ve sunulur.

3.5. GENETİK OPERATÖRLER

GA'da çözüm yığını belirli noktalardan sonra nesil çeşitliliği ulaşabilmek için dizilere(Kromozom) bir takım operatörler kullanılarak işlemler yapılır. Belirli yüzdelik oranlarıyla uygulanarak nesil çeşitliliği sağlanır. Böylece algoritma belirli noktalara gelip takılması önlenmesi amaçlanır.

3.5.1. Seçilim Operatörü

Seçilim, genetik algoritmaların temel bir bileşenidir ve evrimsel algoritma sürecinde bireylerin gelecek nesillere hangi bireylerin geçeceğini belirler. Bu süreç, bireylerin uygunluk değerlerine dayanarak gerçekleşir. Uygunluk değeri, bir bireyin problemi çözme kapasitesini ölçen bir metriktir ve genellikle bir uygunluk fonksiyonu ile hesaplanır. Seçilim sürecinde, yüksek uygunluk değerine sahip bireyler, daha büyük bir olasılıkla seçilir ve gelecek nesile aktarılır. Bu, genetik algoritmanın zaman içinde giderek daha uygun çözümler üretmesini sağlar. Seçilim süreci, rulet tekerleği seçimi, turnuva seçimi ve rastgele seçim gibi çeşitli yöntemlerle gerçekleştirilebilir. Her yöntem, algoritmanın performansını ve çözümün kalitesini farklı şekillerde etkileyebilir, bu nedenle seçilim yönteminin dikkatlice seçilmesi önemlidir.

3.5.1.1. Rulet Tekerleği Seçilimi

Bu yöntemde, her bireyin seçilme olasılığı, toplam uygunluk değerine oranla kendi uygunluk değerine bağlıdır. Bu, bir rulet tekerleğinin döndürülmesine benzetilir, burada her bireyin tekerlek üzerinde bir alanı vardır ve alanın büyüklüğü bireyin uygunluk değerine orantılıdır [81].

3.5.1.2. Turnuva Seçilimi

Bu yöntemde, rastgele seçilen bir grup birey arasından en uygun olanı seçilir. Turnuva boyutu, seçilim basınçını kontrol eder: daha büyük turnuvalar daha fazla seçilim basınçı üretir [81].

3.5.1.3. Sıralama Tabanlı Seçilim

Bireyler uygunluk değerlerine göre sıralanır ve daha yüksek sıradaki bireyler daha büyük bir olasılıkla seçilir. Bu yöntem, uygunluk değerlerinin aşırı değerlerinin seçilim sürecini bozmasını önler [80].

3.5.2. Çaprazlama Operatörü

Çaprazlama operatörü, karşılıklı kromozomların gen yapılarının değişimi ile yeni dizilerin oluşumunu sağlayan operatördür. Çaprazlanan genler sayesinde algoritma bir çözümde takılıp kalmaz. Böylece, ata kromozomun yerlerini değiştirilerek çocuk kromozomlar üretilir ve daha yüksek uygunluklu çocuk kromozomlar oluşur. Uygulanma olasılığı %50-%95 arasında değişir. Çalışmada çaprazlama oranı %65 olarak belirlenmiştir. Çünkü %50 veya altında değerler çözümlerin bir yerden çıkmamasına neden olabilir. Bu çalışmada PMX çaprazlama operatörü kullanılmıştır.

3.5.2.1. Tek noktalı çaprazlama

Bu yöntemde, ebeveynlerin kromozomları rastgele seçilen bir noktada kesilir ve kesim noktasının bir tarafındaki genler birbiriyle değiştirilir. Bu, genetik bilginin bir ebeveynden diğerine aktarılmasını sağlar.

Kromozom 1 **AABBCC**

Kromozom 2 **XXYYZZ**

Çocuk 1 **AAXXZZ**

Çocuk 2 **XXBBCC** [82].

3.5.2.2. Çok Nokta Çaprazlama (Multi-Point Crossover):

İki veya daha fazla kesim noktası kullanılır ve bu kesim noktaları arasındaki gen segmentleri değiştirilir. Bu yöntem, genetik çeşitliliği artırmaya yardımcı olabilir.

Kromozom 1 **AABBCC**

Kromozom 2 **XXYYZZ**

Çocuk 1 **AAYYCC**

Çocuk 2 **XXBBZZ**-[83].

3.5.2.3. Düzgün Çaprazlama (Uniform Crossover)

Her gen pozisyonu için, bir çocuğun hangi ebeveynden gen alacağına rastgele karar verilir. Bu, genetik çeşitliliği daha da artırabilir.

Kromozom 1 **AABBCC**

Kromozom 2 **XXYYZZ**

Çocuk 1 **AXBYCZ**

Çocuk 2 **XXBBZZ** [84,85]

3.5.2.4. PMX çaprazlama operatörü

Bu yöntem gezgin satıcı problemi ve araç rotalamada problemlerinde kullanılır. Bu problemler için özel bir operatördür çünkü dizileri oluşturan genlerin aynı dizi içinde tekrar etmemesi istenir. Bu çalışmada çaprazlama operatörü olarak seçilmiştir [86].

• Kromozom-1: **123|456|789**

• Kromozom-2: **457|891|236**

• Çocuk-1: **123|891|789**

• Çocuk-2: **457|456|236**

• Çocuk-1 Son: **123|891|789** (Genler tekrar etmesin diye tekrar edenler değiştirilir)

• Çocuk-2 Son: **457|328|916**

3.5.3. Elitizm

Elitizm, genetik algoritmalarda kullanılan bir tekniktir ve bu teknik, en iyi bireylerin veya çözümlerin gelecek nesillere aktarıldığı bir stratejiyi ifade eder. Bu, genetik algoritmanın

her iterasyonunda, en iyi performans gösteren bireylerin bir sonraki nesile direkt olarak kopyalandığı anlamına gelir. Ancak, elitisizm genetik çeşitliliği azaltabilir ve bu da algoritmanın yerel optimumlarda takılıp kalmasına neden olabilir. Elitizm uygulanırken, genetik algoritmanın parametreleri (örneğin, elit bireylerin sayısı) dikkatlice seçilmelidir. Elit bireylerin sayısı çok fazla olursa, genetik çeşitlilik hızla azalabilir ve algoritma yerel optimumlarda takılıp kalabilir [87].

3.5.4. Mutasyon(Değişim) Operatörü

Kromozomların kendi genleri veya genleri oluşturan küçük birimleri üzerinde değişiklik yapılmasını sağlayan operatördür. GA'da mutasyonun sağladığı avantaj, problemin çözüm alanını araştırmada yön değişikliklerini sağlayarak mutasyon yardımıyla araştırmanın kısır döngüye girmesini önlemektir (Lokal Minimum). Uygulanma olasılığı %0,5-%15 arasında değişir. Çalışmada mutasyon oranı %15 olarak belirlenmiştir. Mutasyon oranın %15'ten fazla olması çözüm yanlış yerlerde aranmasına sebep olabilir [80].

3.5.4.1. Mutasyon çeşitleri

3.5.4.1.1. Bit Tersleme Mutasyonu

Bit Flip Mutation, genetik algoritmaların bir çeşidi olan ikili kodlanmış genetik algoritmalar için kullanılan bir mutasyon operatörüdür. Bu operatör, seçilen bir bireyin kromozomundaki tek bir biti veya birden fazla biti rastgele seçer ve değerini değiştirir. Örneğin, bir bit 0 ise 1 yapılır; eğer 1 ise 0 yapılır. Bu basit operatör, genetik algoritmanın arama alanındaki çeşitliliği artırır ve algoritmanın yerel optimumlarda takılıp kalmamasına yardımcı olur.

• Mutasyondan önce: 01|0|11001

• Mutasyondan sonra: 01|1|11001

3.5.4.1.2. Swap Mutasyonu

Swap Mutation, genetik algoritmaların bir mutasyon operatörüdür ve genellikle permütasyon tabanlı genetik algoritmalar için kullanılır. Bu operatör, bir bireyin kromozomundaki iki rastgele pozisyonu seçer ve bu pozisyonlardaki genlerin değerlerini birbiriyle değiştirir. Bu, algoritmanın çeşitliliğini artırır ve yeni çözüm alanlarını keşfetmesine yardımcı olur [80,81].

• Mutasyondan önce: 01|2|345|6|789

• Mutasyondan sonra: 01|6|345|2|789

3.5.4.1.3. Scramble Mutasyonu

Scramble Mutasyonu, genetik algoritmalar için kullanılan bir diğer mutasyon operatörüdür ve özellikle permütasyon tabanlı genetik algoritmalar için uygundur. Bu mutasyon türü, kromozomun bir bölümünü alır ve bu bölümdeki genlerin sırasını rastgele karıştırır. Bu, çözüm uzayındaki çeşitliliği artırmaya ve genetik algoritmanın daha geniş bir çözüm uzayını araştırmasına yardımcı olur [80].

• Mutasyondan önce: 01|23456|789

• Mutasyondan sonra: 01|45263|789

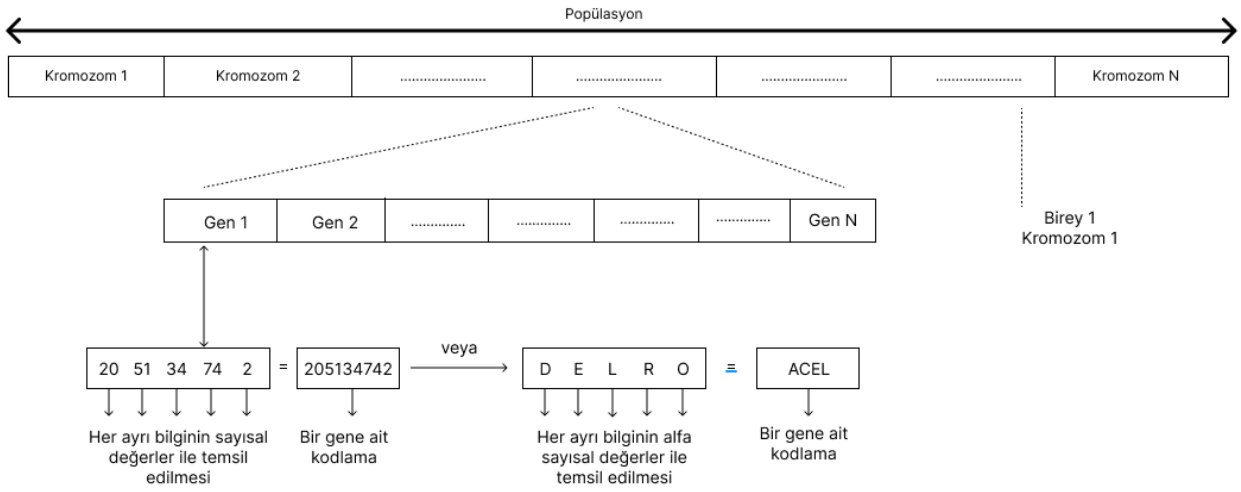
3.5.4.1.4. Inversion Mutasyonu

Inversion Mutasyonu, genetik algoritmalar için kullanılan bir mutasyon tekniğidir. Bu teknik, kromozomun bir bölümünü alıp ters çevirerek çalışır. Bu, genetik algoritmanın çeşitliliğini artırır ve algoritmanın geniş bir çözüm uzayını keşfetmesine yardımcı olur. Inversion mutasyonu, özellikle permütasyon tabanlı problemler için kullanışlıdır [84].

• Mutasyondan önce: 01|23456|789

• Mutasyondan sonra: 01|65432|789

Genetik algoritmanın genel şeması şekil 3.1'de verilmiştir.



Şekil 3.1. Genetik Algoritma Popülasyon, kromozom ve gen şeması.

Çalışmada, PoO algoritması için Gezgin Satıcı Problemi (GSP) gibi bir optimizasyon problemi seçilmiştir. Neden bu tercihin yapıldığı aşağıda açıklanmıştır.

Birincisi, GSP NP-zor bir problem olarak sınıflandırılır ve problem boyutu arttıkça hesaplama zorluğu artar. Bu, PoO'nun daha fazla hesaplama gücü gerektiren bir problemi ele alarak, ağı kötü niyetli faaliyetlere karşı daha güvenli hale getirebileceği anlamına gelir. Ayrıca, GSP'nin doğası gereği çözümünü önceden tahmin etmek zordur, bu da PoO'nun öngörülemeyen bir şekilde seçimler yapmasına olanak tanır ve bu da ağın güvenliğini artırabilir.

İkincisi, GSP, birçok optimizasyon probleminin bir modelini sunar ve PoO'nun bu problemi ele alması, işlemleri optimize etme ve en iyi çözümü elde etme hedefine hizmet edebilir. Bu, PoO'nun optimizasyon odaklı bir yaklaşım benimsemesini sağlar.

Üçüncüsü, GSP gerçek dünyada birçok alanda kullanılan bir problem türüdür. Bu, PoO'nun somut bir uygulama senaryosuna sahip olmasını kolaylaştırabilir ve gerçek dünyadaki problemleri ele almak için pratik bir temel sunabilir.

3.6. GEZGİN SATICI PROBLEMİ (GSP)

GSP problemi tanımlanmak istenirse bir seyyar satıcının mallarını, n şehirde satmak istemesi ve öte yandan bu satıcı bu şehirleri mümkün olan en kısa şekilde ve her bir şehre maksimum bir kere uğrayarak seyahat etmek istemektedir. Amacı satıcıya en kısa yolu sunmaktır [53].

GSP, NP-zor bir problem olarak bilinir, yani polinom zamanlı bir algoritma ile çözülemeyen bir problem kategorisine aittir. Bu, GSP'nin çözümünün, problem boyutunun artmasıyla birlikte zorlaşacağı anlamına gelir [88].

GSP, kombinatorik optimizasyon alanında klasik bir problem olup, bir satıcının bir dizi şehri tam olarak bir kez ziyaret edip başlangıç noktasına geri dönmesi gerektiği bir durumu modellemektedir. Ana hedef, satıcının toplam seyahat mesafesini veya maliyetini minimize etmesidir. Bu problem, birçok farklı alanda, örneğin lojistik, taşımacılık ve üretim planlamada karşılaşılan birçok gerçek dünya problemine uygulanabilir [89].

GSP'nin iki ana varyasyonu vardır: Simetrik ve Asimetrik GSP'dir. Simetrik GSP'de, iki şehir arasındaki mesafe her iki yönde de aynıdır, yani şehir A'dan şehir B'ye olan mesafe ile şehir B'den şehir A'ya olan mesafe eşittir. Asimetrik GSP'de ise, iki şehir arasındaki mesafe her iki yönde de farklı olabilir.

GSP'nin çözümü için birçok farklı algoritma ve yöntem kullanılmaktadır. Bunlar arasında heuristik yöntemler, metaheuristik yöntemler ve tam çözüm yöntemleri bulunmaktadır.

Heuristik yöntemler, genellikle hızlı bir çözüm sağlar. Ancak her zaman optimal olmayabilir. Metaheuristik yöntemler, daha iyi çözümler bulma potansiyeline sahiptir ve genellikle daha büyük problem çeşitleri için kullanılır. Tam çözüm yöntemleri, optimal bir çözüm bulur, ancak hesaplama süresi problem boyutuyla birlikte artar ve bu nedenle genellikle yalnızca küçük veya orta ölçekli problem çeşitleri için uygundur.

GSP problemi optimizasyon konusunda derinliği olan bir problemdir. Matematiksel olarak 1930'larda formüle edilmiştir. Hesaplama karmaşıklığına göre çözümü NP-zor olan önemli bir optimizasyon problemidir. Bu nedenden dolayı GSP problemini etkin çözebilecek bir algoritma olmadığı düşünülmektedir. Yani bilgisayarlar çözümleme yaparken hesaplama sayılarının şehir sayıları arttıkça üstel olarak artmasıdır. Bu da bazı durumlarda belki yıllar alabilecek olmasına neden olur.

Problem çözüm adımları

- Başlangıç için seçilebilecek n adet şehir vardır.
- İlk şehire gelindiğinde n-1 adet değişik şehir arasında seçim hakkı vardır.
- İkinci şehire gelindiğinde n-2 adet şehir arasından şeçim hakkı vardır.

Sonuç olarak satıcı n! Adet şehir arasından seçim yapacaktır. Bu 100 şehir için $9{,}3x10^{157}$ tur etmektedir. Problemin karmaşıklığı $O(N^2 + 2^n)$ olarak belirlenmiştir. Şu ana kadar en iyi çözüm dinamik programlama ile yapılmıştır [88,90].

3.6.1. GSP'nin Çeşitleri

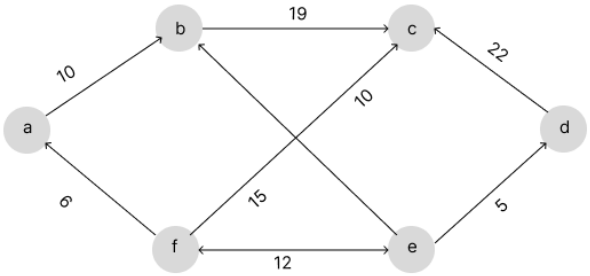
3.6.1.1. Simetrik GSP

Simetrik GSP, operasyonel araştırma ve kombinatorik optimizasyon alanlarında klasik ve önemli bir problemdir. Bu problem, bir satıcının bir dizi şehri ziyaret etmesi ve başlangıç şehrine dönmesi gerektiği durumu modellemektedir. Amaç, toplam seyahat mesafesini veya maliyetini minimize etmektir [91].

3.6.1.2. Asimetrik GSP

İki şehir arasındaki mesafe veya maliyet her iki yönde de farklı olabilir. Bu, pratikte, bir yönde daha fazla trafik veya daha fazla yol çalışması gibi faktörler nedeniyle bir rota üzerindeki seyahat süresinin veya maliyetinin diğer yönde farklı olabileceği anlamına gelir. GSP, hem teorik hem de pratik açıdan önemli bir optimizasyon problemidir. Çeşitli endüstrilerde ve uygulama alanlarında kullanılarak, operasyonel maliyetlerin ve zamanın

önemli ölçüde tasarruf edilmesine yardımcı olabilir. Ancak, GSP'nin çözümü, problem boyutunun artmasıyla birlikte zorlaşmaktadır, bu da daha sofistike ve etkili çözüm yöntemlerinin geliştirilmesini gerektirmektedir [90].



Şekil 3.2. Örnek GSP şeması

Çalışmada düğümlerin çözmesi için verilen problemin GSP olarak belirlenmiştir. Bunun nedeni problemi şehir sayısına göre zorluk derecesi artırılabilen bir problemdir. Ayrıca optimizasyon algoritmalarının kullanabilmesi için bir optimizasyon problemi gereklidir. Kısaca PoW ‘daki matematiksel özet hesaplamalarının yerine GSP optimizasyon hesaplamaları konularak blokzincir için yeni bir model sunulmuştur.

3.7. POO KONSENSÜS ALGORİTMASI

PoW algoritmasının birçok zayıf yönü vardır. Bunlar arasında en önemlisi düşük işlem gücü ve ademi merkeziyetçilik değerinin oldukça düşük olmasıdır. PoW algoritması, karma fonksiyonun zorluğunu artırmak üzere özellikle tasarlanmıştır. Bu, işlem gücü gereksinimlerini artırır ve bu nedenle daha yüksek enerji tüketimine yol açar. Bu nedenle, PoW tabanlı blokzincir ağları, yüksek enerji faturaları ve çevresel etkisi nedeniyle eleştirilmektedir. Başka bir zayıf yönü ise merkezsizliğinin eksikliğidir. PoW algoritması, işlem gücüne dayalı olarak blok üreten madencilere teşvikler sağladığı için büyük madencilik havuzları oluşabilir. Bu, ağın merkezsizliğini azaltır ve bazı madencilik havuzlarının ağ üzerinde baskın hale gelmesi sonucunu doğurabilir. Bu, blokzincir üzerinde kontrolü olan birkaç büyük oyuncuya bağımlılık yaratabilir ve ağın güvenliğini tehlikeye atabilir. Ayrıca, PoW algoritması işlem onaylama süresini uzatabilir. Yeni bir blok oluşturmanın süresi, karma fonksiyonun zorluğuna bağlı olarak değişebilir. Bu nedenle, işlem onaylama süresi değişkenlik gösterebilir ve bazen uzun zaman alabilir. Bu, blokzincir ağının işlem kapasitesini azaltabilir ve uygulama geliştiricileri için zorluklar

yaratabilir. Sonuç olarak, PoW algoritması, yüksek enerji tüketimi, merkezsizliğin eksikliği ve işlem onaylama süresindeki değişkenlik gibi zayıf yönleri nedeniyle eleştirilmektedir.

Tez çalışması kapsamında yeni bir proof mekanizması PoW'un yukarıda bahsedilen dezavantajlarını yok etmek için önerilmiştir. Tabiki bu tez çalışması kapsamında ortaya koyduğumuz yapının çok farklı optimizasyon problemleri ile kurgulanıp çözüm algoritmasında da diğer YZ algoritmalarının kullanılarak karşılaştırma analizlerinin yapılmasına olanak sağlamaktadır. Bu çalışma PoO mekanizmasının ortaya konmasında bir öncüllük oluşturmak ve diğer tüm araştırmacılara bu alanda çalışma yapabilmenin fikri ortaya konmaktadır. PoO'nun bir blokzincir modeli oluşturulmasındaki kullanımı, istikrar üzerinde birkaç olumlu etkiye sahiptir. İlk olarak, PoO, yeni bloklar oluşturmak için PoW ile karşılaştırıldığında enerji tüketimini ve gereken hesaplama kaynaklarını azaltır. Bu, madencilerin karmaşık hesaplamalı problemleri çözme yarışına girmeleri için teşviki azaltır ve daha fazla katılımcının ağa katkıda bulunmasına izin verir. İkinci olarak, PoO, blok ödüllerinin daha adil bir dağılımını sağlar ve madencilik gücünün merkezileşmesini azaltır, bu da daha güvenli ve merkezi olmayan bir blokzincir ağına yol açabilir. Üçüncü olarak, PoO'daki GSP'nin kullanımı, PoW'dan daha esnek ve uyarlanabilir bir yaklaşımla uzlaşma sağlar. Genel olarak, PoO'nun kullanımı, bir blokzincir ağının verimliliğini, adil olmasını ve güvenliğini artırabilir, daha istikrarlı ve sürdürülebilir bir sisteme yol açabilir. Bu durumlarında ispatı tezin uygulama ve bulgular başlığında uygulamalı olarak test edilerek ortaya çıkarılmıştır.

Bu tez çalışması kapsamında PoO, blokzincir sistemlerinde kullanılan PoW uzlaşı algoritmasının sınırlılıklarından motive edilmiştir. PoW, hesaplamaya dayalı olarak yoğun ve enerji tüketimine eğilimli olduğu bilinmektedir. Bu da ölçeklenebilirliğini ve daha geniş bir katılımcı yelpazesine erişilebilirliğini sınırlar. PoO, Gezgin Satıcı Probleminin (GSP) ve çözüm algoritması olarak da genetik algoritmaları kullanarak uzlaşma elde etmek için yeni bir yaklaşım tanıtarak bu sınırlılıkları ele almaya çalışır. GSP, onlarca yıldır incelenen bilinen bir optimizasyon problemidir. Genetik algoritmalar da çeşitli optimizasyon problem alanlarına başarıyla uygulanan güçlü bir optimizasyon tekniğidir. Bu teknikleri kullanarak, PoO, uzlaşmayı daha verimli ve ölçeklenebilir bir şekilde gerçekleştirebilir niteliğe sahiptir. Bu da onu PoW'ya göre umut verici bir alternatif oluşturduğu tez çalışması kapsamında ortaya konmuştur. Ek olarak, PoO'nun yaklaşımı, katılımcılar için ödüllerin dağılımında bir dereceye kadar adillik

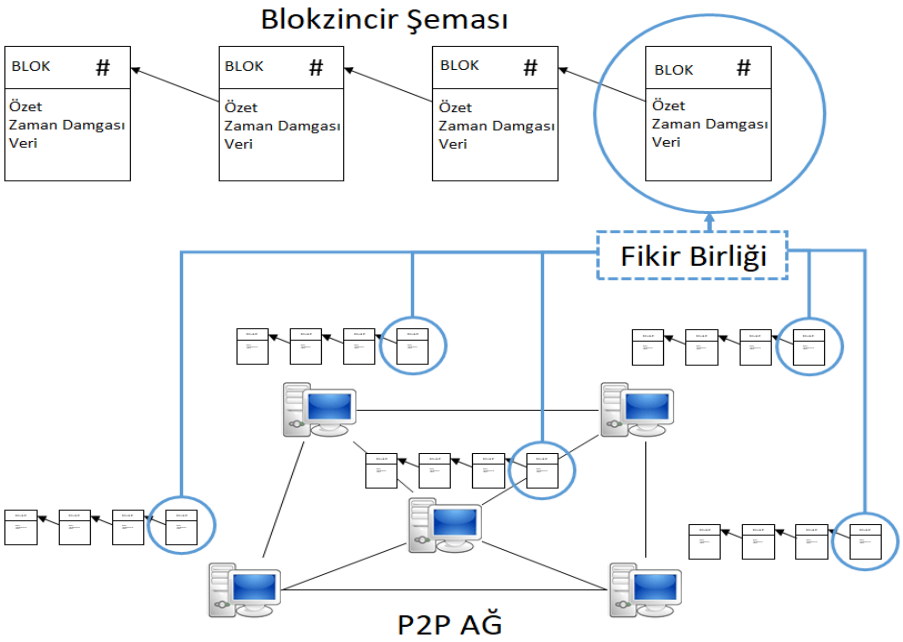
sağlamaktadır. Bu da onun bir uzlaşma algoritması olarak cazibesini daha da artırır. Genel olarak, PoO'nun motivasyonu, blokzincir sistemlerinin ölçeklenebilirliğini, erişilebilirliğini ve adil olmasını artırmaktadır. Böylece PoO mekanizması ve diğer PoO mekanizmalarının gelecekte yaygın kabul görmesi ve başarısı için kritik öneme sahiptir. İlerleyen bölümlerde aşağıdaki konular detaylı bir şekilde açıklanmıştır. PoW'de harcanan işlem gücünü uygun verimli sonuçlar bulmak için optimizasyon algoritmalarına yönlendirmeyi amaçlayan yeni bir blokzincir uzlaşı modeli olan PoO detaylandırılmıştır.

Blokzincir ağında genel olarak, müşteriler, yani düğümler, blok oluşturmak için yarışırlar. Bu çalışmada, çözülmesi gereken problem türü olarak GSP problemi örnek olarak seçilmiştir. PoO modeli çerçevesinde çözüm algoritması olarak genetik algoritma dikkate alınmıştır. Belirlenen popülasyon, iterasyon ve diğer parametrelerle problemin çözülmesi gerçekleştirilmiştir. Ayrıca, algoritma ve model yapısı merkezsizlik aracılığıyla test edilerek ademi merkeziyet unsuru incelenmiştir.

Çalışma genetik algoritmanın blokzincir konsensüs mekanizmasına katkısı incelenmiştir. İlerleyen çalışmalarda farklı optimizasyon problemleri ve sezgisel algoritmalar eklenebilir olarak değerlendirilmiştir. Kullanılan genetik algoritma kısaca anlatılmıştır. Çalışmada amaç GSP probleminin optimizasyonu değildir. Zaten literatürde bununla ilgili birçok çalışma bulunmaktadır. Bu çalışmada asıl amaç genetik algoritmanın konsensüs mekanizmasına adaptasyonunun sağlanmasıdır.

Blokzincir yapısındaki literatürde belirtilen problemlerin çözümü için yeni konsensüs yöntemi ortaya konulmuştur. Amaç düğümlerin genetik algoritma kullanarak blokzincirdeki problemleri çözmesine yardımcı olmasını hedeflemektedir. PoO temel adımları aşağıda listelenmiştir.

1. Düğümler çözüm üretmesi için bir GSP problemi belirlenir.
2. Düğümler problemleri çözer. Uygunluk değeri, yoğunluk ve süre değerlenir geri döndürürler.
3. Döndürülen değerler üzerinde bir denklem üzerinden tekrar değerlendirilirler. Bu değerlendirme üzerinden bloğu oluşturacak düğüm belirlenir.
4. Belirlenen düğüm blok oluşturur. Diğer düğümlere oluşturulan blok dağıtılır.



Şekil 3.3. PoO Genel şema

PoO'da 2 genel amaç hedeflenmiştir. Blok süresinin PoW'a nazaran kontrol altında tutulmasını sağlamak ki böylece daha yüksek işlem çıktısı elde edilebilir. Diğeri ise decentralization donanım konfigürasyonu güçlü olan makinenin blokzinciri domine edilmesinin önüne geçilmesidir. Böylelikle adem-i merkeziyetçiliği daha yüksek bir sistem önerilmektedir.

4. BULGULAR VE TARTIŞMA

Önerilen PoO onay mekanizmasında veri düğümü ve konsensüs düğümü olmak üzere 2 adet düğüm tipi bulunur. Veri düğümü, konsensüs düğümlerine GSP görevi sağlamak için kullanılır. Konsensüs düğümü ise GSO problemlerini belli iterasyon, popülasyon, çaprazlama ve mutasyon oranı ile çözmekle görevlidir. Kazanan düğüm veri düğümü tarafında sunulan ödülü alır. Geliştirilen model çözüm üreten düğümlere ödül dağıtmanın yanısıra, şifrelenmiş veriler ve transfer işlemlerinin merkezi olmayan bir veri deposu olarak da işlev görür. Sistemin ana bileşenlerini aşağıdaki gibi oluşturulmuştur.

4.1. VERİ DÜĞÜMLERİ

Veri düğümlerinin amacı fikir birliği düğümlerine GSP problemi çözme görevi sağlamaktır. Veri düğümü tarafından sağlanan GSP görevi içinde iterasyon, popülasyon, çaprazlama, mutasyon, optimal çözümü ve ödülü içermektedir. Ödül, veri düğümünün hesabından, en iyi genelleme performansına sahip kazanan mutabakat düğümüne ödeme yapacak olan sanal bir rezervuar hesabına hemen aktarılır. Dağıtık bir sistemdeki düğümlerin mükemmel senkronize edilmiş saatleri olmasa da zaman tutarsızlığı bir optimizasyon problemi çözmek için gereken süreden daha azdır. Çözülen problemin verileri düğümlerin rekabetçi bir şekilde oluşturulan blokta saklanır.

4.2. KONSENSÜS DÜĞÜMLERİ

Aslında madenci olan konsensüs düğümleri ağdaki asıl iş gücü olarak tanımlanabilir. Madenciler, veri düğümleri tarafından verilen GSP görevlerinin çözmek için rekabet eder ve sonuç olarak ödüllendirilir. Düğümler GSP görevlerini çözmek için genetik algoritma kullanırlar. Tabiki bu tezde önerdiğimiz PoO olarak isimlendirilen yeni onay mekanizmasında problem türü olarak istenen herhangi bir optimizasyon problemi seçilip çözüm algoritması olarak da istenen optimizasyon algoritmaları seçilip geniş çaplı ağ üzerinde n adet farklı senaryolar oluşturularak geniş bir blokzincir üzerinden de yapılabilir. Ancak bu tez çalışmasında ağ ve donanım olarak sınırlı kaynaklar var olduğundan geniş uzunlukta ağ yapısı oluşturulamamıştır. Tabi ki önerisini yaptığımız yeni konsensüs mekanizmasının başka araştırmacılar tarafından yapılarak geniş bir literatür oluşturulmasının mümkün olduğunu düşünülmektedir. Konsensüs düğümlerinin

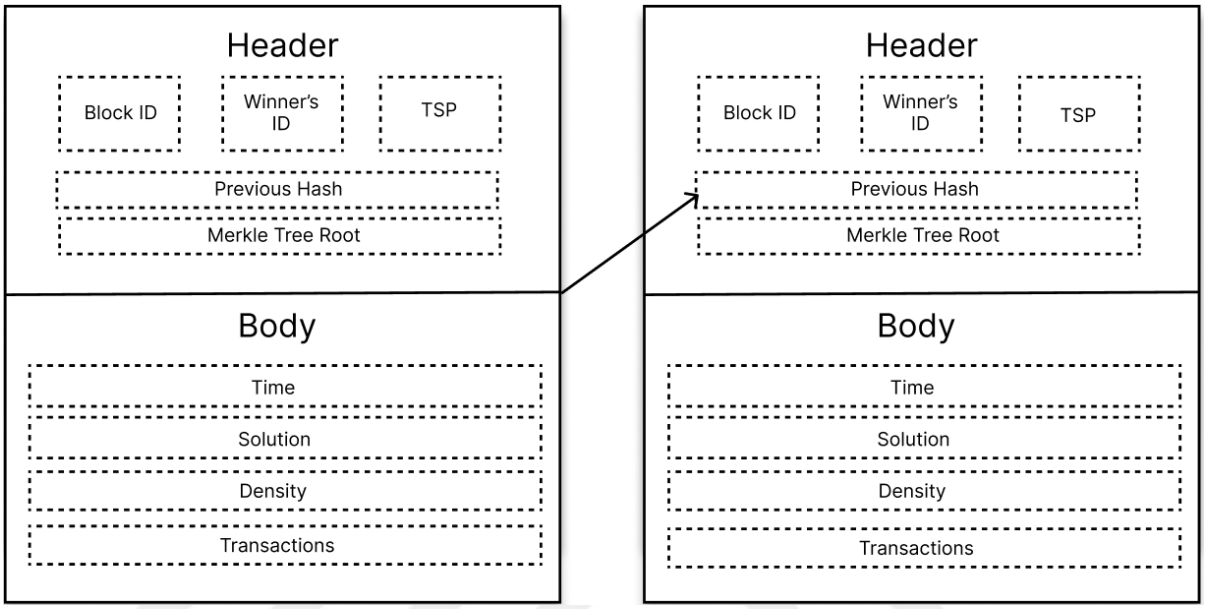
davranışı Algoritma 1 ‘de açıklanmıştır.

Algoritma 1. PoO

Girdiler: İterasyon: Genetik algoritma için iterasyon
Popülasyon: Genetik algoritma için popülasyon
Çaprazlama: Çaprazlama oranı
Mutasyon: Mutasyon oranı
GSPgörev: Düğümlerin çözmesi için GSP problemi
Output: blok: Yeni oluşturulmuş blok
While true
GSP görevi görevini al
Uygunluk değerinin hesapla
Yoğunluk değerini hesapla
Süre değerini hesapla
Kazanmakatsayisini denklem 4.1’e göre hesapla
Kazanma katsayisi en iyi olanı belirle
Katsayi diğer düğümlerden iyiyse blok oluştur.
Blok= Blokolustur(algoritma,uygunluk,yoğunluk,sure)
Blok diğer düğümlere gönder
İf yeni_blok geldiyse then
Kabulet(yeni_blok)
end while

Düğümler çözecekleri GSP problemlerini alır ve çözümleri üretmeye aynı anda başlarlar. Blok oluşturmak için sadece iyi çözüm bulmak yetmez. Aynı zamanda genetik algoritma genlerinin çözüm etrafında ne derece toplandığına ve çözümün ne kadar kısa sürede bulunduğunu göz önüne alınır. Madenciler, genetik algoritma kullanarak GSP problemlerine çözüm ararlar. Çözüm bulan konsensüs düğümü kazanma katsayısını denklem 4.1’e göre hesaplayıp diğeri konsensüs düğümlerine gönderir. Düğümler sıralama yapıp kazanma katsayısı en yüksek olan düğümü kendi aralarında seçerler. Kazanan düğüm kazandığını ilan eder ve bloğunu yayınlar. Sonrasında düğüm ürettiği bloğu diğer düğümlere yayar. Bloğu kendi oluşturduğunu ilan eder. Şekil 4.1’de örnek PoO blok verisinin yapısı verilmiştir. Bloğun veri yapısı, normal blokzincirdeki benzer bir başlık ve gövdeden oluşur. Blok başlığı kazanan kimliğini, GSP problemini, önceki

blok özetini ve merkle ağacı kökünü içermektedir. Blok gövdesinde GSP çözümünün kaç saniyede bulunduğu, çözüm yoğunluğunu ve çözümü içermektedir. Transactions kısmında doğrulanan işlemler bulunmaktadır. Merkle ağacı kökü ise eşler arası bir ağdan veri bloklarınını orijinal bir şekilde alınarak, sahteciliği önlemek için kullanılır. Hash yani şifreleme yapar. Bu yüzden özet ağacı da olarak bilinir.



Şekil 4.1. PoO Blok şeması

4.2.1. Bloğu oluşturacak düğümün belirlenmesi

Blok oluşturma işlemi blokzincirdeki ademi merkeziyetçiliği sağlanması için yapılır. Amacı işlemlerin güvenilir bir kanaldan doğrulanmasını sağlamaktır. PoW algoritması işlem doğrulamak yani blok oluşturmak için düğüm belirlerken, hashing algoritmalarını kullanmaktadır. Aşağıdaki denklem 4.1'de hangi istemcinin PoO kurgusu içerisinde blok oluşturma hakkına sahip olduğunu gösteren yapıdır. Bu denklemdeki katsayılar tamamen kullanıcılar tarafından farklı senaryoların oluşturulabilmesi için değişkenlik gösterebilir. Bu çalışmada en önemli unsurlardan biri olan optimizasyon probleminin çözülmesine dair en uygun çözümü gerçekleştiren istemcinin blok oluşturma hakkına sahip olması hedeflenmiştir. Ancak PoO kurgusunun temel prensiplerinden olan ağın domine edilmesini engellemek için çözüm bulma süresi ve yoğunluk parametrelerinden belli etki dereceleri (katsayı) ile sisteme dahil olmaları sağlanmıştır. Deneylerde algoritmaların uygunluk ve yoğunluk değerlerinin grafikleri ilerleyen kısımlarda verilmiştir.

$$KazanmaKatsayısı = 0.6\ x\ enuyguncozum\ + 0.20\ x\ \text{çözümü}bulmasuresi + 0.20x\ yogunluk \quad (4.1)$$

4.2.2. Kazanma katsayısının belirlenmesinde kullanılan değerler

4.2.2.1. Uygunluk değeri

Uygunluk değeri yeni popülasyonun bireylerinin oluşturulmasında kullanılan bir araçtır. Her iterasyonda popülasyonun bireylerin mevcut değerleri hesaplanır. Uygunluk değeri sadece çözüme ne kadar yakın olduğunu göstermez aynı zamanda optimal çözüme yakınlığı da gösterir.

4.2.2.2. Yoğunluk oranı

Yoğunluk oranı problemin çözümümün ne kadar iyi yapıldığının göstergesidir. Popülasyonun çözüm etrafında ne derece toplandığının hesaplanmasıdır. Formülü Öklid denklemi üzerine kurgulanmış ve denklem 4.2'de verilmiştir[92]. 0 ile 1 arasında bir değer almaktadır. Olabilecek en iyi değer 1 'dir

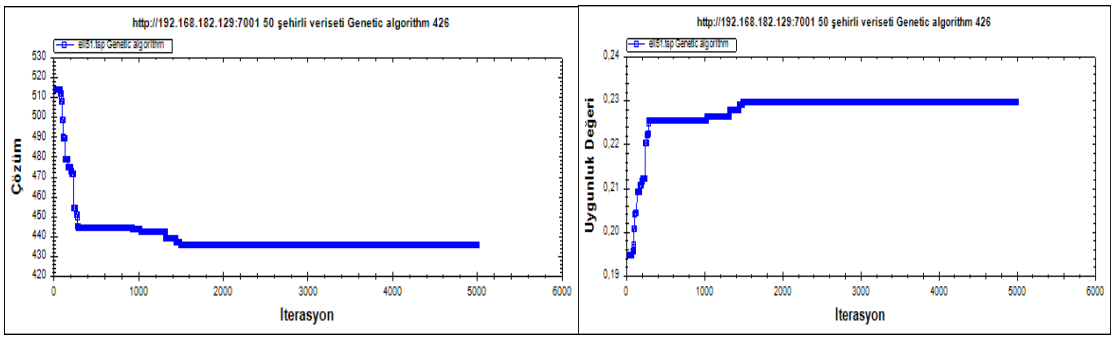
$n =$ dizi elemanı sayısı

$i =$ dizi elemanı

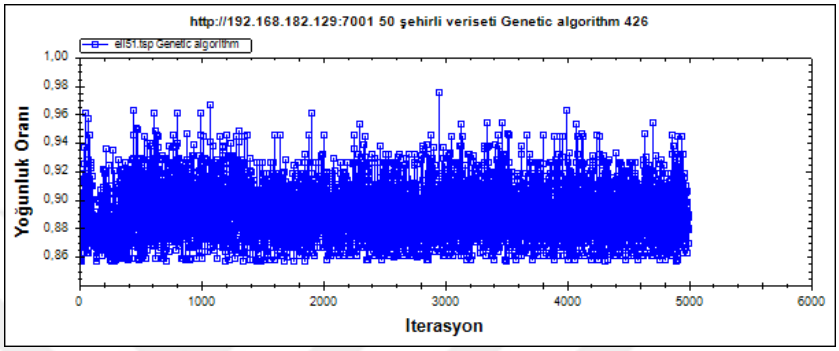
$n_{best} = dizinin\ en\ iyi$ çözümü

$$yogunluk\ oranı = \sqrt{\left(n_{i[1]} - n_{best[1]}\right)^2 + \left(n_{i[2]} - n_{best[2]}\right)^2 + \cdots + \left(n_{i[m]} - n_{best[m]}\right)^2} \quad (4.2)$$

Madencilerin çözümleri, uygunluk ve yoğunluk grafikleri aşağıda Şekil 4.2-4.31'da gösterilmiştir.

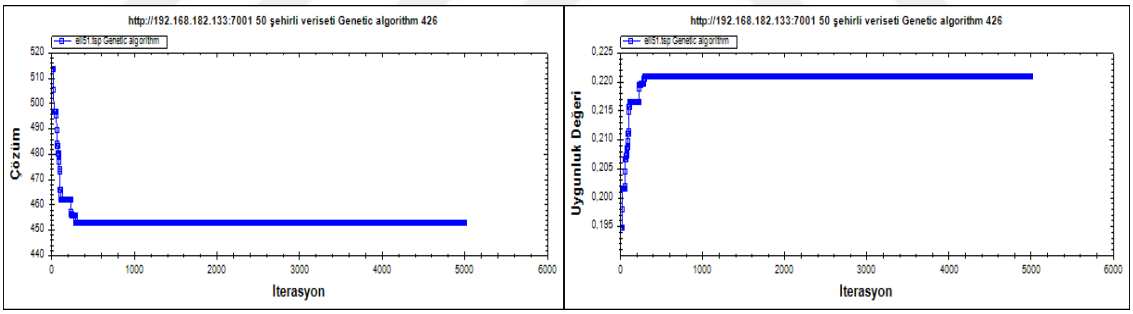


a) b)

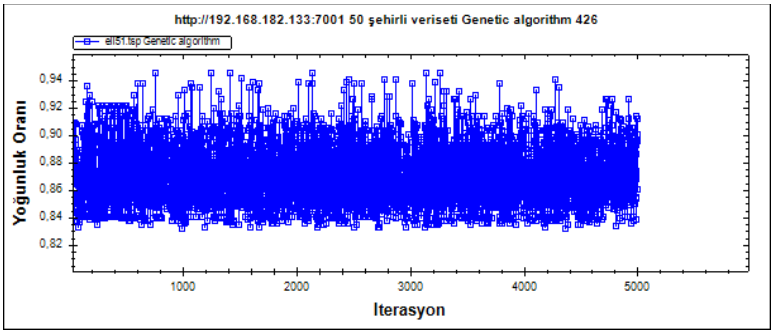


c)

Şekil 4.2. 50 şehirli problem için 129'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

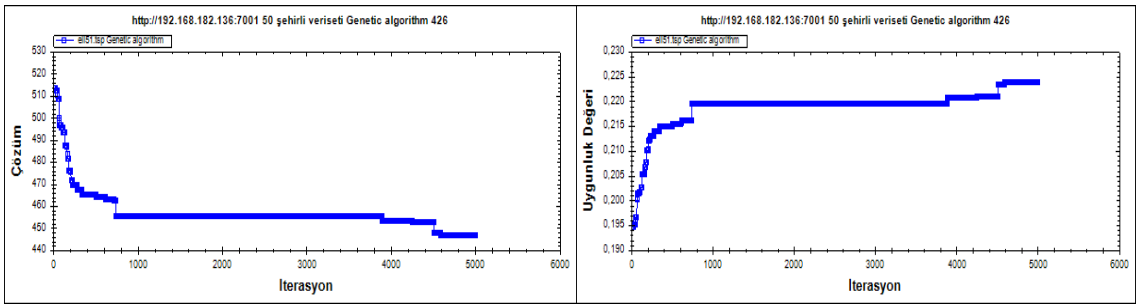


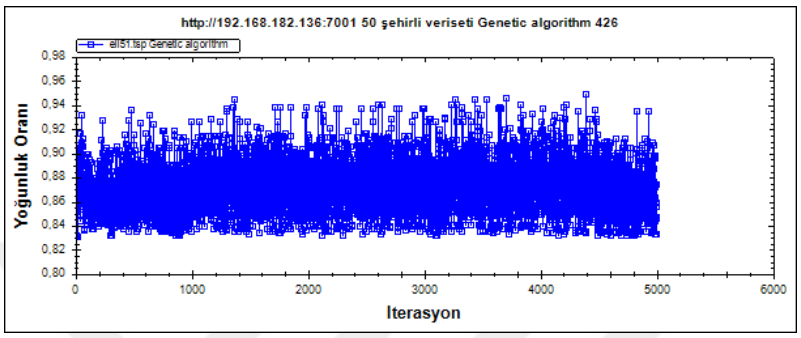
a) b)



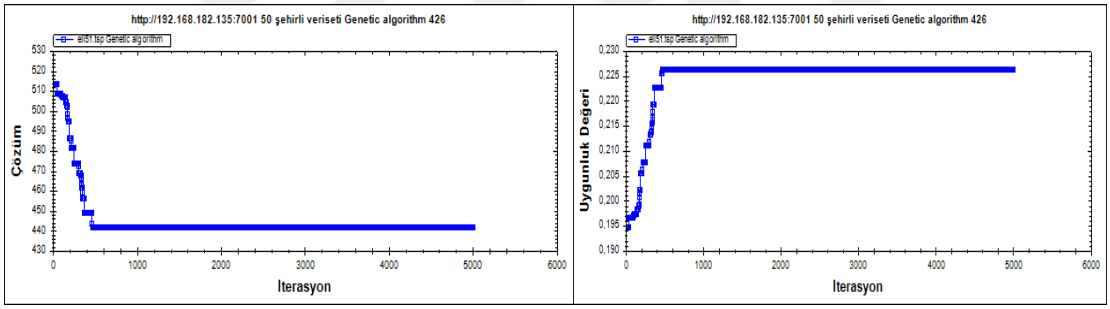
c)

Şekil 4.3. 50 şehirli problem için 133'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

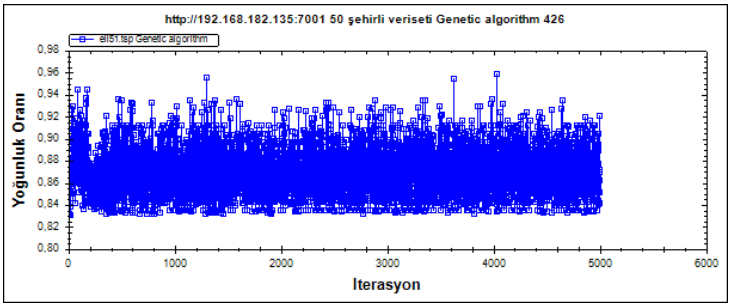




Şekil 4.4. 50 şehirli problem için 136'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

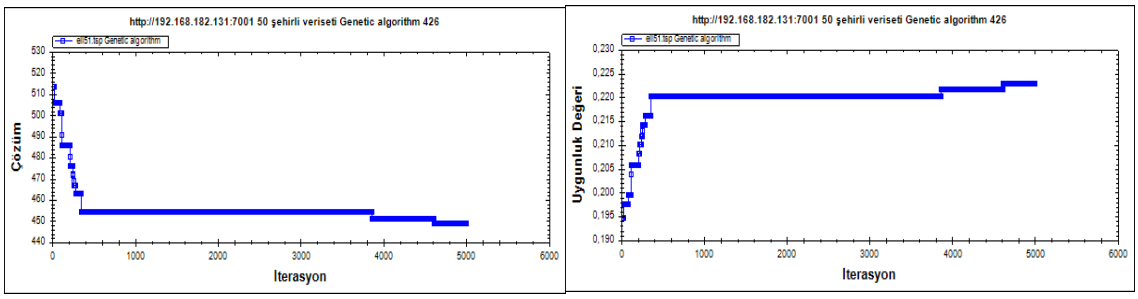


a) b)

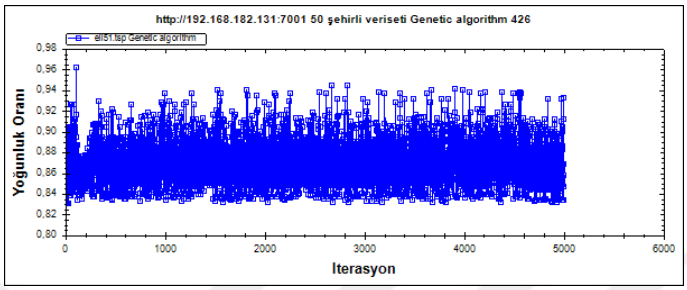


c)

Şekil 4.5. 50 şehirli problem için 135'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

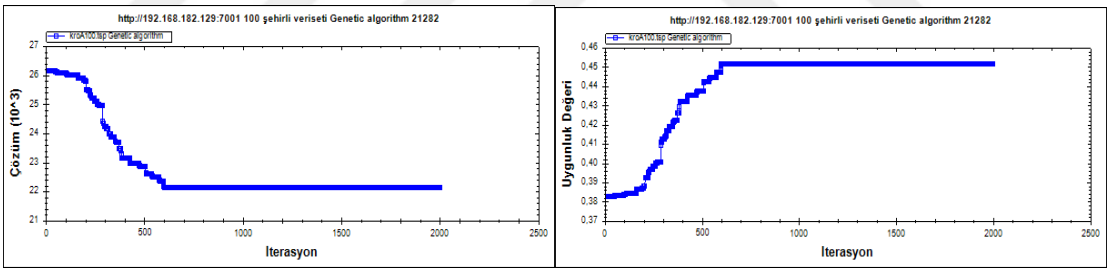


a) b)

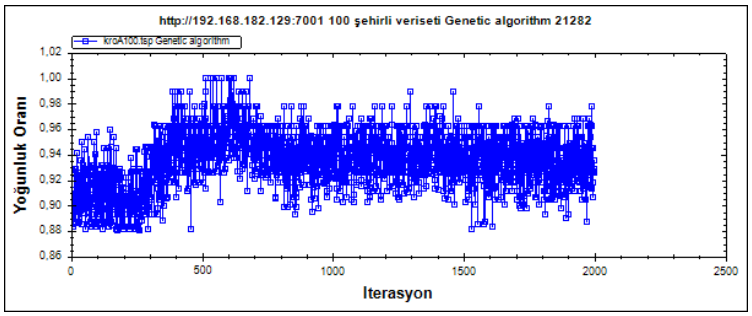


c)

Şekil 4.6. 50 şehirli problem için 131'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

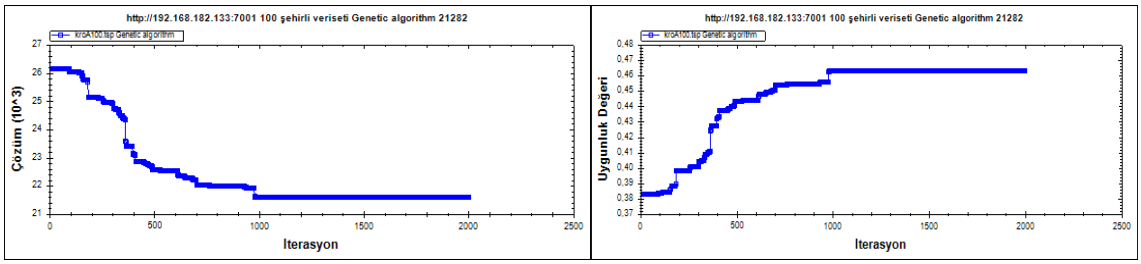


a) b)

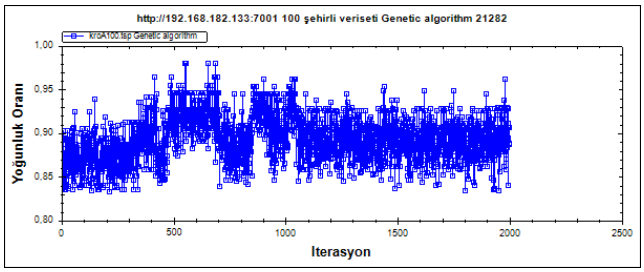


c)

Şekil 4.7. 100 şehirli problem için 129'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

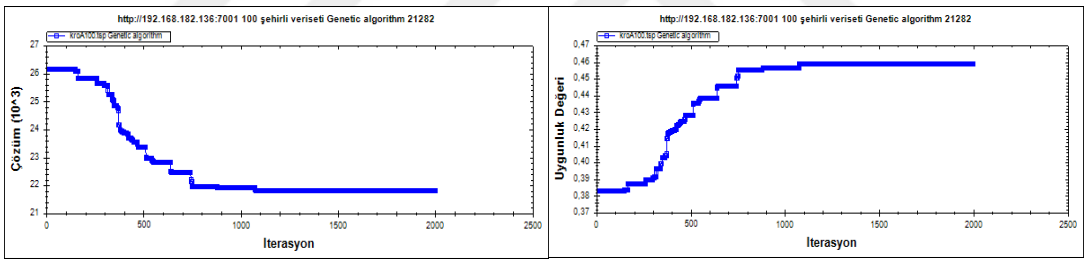


a) b)

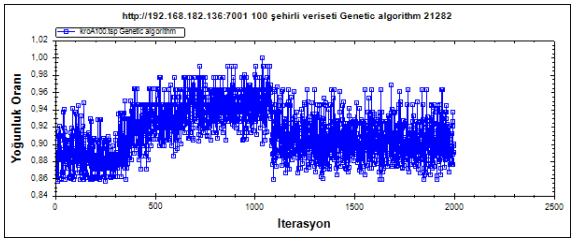


c)

Şekil 4.8. 100 şehirli problem için 133'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

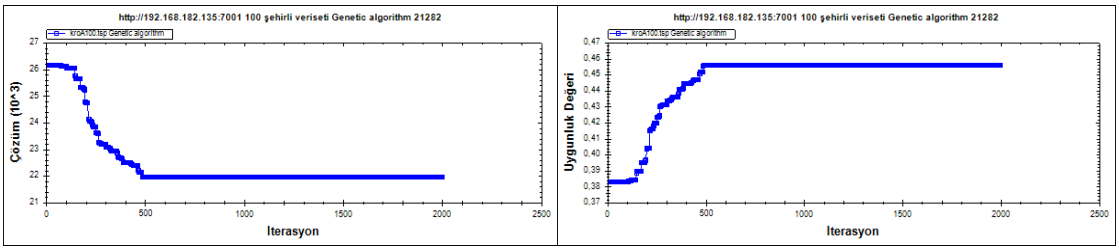


a) b)

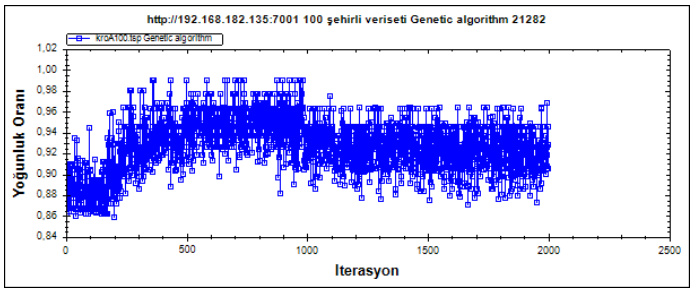


c)

Şekil 4.9. 100 şehirli problem için 136'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

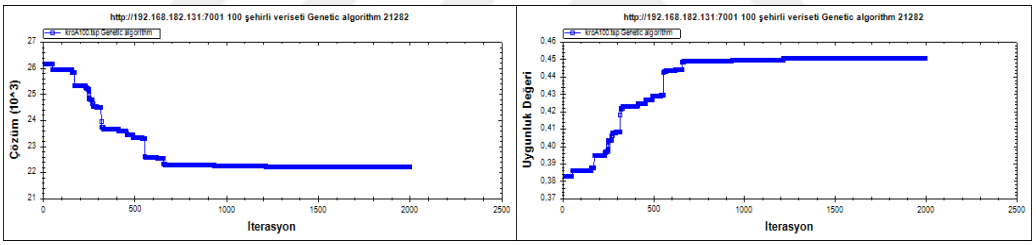


a) b)

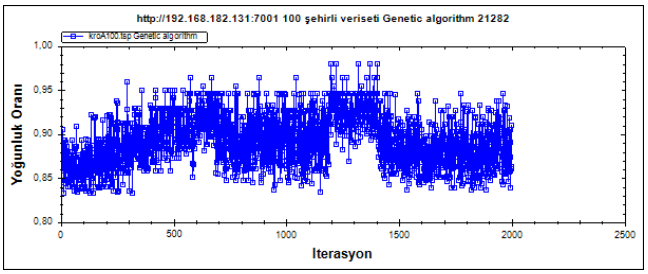


c)

Şekil 4.10. 100 şehirli problem için 135'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

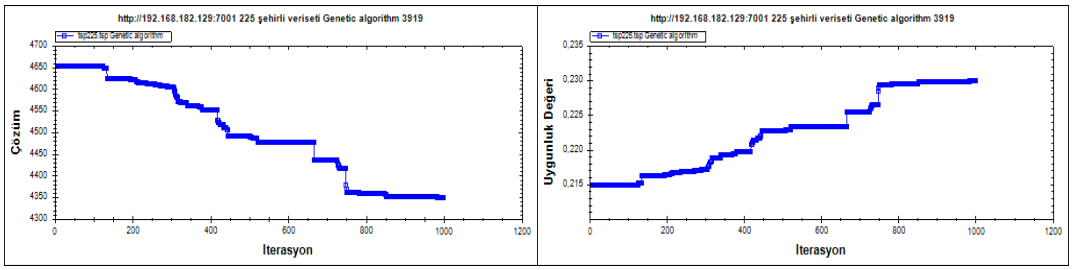


a) b)

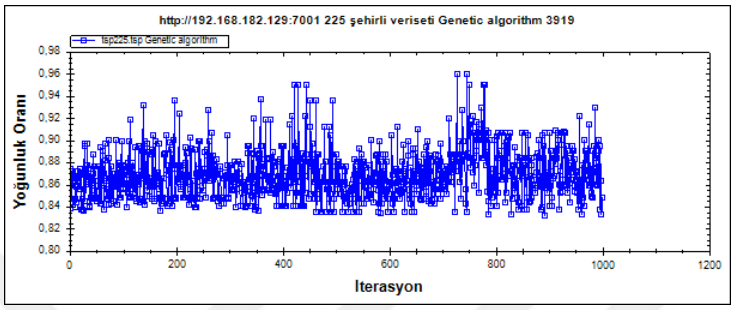


c)

Şekil 4.11. 100 şehirli problem için 131'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

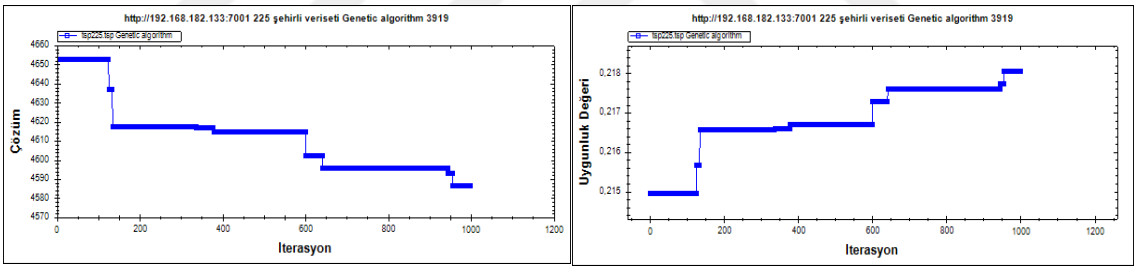


a) b)

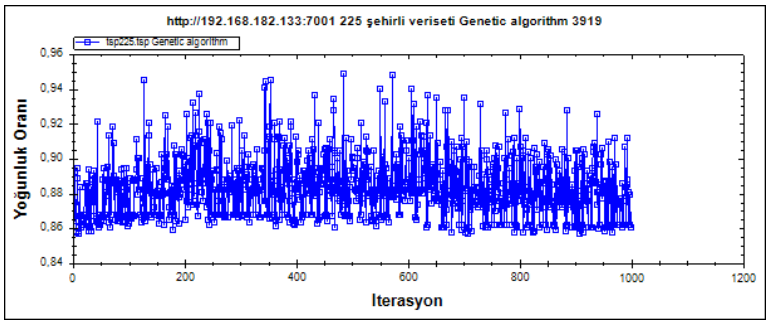


c)

Şekil 4.12. 225 şehirli problem için 129'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

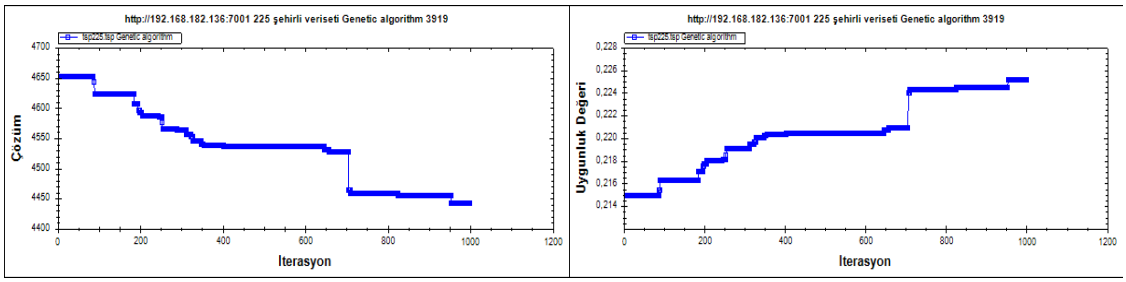


a) b)

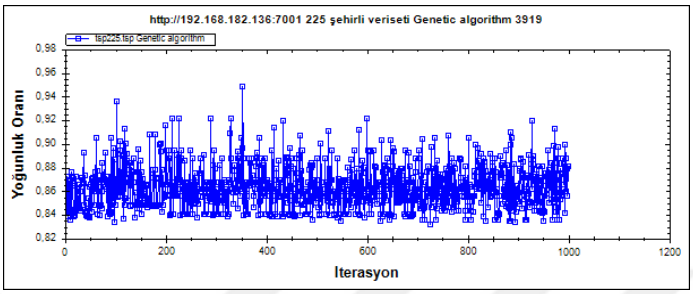


c)

Şekil 4.13. 225 şehirli problem için 133'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

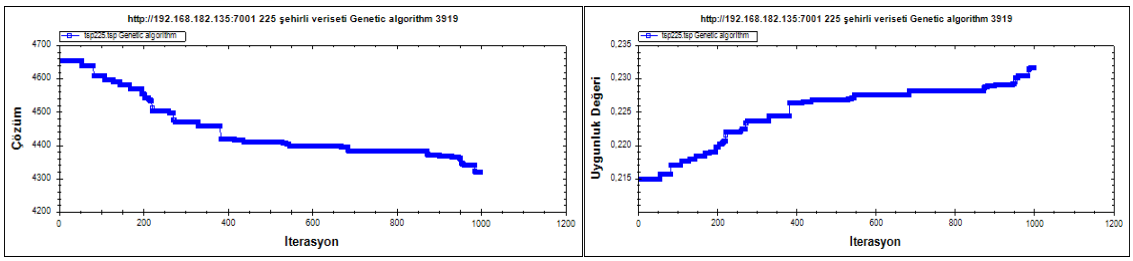


a) b)

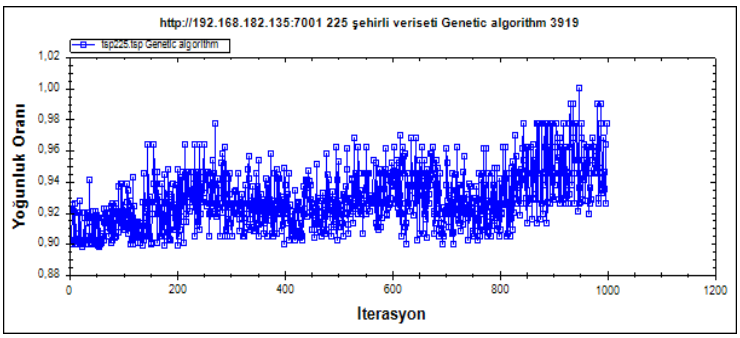


c)

Şekil 4.14. 225 şehirli problem için 136'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

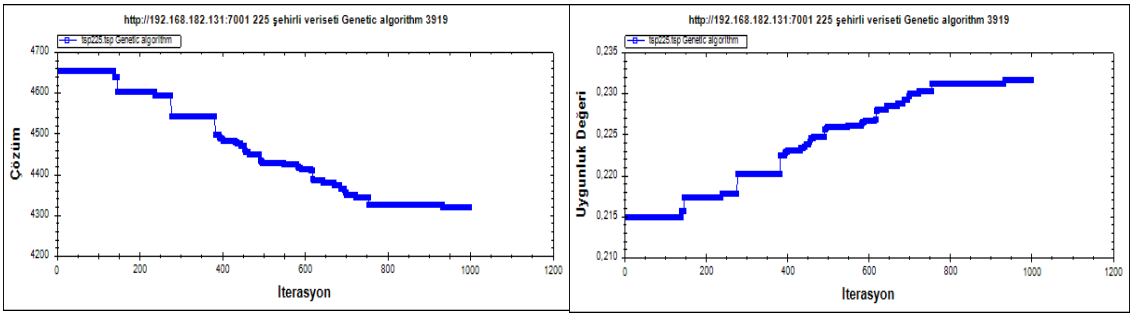


a) b)

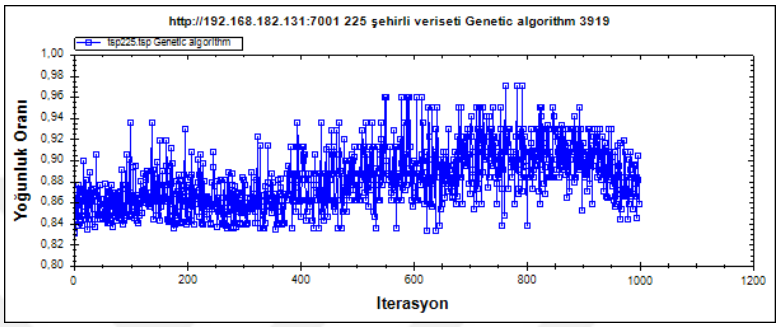


c)

Şekil 4.15. 225 şehirli problem için 135'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

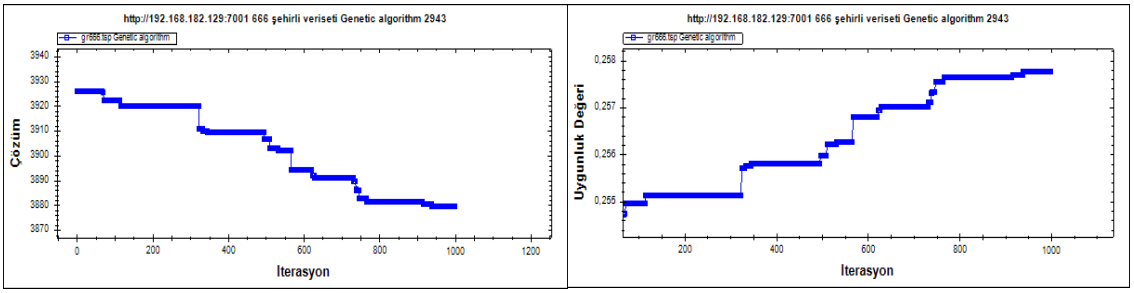


a) b)

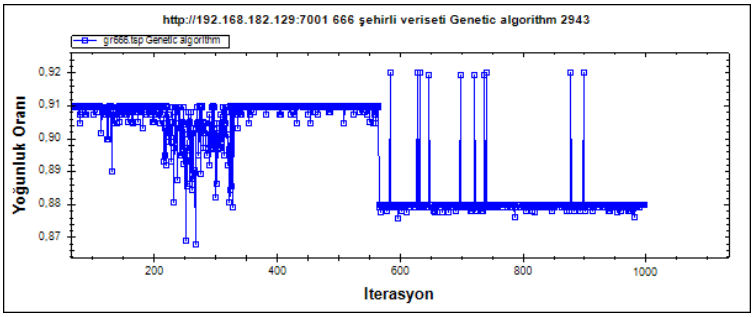


c)

Şekil 4.16. 225 şehirli problem için 131'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

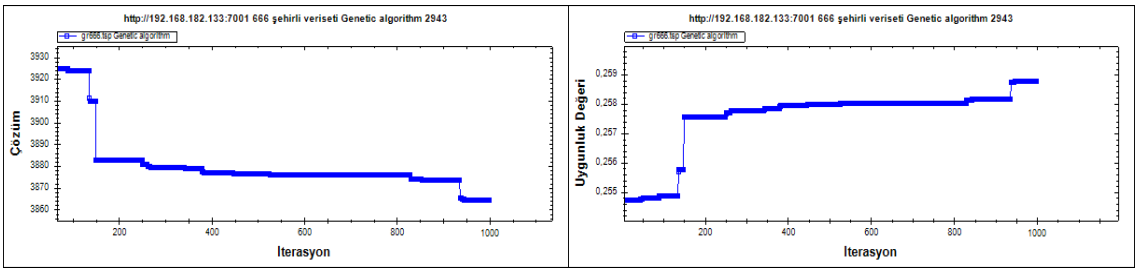


a) b)

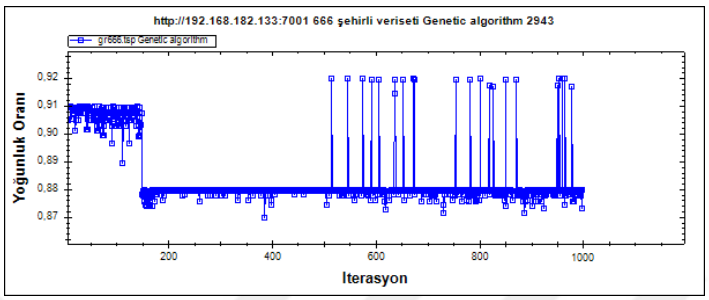


c)

Şekil 4.17. 666 şehirli problem için 129'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

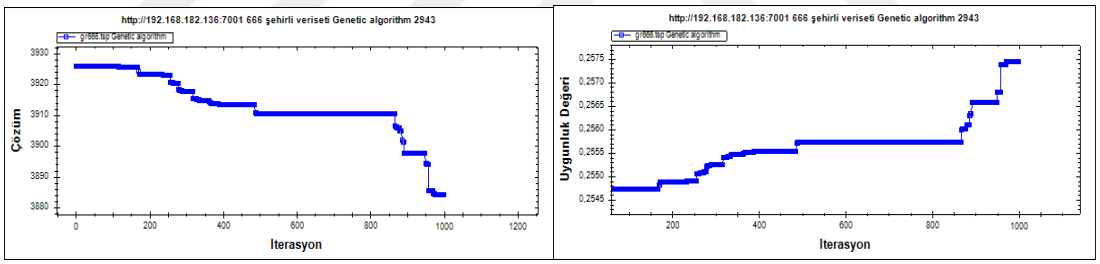


a) b)

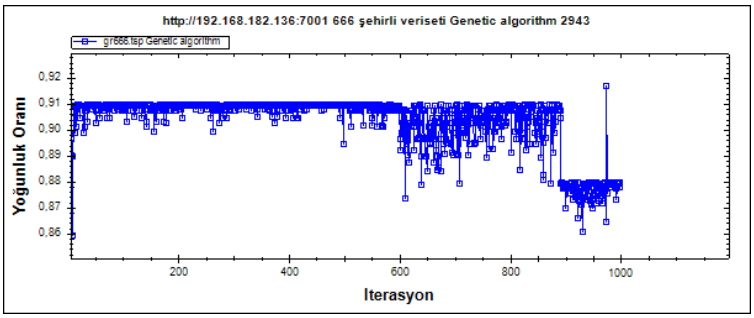


c)

Şekil 4.18. 666 şehirli problem için 133'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

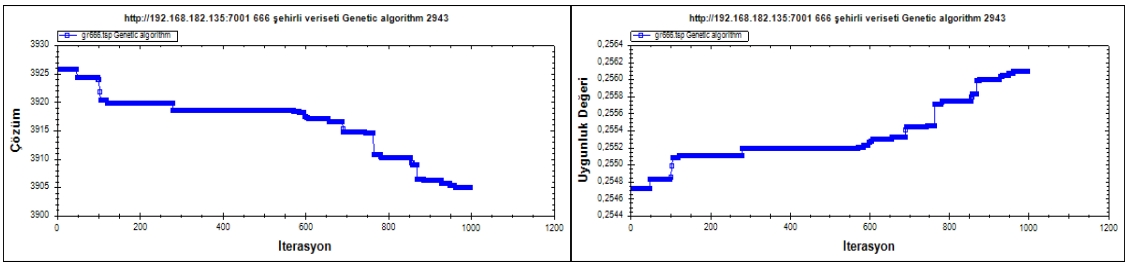


a) b)

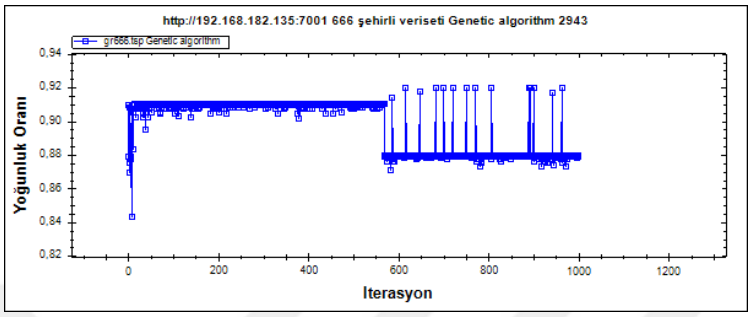


c)

Şekil 4.19. 666 şehirli problem için 136'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

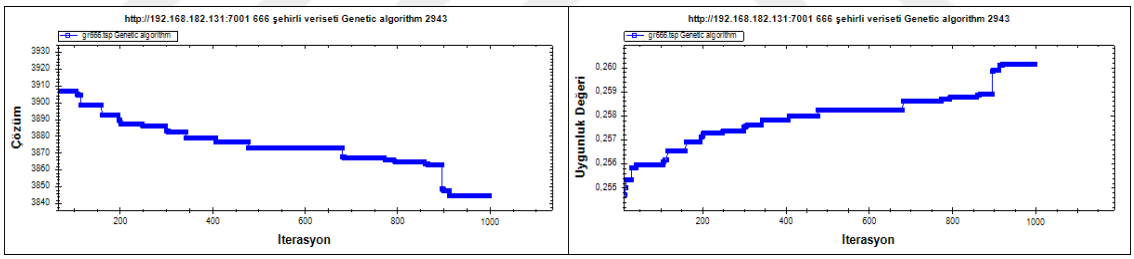


a) b)

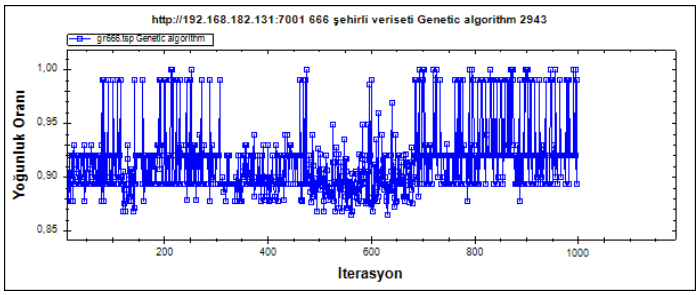


c)

Şekil 4.20. 666 şehirli problem için 135'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

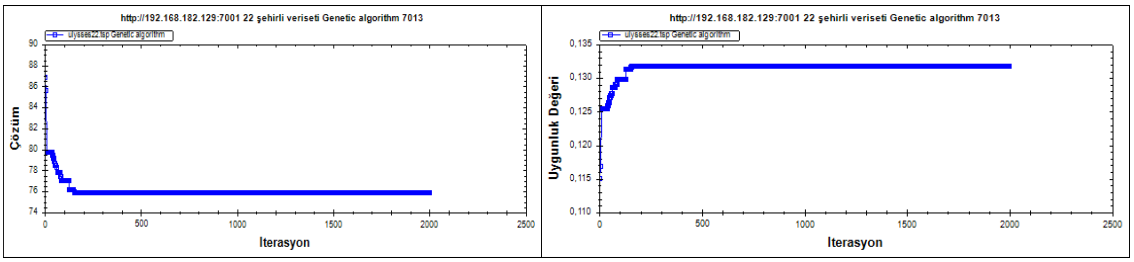


a) b)

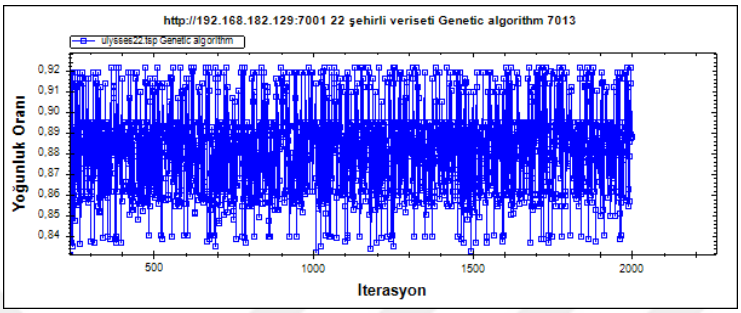


c)

Şekil 4.21. 666 şehirli problem için 131'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

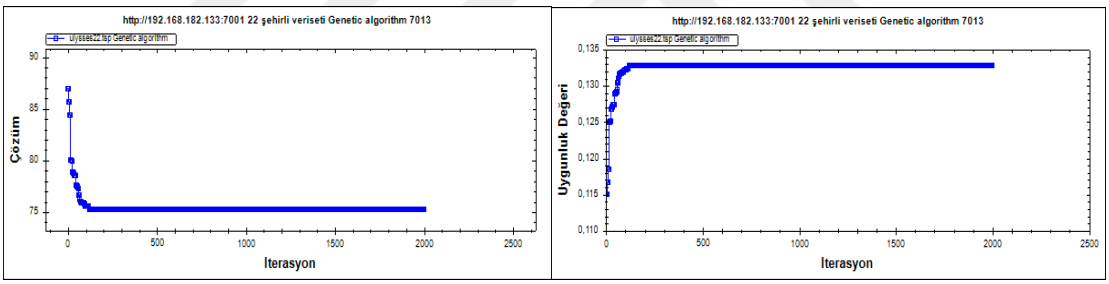


a) b)

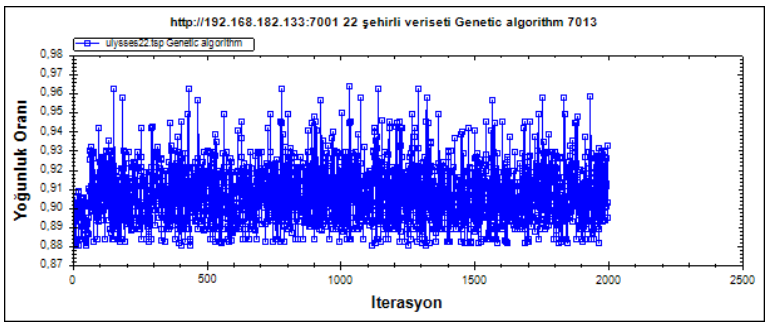


c)

Şekil 4.22. 22 şehirli problem için 129'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

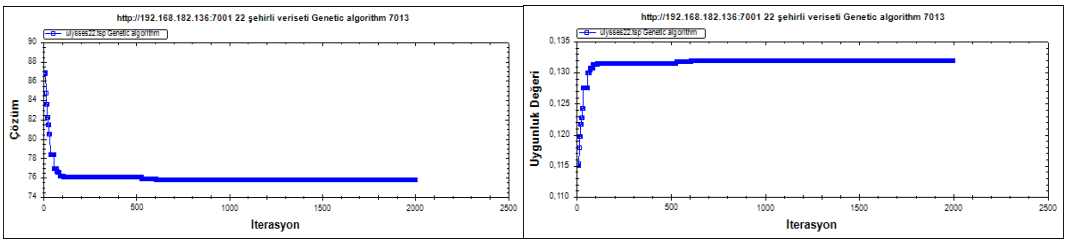


a) b)

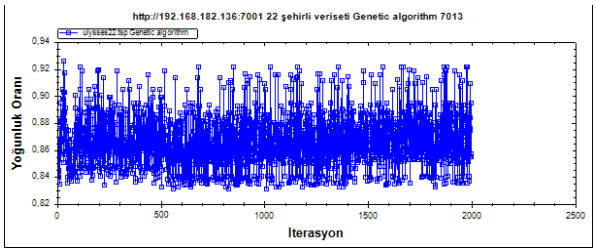


c)

Şekil 4.23. 22 şehirli problem için 133'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

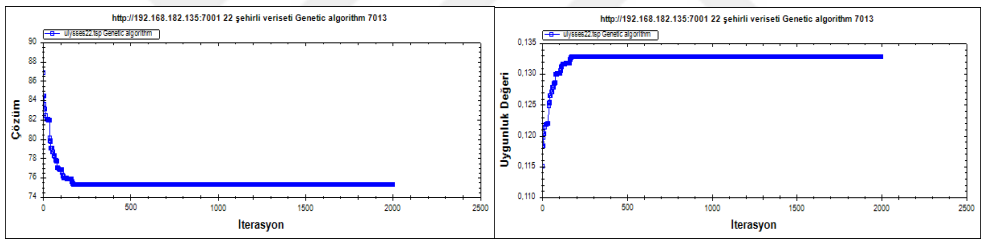


a) b)

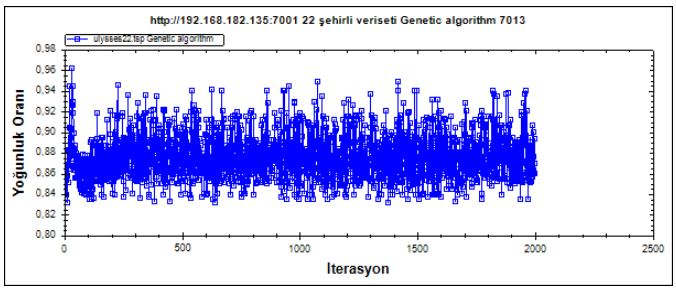


c)

Şekil 4.24. 22 şehirli problem için 136'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

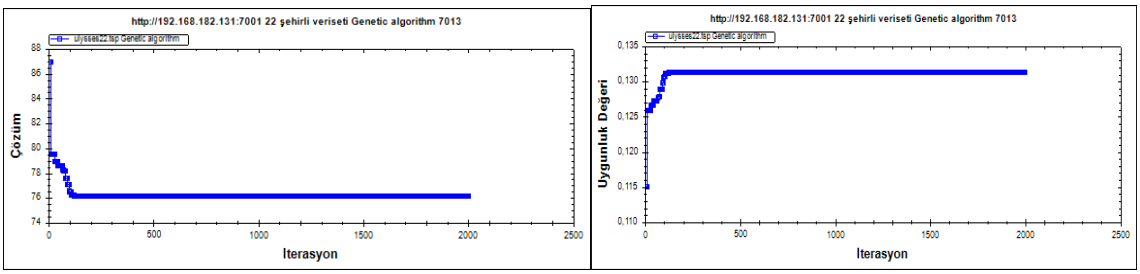


a) b)

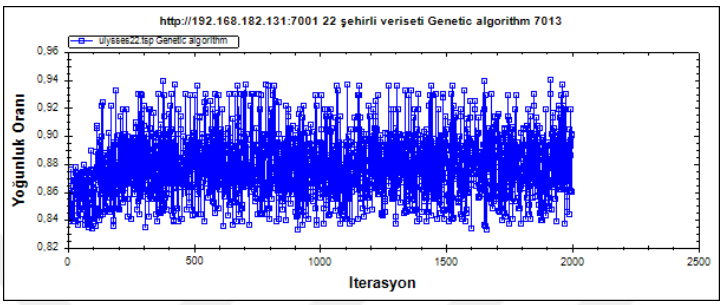


c)

Şekil 4.25. 22 şehirli problem için 135'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

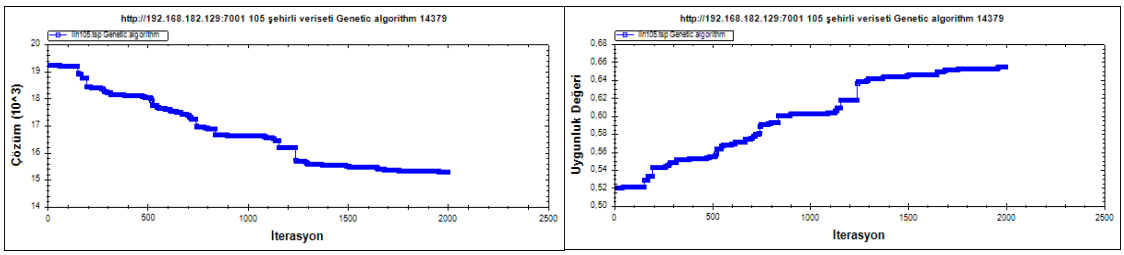


a) b)

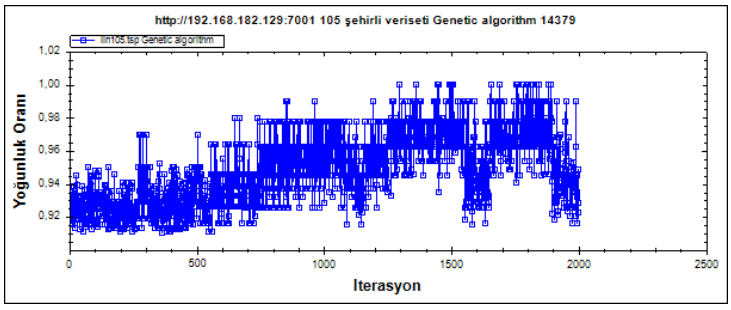


c)

Şekil 4.26. 22 şehirli problem için 131'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

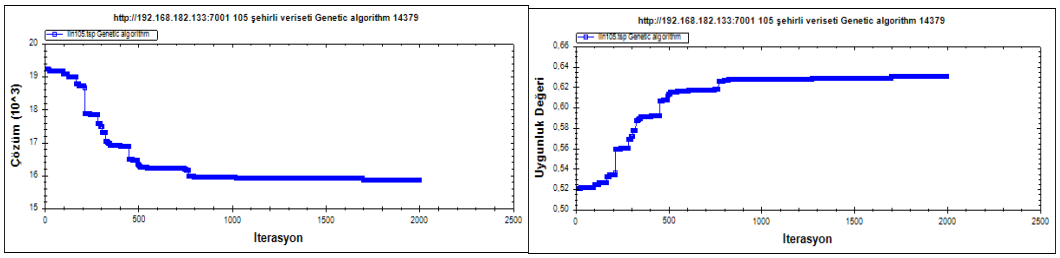


a) b)

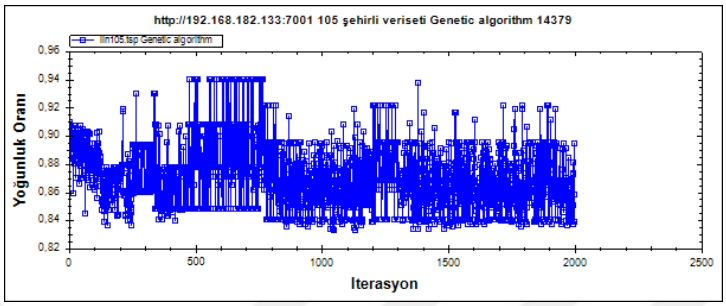


c)

Şekil 4.27. 105 şehirli problem için 129'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

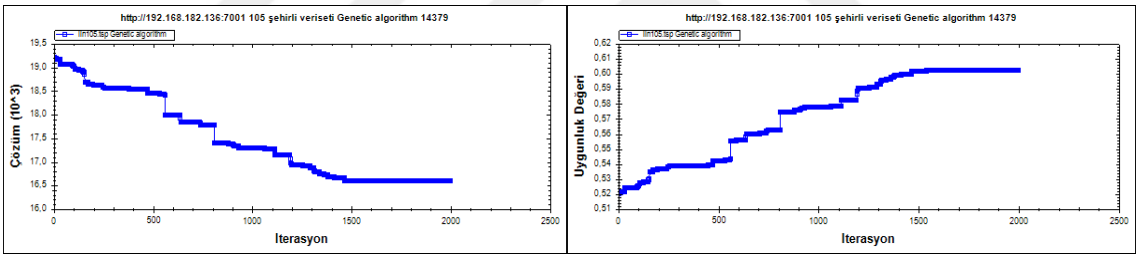


a) b)

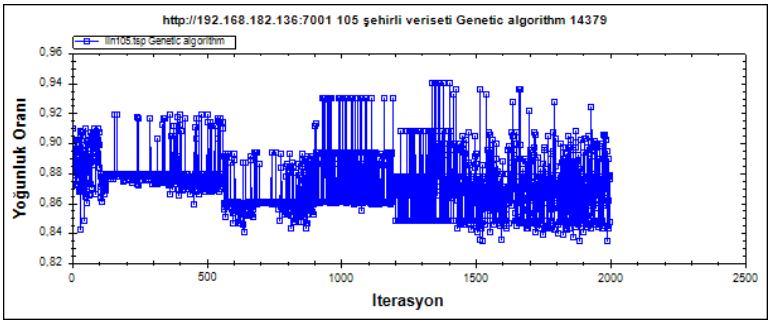


c)

Şekil 4.28. 105 şehirli problem için 133'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

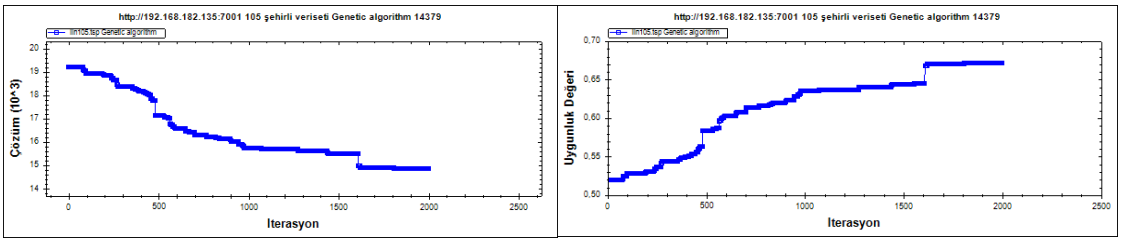


a) b)

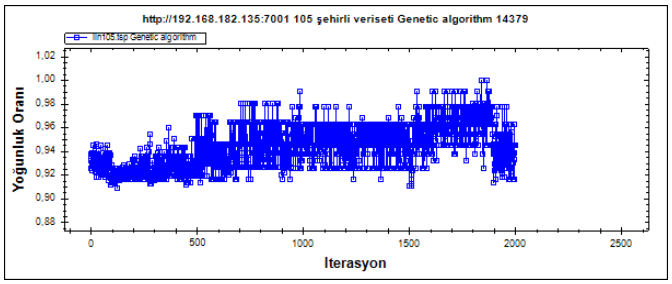


c)

Şekil 4.29. 105 şehirli problem için 136'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

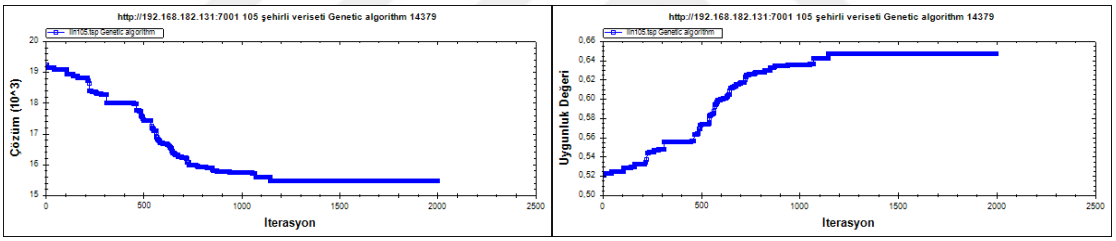


a) b)

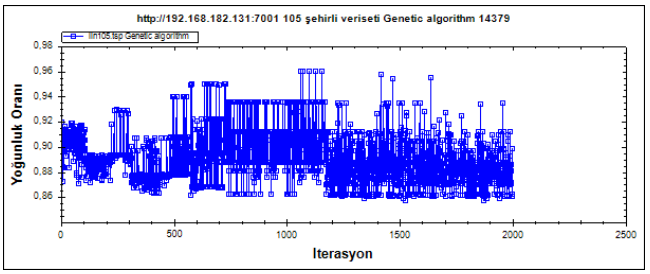


c)

Şekil 4.30. 105 şehirli problem için 135'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği



a) b)



c)

Şekil 4.31. 105 şehirli problem için 131'nolu madencinin a) çözüm grafiği b) uygunluk değeri grafiği c) yoğunluk oranı grafiği

4.3. SİMULASYON ORTAMI

Daha önceki yapılan çalışmalardan referans alınarak simulasyon ortamları incelenmiştir. Blokzincir için birçok simülasyon ortamı olmasına rağmen donanım etkinliğini daha iyi gözlenmesi için özel bir simülasyon ortamının oluşturulması zorunlu olmuştur. Algoritmanın testi belirlenen 2 adet kritere göre yapılmıştır. Bu kriterler SBİ ve Ademi-merkeziyetçilik indeksidir. Genetik algoritmaya ait parametrelerde iterasyonu 1000, 2000 ve 5000 olmak üzere ayrı parametreler, popülasyon ise 30 olarak belirlenmiştir. Çaprazlama değeri 0.65 ve mutasyon değeri ise 0.15 olarak belirtilmiştir. Tabi bu parametrelerle oynanarak çok farklı senaryolara ait çok fazla uygulama gerçekleştirilerek gözlemler kayıt altına alınabilir. Ancak buradaki varyasyon sayısının çok fazla olmasından kaynaklı olarak tez çalışmasında sınırlı değerler üzerine çalışma yapılmıştır. Belli sürede iterasyon verilmesinin sebebi ise blok oluşturulma süresinin kontrol altında tutulmak istenmesidir. PoW algoritmasının ise özet zorluk derecesi 28 bit olarak belirlenmiştir. Madenci sayısı bir takım donanım kısıtlamaları nedeniyle 1'i veri düğümü diğer 5'i consensus düğümü olmak üzere 6 adettir. Madenciler sanal bilgisayar sistemleri kullanılarak Intel(R) Core(TM) i7-9700 CPU @3.00 GHz 32 GB ram bilgisayar üzerinde VMWare yazılımı kullanılarak ile kaynakları paylaştırılarak yapılmıştır. Madenciler 2 adet farklı senaryoda çalıştırılmıştır. 1. senaryoda madencilerin donanım ayarlamaları bir adet yüksek seviye bilgisayar ve daha düşük 4 adet diğer bilgisayarda oluşmaktadır. 2. Senaryoda bilgisayarlar biraz daha yakın donanımlarda çalıştırılmıştır. 1. Senaryoda kullanılan sanal bilgisayarların donanım kaynakları aşağıda Çizelge 4.1. 'de, 2. Senaryoda sanal bilgisayarların donanım kaynakları Çizelge 4.2. 'de verilmiştir.

Çizelge 4.1. 1.senaryo düğümlerin donanım konfigürasyonu

Donanım Konfigurasyonları	
192.128.135.131 ip 'li bilgisayar	2 gb Ram 2Ghz işlemci
192.128.135.133 ip 'li bilgisayar	2 gb Ram 2Ghz işlemci
192.128.135.135 ip 'li bilgisayar	8 gb Ram 16Ghz işlemci
192.128.135.129 ip 'li bilgisayar	2 gb Ram 2Ghz işlemci
192.128.135.136 ip 'li bilgisayar	2 gb Ram 2Ghz işlemci

Çizelge 4.2. 2. senaryo düğümlerin donanım konfigürasyonu

Donanım Konfigurasyonları	
192.128.135.131 ip 'li bilgisayar	4 gb Ram 4Ghz işlemci
192.128.135.133 ip 'li bilgisayar	4 gb Ram 4Ghz işlemci

Çizelge 4.2. 2. senaryo düğümlerin donanım konfigürasyonu(devamı)

192.128.135.135 ip 'li bilgisayar	8 gb Ram 8Ghz işlemci
192.128.135.129 ip 'li bilgisayar	2 gb Ram 2Ghz işlemci
192.128.135.136 ip 'li bilgisayar	2 gb Ram 2Ghz işlemci

Problem olarak kullanılan GSP data setleri Çizelge 4.3 'te listelenmiştir.

Çizelge 4.3. GSP Problemleri

GSP Problemleri	Şehir Sayısı
22 şehirli problem	22
51 şehirli problem	51
105 şehirli problem	105
100 şehirli problem	100
225 şehirli problem	225
666 şehirli problem	666

4.3.1. Saniye başına işlem(SBİ)

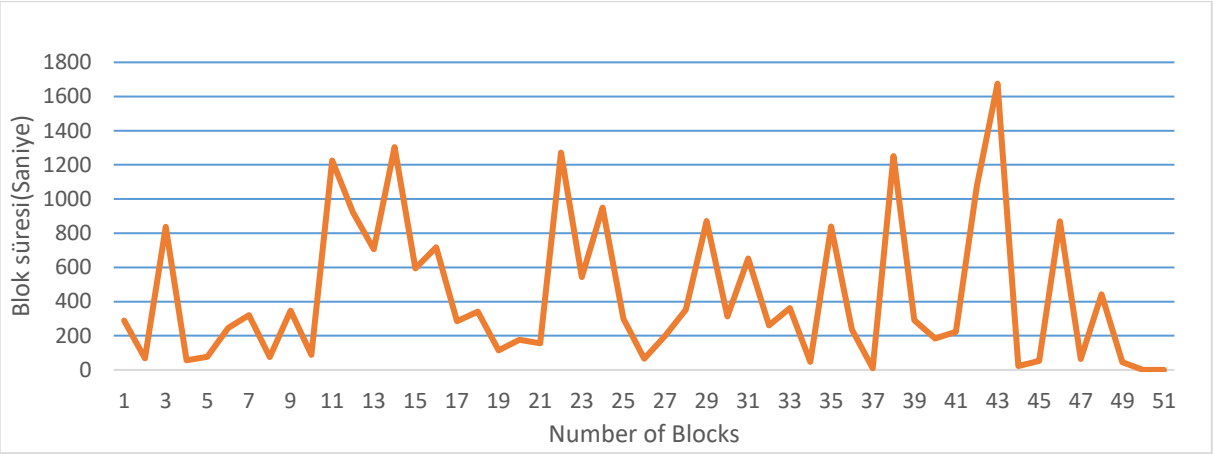
Saniye başına işlem metriği blokzinciri değerlendirmesi için yaygın bir terimdir. Başka bir deyişle SBİ, gerçekleşen işlem sayısıdır. Saniye başına işlem sayısı blokzincirin performansını hesaplamak için kullanılır. Saniyedeki işlem sayısı ne kadar yüksek olursa işlemler oldukça hızlı yürütülür. Bu metrik konsensüs algoritmasına bağlıdır. Saniye başına işlem hesaplama için denklem 4.3 kullanır.

$$SB\dot{I} = \frac{işlem\ sayısı}{blok\ oluşturma\ süresi} \tag{4.3}$$

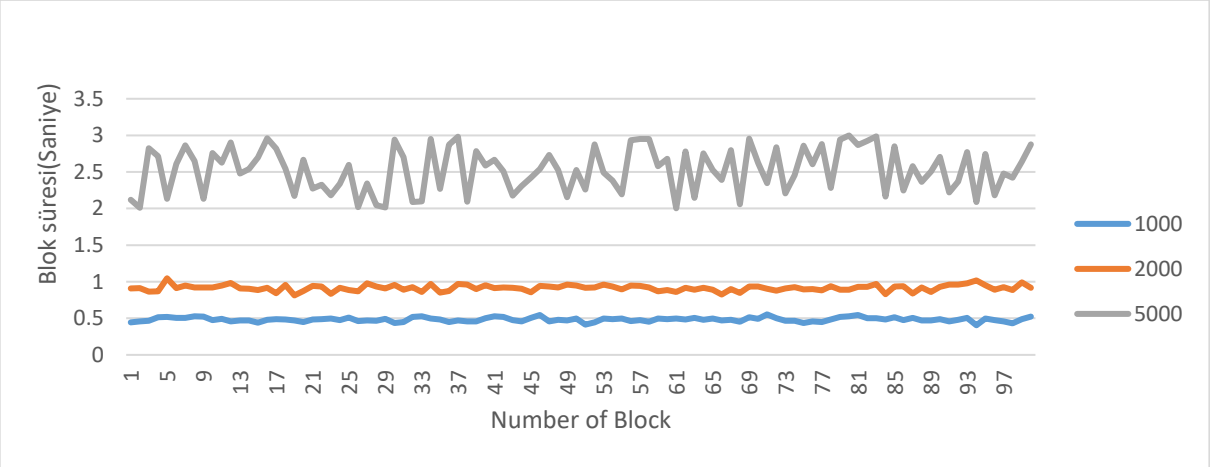
İşlem sayısı düğümlerin herhangi birinin havuzunda tuttuğu işlem sayısıdır. İşlemler çalıştırılmadan önce işlem havuzunda tutulur. Kazanan düğüm işlemleri çalıştırıp onaylar.

4.3.2. Blok oluşturma süresi

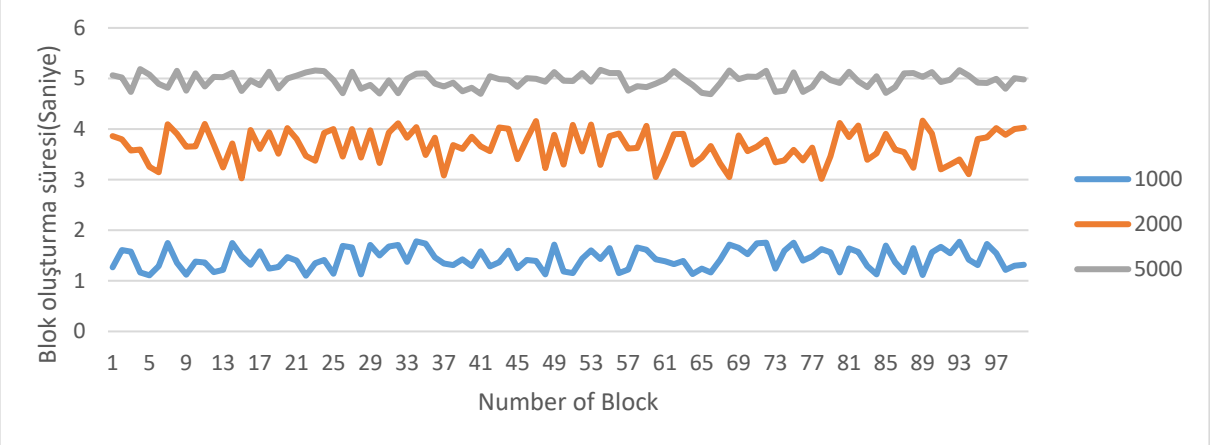
Blok oluşturma süresi blokzincirin etkinliğini dolaylı olarak test edilmesini sağlayan bir terimdir. Kısaca blok oluşturma süresidir. Blok oluşturma süresi SBİ kriterine direk etkilemektedir. O yüzden blokzincir performansı için önemli bir kriterdir. Blok oluşturma süresi ile ilgili PoW ve PoO testleri aşağıda verilmiştir. Testler sistem 100 blokluk oluşturacak şekilde yapılmıştır. Aşağıdaki 4.32 ve 4.38 'daki şekillerde PoW ve PoO verisetlerine blok oluşturma süreleri verilmiştir.



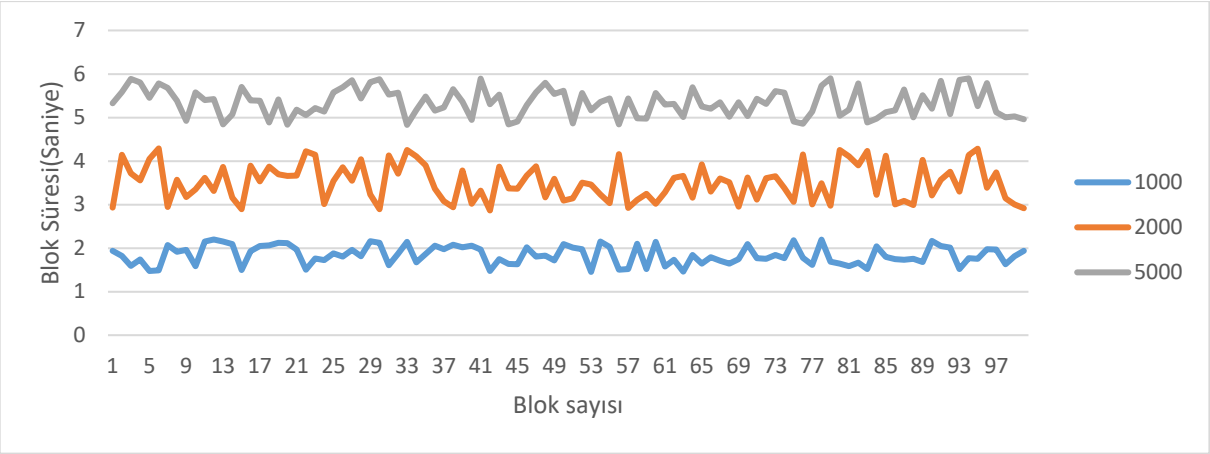
Şekil 4.32. PoW Blok oluşturma süresi grafiği



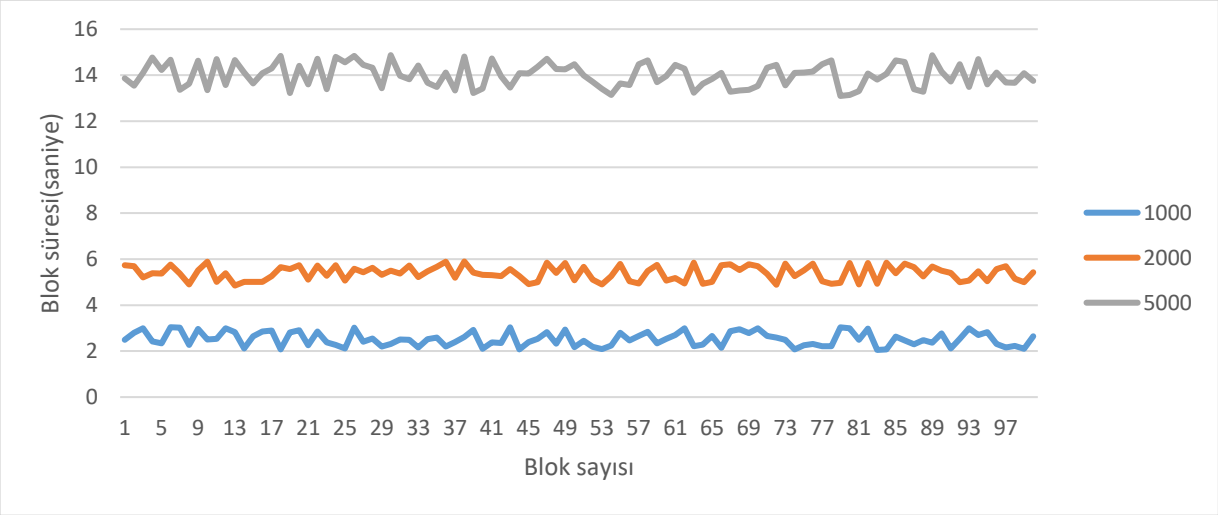
Şekil 4.33. PoO(22 şehirli) Blok oluşturma süresi grafiği



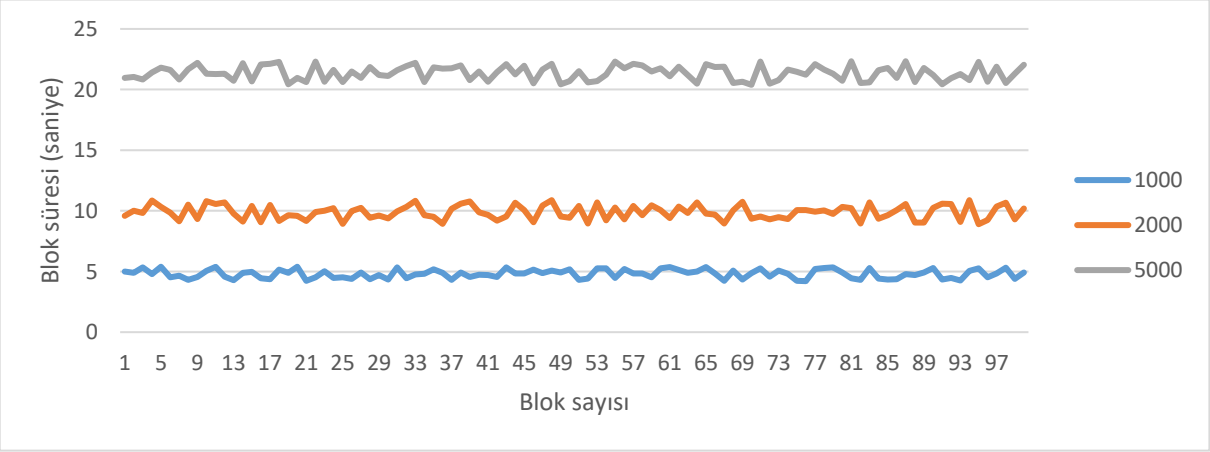
Şekil 4.34. PoO(50 şehirli) Blok oluşturma süresi grafiği



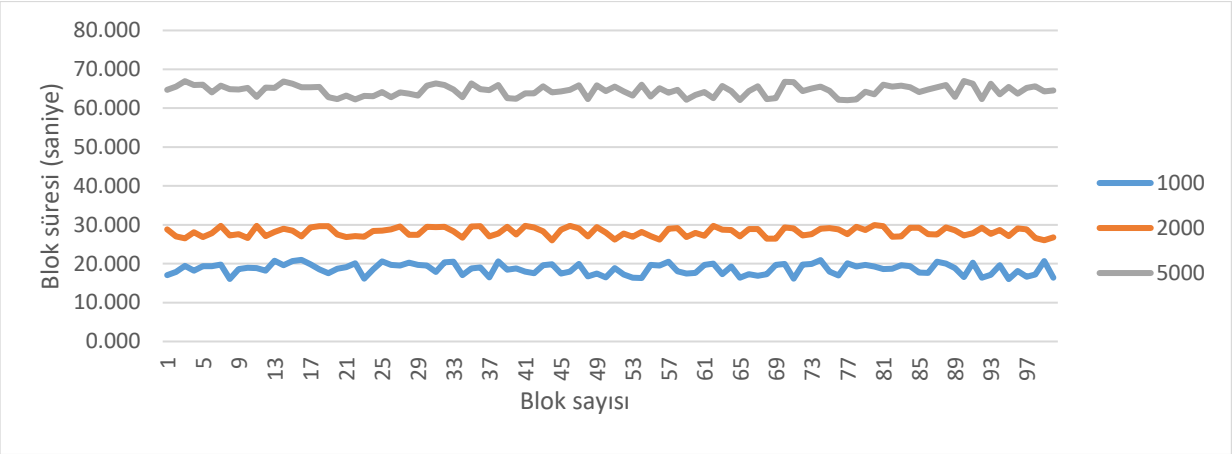
Şekil 4.35. PoO(105 şehirli) Blok oluşturma süresi grafiği



Şekil 4.36. PoO(100 şehirli) Blok oluşturma süresi grafiği



Şekil 4.37. PoO(225 şehirli) Blok oluşturma süresi grafiği



Şekil 4.38. PoO(666 şehirli) Blok oluşturma süresi grafiği

Şekil 4.32'te PoW'un blok oluşturma süresi, Şekil 4.33-4.38'de PoO blok oluşturma süresinin datasetlere göre dağılımı gösterilmektedir. PoO'nun blok oluşturma süresinde PoW'dan çok daha düşük bir dağılım sahip olduğunu gözlemlenmektedir. 1000, 2000 ve 5000 iterasyonda, 30 popülasyonda GSP datasetlerinde blok oluşturma sürelerinde farklılıklar gözlenebilmektedir.

4.3.3. Adem-i merkeziyetçilik

Blokzinciri performansını belirleyen önemli bir konuda onun adem-i merkeziyet derecesidir. Blokzincir varolan fikir birliği algoritması sanal bir demokrasi gibi çalışır. Örneğin bitcoindeki sistem bireylerin sistemdeki işlemleri doğrulaması için bazı kuralları oluşturur. Kurallar sayesinde ademi merkeziyetçi bir ortamı sağlanır. Bir blok zincirinin fikir birliği algoritması ile doğrudan ilgili olmasa da, bir kripto para biriminin ademi merkeziyetçilik seviyesini tanımlamada çok önemli bir faktördür. Örneğin, Bitcoin'deki adem-i merkeziyetçilik, bireylerin ödemeleri ve blokları doğrulamak için bu kuralları benimsemesi için blok doğrulama kurallarının kararlaştırıldığı ve belirlendiği süreç anlamına gelir. Bunun gibi, geleneksel bir bankada bu kararlar banka sahipleri tarafından verilir. Adem-i merkeziyetçiliğin kriterinin matematiksel olarak ifadesi aşağıda denklem 4.4'te verilmiştir[93].

$$F(X) = \frac{(\sum_{i=1}^{i=N} p_i)^2}{N \sum_{i=1}^{i=N} p_i^2}, \tag{4.4}$$

Burada p_i bir i düğümü tarafından çıkarılan toplam blokların kesridir. N, madencilerin

sayısıdır. Bir sistem tamamen dağıtıldığında, adalet dengesi tam sağlandığında olacak değer 1'dir. Tamamen merkezi olduğunda, adalet 1/N olacaktır[93]. PoW algoritmasını kullanan Bitcoinin adem-i merkeziyetçilik değerlendirmesi Çizelge 4.3.'de verilmiştir. Aynı madenciler ile oluşturulmuş ortamda elde edilen değerler sunulmuştur.

Çizelge 4.4. PoW Adalet indeksi ve Blok oluşturma yüzdeleri tablosu

POW	TPS	Blok oluşturma süresi (Ort)	Ademi-Merkeziyetçilik indeksi	Miner135	Miner130	miner132	Miner129	miner131
	22.6014	575.92393	0,44	65	10	8	10	7

Çizelge 4.4.'teki değerlere göre 100 blok oluşturulması istenen PoW algoritmasının sistemi daha merkeziyetçi olmasına neden olduğu görülebilir. PoW dezavantajlarından biri hash değeri yüksek donanımlı makinenin (miner135) oluşturulan blokların neredeyse yarısını oluşturmuştur. Donanım gücü yüksek olan düğüm özet fonksiyonlarını daha hızlı çözmektedir. Bu nedenle adalet indeksi düşmektedir.

4.3.4. Senaryo 1

Bu çalışmada çizelge 4.1'deki donanım ayarlamalarına çizelge 4.5 ve 4.6'daki bilgiler veriler oluşturulmuştur. Çizelgede PoO' onay mekanzimasının SBİ blok oluşturma yüzde ve blok süresi gibi veriler verilmiştir.

Çizelge 4.5. PoO Adalet indeksi ve Blok oluşturma yüzdeleri tablosu

iterasyon	GSP	SBİ	Simülasyon Süresi	Adalet İndeksi	Miner 130	Miner 135	Miner 132	Miner 129	Miner 131	Blok süresi
1000	22 şehirli	3844,79	72.5	0.9756	15	22	24	18	21	0.52
	51 şehirli	1836	112.34	0.9832	24	21	16	20	19	1.0864
	105 şehirli	1193,93	167.87	0.988	18	21	17	23	21	1.67
	100 şehirli	772,08	259.04	0.965	25	15	18	23	19	2.5
	225 şehirli	425,22	470.58	0.963	21	23	24	13	19	4.7
	666 şehirli	124,43	1628.6	0.9557	29	19	17	17	18	16.07
2000	22 şehirli	2032,18	98.54	0.9633	17	16	19	21	27	0.98
	51 şehirli	1052,47	190.33	0.971	15	22	22	17	24	1.9
	105 şehirli	502,313	398	0.927	23	12	27	23	15	3.98
	100 şehirli	362,34	532	0.97	18	25	16	18	23	5.51
	225 şehirli	202,34	989.03	0.94	24	24	22	11	19	9.9
	666 şehirli	70,173	2620.13	0.95	26	14	22	17	21	28.5

Çizelge 4.6. PoO Adalet indeksi ve Blok oluşturma yüzdeleri tablosu(devamı)

5000	22 şehirli	804,232	248.09	0.954	13	17	23	25	22	2.48
	51 şehirli	354,612	503,233	0.99	20	21	23	17	19	5.63
	105 şehirli	258,98	752,354	0.97	20	16	25	18	21	7.72
	100 şehirli	144,94	1379.3	0.98	15	17	27	19	22	13.79
	225 şehirli	93,64	2102.78	0.94	13	15	25	23	24	21.35
	666 şehirli	30,46	6124.74	0.99	23	19	21	20	17	65.65

Çizelge 4.5'te iterasyon ve GSP problemi giriş parametre olarak sisteme verilmektedir. Çıkış değerleri SBİ, adalet indeksi, madencilerin ne kadar blok oluşturduğu ve ne kadar sürede blok oluşturulduğudur. Madencilere belli ayrı ayrı iterasyonda 100'er adet blok oluşturulmaları istenmiştir. Değerlere bakılıp çizelge 4.4.'te PoW'un değerleri ile karşılaştırıldığında PoO onay mekanizmasının olay örgüsünün başarılı olduğunu söylemek mümkündür. Çizelge 4.4'te PoW SBİ değeri blok oluşturma süresinin uzunluğuna bağlı olarak düşük seyretmektedir. Bunu nedeni PoW'un temel özelliklerinden bir tanesi özet fonksiyonlarıdır. Özet fonksiyonları doğası gereği "collision-free"'dir. Yani iki girdi bilgisini aynı çıktıya eşlenememesidir. Bir diğer özelliği ise önceden tanımlanmış bir çıktıyı sağlayan bir girdiyi seçmek zordur. Bu nedenle, girdi mümkün olduğu kadar geniş bir dağılımdan seçilir. Dolayısıyla tüm bunlardan dolayı PoW'da blok oluşturma süresi uzamaktadır. PoO'da ise 1000, 2000 ve 5000 iterasyonda, 30 popülasyonda GSP datasetlerinde blok oluşturma sürelerinde farklılıklar gözlenebilmektedir. Bunun GSP probleminde iterasyon ve şehir sayısı arttıkça madencilerin çözümü bulmakta sürelerinin artış göstermesidir. Bu da blok oluşturma süresine doğrudan etki etmektedir.

Çizelge 4.7. Verilen zamana göre PoO Adalet indeksi ve Blok oluşturma yüzdeleri

Time (saniye)	tsp	TPS	toplam iterasyon	fairness	miner129	miner130	miner135	miner131	miner132	blok sayisi
500	22 şehirli	4017,13	118000	0,97	32	13	32	20	21	118
	51 şehirli	2036,17	32000	0,975	7	5	6	8	6	32
	105 şehirli	518,07	29000	0,899	5	4	4	7	9	29
	100 şehirli	776,48	22000	0,98	6	5	5	7	5	22
	225 şehirli	548,3	17000	0,849	2	2	5	6	4	17
	666 şehirli	30,51	3000	0,36	0	1	0	2	0	3
1000	22 şehirli	4266,93	158000	0,91	32	24	40	30	32	158
	51 şehirli	2395,38	161000	0,96	27	41	38	29	26	161
	105 şehirli	957,54	11000	0,83	2	2	2	4	1	11

Çizelge 4.6. Verilen zamana göre PoO Adalet indeksi ve Blok oluşturma yüzdeleri(devamı)

	100 şehirli	888,36	10000	0,71	3	0	3	1	3	10
	225 şehirli	482,98	24000	0,822	3	3	7	3	8	24
	666 şehirli	83,4	5000	0,71	1	0	1	1	2	5
1500	22 şehirli	3925,45	451000	0,90	58	63	136	106	88	451
	51 şehirli	2199,32	172000	0,97	40	25	30	40	37	172
	105 şehirli	973,44	78000	0,97	12	18	15	20	13	78
	100 şehirli	738,5	22000	0,83	5	1	7	5	4	22
	225 şehirli	539.038	38000	0,96	8	10	6	7	7	38
	666 şehirli	82.028	9000	0,77	1	1	3	1	3	9

Çizelge 4.6'de Çizelge 4.5'ten farklı olarak belli bir süre ve GSP problemi parametre verilerek 1000'er iterasyon ile ne kadar blok oluşturabildikleri gözlemlenmiştir. Sonuçlar SBİ 4017 ila 82 arasında değişmektedir. Adalet indeksi ise 0,97 ile 0,36 arasında değişmektedir. Bu gösterirki ne kadar blok oluşturulursa adalet indeksi 1 'e yakınsamaktadır. Söylemek gerekir ki adalet indeksi ve SBİ değerleri dataset değişebilmektedir ama genetik algoritmanın rastgeleselliğinin bu noktada mutlaka bir etkisi vardır.

4.3.5. Senaryo 2

Bu çalışmada çizelge 4.2'deki donanım ayarlamalarına çizelge 4.7 ve 4.8'deki bilgiler veriler oluşturulmuştur. SBİ, adalet indeksi, blok oluşturma yüzdeleri ve blok süresi gibi veriler çizelgede verilmiştir.

Çizelge 4.8. Senaryo 2 PoO Adalet indeksi ve Blok oluşturma yüzdeleri tablosu

iterasyon	GSP	SBİ	Adalet indeksi	miner130	miner135	miner132	miner129	miner131	Blok süresi
1000	22 şehirli	4304.17	0,98	22	22	20	21	15	0.46
	51 şehirli	2522.2	0,89	19	18	11	32	20	0.79
	105 şehirli	1035.45	0,98	25	19	20	18	18	1.93
	100 şehirli	836.99	0,98	22	18	19	18	23	2.38
	225 şehirli	560.7	0,94	24	16	13	23	24	3.56
	666 şehirli	133.72	0,97	25	16	22	17	20	14.9
2000	22 şehirli	2098.45	0,84	11	15	36	18	20	0.95
	51 şehirli	1263.8	0,95	17	17	27	16	23	1.58

Çizelge 4.7. Senaryo 2 PoO Adalet indeksi ve Blok oluşturma yüzdeleri tablosu(devam)

	105 şehirli	512	0,99	17	19	22	21	21	3.90
	100 şehirli	408.45	0,97	15	19	24	23	19	4.89
	225 şehirli	224.4	0,95	19	20	26	22	13	8.91
	666 şehirli	66.25	0,98	17	23	20	22	18	30.18
5000	22 şehirli	926.54	0,70	10	1	37	30	22	2.1
	51 şehirli	1442.25	0,96	16	17	20	20	27	1.38
	105 şehirli	243.4	0,97	19	20	18	17	26	8.216
	100 şehirli	200.18	0,94	22	25	17	12	24	9.991
	225 şehirli	70.2	0,95	16	15	26	19	24	28.490
	666 şehirli	26.74	0,95	24	16	14	23	23	74.794

Çizelge 4.7'deki parametre olarak iterasyon ve GSP parametreleri verilmiştir. Çizelge 4.2'de verilen donanım ayarlamalarına göre test sonuçlarında genel bir değişiklik olmadığını saptayabiliriz. Burada anlaşılabilecektir ki donanım gücüne bağlı olarak blok oluşturma işlemleri yerini akıllı yöntemlerle blok oluşturma işlemlerine bırakabilecektir. Bu sayede enerji ve kaynak israfının önüne geçilebilir.

Çizelge 4.9. Senaryo 2 Verilen zamana göre PoO Adalet indeksi ve Blok oluşturma yüzdeleri

Zaman (saniye)	GSP	SBİ	toplam iterasyon	Adalet indeksi	miner129	miner130	miner135	miner131	miner132	blok sayisi
500	22 şehirli	3785,54	147000	0,89	47	23	22	35	20	147
	51 şehirli	2218,35	50000	0,86	15	4	7	11	13	50
	105 şehirli	1031,51	27000	0,78	3	2	6	10	6	27
	100 şehirli	886,5	11000	0,83	3	1	3	1	3	11
	225 şehirli	282,130	16000	0,88	3	2	4	2	5	16
	666 şehirli	70,24	3000	0,36	0	1	0	2	0	3
1000	22 şehirli	3926,35	298000	0,89	32	90	74	49	53	298
	51 şehirli	2209,48	115000	0,97	21	23	20	30	21	115
	105 şehirli	954,6	52000	0,97	13	11	8	10	10	52
	100 şehirli	922,88	30000	0,94	5	6	4	7	8	30
	225 şehirli	464,26	33000	0,77	8	1	12	6	6	33
	666 şehirli	85,3	6000	0,51	0	0	2	1	3	6
1500	22 şehirli	3852,1	441000	0,92	101	129	60	86	65	441
	51 şehirli	2138,63	171000	0,95	41	38	21	41	30	171
	105 şehirli	1014,98	53000	0,99	9	10	11	12	11	53

Çizelge 4.8. Senaryo 2 Verilen zamana göre PoO Adalet indeksi ve Blok oluşturma yüzdeleri (devam)

	100 şehirli	630,45	45000	0,83	10	12	6	3	14	45
	225 şehirli	495,47	49000	0,92	9	15	7	9	9	49
	666 şehirli	80,075	8000	0,58	0	0	2	3	3	8

Çizelge 4.8'de ise GSP ve zaman parametre olarak verilmiştir. Çıktı olarak ise GSP problemlerine göre madencilerin blok dağılımları, SBİ, adalet indeksi ve blok sayısı listelenmiştir. Datasetlerin zorluğuna göre blok sayıları azalma gösterdiği gözlenmektedir. Bu da düşük adalet indeksine neden olmaktadır. Oluşturulan blok sayıları ise dataset zorluğuna göre değişmiştir. Bunun nedeni ise datasetlerin zorluğu ve kullanılan algoritmanın rastgeleselliğinin neden olduğu düşünülebilir.

4.3.6. Simulasyon Detayları

Simulasyon ortamı c# ve pythonla yazılmış olup c# ile yazılan arayüzden python blokzincir arayüzü yazılmış olan düğümlere bağlanımaktadır. Toplamda VMWare sistemine 5 adet, 1 adet çok yüksek ve 4 adet çok düşük donanımlı sanal bilgisayar oluşturulmuştur. Pythonla yazılan düğümlerin hangi metodlara sahip oluğu aşağıda verilmiştir. Arayüz ise c# Windows form üzerinde bu arayüzlere bağlanan bir platformda yazılmıştır.

Create_transaction: Bu metod işlem oluşturmak için gereklidir. Bir transaction işelm özeti , Size, işlem girişi, İşlem çıkışı ve İşlem ücreti değerlerini içerir.

Broadcast_transaction: Oluşturulan işlmlerin diğer kayıtlı düğümlere yayınlanmasını sağlar.

Register_node: Düğümlerin birbirlerine kayıt olup haberleşmesini sağlar.

Create_solution: Düğümlerde çözüm oluşturulmasını sağlamaktadır

Create_block: Düğümlerinin çözümden sonra düğüm oluşturmasını sağlar.

broadcast_block: Düğüm blok oluşturduktan sonra bloğu diğer düğümlere yaymasını sağlar .

Bu simulasyon gerekli blokzincir ölçümleri uygulanmıştır. Çalışma sanal makinalar çalıştırılmakta ve içindeki uygulama hesapları aktif hale getirilmektedir. Yazılan blocksim isminde win form uygulaması bu sanal makinelere bağlanır. Onları blokzincir çalışma ortamına entegre eder. Çalışma ortamına ait resimler ve algoritma aşağıda verilmiştir.

Algoritma 2. Blocksim algoriması

Başla

Blocksim uygulaması çalıştırılır.

Blocksime sanal makineleri birbirine kayıt eder blokzincir ortamı aktif edilir.

Blocksim sanal makinelere GSP görevlerini gönderir. Sanal makineler GSP problemlerini çözerler.

Blocksim değerleri alır kazananı belirler

Kazanan sanal makine yani düğüm ödülü blok oluşturmaya hak kazanır.

Kazanan düğüm blokları yayınlar.

Bitir.

5. SONUÇ

Şekil 4.32 PoW'un blok oluşturma süresi Şekil 4.33-4.38'de PoO ve PoW'un blok oluşturma süresinin datasetlere göre dağılımı gösterilmektedir. 1000, 2000 ve 5000 iterasyonda, 30 popülasyonda GSP datasetlerinde blok oluşturma sürelerinde farklılıklar gözlenebilmektedir. Bunun GSP probleminde iterasyon ve şehir sayısı arttıkça madencilerin çözümü bulmakta sürelerinin artış göstermesidir. Bu da blok oluşturma süresine doğrudan etki etmektedir. PoW daki yüksek dağılımın sebebi ise özet fonksiyonlarının "collision-free" özelliğinden kaynaklanmaktadır. Yani iki girdi bilgisini aynı çıktıya eşlenememesidir. Bir diğer özelliği ise önceden tanımlanmış bir çıktıyı sağlayan bir girdiyi seçmek zordur. Bu nedenle, girdi mümkün olduğu kadar geniş bir dağılımdan seçilir. Dolayısıyla tüm bunlardan dolayı PoW'da blok oluşturma süresi uzamaktadır.

Deneylerde iterasyon ve popülasyon değerleri sabit bir değer olarak belirlenmiştir. Bunun amacı madencilerin blokzincir değerlerini iyileştirmesini sağlamaktır. GSP için daha optimum çözümler elbette bulunabilir. Çalışmanın amacı GSP çözmek değil GSP çözümlerinin blokzincir sorunlarına çözüm bulmasını sağlamaktır.

Çizelge 4.4'te adalet indeksinin blok sayısına göre değeri gösterilmiştir. PoW algoritmasını kullanan bilgisayarlar donanımları ölçüsünde blokzincir sistemini daha merkeziyetçi davranmaya itebilmektedir. PoO algoritması matematiksel kazanması katsayısı ile belirlendiğinden PoW gibi sadece donanıma dayalı olabilecek değerlendirmelerden kısmen kaçınmaktadır. Optimizasyon algoritmaları çözümü metodu nedeniyle bu çalışmada ortaya koyulan onay mekanizması modelinde blokzincir çözümlerine yardımcı olabilmektedir. Bu PoO modeline göre değerlendirme kriteri sadece donanıma dayalı olabilecek blok oluşturma yerine akıllı yöntemlerle blok oluşturma işlemini gerçekleştirmiş olduğu ortaya çıkmaktadır. Bu yöntemi destekleyici çalışmalar literatürde mevcuttur [24,94,95,96,97].

Çizelge 4.5'te girdi olarak iterasyon ve GSP belirlenmiştir. Girdilerin bu şekilde belirlenmesinin nedeni iterasyon ve GSP datasetinin çıktılar üzerinde ne gibi bir etkiye sahibi olabileceğini gözlemlemektir. 1000 iterasyonda daha iterasyonlara nispeten daha düşük blok adalet indeksine rastlanmaktadır. Çünkü iterasyondaki artış nispeten daha iyi çözümlerin bulunmasına olanak sağlamaktadır. Bu daha iyi çözümlerin, daha iyi yoğunluk oranların bulunabilmesine neden olmuştur. Yinede söylemek gerekir ki blok

oluşturma yüzdeleri bu şekilde olmasında genetik algoritmanın rastgeleselliğin rolü mutlaka vardır. Saniye başına işlemdeki artışın yüksek iterasyonda fazla olmasının sebebi iterasyonla bağlantılıdır. Daha az iterasyon blok süresini düşürmekte ve yüksek saniye başına işlem elde edilmesine olanak sağlamaktadır. Tüm yapılan işlemleri genel olarak açıklarsak blokzincirde blok oluşturmada akıllı yöntemlerin geliştirilmesi şeklinde değerlendirmek gerekir. Yani madenci cihazlarının gücüne ihtiyaç duyulmayan düşük seviye donanımla yapılabilecek bir algoritma düzeneği olarak düşünülebilir. Çizelge 4.7'de ise aynı şekilde deneyler yapılmıştır. Çizelge 4.5'ten farklı olarak sadece 22 şehirli probleminde adalet indeksi 70%'e kadar düşmüştür.

Çizelge 4.6'de ise deney bir zaman dilimi verilerek gerçekleştirilmiştir. 1000 iterasyon üzerinden her satırda 100'er blok oluşturulması istenmiştir. Oluşturulan blok sayılarının adalet indeksi bağlantısı incelenmiştir. Oldukça tahmin edilebilir bir biçimde blok sayısı ile adalet indeksinin doğru orantılı olduğu gözlemlenebilir. Dahası adalet indeksinin yani blok dağılım yüzdelerinin yer yer %36'ya kadar düştüğü görülür. Çizelge 4.8'de ise datasetlerin zorluğuna göre blok sayıları azalma göstermiştir. Bu da düşük adalet indeksine neden olmaktadır. Oluşturulan blok sayıları ise dataset zorluğuna değişmiştir. Bunun nedeni ise datasetlerin zorluğu ve kullanılan algoritmanın rastgeleselliğinin neden olduğu düşünülebilir.

Optimizasyon algoritmalarının kullanılması hesaplama gücü açısında yeni bir araştırma alanı ortaya çıkarmıştır. PoW algoritması enerji kaynaklarının boşa harcama pahasına da olsa blok oluşturma işlevini yerine getirebilir ama bir optimizasyon algoritmasının çözümü bulmasından faydalanılarak blokzincirde boşa harcanan kaynakları genetik algoritmanın pratik faydalarına yönlendiren yeni bir fikir birliği algoritması önerilmektedir. Ayrıca önerilen sistemin bir âdem-i merkeziyetçilik seviyesini daha adaletli bir şekilde tasarlanmıştır. Deneysel sonuçlar PoO algoritmasının merkezi olmayan bir şekilde blok üretme yeteneğine sahip olduğunu doğrulamaktadır.

PoO algoritması boşa harcanan enerjinin herhangi bir kullanıcı tarafından sunulan optimizasyon problemlerini çözmek için kullanılabileceği blokzinciri için bir fikir birliği protokolü olarak önerilmiştir. PoW dayalı blokzincirlerin yerine alternatif olabilecek bir algoritma olarak tasarlanmıştır. Araştırmanın amacı, büyük miktarda bilgi işlem gücünü optimizasyon algoritmalarının pratik faydalarına yönlendirerek blok oluşturma işlemlerini yüksek donanımdan bağımsız daha hale getirmektir. Ayrıca protokoldeki hesaplama tarzı sıradan CPU'larla madenciliği daha karlı hale getirir. Hesaplama kaynakları artık boşa gitmediğinden, kamu kuruluşlarının ve daha genel kullanıcıların

katılması beklenebilir. Bu, protokolümüzle bir blok zincirini mevcut olanlardan daha merkezi olmayan hale getirecektir.

Ortaya konulan süreç optimizasyon algoritmalarıyla veya YZ ile akıllı yöntemlerle blok oluşturabilmenin önünü açmaktadır. Dahası katılımcılar sistemde problemlere çözüm ararken bulabildikleri en iyi çözümü şeffaf bir şekilde ilan ederler. Böylece çözümlerin paylaşıldığı dağıtık bir optimizasyon sisteminde söz edilebilir. Ancak çalışmanın amacı GSP'ye çözüm bulmak değil GSP optimizasyonun blokzincir sistemlerine yardımcı olmasını sağlamaktır.

5.1. GELECEKTEKİ ÇALIŞMALAR

Gelecek çalışmalarda diğer optimizasyon algoritmaları ve optimizasyon problemleri kullanılarak bunların fikir birliği modelleri üzerindeki etkilerinin deneysel olarak incelenmesi için çeşitli değerlendirmeler yapılabilir. Çalışma bir dağıtılmış optimizasyon yöntemi olarak düşünülebilir. Optimizasyon problemi veya algoritması değiştirilerek çözümler çeşitlendirilebilir.

blokzincir teknolojisinin ve konsensüs algoritmalarının gelişmesiyle birlikte pek çok farklı algoritmanın bir araya getirilmesiyle oluşan pek çok farklı varyasyonu görmek mümkündür. Örneğin PoO'nun genetik algoritmaları kullanması, blokzincir fikir birliğine yeni bir yaklaşım sunmaktadır ~~sunuyor~~. Bu yaklaşım farklı optimizasyon problemlerini çözmek için kullanılabilir. Ayrıca blokzincir konsensüsü için başka algoritmalar da kullanılabilir. Bu algoritmalar GSP analizi yerine farklı problemleri çözmek için kullanılabilir. Dolayısıyla PoO yapısı blokzincir teknolojisine yeni ve farklı bir yaklaşım sunuyor ve diğer algoritmalarla birleştirilerek daha da geliştirilebilir

Ayrıca gelecekte PoS ve PoO arasındaki karşılaştırmalar da incelenebilir. Madencilik için belirli bir miktarda kripto para sahipliği gerektiren PoS'tan farklı olarak PoO, madencinin optimizasyon problemini çözme becerisine odaklanır. Bu, bir madenci ne kadar yetenekliyse, bir sonraki blokta madencilik yapmak için seçilme olasılıklarının da o kadar yüksek olduğu anlamına gelir. Gelecek çalışmalarda PoS ve PoO karşılaştırması detaylı olarak yapılabilir.

PoO, geleneksel kripto para birimi madenciliği konsensüs algoritmalarından farklı bir yaklaşım sunan yenilikçi bir blok zinciri konsensüs modelidir. PoO'nun temel amacı, matematiksel optimizasyon problemlerini çözerken oluşturulan işlem yoğunluğunu

kullanarak kripto para birimi ödüllerini dağıtmaktır. Bu, PoO'nun enerji verimliliği ve çevresel sürdürülebilirlik açısından önemli bir avantaj sunmasını sağlar.

PoO, genetik algoritmalar gibi optimizasyon tekniklerini kullanarak karmaşık matematiksel problemleri çözmek için katılımcıların bilgisayar kaynaklarını kullanır. Bu yaklaşım, PoO'nun ölçeklenebilirliği ve hızlı işlem yetenekleri sunmasını sağlar. Ayrıca, PoO, katılımcıları kendi katkılarına dayalı olarak ödüllendiren ve enerji tüketimini önemli ölçüde azaltan bir model sunar.

PoO'nun gelecekteki potansiyeli oldukça büyüktür. Daha karmaşık optimizasyon problemlerini çözmek için farklı optimizasyon algoritmalarıyla entegre edilebilir. Ayrıca, PoO'nun farklı endüstrilerdeki uygulamaları araştırılabilir, bu da kripto para birimi teknolojisinin ötesinde kullanım alanlarına işaret edebilir.

Sonuç olarak, PoO, hızlı işlem yetenekleri ve adem-i merkeziyetçilik açısından önemli avantajlar sunan yenilikçi bir konsensüs modelidir. Gelecekteki çalışmalar, PoO'nun potansiyelini daha da keşfetmek ve bu modeli daha geniş bir uygulama yelpazesi için optimize etmek için odaklanabilir.

6. KAYNAKLAR

[1] S, Nakamoto. "Bitcoin: A peer-to-peer electronic cash system Bitcoin: A Peer-to-Peer Electronic Cash System." Bitcoin. org. Disponible en https://bitcoin. org/en/bitcoin-paper 2009.

[2] H. Atabaş, "Technology of Blockchain", Ceres Publications. 2. baskı.c. 4 Ss 53-62

[3] V. Güven and E. Şahinöz Blockchain cryptocurrency bitcoin, Kronik Publications 2018 , c. 14, sayı 2, ss. 267–277, 2006.

[4] F. Tschorsch and B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084-2123, thirdquarter 2016,

[5] L. Cai, Q. Li, and X. Liang, "Introduction to Blockchain Basics," *Advanced Blockchain Technology*. Springer Nature Singapore, ss. 3–43, 2022.

[6] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services,* c. 14, sayı 4, p. 352, 2018

[7] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur and H. -N. Lee, "Ethereum Smart Contract Analysis Tools: A Systematic Review," in *IEEE Access*, c. 10, ss. 57037-57062, 2022

[8] T. Cutts, "Smart Contracts and Consumers," *SSRN Electronic Journal. Elsevier BV*, 2019

[9] F. Hofmann, S. Wurster, E. Ron and M. Böhmecke-Schwafert, "The immutability concept of blockchains and benefits of early standardization," 2017 *ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K),* Nanjing, China, 2017, ss. 1-8

[10]Mougayar, W. (2016). The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology.

[11] E. Fathalla, C. Wang, X. Li, R. Gazda, and H. Wu, "Redactable Distributed Ledgers: A Survey," *Distributed Ledger Technologies: Research and Practice*, c. 2, sayı. 3. Association for Computing Machinery (ACM), ss. 1–26, Sep. 18, 2023.

[12] Bamakan, S. M. H., Motavali, A., & Babaei Bondarti, A. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *In Expert Systems with Applications* (c. 154, 113385). Elsevier BV.

[13] L. M. Bach, B. Mihaljevic and M. Zagar, "Comparative analysis of blockchain consensus algorithms," 2018 41st *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija,

Croatia, 2018, ss. 1545-1550.

[14] Haojun Huang, Jialin Tian, Geyong Min, and Wang Miao, “Introduction to blockchains,” *Blockchains for Network Security: Principles, technologies and applications. Institution of Engineering and Technology,* ss. 1–21, Nov. 10, 2020. doi: 10.1049/pbpc029e_ch1.

[15] X. Fu, H. Wang, and P. Shi, “A survey of Blockchain consensus algorithms: mechanism, design and applications,” S*cience China Information Sciences*, c. 64, sayı. 2. Springer Science and Business Media LLC, Nov. 18, 2020

[16] N. Chaudhry and M. M. Yousaf, "Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities," 2018 *12th International Conference on Open Source Systems and Technologies (ICOSST)*, Lahore, Pakistan, 2018, ss. 54-63

[17] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, “On the Security and Performance of Proof of Work Blockchains,” *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Oct. 24, 2016

[18] F. Saleh, “Blockchain without Waste: Proof-of-Stake,” *The Review of Financial Studies*, c. 34, sayı. 3. *Oxford University Press (OUP),* pp. 1156–1190, Jul. 07, 2020. doi: 10.1093/rfs/hhaa075.

[19] G. Shapiro, C. Natoli and V. Gramoli, "The Performance of Byzantine Fault Tolerant Blockchains," *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, USA, 2020, ss. 1-8, doi: 10.1109/NCA51143.2020.9306742.

[20] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, “Proof of Activity,” ACM SIGMETRICS Performance Evaluation Review, c. 42, sayı. 3. *Association for Computing Machiner*y (ACM), ss. 34–37,

[21] K. Karantias, A. Kiayias, and D. Zindros, “Proof-of-Burn,” *Financial Cryptography and Data Security. Springer International Publishing*, pp. 523–540, 2020.

[22] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, “On Security Analysis of Proof-of-Elapsed-Time (PoET),” *Lecture Notes in Computer Science. Springer International Publishing*, pp. 282–297, 2017

[23] C. Zhang, C. Wu, and X. Wang, “Overview of Blockchain Consensus Mechanism,” *Proceedings of the 2020 2nd International Conference on Big Data Engineering.* 2020.

[24] D. Tanana, "Avalanche blockchain protocol for distributed computing security," 2019 *IEEE International Black Sea Conference on Communications and Networking* (BlackSeaCom), Sochi, Russia, 2019, ss. 1-3

[24]K. Salah, M. H. U. Rehman, N. Nizamuddin and A. Al-Fuqaha, "Blockchain for AI:

Review and Open Research Challenges," in *IEEE Access*, c. 7, ss. 10127-10149, 2019

[25]K. Wang, J. Dong, Y. Wang and H. Yin, "Securing Data With Blockchain and AI," in *IEEE Access*, c. 7, s. 77981-77989, 2019

[26]E. C. Ferrer, "The blockchain: a new framework for robotic swarm systems," arXiv, 2016.

[27] G. A. Montes and B. Goertzel, "Distributed, decentralized, and democratized artificial intelligence," *Technological Forecasting and Social Change*, c. 141.ss. 354–358,

[28]K. R. Özyilmaz, M. Doğan and A. Yurdakul, "IDMoB: IoT Data Marketplace on Blockchain," 2018 *Crypto Valley Conference on Blockchain Technology (CVCBT)*, Zug, Switzerland, 2018, ss. 11-19,

[29] J. D. Harris and B. Waggoner, "Decentralized and Collaborative AI on Blockchain," *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, 2019, ss. 368-375

[30]Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He and Y. Zhang, "Blockchain and Deep Reinforcement Learning Empowered Intelligent 5G Beyond," in *IEEE Network*, c. 33, sayı. 3, ss. 10-17, 2019

[31] S. Teerapittayanon and H. T. Kung, "DaiMoN: A Decentralized Artificial Intelligence Model Network," 2019 *IEEE International Conference on Blockchain (Blockchain)*, 2019, ss. 132-139

[32] S. K. Singh, S. Rathore, and J. H. Park, "BlockIoTIntelligence: A Blockchain-enabled Intelligent IoT Architecture with Artificial Intelligence," *Future Generation Computer Systems*, c. 110. Elsevier BV, ss. 721–743, 2020

[33] J. Chen, K. Duan, R. Zhang, L. Zeng, and W. Wang, "An AI Based Super Nodes Selection Algorithm in BlockChain Networks." arXiv, 2018.

[34] M. Qiu, X. Liu, Y. Qi, H. Zhao, and M. Liu, "AI Enhanced Blockchain (I)," *2020 3rd International Conference on Smart BlockChain (SmartBlock)*. *IEEE*, 2020.

[35]A. Singh, A. Saxena, S. Ramani, and M. Karuppiah, "Blockchain in Artificial Intelligence," *Data Analytics, Computational Statistics, and Operations Research for Engineers*. CRC Press, ss. 119–145,2022.

[36] A. Garg, "Blockchain for Artificial Intelligence," International Journal for Research in Applied Science and Engineering Technology, *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, ss. 2414–2417, Nov. 23, 2017.

[37] S. Khan and O. Mangde, “Application of Blockchain in Artificial Intelligence,” *International Journal for Research in Applied Science and Engineering Technology*, c. 10, sayı. 6. 2022.

[38] S. Muthukrishnan and B. Duraisamy, “Blockchain Technologies and Artificial Intelligence,” *Studies in Big Data. Springer Singapore*, ss. 243–268, 2019.

[39] A. Imteaj, M. Hadi Amini, and P. M. Pardalos, “Leveraging Blockchain Technology for Artificial Intelligence,” *Foundations of Blockchain. Springer International Publishing*, ss. 51–58, 2021,

[40] A. S. Masurkar, X. Sun, and J. Dai, “Using Blockchain for Decentralized Artificial Intelligence with Data Privacy,” *2023 International Conference on Computing, Networking and Communications (ICNC). IEEE*, 2023.

[41] R. Wang, M. Luo, Y. Wen, L. Wang, K.-K. Raymond Choo, and D. He, “The Applications of Blockchain in Artificial Intelligence,” *Security and Communication Networks*, 2021. Hindawi Limited, ss. 1–16,2021.

[42]B. Xing and T. Marwala, “The Synergy of Blockchain and Artificial Intelligence,” *SSRN Electronic Journal. Elsevier BV*, 2018.

[42] Mehak, Rahul Kumar, and Dr. Ashima Mehta, “Artificial Intelligence,” *International Journal of Advanced Research in Science, Communication and Technology*. Naksh Solutions, ss. 20–30, 2023.

[43] M. Salimitari, M. Joneidi and M. Chatterjee, "AI-Enabled Blockchain: An Outlier-Aware Consensus Protocol for Blockchain-Based IoT Networks," *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, ss. 1-6

[44]Z. Zheng, H.-N. Dai, and J. Wu, “Overview of Blockchain Intelligence,” Blockchain Intelligence. Springer Singapore, ss. 1–14, 2021.

[45]Abdulrahman Yarali, "Artificial Intelligence and Blockchain," in *Intelligent Connectivity: AI, IoT, and 5G, IEEE, 2022*, ss.165-190,

[46]S. S. Panda and D. Jena, “Decentralizing AI Using Blockchain Technology for Secure Decision Making,” *Algorithms for Intelligent Systems. Springer Singapore*, ss. 687–694, 2020.

[47]R. Wang, M. Luo, Y. Wen, L. Wang, K.-K. Raymond Choo, and D. He, “The Applications of Blockchain in Artificial Intelligence,” *Security and Communication Network*s, ss. 1–16, 2021.

[48]P. Tagde et al, “Blockchain and artificial intelligence technology in e-Health,” Environmental Science and Pollution Research, c. 28, sayı. 38. *Springer Science and Business Media LLC*, ss. 52810–52831, 2021.

[49]J. Gupta, I. Singh, and K. P. Arjun, “Artificial Intelligence for Blockchain I,” *Blockchain, Internet of Things, and Artificial Intelligence. Chapman and Hall/CRC*,

ss. 109–140, 2021.

[50]Ch. V. N. U. B. Murthy and M. L. Shri, "Artificial Intelligence for Blockchain II," *Blockchain, Internet of Things, and Artificial Intelligence. Chapman and Hall/CRC,* ss. 141–154, 2021.

[51]A. A. Hussain and F. Al-Turjman, "Artificial intelligence and blockchain: A review," *Transactions on Emerging Telecommunications Technologies,* c. 32, sayı. 9. Wiley, 2021.

[52]F. Muheidat and L. Tawalbeh, "Artificial Intelligence and Blockchain for Cybersecurity Applications," *Studies in Big Data. Springer International Publishing,* ss. 3–29, 2021

[52] P. Kora and P. Yadlapalli, "Crossover operators in genetic algorithms: A review," International Journal of Computer Applications, c. 162, no. 10, 2017.

[53]P. Saigal, "Merger of Artificial Intelligence and Blockchain," *Blockchain Technology and Applications. Auerbach Publications,* ss. 139–158, 2020.

[54]J. D. Harris, "Analysis of Models for Decentralized and Collaborative AI on Blockchain," *Blockchain – ICBC 2020. Springer International Publishing,* ss. 142–153, 2020.

[55]P. Gulati, A. Sharma, K. Bhasin, and C. Azad, "Approaches of Blockchain with AI: Challenges ; Future Direction," S*SRN Electronic Journal. Elsevier BV,* 2020.

[56]N. Kshetri, "Complementary and Synergistic Properties of Blockchain and Artificial Intelligence," IT Professional, c. 21, sayı. 6. *Institute of Electrical and Electronics Engineers (IEEE),* pp. 60–65,2019

[57]D. Senthilkumar, "Cross-Industry Use of Blockchain Technology and Opportunities for the Future," *Advances in Data Mining and Database Management. IGI Global,* ss. 64–79, 2020.

[58]K. Sgantzos and I. Grigg, "Artificial Intelligence Implementations on the Blockchain. Use Cases and Future Applications," *Future Internet,* c. 11, sayı. 8. 2019.

[59]B. Parker and C. Bach, "The Synthesis of Blockchain, Artificial Intelligence and Internet of Things," *European Journal of Engineering Research and Science, c. 5, sayı. 5. European Open Access Publishing (Europa Publishing),* ss. 588–593, 2020

[60]D. N. Dillenberger et al., "Blockchain analytics and artificial intelligence," *IBM Journal of Research and Development,* c. 63, sayı. 2/3. 2019

[61]B. Xing and T. Marwala, "The Synergy of Blockchain and Artificial Intelligence," *SSRN Electronic Journal. Elsevier BV,* 2018.

[62] C. Rajesh Babu and B. Amutha, "Blockchain and extreme learning machine based

spectrum management in cognitive radio networks," *Transactions on Emerging Telecommunications Technologies*, cil. 33, no. 10. 2020.

[63]K. Saritha, M. Kurni, K. Madhavi, and D. Nagadevi, "Integration of Artificial Intelligence and the Internet of Things with Blockchain Technology," *Lecture Notes in Networks and Systems. Springer Singapore*, ss. 449–457, 2021

[64]M. Swan, "Blockchain for Business: Next-Generation Enterprise Artificial Intelligence Systems," *Advances in Computers. Elsevier*, ss. 121–162, 2018

[65]A. Goel, B. Bhushan, B. Tyagi, H. Garg, and S. Gautam, "Blockchain and Machine Learning: Background, Integration Challenges and Application Areas," *Emerging Technologies in Data Mining and Information Security*. Springer Singapore, ss. 295–304, 2021.

[66]Z. Zheng, H.-N. Dai, and J. Wu, "Overview of Blockchain Intelligence," *Blockchain Intelligence*. Springer Singapore, ss. 1–14, 2021.

[67] Mehak, Rahul Kumar, and Dr. Ashima Mehta, "Artificial Intelligence," *International Journal of Advanced Research in Science, Communication and Technology. Naksh Solutions*, ss. 20–30, Apr. 26, 2023

[68] Wang, X., Han, Y., Leung, V.C.M., Niyato, D., Yan, X., Chen, X. *Fundamentals of Artificial Intelligence. In: Edge AI.* 2020

[69] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, Artificial Intelligence Review, c. 13, no. 2. *Springer Science and Business Media LLC*, ss. 129–170, 1999

[70] Shaveta, "A review on machine learning," *International Journal of Science and Research Archive*, c. 9, sayı. 1. GSC Online Press, ss. 281–285, 2023.

[71] K. K, "Machine Learning," *International Scientific Journal of Engineering and Management*, c. 02, sayı. 04, 2023.

[72] S. Yadav, S. Yadav, S. P. Srivastava, S. K. Gupta, and S. Mishra, "Machine Learning," *Deep Learning for Targeted Treatments*. Wiley, ss. 407–430, 2022.

[73] T. Dahiya, N. Vashishth, D. Garg, A. K. Shrivastava, and P. K. Kapur, "Novel Heuristic Algorithm its Application for Reliability Optimization," *International Journal of Mathematical, Engineering and Management Sciences*, c. 8, sayı. 4., ss. 755–768, 2023.

[74] C. Sathiyaraj Mr, M. Ramachandran prof, M. Amudha Mrs, and R. Kurinjimalar Miss, "A Review on Hill Climbing Optimization Methodology," *Recent trends in Management and Commerce*, c. 3, sayı. ss. 1–7, Ocak. 31, 2022.

[75] A. Y. Zomaya and R. Kazman, 'Simulated Annealing Techniques', in *Algorithms and Theory of Computation Handbook: General Concepts and Techniques*, c. 2., 2010.

[76] K.-L. Du and M. N. S. Swamy, “Tabu Search and Scatter Search,” Search and Optimization by Metaheuristics. *Springer International Publishing*, ss. 327–336, 2016.

[77] A. Kaveh, “Particle Swarm Optimization,” *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Springer International Publishing, ss. 9–40, 2014.

[78] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," in *IEEE Computational Intelligence Magazine*, c. 1, sayı. 4, ss. 28-39, 2006,

[79] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Advances in Engineering Software*, c. 69, pp. 46–61, 2014.

[80] M. Kumar, D. M. Husain, N. Upreti, and D. Gupta, “Genetic algorithm: Review and application*,” Available at SSRN 3529843*, 2010.

[80] K.-S. Tang, K.-F. Man, S. Kwong, and Q. He, “Genetic algorithms and their applications,” *IEEE signal processing magazine*, c. 13, sayı. 6, pp. 22–37, 1996.

[81] A. Lambora, K. Gupta, and K. Chopra, “Genetic algorithm-A literature review,” in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, 2019, ss. 380–384.

[82] A. J. Umbarkar and P. D. Sheth, “Crossover operators in genetic algorithms: a review.” *ICTACT journal on soft computing*, c. 6, sayı. 1, 2015.

[83] G. Syswerda and others, “Uniform crossover in genetic algorithms.,” in ICGA, 1989, c. 3, ss. 2–9.

[84] J. R. Koza, “Survey of genetic algorithms and genetic programming,” in *Wescon conference record*, 1995, ss. 589–594.

[85] J. H. Holland, “Genetic algorithms and adaptation,” *Adaptive control of ill-defined systems*, ss. 317–333, 1984.

[86] Kora, P., & Yadlapalli, P. (2017). Crossover operators in genetic algorithms: A review. *International Journal of Computer Applications*, 162(10).

[87] Bhandari, C. Murthy, and S. K. Pal, “Genetic algorithm with elitist model and its convergence,” *International journal of pattern recognition and artificial intelligence*, c. 10, sayı. 6, ss. 731–747, 1996.

[88] Gavish, B., & Graves, S. C. (1978). The travelling salesman problem and related problems.

[89]Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H., & Shmoys, D. B. (Eds.). (1985). *The traveling salesman problem: A guided tour of combinatorial optimization*. John Wiley & Sons, Inc.

[90]Papadimitriou, C. H., & Steiglitz, K. (1982). Combinatorial optimization: algorithms and complexity. *Courier Corporation.*

[91] Strutz, T. Redesigning the Wheel for Systematic Travelling Salesmen. *Algorithms* 2023, 16, 91

[92] H. T. Kahraman, S. Aras, and E. Gedikli, "Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms," *Knowledge-Based Systems*, c. 190. ss. 105169, 2020.

[93] S. P. Gochhayat, S. Shetty, R. Mukkamala, P. Foytik, G. A. Kamhoua and L. Njilla, "Measuring Decentrality in Blockchain Based Systems," in *IEEE Access*, c. 8, ss. 178372-178390, 2020,

[94] Y. Liu, Y. Lan, B. Li, C. Miao, and Z. Tian, "Proof of Learning (PoLe): Empowering neural network training with consensus building on blockchains," *Computer Networks*, c. 201. Elsevier BV, ss. 108594, Dec. 2021. doi: 10.1016/j.comnet.2021.108594.

[95] J. Chen, K. Duan, R. Zhang, L. Zeng, and W. Wang, "An AI Based Super Nodes Selection Algorithm in BlockChain Networks." *arXiv*, 2018.

[96] F. Bizzaro, M. Conti and M. S. Pini, "Proof of Evolution: leveraging blockchain mining for a cooperative execution of Genetic Algorithms," *2020 IEEE International Conference on Blockchain (Blockchain), Rhodes*, 2020, cc. 450-455,

[97] N. Shibata, "Proof-of-Search: Combining Blockchain Consensus Formation With Solving Optimization Problems," *IEEE Access*, c. 7, ss. 172994-173006, 2019,

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı :Fatih Kürşad GÜNDÜZ

Yabancı Dili :İngilizce

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Elektrik-Elektronik ve Bilgisayar Mühendisliği	Düzce Üniversitesi	2023
Y. Lisans	Enerji Sistemleri Müh.	Süleyman Demirel Üniversitesi	2016
Lisans	Bilgisayar Sistemleri Öğrt.	Süleyman Demirel Üniversitesi	2011
Lise	Bilgisayar Yazılımı	ATML Lisesi	2006

YAYINLAR

Tezden Çıkan SCI endeksli yayınlar

1. F. Gündüz, S. Birogul, and U. Kose, "Proof of Optimum (PoO): Consensus Model Based on Fairness and Efficiency in Blockchain," *Applied Sciences*, c. 13, no. 18. *MDPI AG*, p. 10149, 2023

SCI Endeskli Yayınlar

1. E. Eriskin, G. F. Turker, F. K. Gunduz, and S. Terzi, "Replacement of signalized traffic network design with Hamiltonian roads: delay? Nevermind," *Soft Computing*, c. 27, ss. 12. Springer Science and Business Media LLC, ss. 8245–8254, 2022.

Uluslararası toplantıda sunularak tam metin olarak yayımlanan bildiri:

1. GÜNDÜZ F. K., KIZILKAN Ö., KÜÇÜKSİLLE E. U. Optimization Of Shell and Tube Heat Exchangers With Wolf Colony Algorithm. *International Conferences on Science and Technology Engineering Science and Technology,* pp 171-180 2019.

Ulusal toplantıda sunularak tam metin olarak yayımlanan bildiriler:

1. GÜNDÜZ, Fatih Kürşad. E-İmza Destekli Elektronik Evrak Sistemi. *Akademik Bilişimi* 2013
2. GÜNDÜZ, Fatih Kürşad. Ve Biroğul, Serdar Blokzincir ne değildir. *IMASCON* 2