

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**ÇOKLU-ERİŞİM KENAR BİLİŞİM SİSTEMLERİNDE İŞLEM YÜKÜ
DEVİRİ KARAR YÖNTEMLERİNİN KARŞILAŞTIRILMASI**

Mehmet ALTINKURT

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2021**

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

ÇOKLU-ERİŞİM KENAR BİLİŞİM SİSTEMLERİNDE İŞLEM YÜKÜ DEVRİ KARAR YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Mehmet ALTINKURT

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Aykut KALAYCIOĞLU

Son yıllarda bir yandan kullanıcıların akıllı cihazlardan performans beklentilerinin yükselmesi diğer yandan nesnelerin interneti cihazlarının oldukça yaygınlaşması ilgili cihazlarda oluşan işlem yüklerinin hızlı ve etkin bir şekilde sonuçlandırılması için bilim insanlarını yeni yöntemler aramaya itmiştir. Bu problemin çözülmesinde uzun yıllar Bulut Bilişim kullanılmış ve halen de kullanılmaya devam etmektedir. Fakat Bulut Bilişim kullanıcıdan uzakta merkezi yapısı nedeniyle beklentileri karşılayamamaktadır. Diğer taraftan Çoklu Erişim Kenar Bilişim sistemleri ağı kullanıcıya yakın kenarında, baz istasyonlarına kurulması nedeniyle ilgili probleme çare oluşturabilecek niteliktedir. Bununla birlikte Kenar Bilişim sunucularının da kapasitesi sınırsız değildir. Bu kapsamda kısıtlı işlemci kapasitesinin kullanıcılar arasında nasıl dağıtılacağı yeni bir problem olarak ortaya çıkmaktadır. Tez çalışması kapsamında söz konusu problemin çözümü için önerilen bazı yöntemler çeşitli kıstaslara göre karşılaştırılmıştır. Bu yöntemler; Azınlık Oyunu Yöntemi, Q-öğrenme, Kazan Kal Kaybet Değiştir Stratejisi, Roth Erev Öğrenme ve Öğrenen Durum Makinaları yöntemleridir. Problem kaynak tahsis problemi olarak modellenmiş ve bu çerçevede benzetim çalışması yapılmıştır. Çalışmada öncelikle kullanılan yöntemlerde yer alan parametrelerin en uygun değerleri bulunmaya çalışılmış sonrasında bu parametreler kullanılarak yöntemler çeşitli kriterlere yöre karşılaştırılmıştır. Gerçekleştirilen çalışmalarda yöntemler kaynaklarda yer alan volatilité, ortalama refah, adalet, bir turun ortalama tamamlanma süresi ve bir turda harcanan ortalama enerji kriterleri bakımından incelenmiştir.

Eylül 2021, 67 sayfa

Anahtar Kelimeler: Çoklu-Erişim Kenar Bilişim, Azınlık Oyunu, Pekiştirmeli Öğrenme, Oyun Teorisi

ABSTRACT

Master Thesis

COMPARISON OF OFFLOADING DECISION SCHEMES FOR MULTI-ACCESS EDGE COMPUTING SYSTEMS

Mehmet ALTINKURT

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Asst. Prof. Aykut KALAYCIOĞLU

In recent years the increase in the performance expectations of the users from smart devices together with the widespread use of the Internet of things devices has led scientists to seek new methods in order to quickly and effectively conclude the processing loads on the devices. Cloud Computing has been used for many years to solve this problem and it still continues to be used. However, Cloud Computing cannot meet the expectations due to its remote centralized structure from the user. On the other hand, Multiple-Access Edge Computing systems are capable of solving the problem due to the fact that they are installed on base stations at the edge of the network close to the end user. However, the capacity of Edge Computing servers is not unlimited. In this context, a new problem emerges: how to distribute the limited processor capacity among users. Within the scope of this thesis, some methods proposed to solve the problem in question were compared according to various criteria. These methods are; Minority Game Method, Q-learning, Win Stay Lose Shift Strategy, Roth-Erev Learning and Learning Automata. The problem was modeled as a resource allocation problem and a simulation study was carried out in this framework. In the studies, first of all, the most appropriate values of the parameters in the methods used were tried to be found, then these parameters were used and the methods were compared according to various criteria. In the studies carried out, the methods were examined in terms of volatility, average welfare, fairness, average completion time of a round and average energy spent in a round.

September 2021, 67 pages

Key Words: Multi-Access Edge Computing, Minority Game, Reinforcement Learning, Game Theory

TEŞEKKÜR

Çalışmalarım sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren, değerli zamanını esirgemeyen, danışmanım Sayın Dr. Öğr. Üyesi Aykut KALAYCIOĞLU'na (Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı) teşekkürlerimi ve saygılarımı sunarım.

Bu süreçte desteklerini hiçbir zaman esirgemeyen eşime teşekkür ederim.

Mehmet ALTINKURT
Ankara, Eylül 2021



İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK	i
ÖZET.....	ii
ABSTRACT	iii
TEŞEKKÜR	iv
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	viii
1. GİRİŞ	1
2. KURAMSAL TEMELLER ve KAYNAK ÖZETLERİ	8
2.1 Kuramsal Temeller	8
2.1.1 İşlem yükü devri.....	8
2.1.2 Oyun teorisi	9
2.1.3 Kaynak tahsis problemi.....	11
2.1.4 Azınlık Oyunu modeli.....	12
2.2 Kaynak Özetleri	14
3. KULLANILAN YÖNTEMLER VE BENZETİM ÇALIŞMASI.....	16
3.1 Sistem Modeli	16
3.2 Karşılaştırma Kriterleri	17
3.2.1 Volatilité.....	18
3.2.2 Ortalama refah.....	20
3.2.3 Adalet	20
3.2.4 Bir turun ortalama tamamlanma süresi	21
3.2.5 Bir turda harcanan ortalama enerji.....	24
3.3 Kullanılan Yöntemler	25
3.3.1 Azınlık Oyunu yöntemi.....	25
3.3.2 Q-öğrenme yöntemi.....	29
3.3.3 Kazan Kal Kaybet Değiştir stratejisi	31
3.3.4 Roth-Erev öğrenme yöntemi	34
3.3.5 Öğrenen Durum Makinaları yöntemi	35
3.3.6 Rastgele karar sistemi.....	38

3.3.7 Merkezi karar sistemi	38
4. ARAŞTIRMA BULGULARI	39
4.1 Yöntem Parametrelerinin Belirlenmesi	39
4.1.1 Azınlık Oyunu yöntemi.....	39
4.1.2 Q-öğrenme yöntemi.....	41
4.1.3 Kazan Kal Kaybet Değiştir stratejisi	43
4.1.4 Roth-Erev öğrenme	44
4.1.5 Öğrenen Durum Makinaları yöntemi	44
4.2 Yöntemlerin Karşılaştırılması.....	46
4.2.1 Volatilité	46
4.2.2 Ortalama refah	48
4.2.3 Adalet	49
4.2.4 Bir turun ortalama tamamlanma süresi	51
4.2.5 Bir turda harcanan ortalama enerji.....	53
5. TARTIŞMA VE SONUÇ.....	55
KAYNAKLAR	62
ÖZGEÇMİŞ.....	67

ŞEKİLLER DİZİNİ

Şekil 1.1 Kenar Bilişim Mimarisi.....	3
Şekil 2.1 El-Farol Bar Problemi zaman içerisinde bara giden kişi sayıları.....	13
Şekil 3.1 Azınlık Oyunu Yöntemi Algoritma Diyagramı.....	28
Şekil 3.2 Q-Öğrenme Yöntemi Algoritma Diyagramı.....	31
Şekil 3.3 Kazan Kal Kaybet Değiştir Stratejisi Algoritma Diyagramı.....	33
Şekil 3.4 Roth-Erev Öğrenme Stratejisi Algoritma Diyagramı	35
Şekil 3.5 Öğrenen Durum Makinaları Yöntemi Algoritma Diyagramı.....	37
Şekil 4.1 Azınlık oyunu yönteminde hafıza miktarı 1, 2 ve 3 iken değişen strateji sayılarına göre volatilitenin değişimi	40
Şekil 4.2 Azınlık oyunu yönteminde hafıza miktarı 6'ya strateji sayısı 25'e kadar iken değişen strateji sayılarına göre volatilitenin değişimi	41
Şekil 4.3 Q-Öğrenme yönteminde öğrenme katsayısı (α) ve keşfetme katsayısı (ϵ) değerlerinin değişimine göre volatilitenin değişimi	42
Şekil 4.4 Kazan Kal Kaybet Değiştir stratejisinde karar değiştirme olasılığı (p) değerinin değişimine göre volatilitenin değişimi	43
Şekil 4.5 Roth-Erev yönteminde indirgeme faktörü (λ) değerinin değişimine göre volatilitenin değişimi	44
Şekil 4.6 Öğrenen Durum Makinaları Yönteminde ödül katsayısı (α) ve ceza katsayısı (β) değerlerinin değişimine göre volatilitenin değişimi	45
Şekil 4.7 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan volatilité değerlerini gösteren grafik	47
Şekil 4.8 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan volatilité değerlerini gösteren grafik (0-0.3 arası).....	48
Şekil 4.9 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan ortalama refah değerlerini gösteren grafik.....	49
Şekil 4.10 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan adalet değerlerini gösteren grafik	50
Şekil 4.11 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan adalet değerlerini gösteren grafik (0-500 aralığı).....	51
Şekil 5.1 21, 1001 ve 2001 oyuncu ile yapılan çalışmalarda bulunan volatilité değerleri	56
Şekil 5.2 21, 1001 ve 2001 oyuncu ile yapılan çalışmalarda bulunan ortalama refah değerleri.....	58
Şekil 5.3 21, 1001 ve 2001 oyuncu ile yapılan çalışmalarda bulunan ortalama refah değerleri.....	59
Şekil 5.4 Bir turun ortalama tamamlanma süreleri.....	60
Şekil 5.5 Bir turda harcanan ortalama enerji.....	61

ÇİZELGELER DİZİNİ

Çizelge 3.1 Azınlık Oyununda 2 bitlik geçmiş hafızasına sahip oyuncunun tam strateji tablosu.....	26
Çizelge 3.2 Azınlık Oyununda 2 bitlik geçmiş hafızasına sahip oyuncunun indirgenmiş strateji tablosu (Strateji Sayısı 3).....	26
Çizelge 4.1 Süre açısından inceleme için kullanılan parametreler.....	52
Çizelge 4.2 süre açısından yapılan incelemede yöntemlerin bir turu ortalama olarak tamamlama süreleri (sn).....	52
Çizelge 4.3 Enerji açısından inceleme için kullanılan parametreler.....	53
Çizelge 4.4 Enerji açısından yapılan incelemede yöntemlerin bir turda oyuncu başına ortalama olarak harcadıkları enerji (Joule).....	54

1. GİRİŞ

Geçtiğimiz 10 yılda akıllı telefonlar tüm dünyada çok hızlı bir şekilde yaygınlaşmıştır. Yapılan bir araştırmaya göre 2016 yılında 2,5 milyar olan akıllı telefon kullanıcısı sayısının 2021 yıl sonu itibariyle 3,8 milyarı bulması beklenmektedir (Statista 2020). Bununla birlikte akıllı cihazların giderek yaygınlaşması kullanıcıların hem veri alışverişi hem de işlem gücü talebini artırmıştır. Öyle ki kullanıcıların akıllı cihazlardan beklediği veri alışverişi ve alınan verinin işlenmesi performansını bu cihazlar artık karşılayamamaktadır. Ayrıca batarya teknolojisi gittikçe gelişse de yüksek performans talebi karşısında yine de yetersiz kalmaktadır (Lin vd. 2014). Yapılan bir araştırma iPhone kullanıcılarının en fazla dikkat ettikleri hususun batarya olduğunu göstermektedir (Paczkowski, 2009).

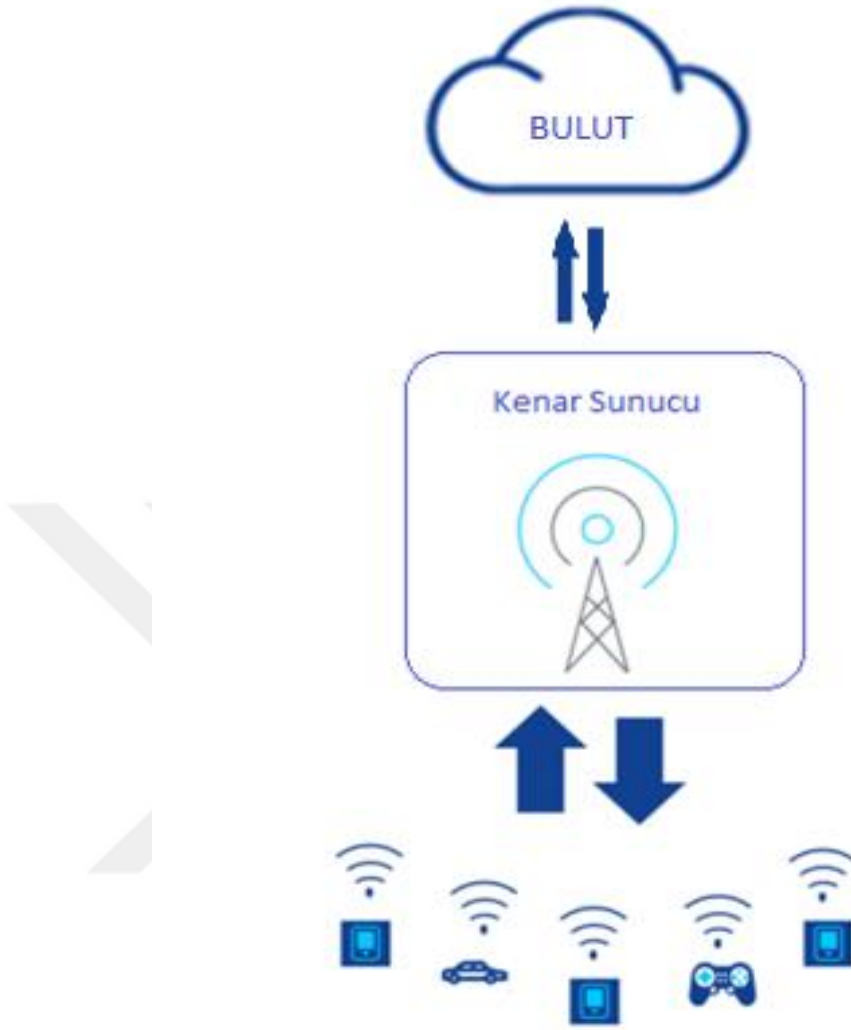
Diğer taraftan günümüzde kullanılan cihaz/ekipmanların internet ortamında birbirleriyle iletişim kurmalarını sağlayan Nesnelerin İnterneti (Internet of Things, IoT) cihazları da hızla yaygınlaşmaktadır. IHS Markit tarafından yapılan bir araştırmaya göre 2017 yılında 27 milyar olan IoT cihaz sayısının 2030 yılına kadar yıllık ortalama %12 artışla 125 milyara yükselmesi beklenmektedir (Anonymous 2017).

Günümüze kadar artan bu tür ihtiyaçlar çeşitli çözümleri de beraberinde getirmiştir. Bunlardan biri bulut bilişimdir. Bulut bilişim bilgisayarların yaptığı her türlü hizmeti merkezi sunucular üzerinden kullanıcıların hizmetine sunan bir olgudur. Örneğin veri depolama kapasitelerini geliştirmek için kurulan büyük veri merkezleri birçok kullanıcının veri depolama ihtiyacını karşılamış, ayrıca kurulan bulut sunucuları üzerinde kullanıcıların büyük işlem kapasitesi gerektiren uygulamalarının işlem ihtiyaçları karşılanmıştır. Bu yönüyle bilgi işlem stratejileri ve altyapı çalışmaları uzun bir süre bulut bilişim yörüngesinde yön bulmuştur.

Bulut bilişimin güçlü işlem hızına karşın veri iletim altyapısı bu değişime yeterince hızlı ayak uyduramamıştır. Bu nedenle bulut bilişim sistemleri için en büyük darboğaz veri iletim hızında olmuştur. Özellikle nesnelerin interneti tarafından oluşturulan devasa verilerin zamanında iletimi sorun haline gelmiştir. Örneğin; Boeing 737 tipi uçağın her uçuşta yarım terabyte veri ürettiği bilinmektedir (Finnegan 2013). Diğer bir örnek olarak sürücüsüz olarak yolda seyredebilen otonom bir arabanın saniyede bir gigabyte veri

ürettiđi bilinmekte ve aracın güvenli olarak sürüşe devam edebilmesi için bu verinin çok kısa bir süre içerisinde işlenmesi gerekmektedir. Bulut bilişimde sunucunun merkezde, kullanıcıdan uzak bir noktada bulunmasından dolayı verinin iletilmesi uzun süre almaktadır. Ang Li ve arkadaşları tarafından yapılan bir çalışmada bulut hizmet sağlayıcılarına çift yönlü veri aktarım süresinin 250 milisaniyeye kadar uzayabildiđi görölmüşür. Özellikle gecikmeye duyarlı uygulamalarda bu süre gereksinimleri karşılamamaktadır. (Li vd. 2010).

Bu noktada Mobil Ağ Kenar Bilişim (Mobile Edge Computing, MEC) kilit bir teknoloji olarak devreye girmektedir. Bulut sistemleri merkezi tek bir noktaya kurulurken MEC sistemleri kullanıcıya yakın olan baz istasyonlarına kurulmaktadır. Bu da kullanıcı taleplerinin çok kısa sürelerde ve daha az hata ile karşılanmasını sağlamaktadır.



Şekil 1.1 Kenar Bilişim Mimarisi

MEC Avrupa Telekomünikasyon Standartları Enstitüsü (European Telecommunications Standards Institute, ETSI) tarafından standartlaştırılan nispeten yeni bir teknolojidir. ETSI tarafından MEC Bilgi İşlem hizmetleri ortamını ve bulut bilişim becerilerini mobil ağın kenarında, Radyo Erişim Ağı (Radio Access Network, RAN) içerisinde ve mobil kullanıcıya yakın olan bir mesafede sağlayan yapıdır şeklinde tanımlanmaktadır (Hu vd. 2015). Bununla birlikte kenar bilişim sistemlerinin sadece mobil kullanıcılara değil aynı zamanda IoT kapsamında pek çok farklı cihaza da hizmet vereceğinden MEC Kısaltması Eylül 2017’de ETSI tarafından Çoklu Erişim Kenar Bilişim (Multi-Access Edge Computing) olarak değiştirilmiştir (Reznik 2017). MEC farklı kaynaklarda Sis Bilişim (Fog Computing), Bulutçuk (Cloudlet), Mikro Verimerkezi (Micro Datacenters) gibi

isimlerle de anılmaktadır (Satyanarayanan 2017). Diğer taraftan Sis Bilişim’de daha çok işletmenin kendi sahip olduğu sunucular kullanılırken MEC sisteminin altyapısı çoğunlukla operatörlere aittir (Tran vd. 2017a).

İlk MEC platformu 2013 yılında Nokia firması tarafından hayata geçirilmiştir (Anonymous 2013a, Satyanarayanan 2017). IBM tarafından yapılan başka bir çalışmada telefon operatörlerinin kenar bilişim kullanarak kullanıcılar ve şirketler için yenilikçi servisler ortaya çıkarabilecekleri belirtilmiştir (Anonymous 2013b). Saguna firması tarafından Open-RAN isimli tamamen sanal bir MEC sistemi kurulmuştur (Anonymous 2016).

MEC sistemlerinde bir baz istasyonuna ya da birkaç baz istasyonunun arasında uygun bir konuma sunucu kurulmakta ve bu sunucu ile kullanıcılara çeşitli hizmetler verilmektedir. MEC sistemleri tarafından sunulan en önemli servisler işlem yükü devri (computation offloading) ve önbellekleme (caching)’dir. İşlem yükü devri bir cihaz tarafından yapılması gereken yüksek işlem kapasitesi gerektiren bir işlemin MEC sunucusu tarafından yapılmasıdır. Bu servis basitçe şu şekilde kullanılır. Cihaz (akıllı telefon ya da IoT cihazı) tarafından işleme ilişkin bilgiler kenar sunucusuna iletilir; sunucu gerekli işlemleri yapar ve sonuçta çıkan bilgileri işlem yükü devri yapan cihaza geri gönderir. Örnek olarak görüntü tanıma işlemi son zamanlarda oldukça yaygınlaşan ve yüksek işlem kapasitesi gerektiren bir uygulamadır. Böyle bir işlemin cihaz tarafından gerçekleştirilmesi yerine işlem gücü çok daha yüksek olan bir sunucu tarafından yapılarak sonucun cihaza geri gönderilmesi çok daha hızlı ve enerji açısından tasarruflu olacaktır. Önbellekleme ise cihaz için gerekli olacak bir verinin belirli bir süre sunucu üzerinde tutulmasıdır. Örnek olarak belirli bir bölgede çok fazla izlenilen bir video MEC sunucusunda önbelleklenerek yeni izleyiciler için zaman kaybetmeden ve ağ omurgasına yük oluşturmada sunulabilmektedir. Tez kapsamında MEC’in işlem yükü devri servisine ilişkin çalışma yürütülmüş olduğundan diğer servislerine ilişkin ayrıca bilgi verilmeyecektir. Bu özelliklere ilaveten MEC’in işlem yoğun uygulamaları ağ tarafında tutarak mobil kaynakların optimize edilmesi, büyük verinin bulut bilişim sunucularına iletilmeden önce önişleme tabi tutulması ve hücre yoğunluğu, kullanıcı konumları ve radyo kanal kaynaklarının tahsisi gibi içeriğe duyarlı servislerin sunulması gibi faydaları

da bulunmaktadır (Tran vd. 2017b). Bu tez çalışması kapsamında kenar bilişimde işlem yükü devri özelliği üzerinde durulacaktır.

İşlem yükünü aktarmanın bulut bilişim açısından etkileri kaynaklarda sıkça tartışılmıştır. Ancak bu çalışmaların birçoğunda bulut bilişimde kaynakların sınırsız olduğu varsayılmaktadır. Bu da devredilen işlem yükünün tamamlanma süresindeki gecikmeleri göz ardı etmektedir. Son zamanlarda gerçekleştirilen çalışmalarda işlem yükünün MEC sunucusuna devredilmesi yönünde çalışmalar yapılmıştır. Shanhe Yi ve arkadaşları tarafından yapılan bir çalışmada, yüz tanıma uygulaması ile bir deneme yapılmış ve bulut bilişim üzerinde cevap süresi 900 ms iken kenar bilişim üzerinde 169 ms'ye düşmüştür (Yi vd. 2015). Kiryong ha tarafından giyilebilir cihazlar üzerine yapılan başka bir çalışmada işlem yükünün MEC sunucusuna devrinde enerji sarfiyatının da %30 - %40 azaldığı görülmüştür (Ha vd. 2014). Diğer taraftan kenar bilişimin market payının da gittikçe artacağı öngörülmektedir. ABI Research tarafından yapılan bir araştırmaya göre yapay zeka uygulamalarının kenar bilişimde işlenme oranı 2017'de %6 iken 2023 yılında bu oranın %43'e çıkması beklenmektedir (Bay, O. 2018).

MEC sunucuları yalnızca bir veya birkaç baz istasyonunun kapsadığı alana servis sağlamaktadır. Bu nedenle kapasitesi bulut bilişim sunucuları kadar yüksek değildir. Buna karşın hizmet verilen bölgelerde sunucuların kapasitesinden çok daha fazla işlem yükü devri talebi olabilmektedir. Bu durumda MEC işlemci sunucu kapasitesinin kullanıcılar arasında etkin paylaşımı çözülmesi gereken önemli bir problemidir. Tez çalışması bu probleme odaklanmaktadır.

MEC sunucusu işlemci kapasitesinin, işlem yükünü sunucuya devretmek isteyen cihazlar arasında etkin bir şekilde nasıl paylaşılacağı problemi literatürde kaynak tahsis (Resource Allocation) problemi olarak ele alınmaktadır. Söz konusu problemin çözümünde kaynaklarda problem Azınlık Oyunu (Minority Game) olarak modellenmektedir. Benzer şekilde söz konusu problem bu tez çalışmasında da Azınlık Oyunu olarak modellenmiş ve literatürde ilgili problemin çözümünde kullanılan bazı yöntemler incelenerek karşılaştırılmıştır. Çalışmada sınırlı kapasiteye sahip tek bir kenar bilişim sunucusuna işlem yükü devri yapmak isteyen birden fazla kullanıcının olduğu bir sistem varsayımı

yapılmıştır. Ayrıca problemin sadeleştirilmesi açısından tüm kullanıcıların aynı anda devredilebilecek işlem yükü oluşturduğu ve her bir turun aynı anda başladığı bir sistem tasarlanmıştır. Kuyruklama (Queing) yöntemi olarak Kleinrock tarafından anlatılan (1967) İşlemci Paylaşımı (Processor Sharing) yöntemi kullanılmıştır.

Tez çalışması kapsamında problemin çözümü için bazı Oyun Teorisi (Game Theory) yöntemleri ile Pekiştirmeli Öğrenme (Reinforcement Learning) yöntemleri incelenmiştir. İncelenen Oyun Teorisi yöntemleri: Azınlık Oyunu Yöntemi (Arthur 1994, Challet ve Zhang 1997) ve Kazan Kal Kaybet Değiştir Stratejisi (Reents vd. 2001), Pekiştirmeli Öğrenme yöntemleri ise: Q-öğrenme (Watkins and Dayan 1992), Roth-Erev Öğrenme (Roth ve Erev 1995) ve Öğrenen Durum Makinaları (Cattueuw ve Manderick 2011) yöntemleridir. Ayrıca karşılaştırma yapılabilmesi için kararların merkezi bir noktadan alındığı varsayılan bir sistem ile oyuncuların kararlarını düzgün dağılımlı olarak rastgele verdiği bir sistem benzetimi yapılmıştır.

Karşılaştırılan yöntemlerde oyuncular çeşitli kurallar kullanarak karar vermektedir. Bu kurallar çoğunlukla belirli sabit parametreler içeren matematiksel denklemlerden oluşmaktadır. Tez kapsamında kullanılan yöntemlerde parametrelerin sonuca etkisi incelenmiş, belirli şartlar altında daha uygun parametrelerin seçimine özen gösterilmiştir.

Yöntemlerin en iyi sonucu elde edebilmesi için en uygun sabitlerin seçilebilmesi amacıyla çalışma yapılmış olup sabitler için elde edilen değerler yöntemlerin karşılaştırılması aşamasında kullanılmıştır.

Çalışmada tüm oyuncuların aynı anda işleme başladığı bir sistem tasarlandığı için Dinamik Sıralı Oyun (Dynamic Sequential Game) (Deng vd. 2016) gibi sıra temelli yöntemler ve tüm oyuncular birbirinin eşdeğeri olarak tasarlandığı için Stackelberg Game (Zheng vd. 2017) gibi lider temeli sistemler kullanılmamıştır.

Çalışma kapsamında Kenar Bilişim için birden fazla kullanıcının ve tek bir kenar bilişim sunucusunun bulunduğu bir benzetim ortamı oluşturulmuştur. Kodlama dili olarak

Python kullanılmıştır. Literatürde çeşitli kenar bilişim benzetim ortamları bulunmakla birlikte, bu ortamlar çoğunlukla kompleksir ve hizmet sunulan cihazların benzetimi, ağın ve kanalın durumu, ağda kullanılan protokol gibi birçok olgu hesaba katılmaktadır (Jha vd. 2020). Oysa bu tez çalışmasında işlem yükü devri kararı için kullanılan yöntemler karşılaştırılmaktadır ve bu karşılaştırma için oldukça basit bir benzetim ortamı yeterlidir. Bu nedenle hazır benzetim ortamlarının kullanılması yerine basit bir benzetim ortamının oluşturulması daha uygun görülmüştür.

Yapılan çalışmalarda Çoklu Erişim Kenar Bilişim konusunun kullanıcıya olan faydaları ele alındığından sunucunun harcadığı enerji, sunucudan faydalanma (server utilization) gibi konulara girilmemiştir.

Yapılan çalışmalar neticesinde farklı yöntemlerin farklı kriterler için daha iyi sonuç verdiğini söylemek mümkündür, yani tüm kriterler için en iyi sonucu veren bir yöntem bulunmamaktadır. Tartışma ve Sonuç bölümünde her bir kriter için çıktılar ayrı ayrı değerlendirilmiştir.

Tez, beş bölümden oluşmaktadır. Birinci bölümde, tez çalışmasının amacı ve kapsamından bahsedilmiştir. İkinci bölümde, konuya ilişkin kuramsal temellere yer verilmiş ve kaynaklarda yer alan bazı çalışmalar anlatılmıştır. Üçüncü bölümde karşılaştırma yapılacak yöntemler hakkında bilgi verilmiş ve oluşturulan benzetim ortamı anlatılmıştır. Dördüncü bölümde elde edilen araştırma bulguları sonuçlarından bahsedilmiştir. Son bölümde ise yapılan çalışma hakkında değerlendirme yapılmıştır.

2. KURAMSAL TEMELLER ve KAYNAK ÖZETLERİ

Bu bölümde benzetim çalışmasına temel oluşturması için kuramsal temellerden ve konu ile ilgili daha önce yapılmış çalışmalardan bahsedilmektedir. Kuramsal temeller bölümü dört kısımdan oluşmaktadır. İlk kısımda İşlem Yüğü Devri, ikinci kısımda Oyun Teorisi, üçüncü kısımda Kaynak Tahsis Problemi dördüncü kısımda ise Azınlık Oyunu Modeli hakkında kısaca bilgi verilmektedir. Kaynak özetleri bölümünde konu ile ilgili yapılmış olan çalışmalara değinilmiştir.

2.1 Kuramsal Temeller

2.1.1 İşlem yüğü devri

Kullanıcı cihazları tarafından üretilen işlem talepleri ya kullanıcı cihazı tarafından işlenir ya da işlenmek üzere kenar bilişim sunucularına iletilir. Sunucuda işlenen veriler sonucunda elde edilen sonuçlar işlem talebini üreten kullanıcı cihazına geri iletilir. İşlemin kenar bilişim sunucusu tarafından işlenerek sonucun kullanıcı cihazına geri iletilmesine işlem yüğü devri denir. İşlem yüğü devri konsepti ilk olarak 1970'lerin başlarında dağıtık sistemlerde yük dengelemek amacıyla kullanılmıştır. Bu kavram kablosuz iletişimin gelişmesiyle geniş anlamda incelenmeye başlanmıştır (Yang vd. 2008).

İşlemin yapılış şekline göre üç çeşit yük devri mevcuttur (Hasan 2018). İkili (binary) yük devrinde kullanıcı cihazında oluşan bir işlem ya tamamen devredilir ya da tamamen kullanıcı cihazında gerçekleştirilir. Parçalı yük devrinde işlemler parçalara ayrılır. Bu parçalardan bir kısmı kullanıcı cihazında gerçekleştirilirken kalan kısmı ise kenar ya da bulut sunucusuna devredilir. Tam yük devrinde ise kullanıcı cihazında oluşan tüm işlem talepleri devredilir. Tez çalışması kapsamında sistemin yalnızca ikili yük devrini desteklediği varsayılmaktadır. Yani kullanıcılar oluşan işlem yükünün tamamını ya kendisi işlemekte ya da MEC sunucusuna devretmektedir.

Belirli bir bölgede birçok kullanıcı aynı kenar sunucusuna işlem yükü devretmek istediğinde, kenar sunucusu kapasite limitine ulaşacağından gecikmeler yaşanacaktır. Böyle bir durumla karşılaşmaktansa kullanıcı işleminin yerel sistem tarafından hesaplanması daha uygun olacaktır.

İşlem yükü devri üç temel adımdan oluşmaktadır: bölütleme, hazırlama ve karar verme (Akherfi vd. 2018). Bölütleme işlemi verinin transfer edilebilecek boyutlara bölünmesi işlemidir. Hazırlama adımında uzak sunucu seçimi, işlemin sunucu tarafında gerçekleştirilebilmesi için kodun sunucuya kurulması, vekalet (Proxy) işlemlerinin tamamlanması gibi işlemler gerçekleştirilir. Karar verme aşamasında ise parçalara ayrılan işlem yükünün sunucu tarafında işlenip işlenmeyeceğine karar verilir. Karar verme işlemi süreç başladığında yani gerçek zamanlı olursa kanal durumu, sunucu yoğunluğu gibi bilgiler sürekli güncelleneceği için bu bilgiler ışığında daha net kararlar verilebilecektir.

2.1.2 Oyun teorisi

Oyun Teorisi kaynak tahsisi konusunda sık kullanılan bir matematiksel modelleme yöntemidir (Ranadheera vd. 2017). Oyun teorisi oyuncular arasındaki stratejik etkileşime dayanır ve bu yönüyle merkezi yöneticiye gerek duymaz (Sastry vd. 1994, Zheng vd. 2015). Oyun teorisinde en temel varsayım tüm oyuncuların rasyonel davrandığıdır. Yani her oyuncu kendi faydasını maksimize etmeye çalışmaktadır. Her bir oyuncu yapacağı hamleyi planlarken diğer oyuncuların yapacağı hamleyi tahmin etmeye çalışmak durumundadır.

Oyun teorisi ilk olarak matematikçi John Von Neumann tarafında 1920'li yıllarda ortaya atılmış olmakla birlikte 1944 yılında Oskar Morgenstern ile birlikte yazdığı Theory of Games and Economic Behavior isimli kitabı ile yaygınlaşmıştır (Von Neumann, 1944). 1950 yılına gelindiğinde John Forbes Nash Jr., Nash Dengesi (Nash Equilibrium) teorisiyle Oyun Teorisi kapsamındaki her oyunda bütün oyuncuların karar değiştirerek daha iyi bir fayda elde edemeyeceği bir denge noktası olduğunu ortaya atmıştır (Nash, 1950)

Oyun teorisinde kullanılan temel kavramlar aşağıdaki gibidir;

Oyun (Game): Belirli bir hedef doğrultusunda stratejik etkileşimin gerçekleştiği her türlü olaya oyun denir.

Oyuncu (Agent): Oyun içerisinde kendi çıkarlarını düşünerek kararlar veren mekanizmalara oyuncu denir. Oyuncunun gerçek kişi olma zorunluluğu yoktur, karar verme yeteneğine sahip her türlü varlık oyuncu olabilir.

Strateji: Oyuncuların oyun içerisinde verebileceği kararlara strateji denir.

Fayda (Payoff-Utility): Oyuncuların oyun içerisinde elde ettiği olumlu sonuçlara fayda denir.

Nash Dengesi (Nash Equilibrium): Bütün oyuncuların karar değiştirerek daha iyi bir fayda elde edemeyeceği duruma Nash Dengesi denir.

Oyun teorisinde kullanılan strateji türleri aşağıdaki gibidir.

Saf Strateji (Pure Strategy): Bir oyuncu her durumda aynı kararı veriyorsa yani bu kararı verme olasılığı %100 ise bu stratejiye oyuncu için saf strateji denir.

Karma Strateji (Mixed Strategy): Oyuncunun farklı durumlarda farklı kararlar verdiği stratejiye karma strateji denir.

Optimal Strateji: Oyuncunun kullanabileceği seçenekler arasında oyuncuya mümkün olan en büyük faydayı sağlayacak olan stratejiye optimal strateji denir.

Baskın Strateji (Dominant Strategy): Bir oyuncuya diğer oyuncuların verdiği karar ne olursa olsun her zaman en yüksek faydayı sağlayacak stratejidir.

Oyun teorisinin iki ana dalı bulunmaktadır: işbirlikçi (cooperative) oyun ve işbirlikçi olmayan (non-cooperative) oyun. İşbirlikçi oyunlarda oyuncular iş birliği sergiler ve her oyuncu için faydalı olan bir davranış belirlemeye çalışırlar. İşbirlikçi olmayan oyunlarda ise her oyuncu rasyonel ve bencil olarak düşünülmekte ve oyuncular birbirlerine karşı yarışmaktadır. Mobil cihazların genellikle gerçek kullanıcıları bulunduğundan, işlem yükü devri problemini oyun teorisi ile modellemek uygun olacaktır.

2.1.3 Kaynak tahsis problemi

Kısıtlı sayıda kaynağın belirli sayıda kullanıcı (oyuncu-ajan) arasında otonom bir şekilde dağıtılması bilgisayar biliminde ve ekonomide sık rastlanan bir araştırma alanıdır. Bu alandaki problemin çözümü için kaynak tahsisi probleminin çözülmesi gereklidir. Kaynak tahsisi sınırlı miktardaki kaynağın belirli sayıdaki kullanıcıya dağıtılması işlemidir (Chevalleyre vd. 2004).

Genel olarak kaynakların yapısı itibarıyla bakıldığında iki çeşit kaynak bulunmaktadır: bölünebilen kaynaklar ve bölünemeyen kaynaklar. Bölünemeyen kaynaklar yalnızca tek bir kullanıcıya hizmet edebilirken bölünebilen kaynaklar aynı anda birden fazla kullanıcıya hizmet verebilmektedir. Bu açıdan bakıldığında MEC sistemleri bölünebilir kaynak olarak değerlendirilebilir (Lam vd. 2007).

Tahsis işlemi karar vericiye göre iki kısma ayrılmaktadır: merkezi (centralized) ve dağıtık (distributed). Merkezi karar alınan sistemde tek bir karar verici kaynağın kullanıcılar arasında nasıl dağıtılacağına karar vermektedir. Dağıtık karar verilen bir sistemde ise her kullanıcı kaynağı kullanıp kullanmamak konusunda belirli kriterlere göre kendi kararını vermektedir.

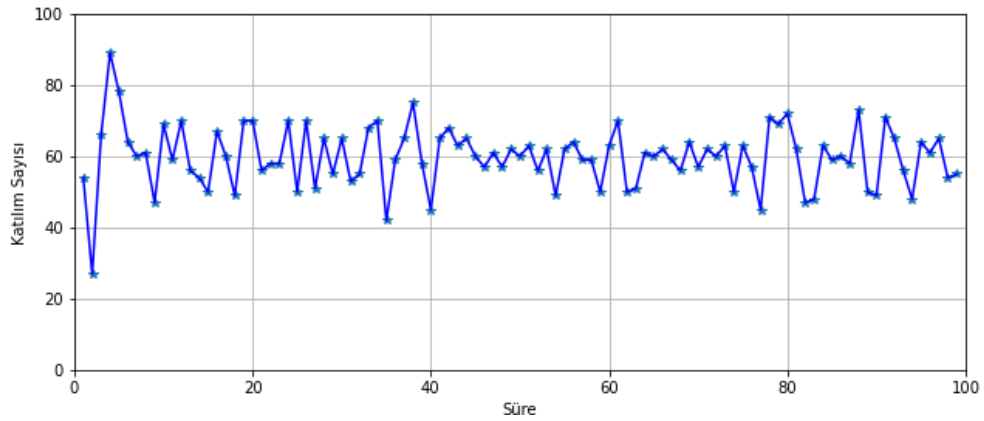
Tezde ele alınan problemin merkezi bir kaynak tahsis sistemi tarafından çözülebilmesi mümkün değildir. Bunun nedeni merkezi sistemlerde dağıtıma karar veren noktada tüm bilginin var olmasının gerekliliğidir. Gerçek bir MEC sisteminde kullanıcıların oluşturacakları yük miktarları, iletim kanalının durumu, ağda bulunan kullanıcı sayısı gibi

etmenler her an deęişebilmektedir. Bu öngörülemeyen ani deęişiklikleri merkezi bir sistemin bilmesi mümkün olmadığından kaynak tahsisi dağıtık bir sistem tarafından yapılmalıdır (Ranadheera vd. 2017).

Tezde bahsedilen problemde tek bir kaynak ve çok sayıda kullanıcı mevcuttur. Eğer kaynağı kullanmak isteyen kullanıcı sayısı kaynağın kapasitesinden fazla olursa kaynak aşırı yüklenmiş olacaktır. Eğer bu sayı kaynak kapasitesinden az olursa kaynağı kullanan kullanıcılar bir fayda elde etmiş olacaklardır.

2.1.4 Azınlık Oyunu modeli

Azınlık oyunu kaynak tahsis problemlerinde kullanılan bir oyun teorisi modelidir. Tek kaynağın çok sayıda kullanıcı tarafından kullanıldığı sistemler ilk olarak 1994 yılında W. Brian Arthur tarafından El Farol Bar Problemi olarak ortaya atılmıştır (Arthur 1994). Bu probleme göre bir kasabada bulunan N sayıda kişi (Örnek olarak $N=100$) kasabada bulunan tek bara gitmek ya da gitmemek konusunda karar vermeye çalışmaktadır. Barın kapasitesi 60 kişidir. Bara giden kişi sayısı 60'tan fazlaysa bar çok kalabalık olacağından bara gitmemek daha uygun olacaktır. Diğer taraftan eğer bardaki kişi sayısı 60'ın altındaysa bara gitmek evde kalmaktan daha iyi bir seçenek haline gelecektir.



Şekil 2.1 El-Farol Bar Problemi zaman içerisinde bara giden kişi sayıları (Arthur 1994)

Makalede yazar çeşitli denemeler sonucunda bara giden kişi sayısının Şekil 2.1’deki gibi olduğunu görmüştür. Bu grafik bara giden kişi sayısının döngü oluşturmeyen bir şekilde değiştiği ve barda bulunan kişi sayısının barın kapasitesi civarında salındığı sonucunu ortaya çıkarmıştır. Salınım başlangıçta geniş aralıkta başlamışken sonrasında dar bir aralıkta devam etmiştir. Buradan halkın geçmiş deneyimleri sonucunda daha doğru kararlar verdikleri sonucu çıkmaktadır. Zaman içerisinde halkın verdiği kararlar neticesinde barın doluluk oranı sürekli olarak barın kapasitesine yakın bir değer çıkmaktadır.

El Farol Bar Problemi ilk kez Challet ve Zhang (1997) tarafından formülize edilmiştir. Yazarlar makalede ilgili problem için oluşturdukları modele azınlık oyunu ismini vermişler ayrıca problemin çözümü için strateji tablolarından oluşan temel bir yöntem ortaya atmışlardır. Sonrasında yazılan kaynaklarda söz konusu oyun teorisi modeli azınlık oyunu modeli, çözümde kullanılan yöntem ise azınlık oyunu yöntemi olarak kullanılmıştır. Ayrıca sonraki çalışmalarda azınlık oyunu modelinin çözümü için farklı yöntemler ortaya atılmış bu yöntemlerin azınlık oyunu yönteminden daha iyi sonuçlar verdiği anlatılmıştır.

Bu tez çalışmasında diğer kaynaklara benzer olarak problemi modellemek için Azınlık Oyunu Modeli kullanılmış ve bu problemin çözümünde Azınlık Oyunu Yöntemi ile

birlikte Q-öğrenme, Kazan Kal Kaybet Değiştir Stratejisi, Roth Erev Öğrenme ve Öğrenen Durum Makinaları yöntemleri incelenerek karşılaştırılmıştır.

2.2 Kaynak Özetleri

MEC sistemlerini ve kaynak tahsis problemini araştıran birçok kaynak mevcuttur. Bazı kaynaklar yalnızca MEC konusunu ya da kaynak tahsis problemini ele alırken MEC sistemlerinde kaynak tahsis problemini inceleyen kaynaklar da mevcuttur.

MEC konusunu inceleyen kaynaklardan (Tran vd. 2017a, Tran vd. 2017b, Taleb vd. 2017) kenar bilişim konusunda genel bakış sunmakta, kenar bilişimin sorunlarını ve kenar bilişimden faydalanılabilecek alanları anlatmaktadır.

Kaynak tahsis probleminin çözümünde pek çok kaynak ilgili problemi Azınlık Oyunu şeklinde modellemektedir. (Savit vd. 1999, Sasidevan 2016) makalelerinde problemin çözümünde kullanılan Azınlık Oyunu Yöntemi, oyuncu sayısı ve hafıza miktarı açısından incelenmiş oyuncu sayısına göre en uygun hafıza miktarları bulunmuştur.

Whitehead tarafından yapılan çalışmada (Whitehead 2008) Azınlık Oyunu Modelinin Roth Erev yöntemiyle çözümü için çalışma yapılmış ve modelin Roth Erev yöntemiyle çözülebileceği sonucuna varılmıştır.

Catteeuw ve Manderick tarafından yapılan çalışmada (Catteeuw ve Manderick 2011) Azınlık Oyununun Modelinin çözümünde Q-öğrenme, Öğrenen Durum Makineleri ve Roth-Erev Öğrenme yöntemleri kullanılmış, yöntemler volatilité, ortalama refah ve adalet bakımından incelenmiştir.

Problemin çözümünde Reents ve Sasidevan tarafından yapılan çalışmalarda stokastik bir yaklaşım sergilenmiştir (Reents vd. 2001, Sasidevan, V. ve Dhar, D. 2014). Bu kaynaklarda her turda kaybeden oyuncular belirli bir olasılıkla karar değiştirerek toplam faydanın artırılması hedeflenmektedir.

MEC sistemlerinde işlem yükü devri de kaynaklarda sıkça karşılaşılan alanlardandır. Örneğin Deng ve arkadaşları tarafından çok hücreli yarı statik ortamlı bir sistem tasarlanmış, işlem yükü devri problemini dinamik sıralı oyun olarak kurgulamışlardır (Deng vd. 2016). Zheng ve arkadaşları tarafından yapılan çalışmada (Zheng vd. 2016) problem stokastik oyun, Zheng ve arkadaşları tarafından yapılan çalışmada (Zheng vd. 2017) Stackelberg oyun, Tekin ve arkadaşları ile Xu ve arkadaşları tarafından yapılan çalışmalarda (Tekin ve van der Schaar 2014, Xu vd. 2017a) Markov Karar Süreci, Aliyu ve arkadaşları tarafından yapılan çalışmada (Aliyu vd. 2017) işbirlikçi oyun, Xu tarafından yapılan çalışmada (Xu 2017b) mezat oyunu, He tarafından yapılan çalışmada (He 2017) derin Q-öğrenme (Deep Q-Learning) olarak kurgulanmıştır.

Ranadheera ve arkadaşları tarafından yapılan çalışmada 21 kullanıcı bir benzetim çalışması yapılmış çalışmada bu tez çalışmasına benzer şekilde Q-öğrenme, Azınlık Oyunu, Kazan Kal Kaybet Değiştir Stratejisi, Öğrenen Durum Makinaları ve Rastgele Karar Sistemi için değişen hafıza miktarına göre volatilité incelenmiştir (Ranadheera vd. 2017).

Xu ve arkadaşları (Xu vd. 2019) ile Jain ve Jat (Jain ve jat 2020) yaptıkları çalışmalarda Kenar Bilişim konusunu süre açısından incelemiş işlem yükü devrinin daha kısa sürede tamamlanması için yöntemler önerilmiştir.

Lin ve arkadaşları (Lin vd. 2014), Zhang ve arkadaşları (Zhang vd. 2016), Mao ve arkadaşları (Mao vd. 2016) ve Xu ve arkadaşları (Xu vd. 2017a) tarafından yapılan çalışmalarda Kenar Bilişim konusunu enerji verimliliği açısından incelenmiştir.

Kaynak araştırması kapsamında Catteeuw ve Manderick tarafından yapılan çalışma dışında Ortalama Refah ve Adalet kavramlarını inceleyen bir kaynağa rastlanmamıştır.

Yapılan kaynak araştırmasında bu çalışmada kullanılan yöntemleri ve karşılaştırma kriterlerini birlikte kullanarak bu derinlikte inceleyen bir kaynağa rastlanmamıştır.

3. KULLANILAN YÖNTEMLER VE BENZETİM ÇALIŞMASI

Bu tez çalışmasında kenar bilişim sistemlerinde yük aktarımı karar problemi bir kaynak tahsis problemi olarak modellenmiştir. Bu bölümde öncelikle benzetim ortamına ilişkin bilgiler verilmiştir. İkinci olarak yöntemlerin karşılaştırılması için kullanılan kriterler anlatılmıştır. Son olarak da tez kapsamında kullanılan yöntemler anlatılmıştır.

3.1 Sistem Modeli

Benzetim çalışması için tasarlanan sistem modelinde çok sayıda oyuncu ile çok sayıda tekrar yapılmaktadır. Böylece yöntemleri kullanan oyuncuların zaman içerisinde daha iyi kararlar verip vermedikleri kontrol edilmektedir. Sistem tek bir kenar bilişim sunucusu ve $2k + 1$ kullanıcıdan oluşmaktadır, yani oyuncu sayısı tektir. Kullanıcılar özdeşdir ve aralarında herhangi bir öncelik sıralaması bulunmamaktadır. Ayrıca oyuncuların her bir turda oluşturdukları işlem yükü miktarları da birbirine eşittir. Sunucunun kapasitesi k kullanıcının işlem talebini karşılayabilecek kadardır. Kaynak tahsis problemi başlığı altında anlatıldığı üzere oyuncuların amacı azınlık olan grupta kalmaktır; yani oyuncuların çoğu işlem yükünü sunucuya devrediyorsa ise işlem yükünü devretmeyen oyuncular, çoğunluk işlem yükünü devretmiyorsa işlem yükünü devreden oyuncular kazanmaktadır. Oyuncu sayısının tek olması nedeniyle eşitlik gibi bir durum söz konusu olmamaktadır. Tüm kullanıcılar sahip oldukları işlem yüklerini ya tamamen kendileri çözmekte ya da tümüyle devretmektedir, yani ikili işlem devri yapmaktadır.

Çalışmada her bir yöntem ele alınırken tüm oyuncuların aynı yöntemi kullandığı varsayılmıştır. Bu açıdan sistem homojendir.

Oyun on bin tekrar ile gerçekleştirilmektedir. Her bir tekrar için tur, iterasyon ve raund tabirleri kullanılabilmeyle birlikte çoğunlukla tur kelimesi kullanılmıştır. Benzer şekilde oyuna katılım sağlayan karar vericilere oyun teorisinden yola çıkarak oyuncu ve ajan, MEC sistemleri göz önünde bulundurularak kullanıcı denilebilmektedir. Bu tez

alışmasında oęunlukla kullanıcı kelimesi kullanılmıştır. İşlem yükü devri yapan oyuncu sayısını ifade etmek için ise katılım sayısı (attendance) ibaresi kullanılmıştır.

On bin tur sonunda bir oyun tamamlanmaktadır. Oyun sonucunda çıkan verilerin doğruluęunu artırmak adına her bir oyun on kez tekrarlanarak çıkan verilerin ortalaması alınmıştır.

Yapılan alışmada sunucu kapasitesi k iken oyuncu sayısı $2k+1$ olarak kabul edilmiştir. Bu durumda her turda hangi kararı veren gruptaki sayı daha az ise o gruptaki kullanıcılar doğru kararı vermiş oęunlukta kalan oyuncular ise yanlış karar vermiş olarak değerdendirilmektedir. Gerekte ise oyuncu sayısı k iken sunucu kapasitesi $2k+1$ olmak durumunda değildir. Tez kapsamında sunucu kapasitesi k olmak üzere işlem yükü devri yapan oyuncuların sayısı k 'yı geçmemişse işlem yükü devri yapan oyuncular, k 'yı geçmiş ise işlem yükü devri yapmayan oyuncular azınlıkta kalan olarak nitelendirilmektedir.

3.2 Karşılaştırma Kriterleri

Kuramsal temeller kısmında bahsedildięi üzere kaynak tahsis probleminde az miktarda kaynaęın ok sayıda oyuncu arasında paylaştırılması söz konusudur. Tez kapsamında varılmaya alışılan nokta, hangi yöntem kullanıldığında söz konusu kaynak tahsisinin en iyi şekilde gerekleşmiş olacağının bulunmasıdır. Bu çerevede yöntemler arasında karşılaştırma yapılabilmesi için öncelikle hangi kriter gerektirmektedir.

Tasarlanan benzetim sisteminde en temelde her bir kullanıcının her bir turda verdiği karara ait veriler tutulabilmektedir. Buradan yola ıkarak her bir turda kaç oyuncunun işlem yükü devrine karar verdiği, turu hangi grubun kazandığı, her bir oyuncunun kaç kez işlem yükü devrine karar verdiği ve kaç kez doğru karar verdiği gibi pek ok veri ayrı değışkenler olarak tutulmaktadır. Buradan yola ıkarak yöntemlerin karşılaştırılması için farklı kriterler kullanılabilmektedir. Bu bölümde tez kapsamında yöntemlerin karşılaştırılmasında kullanılan kriterler incelenmiştir.

3.2.1 Volatilite

Yapılan benzetim çalışmasında bir oyun sonucunda elde edilen en önemli verilerden biri her bir turda kaç oyuncunun işlem yükü devrine karar verdiği diğer bir deyişle katılım bilgisidir (attendance). Şekil 2.1’de El Farol Bar Problemi (Arthur 1994) için yazarın elde ettiği veriler ışığında oluşturulmuş bir grafik mevcuttur. Grafik incelendiğinde katılım sayısının barın kapasitesi olan 60’ın etrafında salındığı söylenebilir. Katılım sayılarının bu kadar geniş aralıkta salınması barın çok verimsiz kullanılmasına neden olmaktadır.

Yukarıdaki örnekten yola çıkarak, bir oyun boyunca her bir turda kullanıcıların katılım miktarları, diğer bir deyişle sunucuyu kullanma miktarları sunucunun verimli kullanılıp kullanılmadığının bir göstergesidir. Her bir turda sunucu kapasitesi kadar kullanıcının işlem yükünü devretmesi sunucunun en verimli şekilde kullanılması anlamına gelmektedir. Böyle bir durumda katılım grafiği düz bir çizgiden oluşacaktır. Diğer taraftan kendi çıkarlarını düşünen oyuncuların oluşan bir sistemde böyle bir durum söz konusu olamayacaktır. Çünkü oyunun bir turunda yanlış karar veren kullanıcılar bir sonraki turda doğru karar vererek azınlıkta olan gruba girmeye çalışacaklar ve bu da oyundaki dengenin bozulmasına neden olacaktır. En kötü durum ise tüm oyuncuların aynı kararı vererek bir turda hepsinin katılım sağlaması, diğer turda hiçbirinin katılım sağlamaması ve oyunun bu şekilde sürüp gitmesi durumudur. Böyle bir durumda katılım grafiği sıfır ile oyuncu sayısı arasında sürekli zikzaklar çizecektir.

Sunucunun verimi açısından en iyi ve en kötü senaryo incelendiğinde katılım grafiğine bakarak kullanıcıların katılım sayısının dar bir alana sıkışmasıyla verimin arttığı, geniş bir alana yayılmasıyla verimin azaldığını söylemek mümkün olur. Bu da katılım sayısı grafiğinin standart sapmasının küçülmesiyle verimin arttığı, büyümesiyle verimin azaldığı sonucunu çıkarmaktadır. Oyun boyunca katılım sayısının varyansının oyuncu sayısına bölünmesiyle volatilite elde edilir.

Kaynak tahsis problemleri incelenirken kaynaklarda en sık kullanılan kriter volatilitedir. Catteeuw ve Manderick ise denklem 3.1’deki gibi göstermiştir (Catteeuw ve Manderick 2011).

$$V = \frac{1}{TN} \sum_{t=t_0}^T (\#_1(t) - N/2)^2 \quad (3.1)$$

Burada $\#_1(t)$; t turunda işlem yükü devrine karar veren oyuncu sayısını, T ise oyundaki toplam tur sayısını ve N oyuncu sayısını ifade etmektedir. Oyuncu sayısı tek olduğundan $2k+1$ şeklinde de ifade edilebilir.

Denklem 3.1'den yola çıkıldığında, ideal durumda işlem yükü devri yapan oyuncu sayısı k olacaktır. Burada volatilité;

$$V = \frac{1}{TN} \sum_{t=t_0}^T (k - ((2k + 1)/2))^2 = \frac{1}{TN} \sum_{t=t_0}^T \left(\frac{1}{2}\right)^2 = \frac{1}{TN} (0.25)T = \frac{0.25}{N}$$

olacaktır.

En kötü durumda ise katılım sağlayan kullanıcı sayısı tek numaralı turlarda N çift numaralı turlarda 0 olacağından;

$$V = \frac{1}{TN} \sum_{t=1,3,5,\dots}^T (N - (N/2))^2 + \frac{1}{TN} \sum_{t=2,4,6,\dots}^T (0 - (N/2))^2 = \frac{1}{TN} \sum_{t=t_0}^T (N/2)^2 = \frac{1}{TN} \left(\frac{N^2}{4}\right) T = \frac{N}{4} = 0.25N$$

olacaktır.

Bu durumda 2001 oyunculu bir sistemde volatilitenin en iyi değeri 0,000125 olur iken en kötü değeri 500,25 olabilmektedir. Diğer taraftan düzgün dağılımlı rastgele olarak karar veren sistemde volatilitenin 0,25 civarında olması beklenmektedir. Rastgele karar verme mekanizmasının uygulanması oldukça kolay olduğundan, bir yöntemin kullanılmaya değer olması için parametrelere uygun değerler verildiğinde volatilitenin 0.25'ten düşük olması beklenir (Catteeuw ve Manderick 2011). Tez çalışmasının dördüncü bölümünde burada bahsedilen uygun parametrelerin neler olduğu belirlenmiştir.

3.2.2 Ortalama refah

Catteeuw ve Manderick tarafından yapılan çalışmada (2011) volatiliteye ek olarak refah (welfare-fitness) kavramı ve buna bağlı olarak da ortalama refah kavramı tanımlanmıştır. Bir oyuncunun refahı (w_i) oyuncunun toplam doğru karar sayısından yanlış karar sayısının çıkarılması ile ortaya çıkmaktadır. Ortalama refah (W) ise tüm oyuncuların refah miktarlarının ortalamasının alınmasıyla bulunmaktadır. Sunucu kapasitesinin k olduğu durumda her bir turda $2k+1$ oyuncu içerisinde en fazla k oyuncu doğru karar verebileceği için ortalama refahın negatif çıkacağı aşîkardır.

Bir turda ideal durumda k kadar oyuncu $+1$ puan geriye kalan $k+1$ oyuncu ise -1 puan alacaktır. Bir turdaki toplam puan -1 olacaktır. Ortalama refah ise $-1/(2k+1)$ olarak hesaplanacaktır.

En kötü durumda ise tüm oyuncular her turda aynı kararı vererek yanlış karar vermiş olacaklardır. Bu durumda tüm oyuncular negatif puan alacağından toplam refah $-1(2k+1)$ olacak, ortalama refah ise $-1(2k+1)/(2k+1)=-1$ olarak hesaplanacaktır.

En iyi durum ve en kötü durum göz önünde bulundurulduğunda ortalama refah ne kadar yüksek ise kullanılan yöntemin de o kadar başarılı olduğu söylenebilecektir.

3.2.3 Adalet

Catteeuw ve Manderick tarafından yapılan çalışmada ayrıca Adalet (Fairness- F) kavramı tanımlanmıştır (Catteeuw ve Manderick 2011). Adalet için hesaplama, Denklem 3.2’de belirtildiği şekilde yapılmaktadır.

$$F = \sqrt{\frac{N}{\sum_{i=1}^N (w_i - W)^2}} \quad (3.2)$$

Burada w_i i numaralı oyuncunun refahını, W ortalama refahı N ise oyuncu sayısını göstermektedir. Oyuncuların bireysel refahları ortalama refahtan farklı olduğunda denklemin payda kısmı büyüyecek böylece adalet miktarı küçülecektir. Bu açıdan değerlendirildiğinde adalet miktarının yüksek olması oyuncular arasında refah miktarının benzer olduğu, düşük olması ise oyuncuların refah miktarları arasında büyük farklılıklar olduğu sonucunu çıkarmaktadır.

En iyi durumda tüm oyuncuların refah miktarı eşit olacağından ortalama refah da her bir oyuncunun refah miktarına eşit olacaktır. Adalet denkleminde kök içerisindeki değerin paydası sıfır çıkacağından adalet miktarı sonsuza yakınsayacaktır.

En kötü durumda ise k oyuncu sürekli kazanırken $k+1$ oyuncu sürekli kaybedecektir. Böylece oyuncuların ortalama refah değerleri bireysel refah değerlerinden çok farklı olacaktır. Adalet denkleminde kök içerisindeki değerin paydası yüksek çıkacağından adalet miktarı oldukça küçülecektir.

3.2.4 Bir turun ortalama tamamlanma süresi

Özellikle gecikmeye duyarlı uygulamalarda gecikmenin azaltılması MEC sistemlerinin kullanılmasının önemli amaçlarından biridir. Bu nedenle MEC sistemlerinde sürenin incelenmesi önemli bir çalışmadır.

Tez çalışması kapsamında bir oyun içerisinde bir turun ne kadar sürede tamamlandığına dair inceleme yapılmıştır. Yapılan incelemede sunucunun kendisine devredilen işlem yüklerinin tamamını işlemci paylaşımı yöntemiyle işlediği yani aynı anda başlayıp aynı anda tamamladığı varsayılmaktadır. Bu durumda sunucuya ne kadar fazla işlem yükü devredilirse sunucunun tüm işlemleri tamamlaması o denli uzun zaman alacaktır. Her kullanıcı bu süreye göre başarılı ya da başarısız olduğuna karar vererek sonraki turlarda buna göre işlem yapacaktır.

Oluşturulan sistemde kullanıcı cihazlarının tek çekirdekli, f_m işlemci frekansına sahip ve birbirine eşdeğer olduğu varsayılmaktadır. Kenar bilişim sunucusunun işlemcisinin ise f_s frekanslı ve n_c çekirdek sayısına sahip olduğu varsayılmaktadır. Ayrıca (Kumar ve Lu 2010)'de belirtildiği gibi fazladan hafıza nedeniyle s katsayısı gelmektedir.

Bir bit başına gerekli olan makine çevrimi sayısı X ile gösterilmiştir. Bu değer işlemin türüne göre değişmekte olup (Miettinen ve Nurminen 2010)'de örnek bir tablo bulunmaktadır. L talep edilen işlem yükünün bit cinsinden miktarı olmak üzere bu yükün işlenmesi için gerekli olan makine çevrimi $L \times X$ kadar olacaktır. Ayrıca kenar bilişim sunucusuna gelen taleplerin eş zamanlı geldiği ve işlemci paylaşımı kuyruklama yöntemiyle aynı anda işlendiği varsayılmaktadır (Kleinrock 1967).

Bir kullanıcı cihazının oluşan bir işlem talebini kendi işlemcisinde gerçekleştirmesi için geçen süre T_l denklem (3.3)'de gösterilmiştir.

$$T_l = \frac{XL}{f_m} \quad (3.3)$$

Bulut bilişimde süre hesaplamasına ilişkin denklemler Melendez ve McGarry (2017) tarafından yapılan çalışmada verilmektedir. MEC sistemini hiç ekstra aktarım noktası olmayan bir bulut bilişim sistemi olarak düşünürsek süre hesaplamasını bu hesaplama yöntemiyle gerçekleştirebiliriz. Buna göre bulut bilişimde yük aktarımı için kullanılan toplam süre; işlemi gerçekleştirme, kuyrukta bekleme, iletim ve yayılım sürelerinin toplamına eşittir. Söz konusu hesaplama denklem (3.4)'de gösterilmiştir.

$$\psi = A + B + \frac{L\alpha}{r} + \frac{l}{\frac{2}{3}c} \quad (3.4)$$

Burada ψ toplam süreyi, A işlem yükünün sunucuda işlenme süresini, B kuyrukta bekleme süresini, L paket boyutunu (Bit), α kanal kodlama oranını, r iletim hızını (bit/sn), l kullanıcı ile sunucu arasındaki mesafeyi ve c ışık hızını göstermektedir. Yayılım hızı $(\frac{l}{\frac{2}{3}c})$ ihmal edilebilecek kadar küçüktür. Kenar bilişimde işlemci paylaşımı kuyruklama

yöntemi kullandığımız için kuyruk bekleme süresini (B) işlem süresi (A) içerisinde değerlendirebiliriz. Bu durumda yalnızca iletim süresi ($\frac{L\alpha}{r}$) ile işlem süresinin (A) hesaba katılması kenar bilişim açısından yeterli olacaktır.

Sunucuda gerçekleşen işlemi şu şekilde özetlemek mümkündür: Sunucu her bir turun başında kendisine yapılan işlem taleplerini toplamakta, topladığı tüm yükü aynı anda işlemekte ve tüm işlemler tamamlandığında sonuçları kullanıcı cihazlara iletmektedir. Yani işlem yükünü devretmek isteyen kullanıcılar işlem yüklerini aynı anda iletmekte ve aynı anda geri almaktadır.

Sonuç olarak işlem yükü devri yapan bir kullanıcı için iki tür süre etkili olacaktır. İlki kullanıcının işlem yükünü iletim süresi, denklem 3.5'te gösterilmektedir. İkinci olarak sunucunun işlem yükünü işleme süresi, denklem 3.6'te gösterilmektedir.

$$T_{tr} = \frac{L\alpha}{r} \quad (3.5)$$

$$T_i = \frac{\sum_{i=1}^N L_i X_i}{sn_c f_s} \quad (3.6)$$

Burada N sistemde işlem yükü devri yapan toplam kullanıcı sayısını göstermektedir. İşlem yükü devri sırasında toplam süre denklem 3.7'te gösterilmektedir.

$$T_o = T_{tr} + T_i = \frac{L\alpha}{r} + \frac{\sum_{i=1}^N L_i X_i}{sn_c f_s} \quad (3.7)$$

Eğer tek bir kullanıcı işlem yükü devrine karar verirse sunucuda bu işlem çok kısa zamanda tamamlanacak, yükü kendi işlemeye karar veren oyuncular karşısında avantaj elde etmiş olacaktır. Eğer çoğunluk işlem yükünü devretmeye karar verirse bu kez tüm işlemlerin sunucuda işlenmesi uzun süreceğinden işlem yükünü kendisi işleyen kullanıcı daha kısa sürede tamamlamış olacaktır. Bir tur sonunda $T_i > T_o$ olursa işlem yükünü

devreden oyuncular, $T_l < T_o$ olursa işlem yükünü devretmeyen oyuncular başarılı olarak sayılacaklardır.

3.2.5 Bir turda harcanan ortalama enerji

İşlem yükü devrinin başka bir kullanım amacı da enerji tasarrufu sağlamaktır. Özellikle Nesnelerin interneti gibi kısıtlı enerji kaynağına sahip cihazlar için işlem yükünü en az enerji ile tamamlamak önem arz etmektedir. Tez çalışması kapsamında yer alan yöntemleri kullanan kullanıcıların bir turda harcadıkları enerji miktarları karşılaştırılmaktadır.

Enerji açısından yapılan inceleme süre açısından yapılan incelemenin devamı niteliğindedir. Süre için yapılan incelemede bulunan her bir süre için, o süre zarfında cihazın harcadığı güç miktarları ile çarpılarak cihazın harcadığı enerji miktarı bulunmuştur.

Kullanıcı cihazı için üç farklı enerji harcamasından bahsetmek mümkündür. İlki cihazın işlem yükünü devretmeyip kendisinin yapması durumunda harcanan işlem enerjisi, ikinci olarak işlem yükü devri sırasında cihazın işlem yükünü aktarırken harcadığı enerji ve son olarak da işlem yükü devri sırasında cihazın işlem yükünü devredip sonucu beklerken harcadığı enerji olarak sıralanabilir.

Enerji ve süre arasındaki ilişki denklem 3.8 ve 3.9'da verilmektedir (Kumar ve Lu 2010).

$$E_l = P_l \times T_l \quad (3.8)$$

$$E_o = (P_{tr} \times T_{tr}) + (P_l \times T_l) \quad (3.9)$$

Burada E_l kullanıcının işlemin kullanıcı cihazında gerçekleştirilmesi durumunda, E_o ise işlem yükü devri yapılması durumunda kullanıcı cihazında harcanan enerjiyi belirtmektedir. T_l denklem 3.3'te T_{tr} denklem 3.5'te ve T_i denklem 3.6'da verilmişti.

Karthik Kumar ve Yung-Hsiang Lu tarafından yapılan çalışmada HP iPAQ PDA için enerji değerlerinin; $P_l \sim 0,9$, $P_{tr} \sim 1,3$, $P_r \sim 0,3$ olduğu bildirilmiştir (Kumar ve Lu 2010). Tez çalışması kapsamındaki enerji hesaplamalarında bahse konu çalışmadan yararlanılarak enerji değerleri hesaplanmıştır.

3.3 Kullanılan Yöntemler

Bu bölümde tez çalışması kapsamında yer alan yöntemler tanıtılmıştır. Kullanılan yöntemlere ek olarak analiz çalışmalarında yöntemlerin başarımını karşılaştırabilmek amacıyla iki sistem tasarlanmıştır. İlki merkezi olarak karar verdiği varsayılan bir sistemdir. Bu sistem belirli bir sıraya göre kullanıcılara işlem yükü devri konusunda bildirimde bulunmaktadır. Merkezi bir sistemin gerçekte var olamayacağı ikinci bölümde anlatılmıştı. İkinci sistem ise tüm oyuncuların herhangi bir etkene bağlı kalmadan rastgele karar verdiği varsayılan bir sistemdir. Bu sistemde oyuncular MEC sunucusunun durumuna bakmaksızın işlem yükü devri yapmaya ya da yapmamaya rastgele karar vermektedir.

3.3.1 Azınlık Oyunu yöntemi

El Farol Bar Problemi ilk kez Challet ve Zhang tarafından formülize edilmiştir (Challet ve Zhang 1997). Problemin çözümünde ilk olarak ortaya atılan yöntem olduğu için çoğunlukla Azınlık Oyunu adıyla bilinir. Bu yöntemde oyuncular geçmiş M oyunun kazanan aksiyonunu hafızada tutarlar. Bu durumda 2^M farklı geçmiş ihtimali oluşacaktır. Her oyuncu bir strateji tablosu oluşturur. Strateji tablosu stratejilerden oluşmaktadır. Stratejiler ise her bir tarih değerine karşılık gelen aksiyonlardan oluşmaktadır. Bu durumda strateji tablosunda 2^{2^M} farklı seçenek oluşacaktır.

Çizelge 3.1 Azınlık Oyununda 2 bitlik geçmiş hafızasına sahip oyuncunun tam strateji tablosu

Geçmiş Değeri	S 1	S 2	S 3	S 4	S 5	S 6	S 7	S 8	S 9	S 10	S 11	S 12	S 13	S 14	S 15	S 16
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Çizelge 3.1’de iki bitlik geçmiş değerine göre hazırlanan strateji tablosunu göstermektedir. Ancak kolayca anlaşılacağı üzere M sayısı arttıkça seçenek miktarı çok hızlı şekilde artacaktır. Bu da işlemin tamamlanma süresini çok artıracaktır. Bu nedenle strateji tablosunun rastgele seçilen stratejilerden oluşturulması daha uygun olur. Buna indirgenmiş Strateji Tablosu denir.

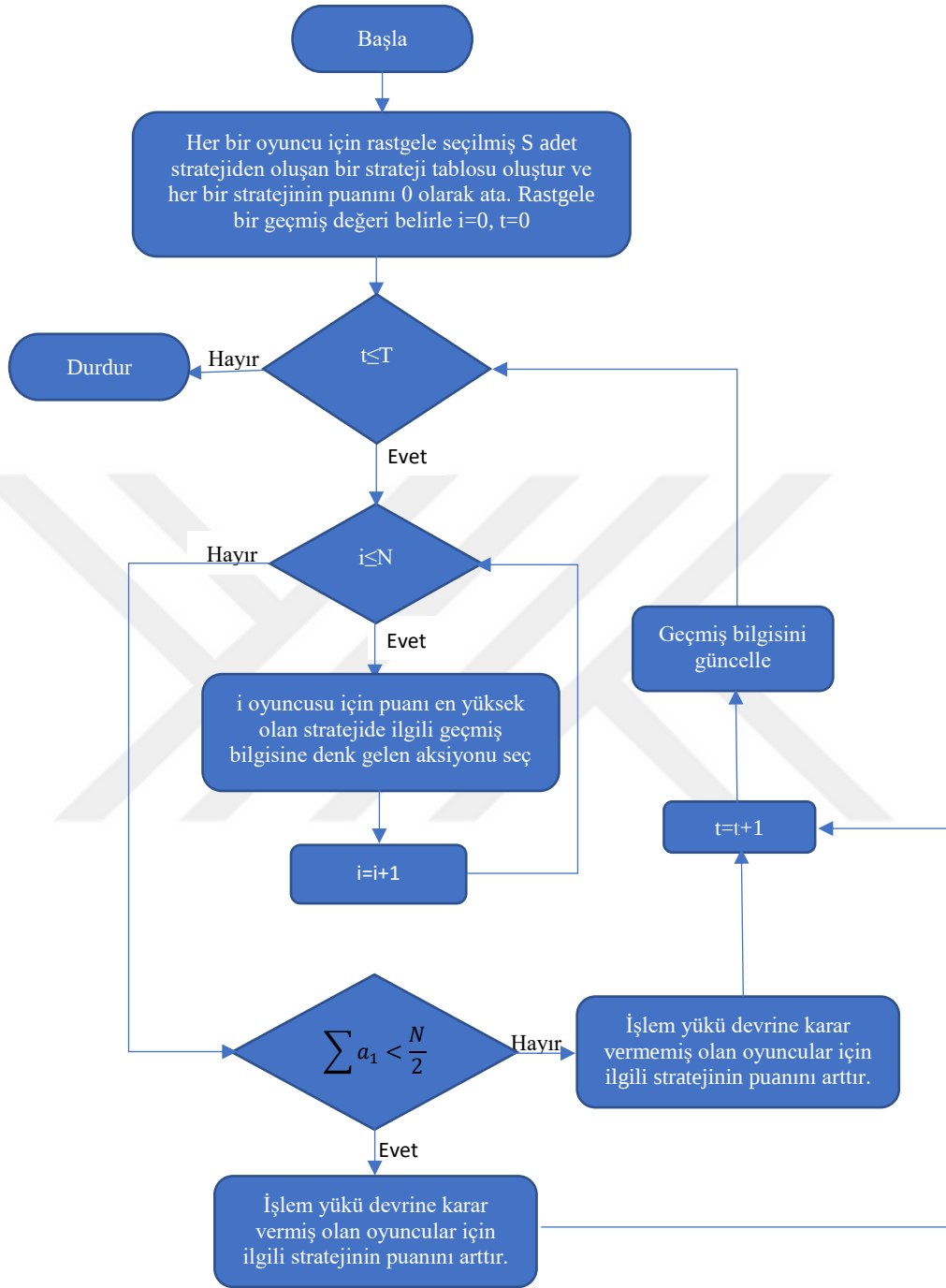
Çizelge 3.2 Azınlık Oyununda 2 bitlik geçmiş hafızasına sahip oyuncunun indirgenmiş strateji tablosu (Strateji Sayısı 3)

Geçmiş Değeri	S 1	S 7	S 12
00	0	0	1
01	0	1	0
10	0	1	1
11	0	0	1

Çizelge 3.2’de örnek bir indirgenmiş strateji tablosu yer almaktadır. Bu şekilde oluşturulan tablolarda oyun ilerledikçe kazanan stratejilerin puanları artırılır. Her bir turda en yüksek puana sahip stratejinin ilgili tarih değerine rastgelen aksiyon oyuncu tarafından seçilir.

Azınlık Oyunu yönteminin algoritma diyagramı şekil 3.1’de verilmektedir. Burada i oyuncu numarasını N oyuncu sayısını, t tur numarasını, T toplam tur sayısını a_1 işlem yükünü devretmeye karar veren oyuncu sayısını ifade etmektedir.





Şekil 3.1 Azınlık Oyunu Yöntemi Algoritma Diyagramı

3.3.2 Q-öğrenme yöntemi

Bu yöntem ilk kez 1989 yılında Watkins tarafından ortaya atılmış (Watkins 1989), 1992 yılında yine Watkins tarafından ispatı yapılmıştır (Watkins and Dayan 1992). Bu yöntemde her bir ajan her bir durum-aksiyon (s,a) ikilisi için bir Q değeri hesaplar. Her oyunda Q değeri daha yüksek olan aksiyon seçilir. Her oyunun sonunda eğer seçilen aksiyon başarılı ise oyuncuya ödül puanı verilir. Bu ödül puanı da hesaba katılarak Q değerleri güncellenir. Q değeri güncelleme denklemi 3.10'da verilmiştir.

$$Q_n(s, a) = \begin{cases} (1 - \alpha_n)Q_{n-1}(s, a) + \alpha_n[r_n + \gamma V_{n-1}(y_n)] & \text{eğer } a = a_n \\ Q_{n-1}(s, a) & \text{diğer durumda} \end{cases} \quad (3.10)$$

Denklem 3.10'de α öğrenme katsayısı γ faydalanma katsayısı r ise ödül değeridir.

Q-öğrenme yönteminin iki temel özelliği keşfetme (ϵ) ve faydalanmadır (γ). Keşfetme özelliği öğrenme sırasında algoritmanın belirli bir döngü içerisinde sıkışıp kalmasını engelleyerek varsa daha uygun sonuçları bulmasını sağlar. Bu özellik ϵ değeri ile belirlenir. Karar vermesi gereken bir oyuncu ϵ ihtimalle sonraki kararını rastgele olarak verir, $1 - \epsilon$ ihtimalle Q tablosuna bakarak Q değeri daha yüksek olan kararı verir. Böylece yeni durumlar keşfetme fırsatı elde etmiş olur.

Faydalanma özelliği ise oyuncunun en yüksek Q değerine doğru gitmesini sağlar. Bu özellik ile Q değeri güncellenirken bir sonraki durumda en yüksek puanlı değer faydalanma katsayısı ile çarpılarak Q değerine eklenir. Denklem 3.10'un faydalanma özelliğine ilişkin kısmı denklem 3.11'de verilmiştir.

$$V_{n-1}(y_n) = \max_b Q_{n-1}(y, b) \quad (3.11)$$

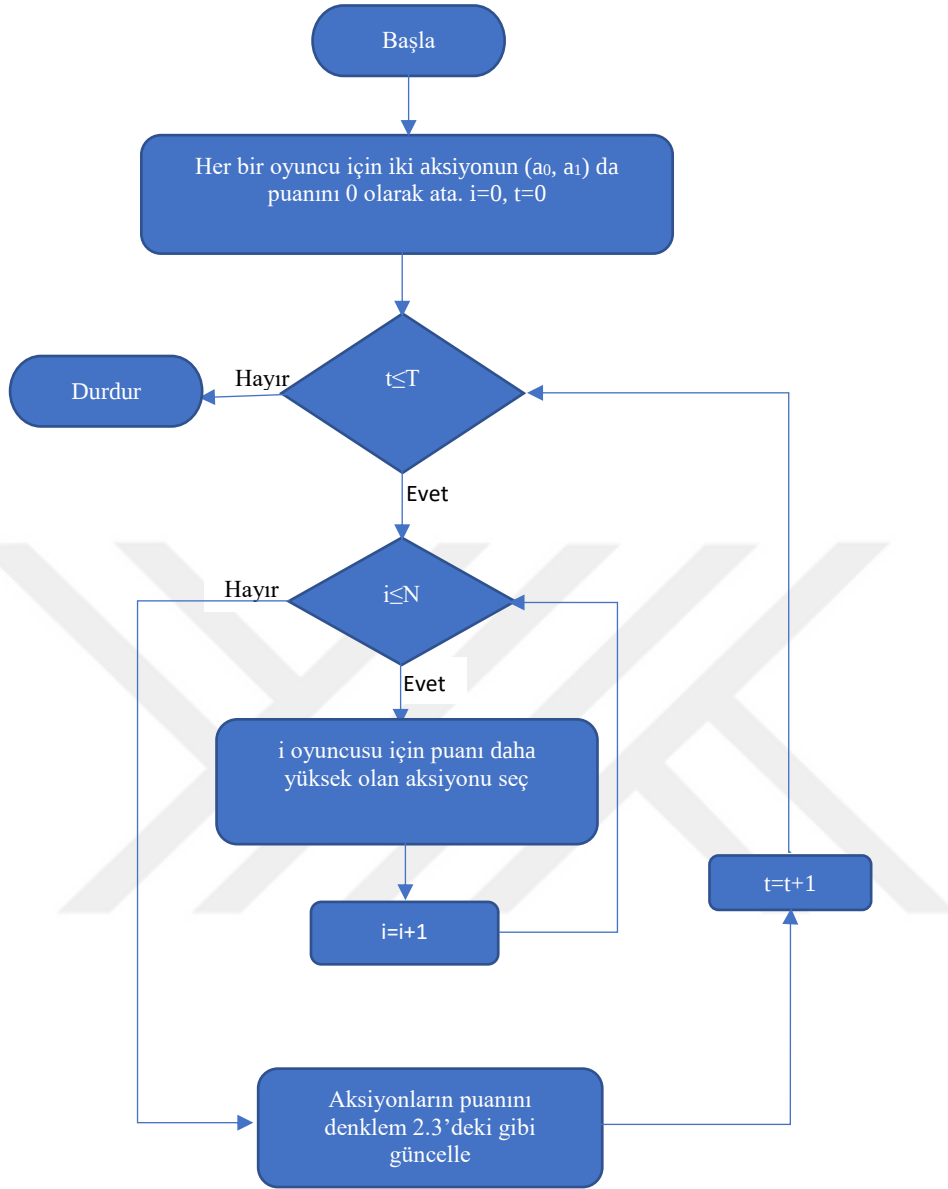
Burada b tüm aksiyonları $V_{n-1}(y_n)$ ifadesi bir aksiyon seçildikten sonra yeni oluşan durumdan bir sonraki adımda alınabilecek en yüksek puanı ifade etmektedir.

Diğer taraftan azınlık oyunu modelinde yalnızca iki aksiyon yani iki durum vardır. Bu nedenle faydalanma özelliğinin kullanılabileceği yeni durumlar olmayacaktır. Sonuç olarak bu özelliğin kullanılmasına gerek olmayacaktır.

Faydalanma özelliği çıkarılınca Q değeri güncelleme denklemi 3.12'deki gibi olur.

$$Q(s, a) = \begin{cases} Q_{n-1}(s, a) + \alpha_n[r_n - Q_{n-1}(s, a)] & \text{eğer } a = a_n \\ Q_{n-1}(s, a) & \text{diğer durumda} \end{cases} \quad (3.12)$$

Q-öğrenme yönteminin algoritma diyagramı şekil 3.2'de verilmektedir. Burada i oyuncu numarasını N oyuncu sayısını, t tur numarasını, T toplam tur sayısını ifade etmektedir.



Şekil 3.2 Q-Öğrenme Yöntemi Algoritma Diyagramı

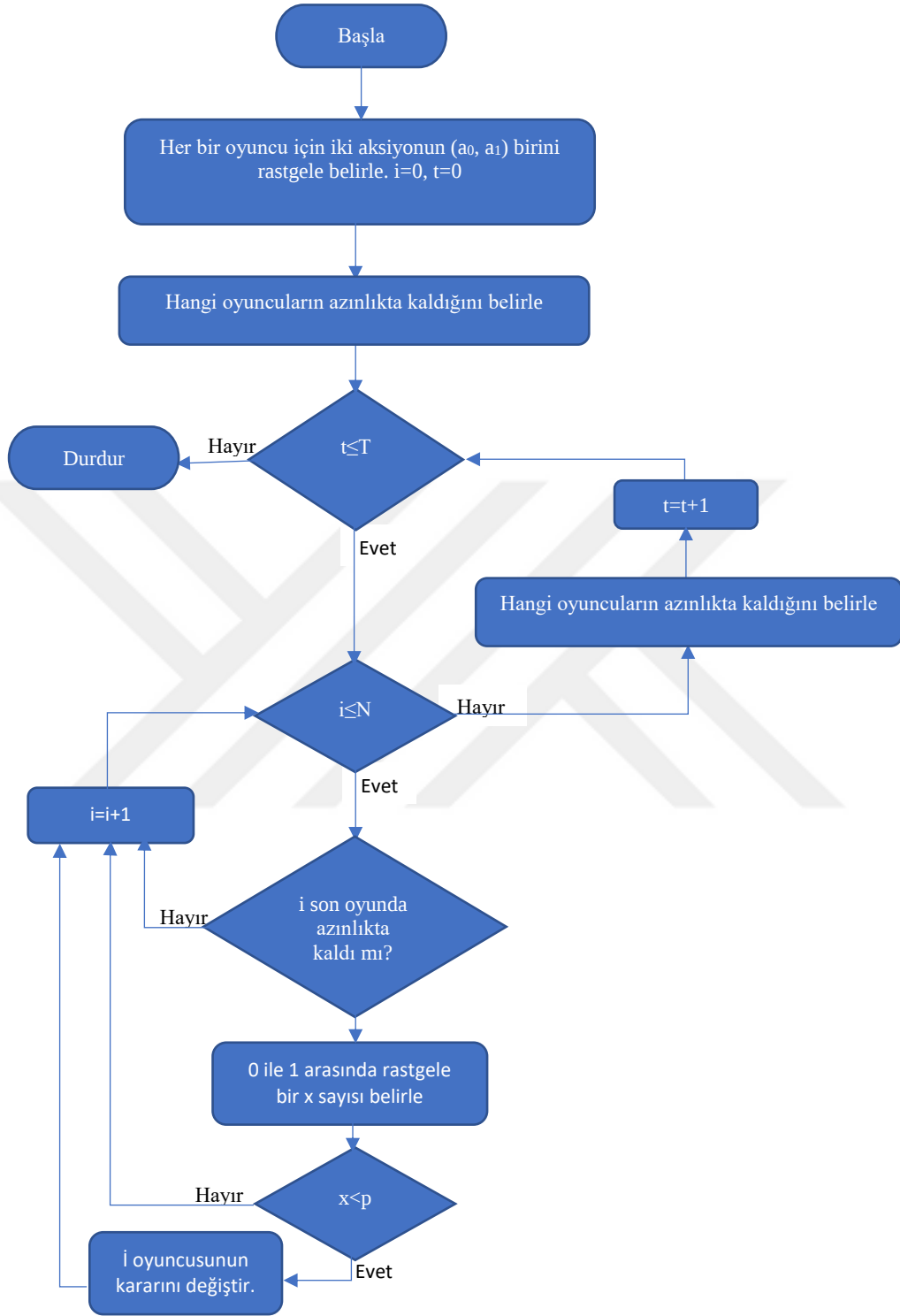
3.3.3 Kazan Kal Kaybet Değiştir stratejisi

Reents ve arkadaşları tarafından yapılan bir çalışmada Azınlık oyunu için stokastik strateji tabanlı oldukça sade bir öğrenme metodu geliştirilmiştir (Reents vd. 2001). Bu metoda göre her oyuncu i , $a_i(t)$ şeklinde bir aksiyon belirler. Eğer oyuncu bir oyunda

kazanırse bir sonraki oyunda yine aynı aksiyonu seçecektir $a_i(t + 1) = a_i(t)$. Eğer oyunu kaybederse sabit bir p ihtimalle diğer aksiyonu seçecek, $1 - p$ ihtimalle yine önceki aksiyonunu seçecektir. Bu durumda $P(a_i(t + 1) = -a_i(t)) = p$ olacaktır. Yapılan çalışmada p değeri yeterince küçük seçilirse sistem optimum noktaya ulaşacaktır.

Burada yakınsama p 'nin ve oyuncu sayısının (N) değerine bağlı olacaktır. x N 'den çok küçük bir sabit olmak üzere $p = x/(N/2)$ dersek oyuncu sayısı büyüdükçe her turda karar değiştiren oyuncu sayısı x 'e yakın bir değer olacaktır. Burada x büyük olursa her turda çok fazla oyuncu karar değiştireceği için ideal noktaya yakınsamayacaktır. Diğer taraftan x değeri gereğinden fazla küçük olursa yakınsama süresi çok uzayacaktır.

Kazan Kal Kaybet Değiştir Stratejisi yönteminin algoritma diyagramı Şekil 3.3'te verilmektedir. Burada i oyuncu numarasını N oyuncu sayısını, t tur numarasını, T toplam tur sayısını ifade etmektedir.



Şekil 3.3 Kazan Kal Kaybet Değiştir Stratejisi Algoritma Diyagramı

3.3.4 Roth-Erev öğrenme yöntemi

Bu yöntem ilk olarak 1995 yılında Alvin E. Roth ve Ido Erev tarafından önerilmiştir (Roth ve Erev 1995). Q-öğrenme'ye benzer şekilde bu yöntemde de oyuncular her bir aksiyon için ağırlık değerleri belirler. Ama bu yöntemde ağırlık değerleri güncellenirken geçmiş ağırlık değerlerinin indirgenmiş haliyle toplanır. Böylece oyun ilerledikçe geçmişteki değerlerin etkisi gittikçe azalır. Q değerleri güncellenme denklemi 3.13'da verilmiştir.

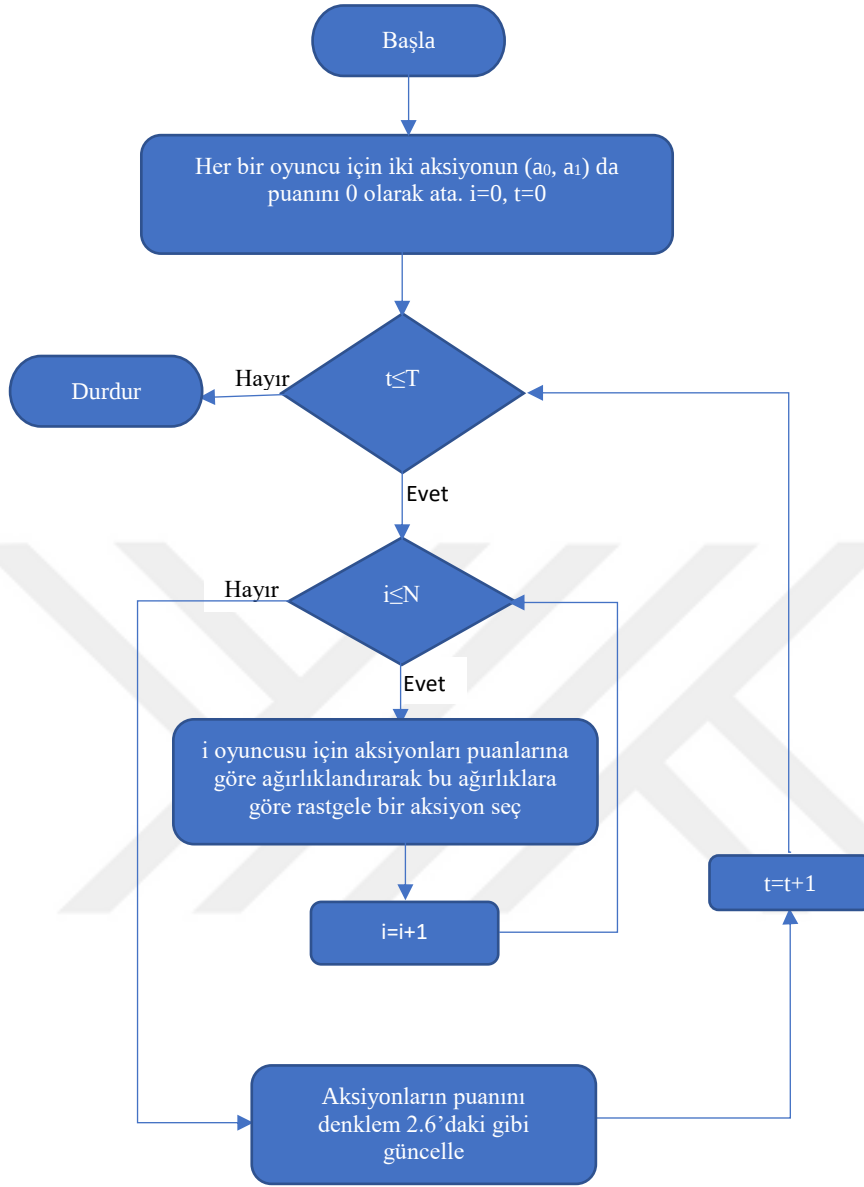
$$q_a(t) = \begin{cases} \lambda q_a(t-1) + u_{i,a}(t-1) & \text{eğer } a_{i,t} = a \\ \lambda q_a(t-1) & \text{diğer durumda} \end{cases} \quad (3.13)$$

Burada λ indirgeme faktörü, $u_{i,a}(t-1)$ değeri ise bir önceki oyunda elde edilen ödül miktarıdır.

Ayrıca Q-öğrenme yönteminde ağırlığı daha yüksek olan aksiyon seçilirken Roth-Erev öğrenme yönteminde farklı olarak aksiyon ağırlık değerleri ile orantılı olarak rastgele seçilir. Her bir aksiyonun seçilme ihtimali denklem 3.14'te olduğu şekilde hesaplanır.

$$p_{a(t)} = \frac{q_a}{\sum_a q_a} \quad (3.14)$$

Roth-Erev Öğrenme yönteminin algoritma diyagramı şekil 3.4'te verilmektedir. Burada i oyuncu numarasını N oyuncu sayısını, t tur numarasını, T toplam tur sayısını ifade etmektedir.



Şekil 3.4 Roth-Erev Öğrenme Stratejisi Algoritma Diyagramı

3.3.5 Öğrenen Durum Makinaları yöntemi

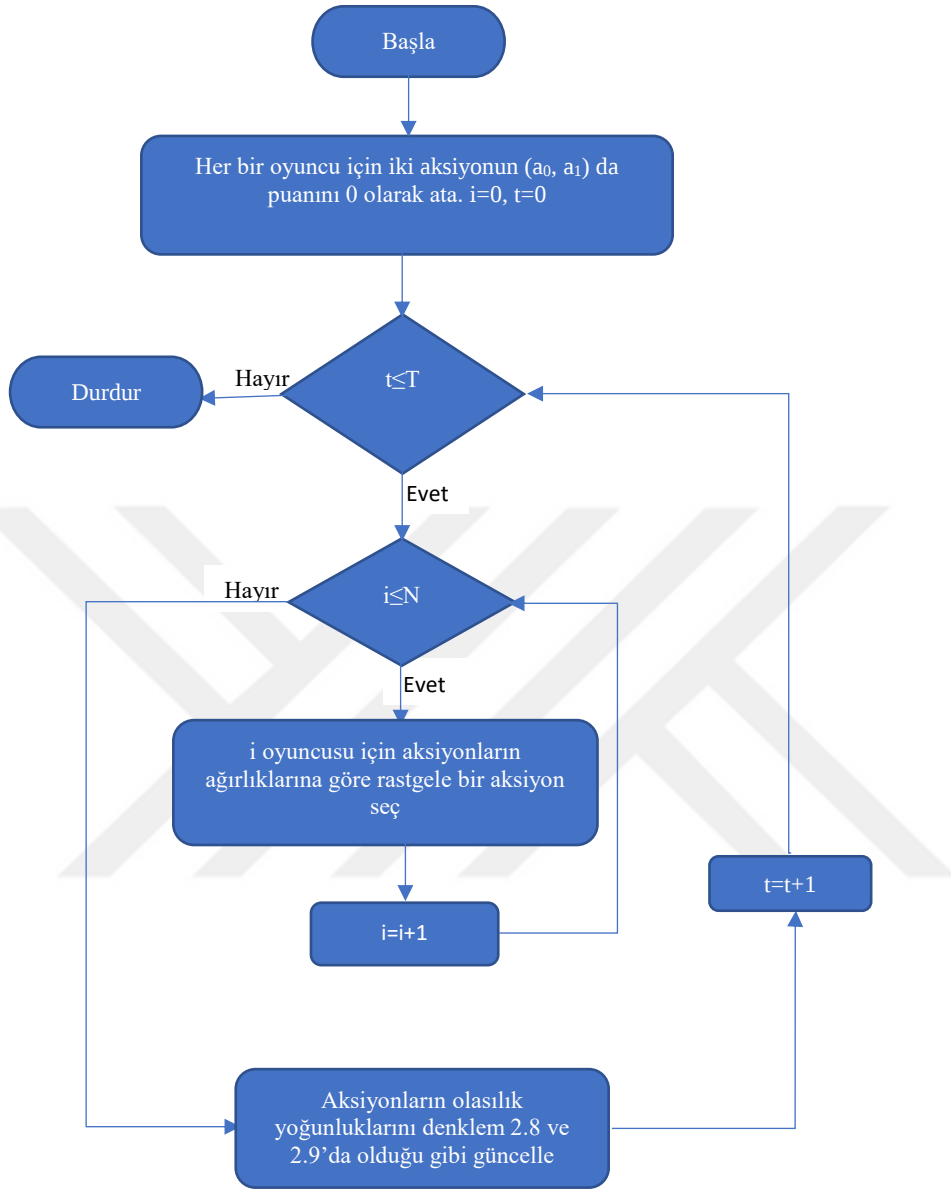
Bu yöntemde aksiyonların kazanma ihtimalini gösteren olasılık dağılımları manipüle edilerek en uygun strateji bulunmaya çalışılır (Cattueuw ve Manderick 2011). Her oyunda her oyuncu kendi olasılık dağılımlarına göre bir aksiyonu seçer ve kazanıp kazanmama durumuna göre aksiyonların olasılık dağılımlarını aşağıdaki denklemlere göre günceller.

$$p_a(t) = p_a(t-1) + \alpha r(1 - p_a(t-1)) - \beta(1-r)p_a(t-1) \quad (3.15)$$

$$p_j(t) = p_j(t-1) + \alpha r p_j(t-1) - \beta(1-r) \left(\frac{1}{n-1} - p_j(t-1) \right) \quad \forall j \neq a \quad (3.16)$$

Burada α ve β sırasıyla ödül ve ceza katsayılarıdır ve sıfır ile bir arasında değer alır. n aksiyon sayısı, r ise ödül miktarıdır. Ödül miktarı da yine sıfır ile bir arasında bir değer alır.

Öğrenen Durum Makinaları yönteminin algoritma diyagramı şekil 3.5'te verilmektedir. Burada i oyuncu numarasını N oyuncu sayısını, t tur numarasını, T toplam tur sayısını ifade etmektedir.



Şekil 3.5 Öğrenen Durum Makinaları Yöntemi Algoritma Diyagramı

3.3.6 Rastgele karar sistemi

Yöntemlerin başarımının gösterilebilmesi ve karşılaştırılabilmesi için literatürde yer alan yöntemlere ek olarak kaynağı kullanıp kullanmamaya rastgele karar veren kullanıcılardan oluşan bir sistem benzetim çalışmasına eklenmiştir.

Rastgele karar alan sistemde kullanıcılar geçmişte ne olmuş olursa olsun düzgün dağılımlı rastgele olarak karar verirler. Doğal olarak hafızaya ihtiyaçları bulunmamaktadır. Ayrıca düzgün dağılımlı rastgele karar verdikleri için sunucu kapasitesi ne olursa olsun ortalama olarak kullanıcı sayısının yarısı kadar kullanıcı işlem yükü devri için karar verecektir.

3.3.7 Merkezi karar sistemi

İkinci bölümde MEC sistemlerinde azınlık oyunu modelinin çözümünde merkezi bir yöntem kullanılamayacağından bahsedilmişti. Bunun nedeninin merkezi noktada tüm kullanıcılara ilişkin bilgilerin bulunmasının mümkün olmaması olduğu açıklanmıştı.

Gerçekte mümkün olmayacak bile olsa karşılaştırma açısından merkezi bir noktadan karar verdiği var sayılan bir sistem tasarlanarak benzetim çalışmasında kullanılmıştır.

Merkezi karar veren $2k + 1$ kullanıcı bulunan bir sistemde her turda k adet oyuncunun işlem yükü devrine izin verilmesi beklenir. İşlem yükü devri yapan oyuncu sayısı sürekli sabit olacağından standart sapma ve dolayısıyla volatilité sıfır olacaktır. Diğer taraftan adalet değerinin de yüksek çıkması için her oyunda farklı kullanıcılar işlem yükü devri kararı vermektedir.

4. ARAŞTIRMA BULGULARI

Tez kapsamında bu kısma kadar kuramsal temeller ve benzetim çalışması hakkında bilgi verilmiştir. Bu kısımda öncelikle kullanılan yöntemlerde yer alan parametreler mümkün olan en uygun şekilde belirlenecek ve belirlenen parametreler kullanılarak yöntemler belirlenen kriterler açısından karşılaştırılacaktır.

4.1 Yöntem Parametrelerinin Belirlenmesi

Tez kapsamında kullanılan yöntemlere ilişkin üçüncü bölümde bilgi verilmişti. Verilen bilgilerden anlaşılacağı üzere yöntemlerde çeşitli parametrelere dayalı çeşitli denklemler kullanılmaktadır. Söz konusu parametrelerin değerleri doğru seçildiğinde yöntemi kullanan kullanıcının doğru karar verme ihtimali de artmış olacaktır.

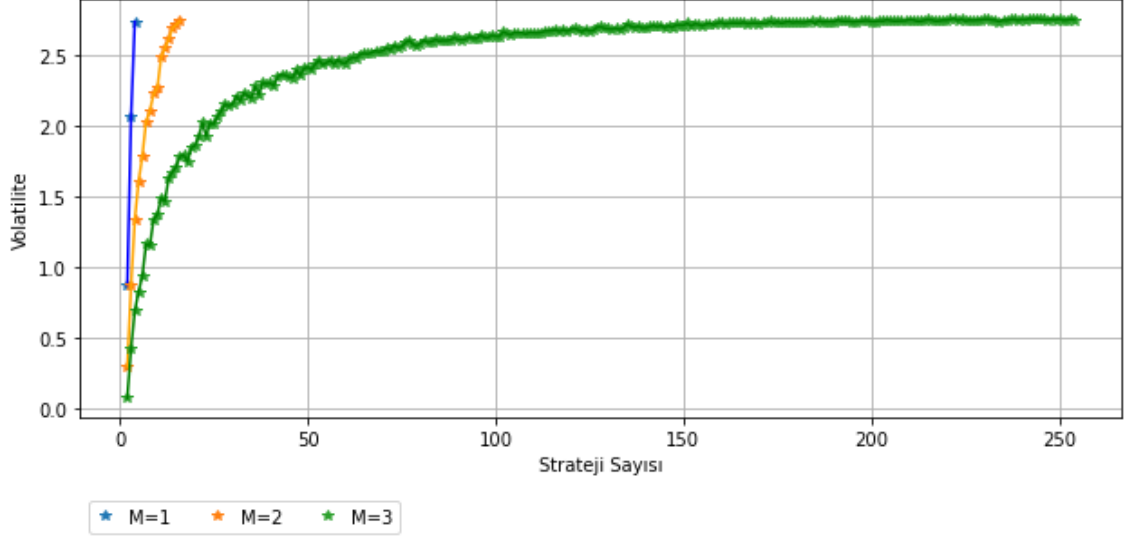
Üçüncü bölümde anlatıldığı üzere sunucunun verimi volatilité ile doğrudan ilintilidir. Yüksek volatilité sunucunun bir turda aşırı dolarken diğer turda büyük oranda boş kalması anlamına gelmektedir. Volatilité azaldıkça sunucuyu kullanmak isteyen kullanıcı sayısı sunucunun kapasitesi civarında salınacağından sunucu kapasitesi verimli olarak kullanılmış olacaktır.

Bu bölümde yöntemlerde yer alan değişkenler için olabilecek tüm ihtimaller tek tek denenerek volatilitenin en düşük çıktığı değişken değerleri belirlenecektir. Parametrelerin belirlenmesi için yapılan çalışmalarda oyuncu sayısı 21 olarak kullanılmıştır.

4.1.1 Azınlık Oyunu yöntemi

Azınlık oyununda volatilitéyi etkileyebilecek iki temel parametre bulunmaktadır. Bunlar hafıza miktarı ve her bir oyuncunun sahip olduğu strateji sayısıdır. Hafıza miktarı 1 iken tam strateji tablosundan seçilebilecek en fazla 4 , 2 iken 16, 3 iken 256, 4 iken 65.536, 5 iken 4.294.967.296 strateji bulunmaktadır. Strateji sayısı arttıkça işlem miktarı çok artacağından hafıza miktarının benzetim çalışmasını tüm strateji sayıları için

gerçekleştirmek mümkün olmayacaktır. Bu kapsamda öncelikle hafıza miktarı 1, 2 ve 3 için tüm strateji sayıları denenerek volatilité değérleri Şekil 4.1’de gösterilmiştir.

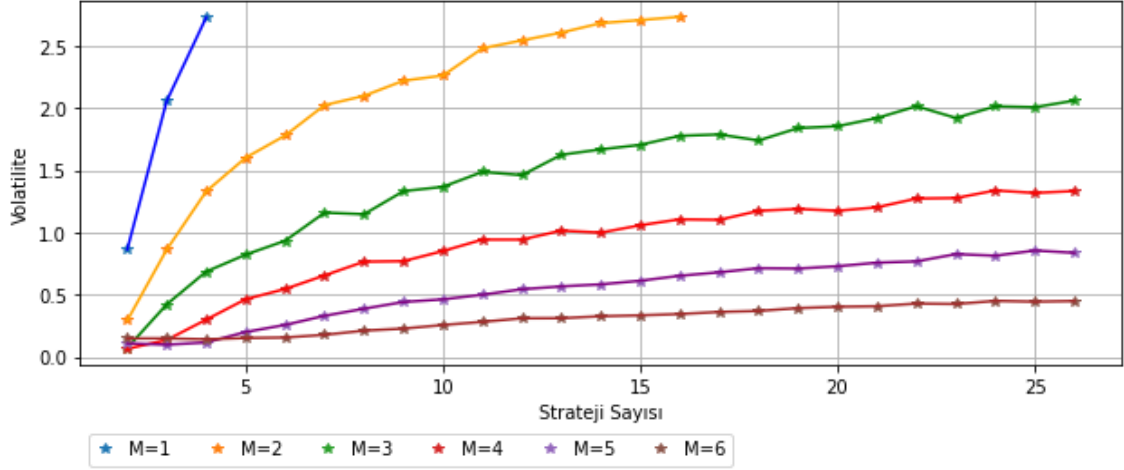


Şekil 4.1 Azınlık oyunu yönteminde hafıza miktarı 1, 2 ve 3 iken değışen strateji sayılarına göre volatilitenin değışimi

Azınlık oyunu yöntemi için oyuncuların strateji sayısının 0 olması gibi bir durum söz konusu olamayacaktır. Strateji sayısının 1 olması durumunda ise tüm oyuncular her oyunda sürekli aynı stratejiyi seçeceğinden karar değıştirmeleri gibi bir durum söz konusu olamayacaktır. Bu nedenle benzetim çalışmalarında strateji sayısının 0 ve 1 olduğu değérler alınmamıştır. Yapılan çalışmada en düşük volatilité değeri strateji sayısı 2 hafıza miktarı 3 iken 0,0268 olarak elde edilmiştir.

Şekil 4.1’den anlaşılacağı üzere oyuncuların seçeceği strateji sayısı arttıkça volatilité değérleri de artmaktadır. Diğer taraftan hafıza miktarı arttıkça volatilité değérinin düştüğü görülmektedir. Bu durum akıllara hafıza miktarının artırılması ve strateji sayısının düşük tutulması durumunda volatilitenin daha da düşüp düşmeyeceği sorusunu getirmektedir. Şekil 4.2’de hafıza miktarı 6’ya kadar olacak şekilde volatilitenin strateji sayısına göre değışimi gösterilmektedir.

Daha önceki grafikte strateji sayısı arttıkça volatilitenin de arttığı bulunduğundan hafıza miktarı 3 ve sonrası için strateji sayısı 26'da kesilmiştir.



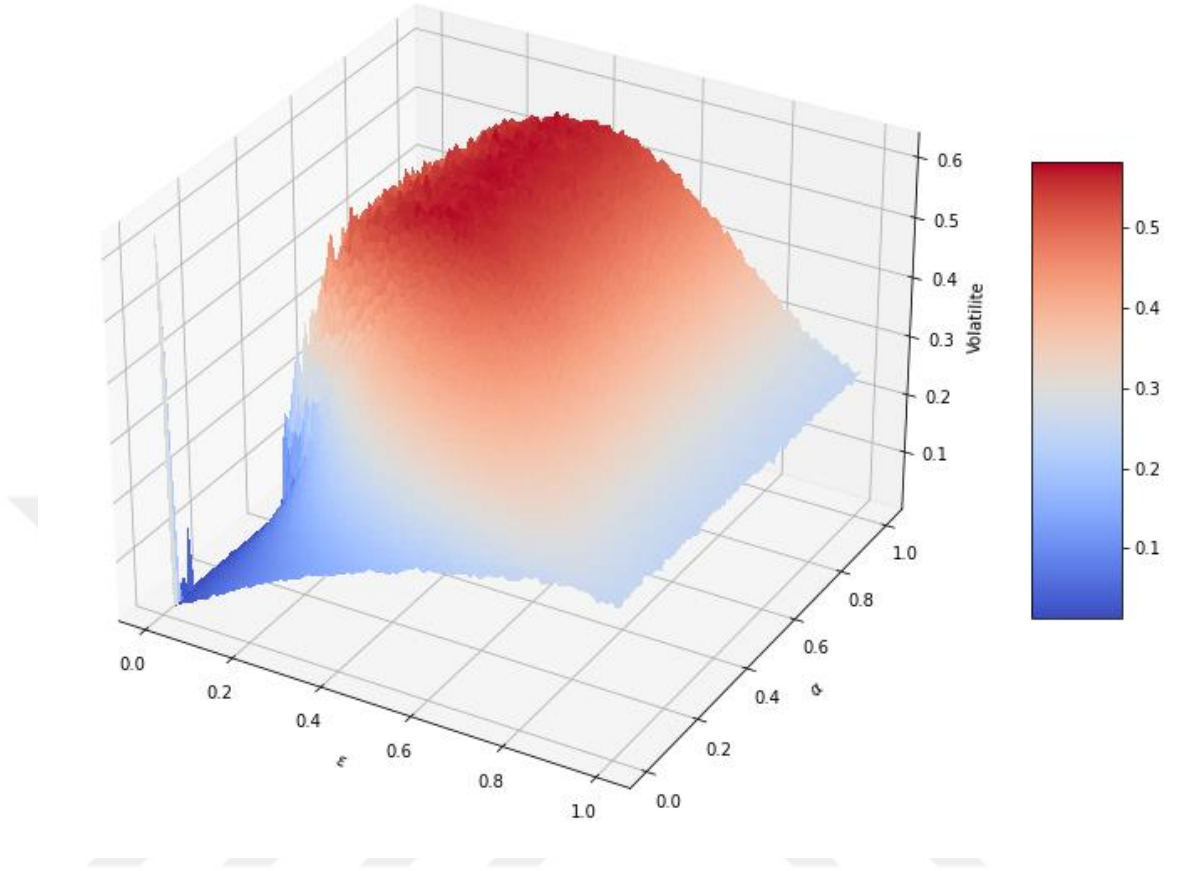
Şekil 4.2 Azınlık oyunu yönteminde hafıza miktarı 6'ya strateji sayısı 25'e kadar iken değişen strateji sayılarına göre volatilitenin değişimi

Şekilden görüldüğü üzere volatilitenin en küçük değeri hafıza miktarı 4 strateji sayısı 2 iken elde edilmiştir. Bulunan en küçük volatilité değeri 0,0628'dir. Hafıza miktarı 4'ü geçtikten sonra Strateji sayısının küçük değerleri için de volatilitenin arttığını söylemek mümkündür.

4.1.2 Q-öğrenme yöntemi

Q-öğrenme yönteminde volatilitéyi etkileyebilecek iki değişken bulunmaktadır. Bunlar: öğrenme katsayısı olan α ile keşfetme katsayısı olan ϵ değerleridir. Bu iki değer 0 ile bir arasındadır. Bu değerler 0,01 aralıklarla tek tek denenmiş ve şekil 4.3'teki üç boyutlu grafik oluşturulmuştur.

Şekilden anlaşılacağı üzere α ve ϵ 'un farklı değerleri için volatilité 0 ile 1 arasında değerler almaktadır.

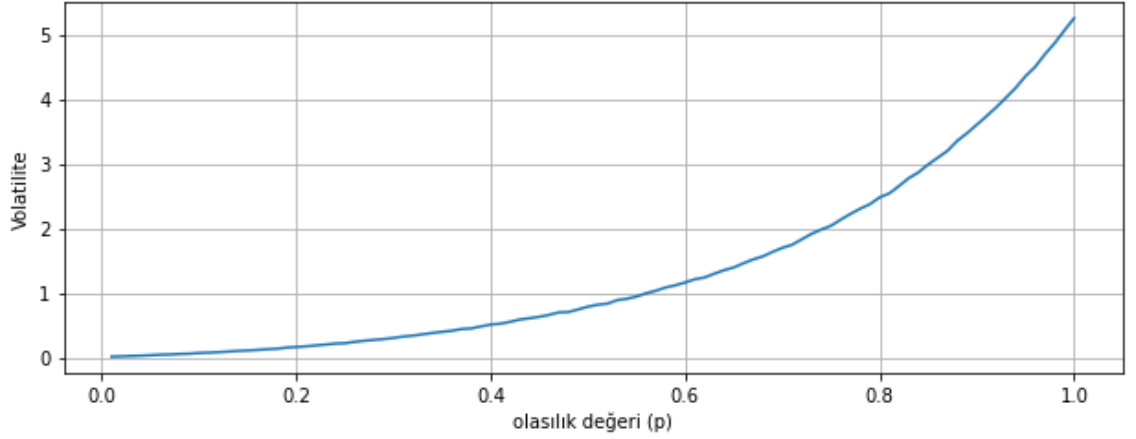


Şekil 4.3 Q-Öğrenme yönteminde öğrenme katsayısı (α) ve keşfetme katsayısı (ϵ) değerlerinin değişimine göre volatilitenin değişimi

Yapılan çalışmada volatilitenin en küçük değeri $\alpha = 0,11$ ve $\epsilon = 0,01$ iken 0,0119 olarak elde edilmiştir. En yüksek olduğu değer ise $\alpha=0,01$ ve $\epsilon=0,01$ iken 0,6274 olarak bulunmuştur. Grafikten görüldüğü üzere Q-öğrenme yönteminde volatilitenin değeri α 'dan ziyade ϵ 'un değişiminden etkilenmektedir. ϵ büyüdükçe keşfetme özelliği artmakta yani rastgele karar verme olasılığı yükselmektedir. $\epsilon=1$ olması yöntemin rastgele karar verdiği anlamına gelmektedir. Grafikte görüldüğü üzere bu bölgede volatilitenin değeri 0,25 civarındadır.

4.1.3 Kazan Kal Kaybet Değiştir stratejisi

Kazan Kal Kaybet Değiştir Stratejisinde volatilitiyi etkileyebilecek tek değişken karar değiştirme olasılığı yani p 'dir. p değeri 0 ile 1 arasında 0,01 aralıklarla tek tek denenmiş ve Şekil 4.4'teki grafik oluşturulmuştur.



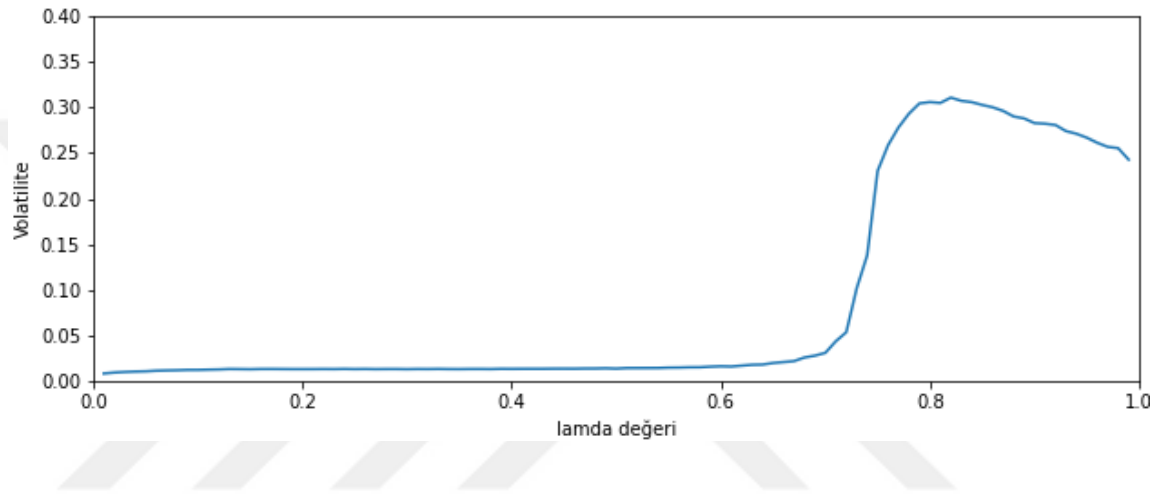
Şekil 4.4 Kazan Kal Kaybet Değiştir stratejisinde karar değiştirme olasılığı (p) değerinin değişimine göre volatilitenin değişimi

Yapılan çalışmada volatilitenin en düşük değeri p 0,01 olduğu durumda 0,017 olarak elde edilmiştir. Grafikten anlaşılacağı üzere p değeri arttıkça volatilité yükselmektedir. Volatilitenin en yüksek değeri olasılığın 1 olduğu durumda 5,25 olarak bulunmuştur.

Üçüncü bölümde anlatıldığı üzere Kazan Kal Kaybet Değiştir stratejisinde bir turda kaybeden oyuncular bir sonraki turda p olasılıkla kararlarını değiştirmektedirler. p değerinin 1 olduğu durumda ilk turda kaybeden bütün oyuncular ikinci turda mutlaka karar değiştirecekler, kazanan oyuncular karar değiştirmedikleri için ikinci turda tüm oyuncular aynı kararı vererek tüm oyuncular kaybetmiş olacaktır. İkinci turdan sonraki tüm turlarda bütün oyuncular karar değiştirerek aynı kararı vermeye devam edeceklerdir. Bu durumda en kötü senaryo gerçekleşmiş olacak ve volatilité 3. bölümde volatilité konusunda anlatıldığı gibi 21 oyuncu için $V = 5,25$ olarak bulunmuş olacaktır. Bu da bulduğumuz sonucu doğrulamaktadır.

4.1.4 Roth-Erev öğrenme

Denklem 3.13 ile belirtilen Roth-Erev Öğrenme yöntemine ait denklem incelendiğinde volatilitiyi etkileyebilecek tek değişkenin indirgeme faktörü (λ) olduğu görülmektedir. Şekil 4.5'te yer alan grafikte indirgeme faktörünün farklı değerleri için elde edilen volatilité değerleri gösterilmektedir. Grafik λ değerinin 0 ile 1 arasındaki değerleri 0,01 aralıklarla tek tek denenerek elde edilmiştir.

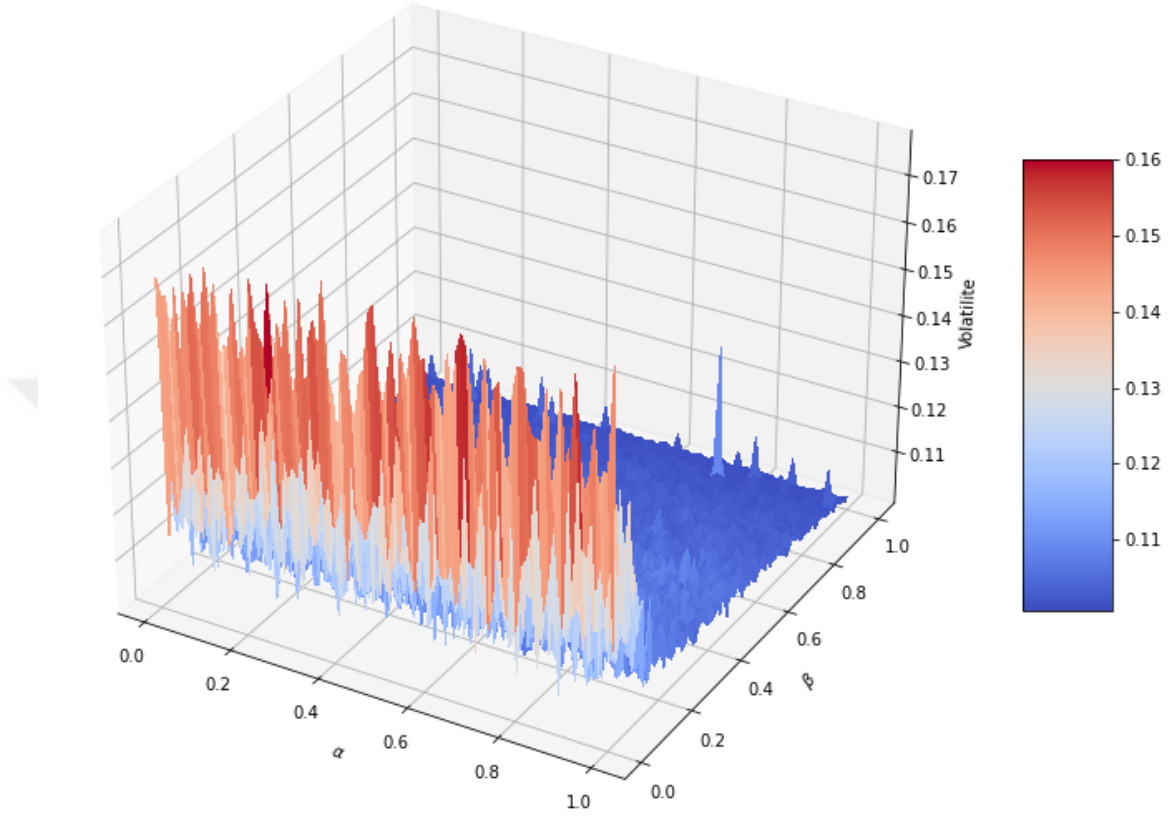


Şekil 4.5 Roth-Erev yönteminde indirgeme faktörü (λ) değerinin değişimine göre volatilitenin değişimi

Grafikte görüldüğü üzere volatilitenin en düşük değeri indirgeme faktörü (λ) 0,01 iken 0,009 olarak gerçekleşmiştir.

4.1.5 Öğrenen Durum Makinaları yöntemi

Öğrenen Durum Makinaları yönteminde kullanılan denklemler 3.15 ve 3.16'da verilmişti. Bu denklemlerden yola çıkarak volatilitiyi etkileyen parametrelerin ödül katsayısı (α) ve ceza katsayısı (β) olduğunu söylemek mümkündür. Bu iki değer 0 ile 1 arasındadır. Bu değerler 0,01 aralıklarla tek tek denenmiş ve şekil 4.6'deki üç boyutlu grafik oluşturulmuştur.



Şekil 4.6 Öğrenen Durum Makinaları Yönteminde ödül katsayısı (α) ve ceza katsayısı (β) değerlerinin değişimine göre volatilitenin değişimi

Şekilden görüldüğü üzere β 'nın küçük değerleri için volatilité yüksek çıkarken diğer değerlerde birbirine çok yakın çıkmıştır. Grafikte volatilité için α değerine bağıllık gözlenmemektedir. Yapılan çalışma sonucunda volatilitenin en düşük değeri için ödül katsayısı, (α) 0,86, ceza katsayısı (β) ise 0,73 olarak bulunmuştur. Bu değerlerle yapılan benzetim çalışması sonucunda volatilité değeri 0,1001 olarak elde edilmiştir. Volatilitenin en yüksek değeri ise $\alpha=0,37$, $\beta=0$ iken 0,1778 olarak bulunmuştur.

4.2 Yöntemlerin Karşılaştırılması

Tez kapsamında buraya kadar Bölüm 3.2’de karşılaştırma kriterleri, 3.3’te yöntemler anlatılmıştı. Ayrıca Bölüm 4.1’de kullanılan yöntemlerde volatilitiyi minimum yapan parametreler belirlenmiş oldu. Bu bölümde 4.1’de bulunan parametre değerleri kullanılarak yöntemler Bölüm 3.2’de anlatılan kriterler açısından karşılaştırılacaktır.

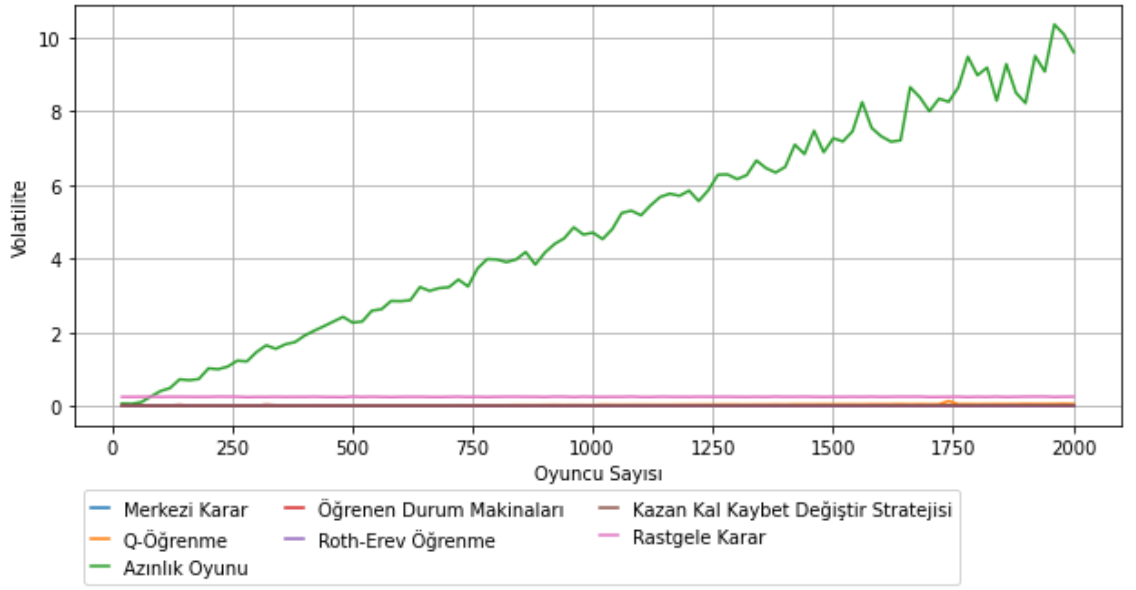
Karşılaştırma yapılırken volatilité, ortalama refah ve adalet kriterlerinde kullanıcı sayıları değiştirilerek kullanıcı sayısının etkileri de izlenecektir. Bir turun ortalama tamamlanma süresi ve bir turda harcanan ortalama enerji kriterlerinde sunucu kapasitesi oyuncu sayısının yarısı kadar değil gerçek değerlere uygun seçilmiştir. Bu nedenle bu kriterler için oyuncu sayısı sabit alınmıştır.

Tezde bu kısma kadar bir parametre olarak oyuncu sayısının yöntemlere etkisinden bahsedilmemişti. Bunun nedeni oyuncu sayısının yöntemlere özgü bir parametre değil, tamamen dış bir etken olmasıdır. Yani gerçek hayatta kullanıcı sayısının belirlenebilmesi gibi bir durum mümkün olmayacaktır.

4.2.1 Volatilité

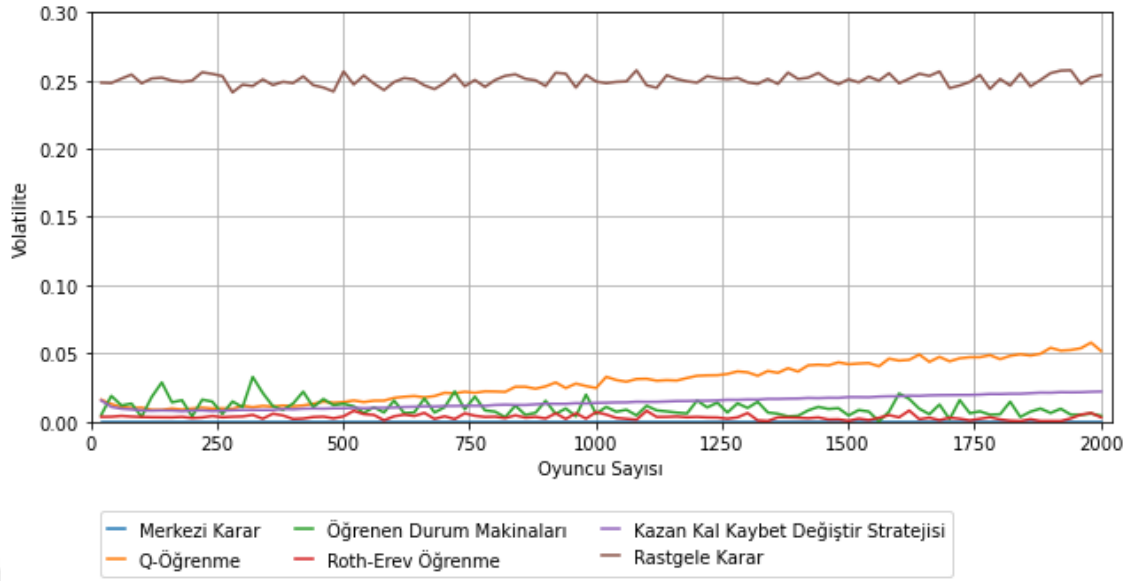
İşlem yükü devri yöntemlerini karşılaştırmada en önemli kriterlerden biri volatilitedir. Bölüm 3.2.1’de volatilité hakkında bilgi verilmiştir.

Şekil 4.7’de değişen oyuncu sayısına göre volatilitenin değişimi gösterilmektedir. Burada 21 oyuncudan 2001 oyuncuya kadar 20’şerli aralıklarla alınan volatilité değerleri yer almaktadır.



Şekil 4.7 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan volatilitte değerlerini gösteren grafik

Grafikte değişen oyuncu sayısından en fazla etkilenen yöntemin azınlık oyunu yöntemi olduğu sonucu ortaya çıkmaktadır. Bu yöntemde oyuncu sayısı arttıkça volatilitte çok yüksek değerler alabilmektedir. Azınlık oyunu yönteminin volatilitte değişimi o kadar yüksektir ki diğer yöntemlerin değişimi görülememektedir. Bu nedenle şekil 4.8’de grafik volatilitenin 0 ile 0,3 olduğu aralıkta yeniden çizdirilmiştir.

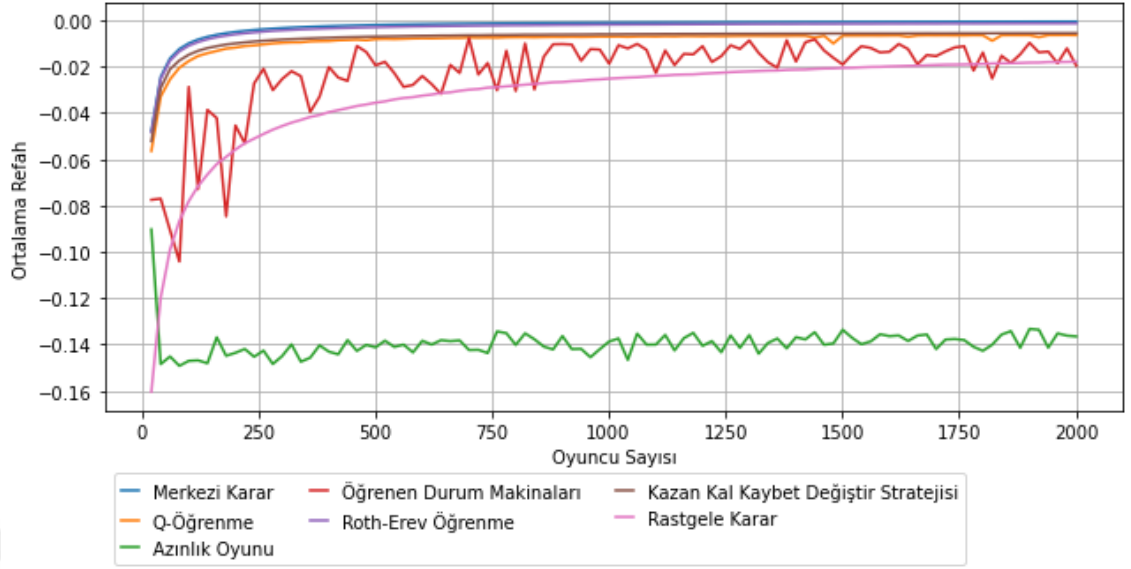


Şekil 4.8 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan volatilitte değerlerini gösteren grafik (0-0,3 arası)

Azınlık Oyunu, Q-öğrenme ve Kazan Kal Kaybet Değiştir Stratejisinde oyuncu sayısı arttıkça volatilitte genel olarak artmaktadır. Yapılan çalışmada bulunan en düşük volatilitte değeri 0,00013 olmuştur. Bu değer 1561 oyunculu sistemde Öğrenen Durum Makinaları yöntemiyle elde edilmiştir.

4.2.2 Ortalama refah

Bölüm 3.2.2’de anlatıldığı üzere ortalama refah bir yöntemin başarısını gösteren bir gösterge olarak kullanılabilir. Ortalama refah değeri negatif bir değerdir ve ne kadar yüksek olursa yöntem o kadar başarılıdır denilebilir. Şekil 4.9’da yöntemlerde elde edilen ortalama refah değerleri gösterilmektedir.



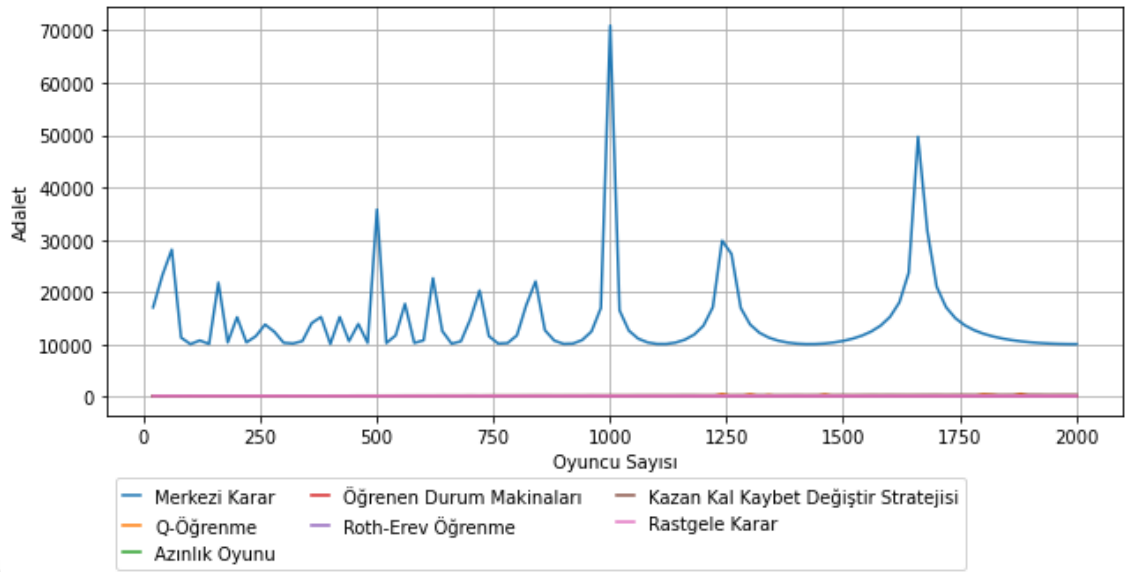
Şekil 4.9 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan ortalama refah değerlerini gösteren grafik

Grafikten anlaşılacağı üzere oyuncu sayısı arttıkça azınlık oyunu yöntemi haricinde tüm yöntemlerde ortalama refah düzeyleri artmaktadır. Azınlık oyunu yönteminde ise oyuncu sayısına bağlı olmaksızın çok farklı sonuçlar vermektedir.

Üçüncü bölümde Ortalama Refah anlatılırken en iyi değer $-1/(2k+1)$ yani $-1/\text{oyuncu sayısı}$ olacağı anlatılmıştı. Yapılan çalışmada kullanılan en yüksek oyuncu sayısı 2001 olduğundan 2001 oyuncu için ideal değer yaklaşık olarak 0,0005'tir. Yapılan çalışmada ortalama refahın en yüksek değeri 2001 oyuncu ile Roth-Erev öğrenme yönteminde -0,0013 olarak bulunmuştur.

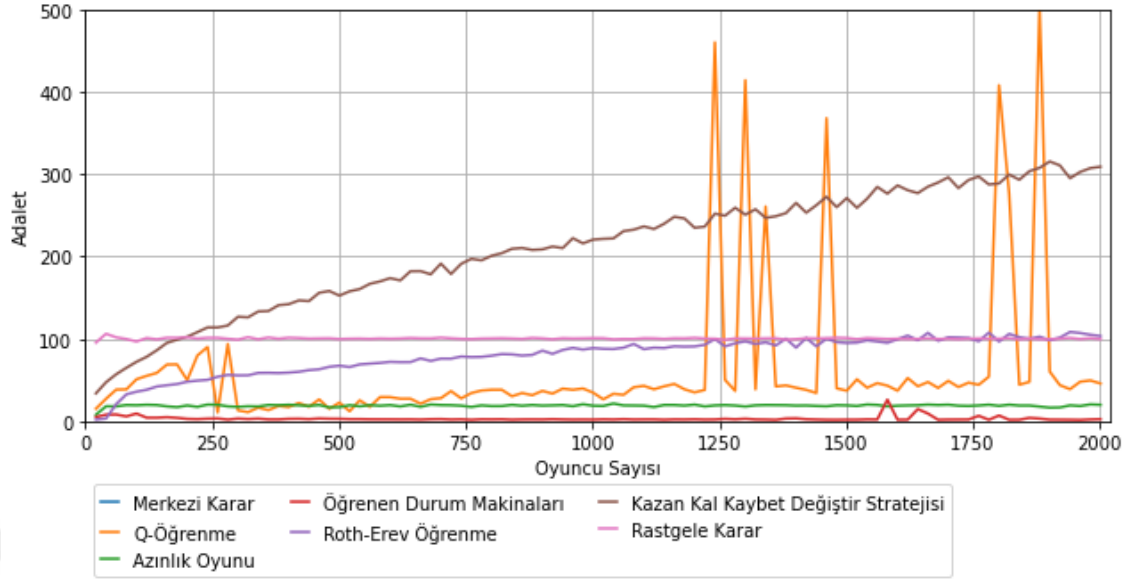
4.2.3 Adalet

Adalet parametresi bölüm 3.2.3'te anlatılmıştı. Şekil 4.10'da oyuncu sayısının değişimine göre adalet değerleri verilmektedir.



Şekil 4.10 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan adalet değerlerini gösteren grafik

Bölüm 3.2.3'te ideal durumda adalet değerinin sonsuza yakınsayacağı anlatılmıştı. Bu doğrultuda tahmin edileceği üzere merkezi karar alan varsayımsal sistemin diğer sistemlere göre adalet değerinin çok yüksek çıktığı görülmektedir. Şekil 4.11'da aynı şekil 0-500 adalet değeri aralığında yeniden çizdirilmiştir.



Şekil 4.11 Değişen oyuncu sayısına göre yöntemlerde ortaya çıkan adalet değerlerini gösteren grafik (0-500 aralığı)

Şekilde görüldüğü üzere adaletin en yüksek olarak bulunduğu yöntem Q-öğrenme yöntemidir. Elde edilen en yüksek değer 1541 oyuncu ile 535,25 olarak elde edilmiştir. Bununla birlikte oyuncu sayısı arttıkça birçok yöntemde adalet değerinin arttığı söylenebilir. Diğer taraftan Öğrenen Durum Makinaları ve Azınlık Oyunu yöntemlerinde adalet değerleri sürekli olarak sifıra yakın olarak bulunmuştur.

4.2.4 Bir turun ortalama tamamlanma süresi

Kenar bilişim sistemlerinde işlem yükü devri işlemlerinde süre hesabının nasıl yapıldığına ilişkin bilgi bölüm 3.2.4'te verilmişti. Bu kısımda örnek değerlerle hesaplama yapılarak yöntemlerde elde edilen sonuçlar gösterilmektedir.

Benzetim çalışmasında kullanılan parametreler çizelge 4.1'de gösterilmektedir. Burada kullanılan parametreler (Hasan 2018)'den yararlanılarak belirlenmiştir.

Çizelge 4.1 Süre açısından inceleme için kullanılan parametreler

Parametre	Değer
Kullanıcı Sayısı	2001
Tur Sayısı	10000
İş Yüğü Boyutu	60 Mb
X (Makine Çevrim/Bit)	600
f_m (Kullanıcı cihazı işlem frekansı)	1 GHz
f_s (Sunucu işlem frekansı)	10 GHz
n_c (Sunucu çekirdek sayısı)	32
s (Hafıza katsayısı)	1
Bant Genişliği (r)	10Mbps
Kodlama oranı (α)	0,01935

Gerçekleştirilen benzetim çalışması neticesinde yöntemlerde elde edilen süre değerleri Çizelge 4.2’de verilmektedir.

Çizelge 4.2 süre açısından yapılan incelemede yöntemlerin bir turu ortalama olarak tamamlama süreleri (sn)

Merkezi Karar Alan Sistem	Azınlık Oyunu	Q-öğrenme	Kazan Kal Kaybet Değiştir Stratejisi	Roth-Erev Öğrenme	Öğrenen Durum Makinaları	Rastgele Karar Alan Sistem
36	56,30	112,27	37,12	36,13	39,27	112,77

Yapılan incelemede bir turun ortalama olarak en kısa sürede tamamlandığı yöntem 36,13 saniye ile Roth-Erev öğrenme yöntemi olmuştur.

4.2.5 Bir turda harcanan ortalama enerji

Bu bölümde işlem yükü devrinde enerji hesaplaması yapılmıştır. Burada oyuncuların en az enerji harcayarak yüklerinin işlenmesi amaç edinilmiştir.

Enerjinin hesaplamasında süreler üzerinden gidilmiştir. Burada kullanıcının işlemi kendisinin sonuçlandırması süresince bir saniyede harcadığı enerji miktarı, işlem yükünü iletirken bir saniyede harcadığı enerji miktarı ve işlemin sonucunu beklerken bir saniyede harcadığı enerji miktarı (Kumar ve Lu 2010)'dan alınmış ve süre hesaplamalarında bulunan süreler ile çarpılmıştır. İşlemlerin nasıl gerçekleştirildiğine ilişkin bilgi ve kullanılan denklemler bölüm 3.2.5'te verilmiştir. Benzetim çalışmasında kullanılan parametreler çizelge 4.3'te gösterilmektedir.

Çizelge 4.3 Enerji açısından inceleme için kullanılan parametreler

Parametre	Değer
Kullanıcı sayısı	2001
Tur sayısı	10000
Hafıza miktarı	3 bit
İş yükü boyutu	60 Mb
X (Makine çevrimi/Bit)	600
f_m (Kullanıcı cihazı işlem)	1 GHz
f_s (Sunucu işlem frekansı)	10 GHz
n_c (Sunucu çekirdek sayısı)	32
s (Hafıza katsayısı)	1
Bant Genişliği (r)	10Mbps
Kodlama oranı (α)	0,01935
İşlem için harcanan güç	0,9 W
İletim için harcanan güç	0,65 W
Beklemede harcanan güç	0,3 W

Gerçekleştirilen benzetim çalışması neticesinde yöntemlerde elde edilen süre değerleri Çizelge 4.4’te verilmektedir.

Çizelge 4.4 Enerji açısından yapılan incelemede yöntemlerin bir turda oyuncu başına ortalama olarak harcadıkları enerji (Joule)

Merkezi Karar Alan Sistem	Azınlık Oyunu	Q- öğrenme	Kazan Kal Kaybet Değiştir Stratejisi	Roth-Erev Öğrenme	Öğrenen Durum Makinaları	Rastgele Karar Alan Sistem
30,96	32,58	33,02	32,90	32,89	33,17	33,17

Çizelgeden görüleceği üzere yöntemlerde elde edilen harcanan ortalama enerji değerleri birbirine oldukça yakın çıkmıştır. Bununla birlikte en düşük enerji değeri 32,58 joule ile Azınlık Oyunu yönteminde elde edilmiştir.

5. TARTIŞMA VE SONUÇ

Bu tez çalışmasında Çoklu-Erişim Kenar Bilişim sistemlerinde işlem yükü devri ele alınmıştır. Bu çerçevede tek sunuculu çok kullanıcılı bir sistem kurgulanmış, kenar bilişim sisteminin işlem gücünün kullanıcılar arasında nasıl dağıtılacağına ilişkin çalışma yapılmıştır.

Çalışmalarda problem kaynak tahsis problemi olarak modellenmiştir. Literatürde yer alan bazı oyun teorisi ve pekiştirmeli öğrenme yöntemleri kullanılarak kaynak tahsisi benzetim çalışması yapılmış ve sonuçları incelenmiştir. Kullanılan yöntemler; Azınlık Oyunu Yöntemi, Q-öğrenme, Kazan Kal Kaybet Değiştir Stratejisi, Roth-Erev Öğrenme ve Öğrenen Durum Makinaları yöntemleridir. Ayrıca karşılaştırmaya eklenmesi açısından merkezi karar alan bir sistem ile oyuncuların tamamen rastgele karar verdiği bir sistem de kurgulanarak çalışmalara eklenmiştir.

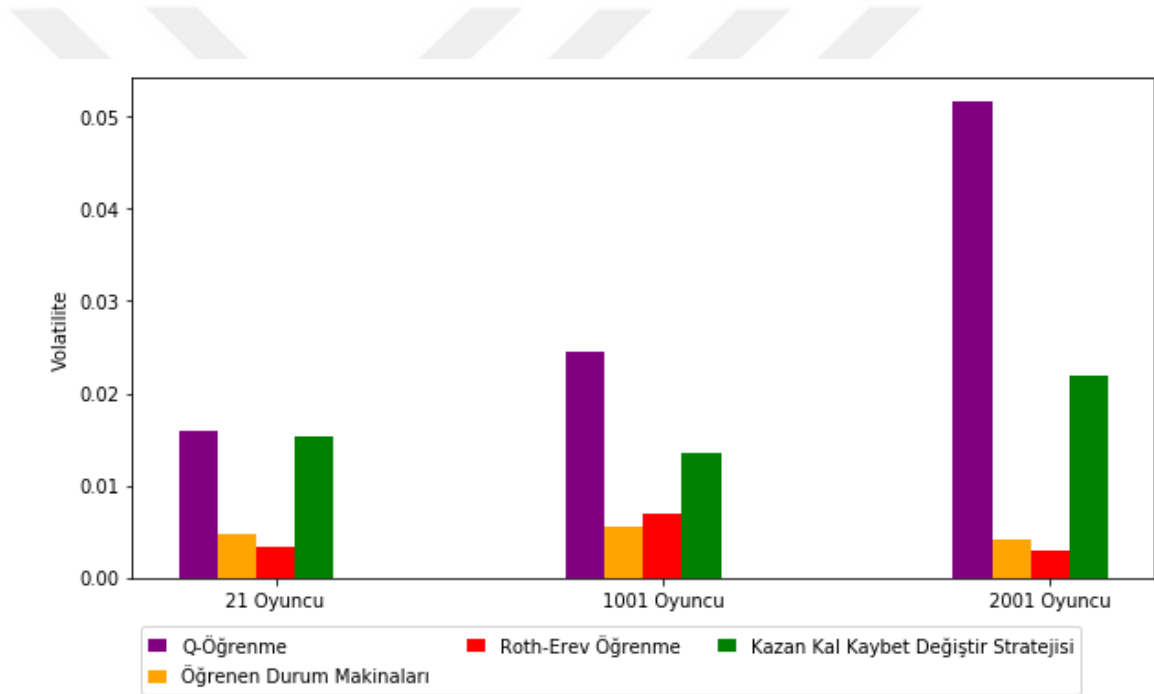
Karşılaştırılan yöntemlerde oyuncular çeşitli kurallar kullanarak karar vermektedir. Bu kurallar çoğunlukla belirli sabit parametreler içeren matematiksel denklemlerden oluşmaktadır. Tez kapsamında kullanılan yöntemlerde parametrelerin sonuca etkisi incelenmiş, belirli şartlar altında daha uygun parametrelerin seçimine özen gösterilmiştir. Söz konusu sabitlerin tespiti ile bulunan sabitler ilgili yöntemlerde kullanılarak yöntemlerin karşılaştırılması gerçekleştirilmiştir.

Yöntemlerin karşılaştırılmasında beş kritere bakılmıştır. Bu kriterler volatilité, ortalama refah, adalet, bir turun ortalama tamamlanma süresi ve bir turda ortalama olarak harcanan enerji kriterleridir.

Kullanılan yöntemler volatilité, ortalama refah ve adalet kriterleri açısından incelenirken oyuncu sayısının değişiminin etkisi de gözlemlenmiştir. Bunun için 21 ile 2001 arasında oyuncu sayıları 20'şer artırılarak çalışma tekrarlanmış ve sonuçlar elde edilmiştir.

Yapılan alıřmalar neticesinde farklı yntemlerin farklı kriterler iin daha iyi sonu verdiđini sylemek mmkndr, yani tm kriterler iin en iyi sonucu veren bir yntem bulunmamaktadır.

Volatilite aısından yapılan incelemede Azınlık Oyunu, Q-đrenme ve Kazan Kal Kaybet Deđiřtir Stratejisinde oyuncu sayısı arttıa volatilite deđerlerinin de arttıđı gzlemlenmiřtir. Deđiřen oyuncu sayısından en fazla etkilenen yntem azınlık oyunu yntemi olmuřtur. Yapılan alıřmada bulunan en dřk volatilite deđerı 0,00013 olmuřtur. Bu deđer 1561 oyunculu sistemde đrenen Durum Makinaları yntemiyle elde edilmiřtir.



řekil 5.1 21, 1001 ve 2001 oyuncu ile yapılan alıřmalarda bulunan volatilite deđerleri

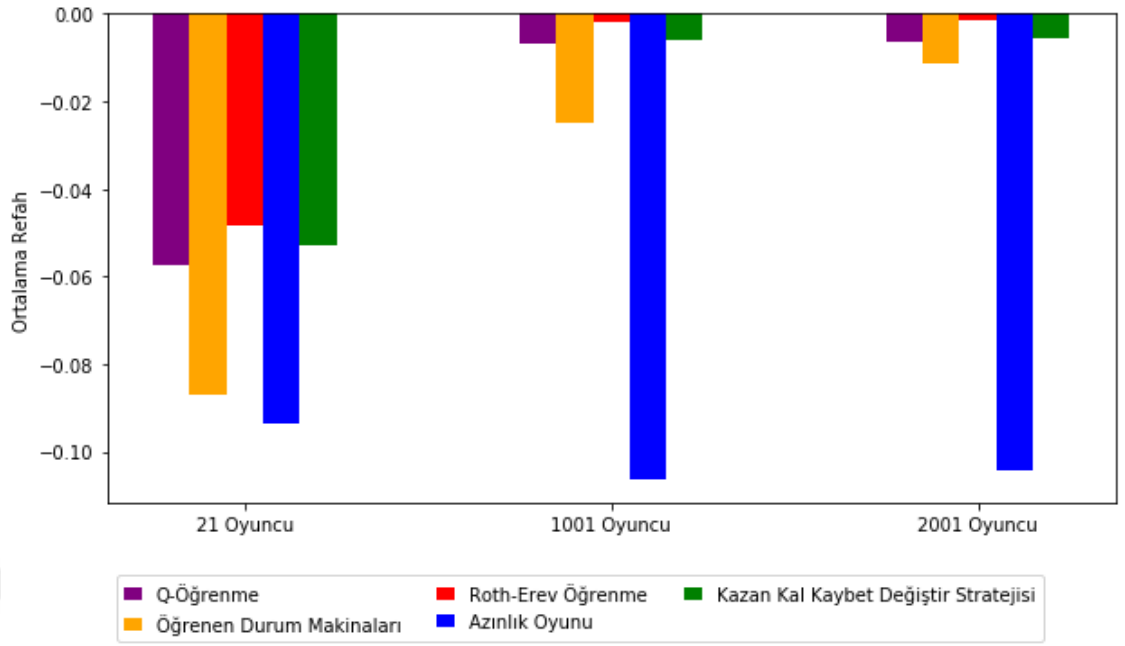
řekil 5.1’de 21, 1001 ve 2001 oyuncu ile yapılan alıřmalarda bulunan volatilite deđerleri stn grafik olarak verilmiřtir. azınlık oyunu ynteminde volatilite oldukça yksek ıktıđı iin grafikte bu ynteme yer verilmemiřtir. Grldđ zere artan oyuncu sayısından en fazla etkilenen yntem Q đrenme yntemidir.

Ranadheera ve arkadaşları tarafından yapılan çalışmada (Ranadheera vd. 2017) Şekil 2’de volatilitenin değişen hafıza miktarına göre grafiği verilmiştir. Şekil incelendiğinde hafıza miktarının 3 olduğu durumda bu tez çalışması kapsamında bulunan sonuçlar ile benzerlik gösterdiği görülmektedir.

Savit ve arkadaşları tarafından yazılan makalede (Savit vd. 1999) Azınlık Oyunu Yönteminde standart sapmanın hafıza miktarına göre değişimi gözlenmiştir. Burada oyuncu sayısı 101, strateji sayısı tezde yer alan çalışmaya benzer şekilde 2 olarak seçilmiştir. Grafikte hafıza miktarının 4 olduğu durumda standart sapmanın ortalamasının 6,5 civarında olduğu görülmektedir. Bu durumda volatilité 0.418 olarak bulunmaktadır. Bu değer tez çalışmamızda bulunan değerle paralellik göstermektedir. Aynı makalede 1001 oyuncu için hafıza miktarı 4, strateji sayısı 2 iken volatilité değeri 5 civarında bulunmuştur. Yine bu değer tez çalışmasında bulunan değerlerle uyumludur.

Catteeuw ve Manderick tarafından 301 oyuncu için yapılan çalışmada Q-öğrenme yöntemi için $\alpha=0,1$ ve $\epsilon=0,01$ alınmış ve sonucu 0,011 olarak bulunmuştur (Catteeuw ve Manderick 2011). Bu değer tez çalışması kapsamında bulunan değerle benzerlik göstermektedir.

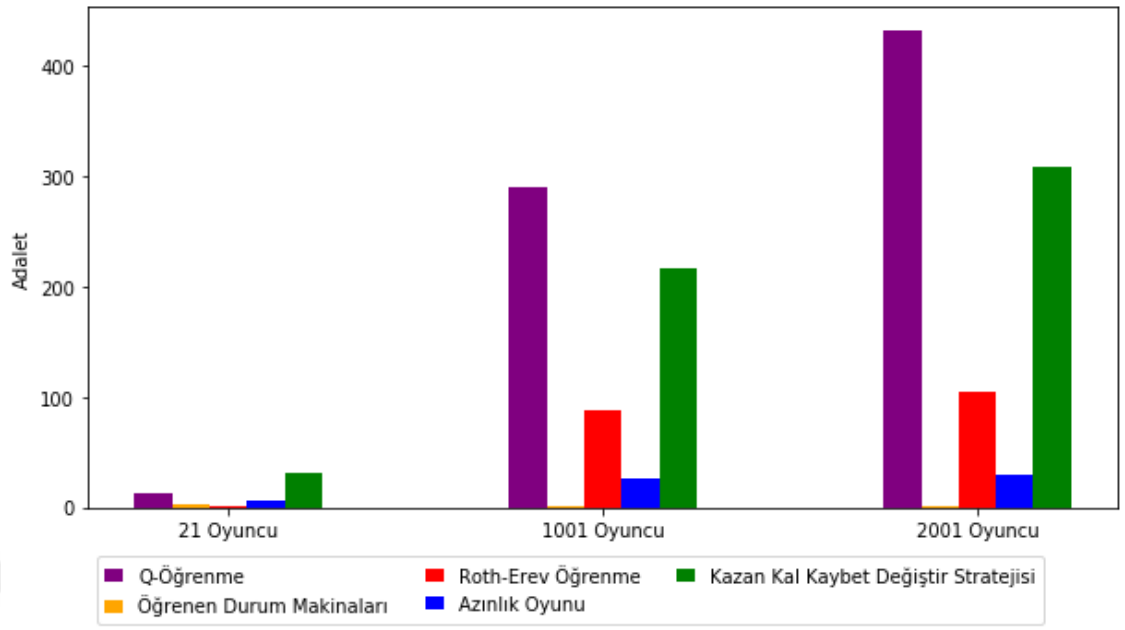
Ortalama refah açısından yapılan incelemede ise oyuncu sayısı arttıkça azınlık oyunu yöntemi haricinde tüm yöntemlerde ortalama refah düzeyleri artmıştır. Azınlık oyunu yönteminde ise oyuncu sayısına bağlı olmaksızın çok değişken sonuçlar vermektedir. Ortalama refahın en yüksek değeri 2001 oyuncu ile Roth-Erev öğrenme yönteminde - 0,0013 olarak bulunmuştur.



Şekil 5.2 21, 1001 ve 2001 oyuncu ile yapılan çalışmalarda bulunan ortalama refah değerleri

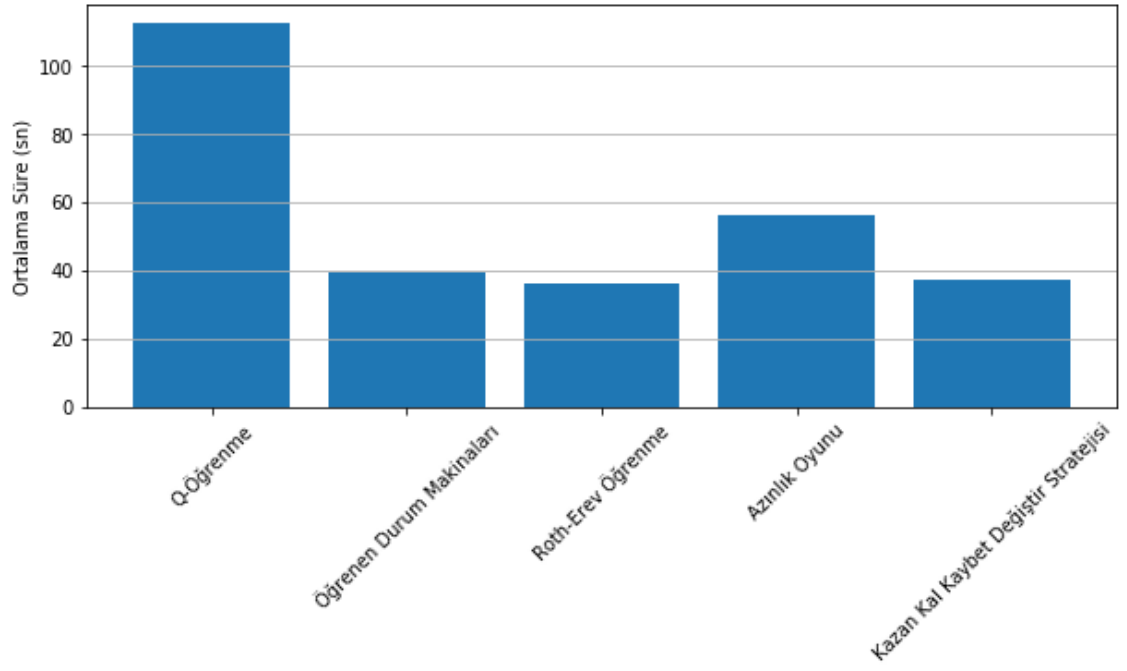
Şekil 5.2’de 21, 1001 ve 2001 oyuncu ile yapılan çalışmalarda bulunan ortalama refah değerleri verilmiştir. Grafikten, azınlık oyunu dışındaki yöntemlerde oyuncu sayısı arttıkça ortalama refah değerinin azaldığı anlaşılmaktadır.

Adalet açısından yapılan incelemede, adalet miktarının en yüksek olarak bulunduğu yöntem Q-öğrenme yöntemidir. Elde edilen en yüksek değer 1541 oyuncu ile 535,25 olarak elde edilmiştir.



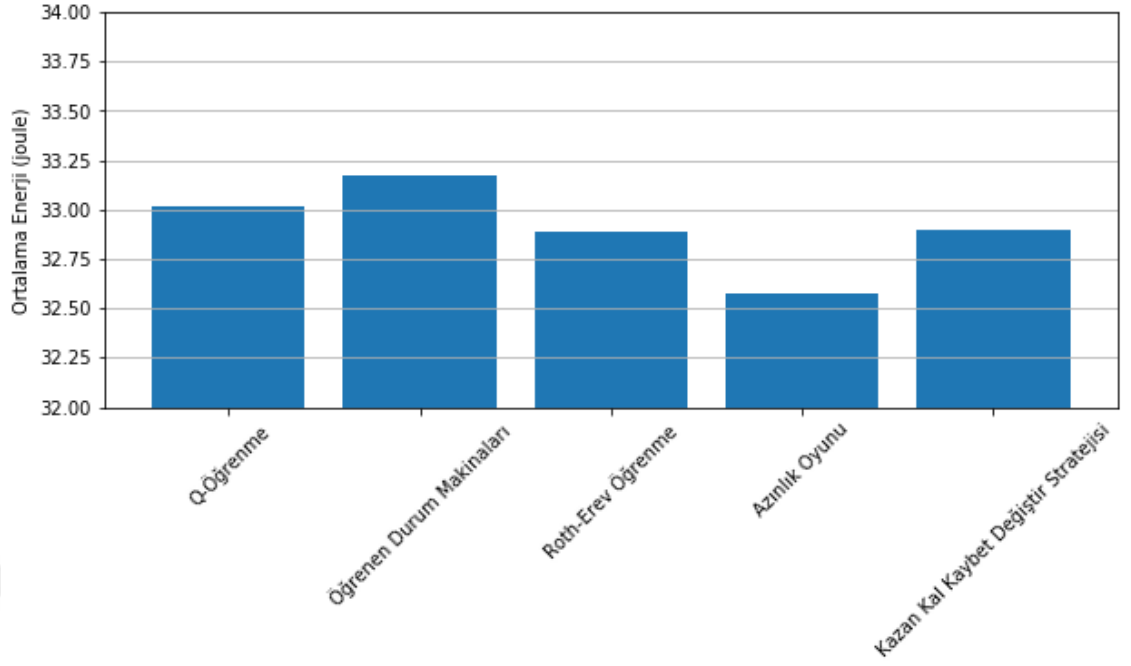
Şekil 5.3 21, 1001 ve 2001 oyuncu ile yapılan çalışmalarda bulunan ortalama refah değerleri

Şekil 5.3'te 21, 1001 ve 2001 oyuncu ile yapılan çalışmalarda bulunan ortalama refah değerleri verilmektedir. Oyuncu sayısı arttıkça birçok yöntemde adalet değerinin arttığı görülmektedir. Diğer taraftan öğrenen durum makinaları ve azınlık oyunu yöntemlerinin oyuncu sayısı değişiminden büyük aranda etkilenmediği söylenebilir.



Şekil 5.4 Bir turun ortalama tamamlanma süreleri

Yapılan çalışmalarda süre bakımından yapılan incelemede en iyi yöntem Roth-Erev Öğrenme yöntemi olarak bulunmuştur. Bir turun ortalama tamamlanma süreleri Şekil 5.4’te verilmiştir. Bu yöntemde bir turun ortalama tamamlanma süresi 36,13 saniye olarak bulunmuştur.



Şekil 5.5 Bir turda harcanan ortalama enerji

Çalışma kapsamında enerji bakımından yapılan incelemelerde ise bulunan en iyi yöntem Azınlık Oyunu yöntemi olmuştur. Bu yöntemde bir turda ortalama olarak harcanan enerji 32,58 joule olarak bulunmuştur. Yöntemlerin harcanan ortalama enerji açısından karşılaştırması Şekil 5.5'te verilmiştir.

İlerleyen zamanlardaki çalışmalarımızda yeni yöntemler eklenerek çalışmalar tekrarlanacaktır. Buna ilave olarak farklı oyuncuların aynı oyun içerisinde farklı yöntemler kullandığı sistemler kurgulanarak oyuncuların heterojen ortamda nasıl davrandığı incelenecektir. Ayrıca oyuncuların yöntemler arasında geçişine izin verilerek en optimum ortamın sağlanması için çalışmalar yapılacaktır.

KAYNAKLAR

- Akherfi K., Gerndt M. and Harroud H. 2018. Mobile cloud computing for computation offloading: Issues and challenges. *Applied Computing and Informatics*, vol. 14, is. 1 pp. 1-16
- Aliyu S. O., Chen F., He Y., and Yang H. 2017. A game-theoretic based qos-aware capacity management for real-time edgeiot applications. 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), July 2017, pp. 386–397.
- Anonymous 2013a. Increasing mobile operators' value proposition with edge computing. Intel and Nokia Siemens Networks Technical Brief. 2013.
- Anonymous 2013b. Smarter wireless networks; add intelligence to the mobile network edge. IBM Corporation Thought Leadership White Paper. 2013.
- Anonymous 2016. Using mobile edge computing to improve mobile network performance and profitability. Saguna and Intel White Paper. 2016.
- Anonymous 2017. The Internet of Things: a movement, not a market. IHS Markit e-book.
- Arthur, W. B. 1994. Inductive reasoning and bounded rationality. *The American Economic Review*, 84(2):406–411, 1994.
- Bay, O. 2018. Hardware Vendors Will Win Big in Meeting the Demand For Edge AI Hardware. <https://www.abiresearch.com/press/hardware-vendors-will-win-big-meeting-demand-edge-ai-hardware/> Erişim Tarihi: 05.10.2020
- Catteeuw, D. and Manderick, B. 2011. Heterogeneous Populations of Learning Agents in the Minority Game. *Lecture Notes in Computer Science*. 7113. 100-113.
- Challet D. and Zhang Y. 1997. Emergence of cooperation and organization in an evolutionary game. *Physica A*, 246:407–418, 1997.
- Chevaleyre Y., Endriss U., Estivie S., and Maudet N. Multi-agent resource allocation with k-additive utility functions. In *Workshop on Computer Science and Decision Theory*, pages 83–100, 2004.
- Deng, M., Tian, H., and Lyu, X., 2016. Adaptive sequential offloading game for multi-cell mobile edge computing. 23rd International Conference on Telecommunications (ICT), May 2016, pp. 1–5.
- Finnegan, M. 2013. Boeing 787s to create half a terabyte of data per flight, says Virgin Atlantic. <https://www.computerworld.com/article/3417915/boeing-787s-to-create-half-a-terabyte-of-data-per-flight--says-virgin-atlantic.html>. Erişim Tarihi: 15.05.2020
- Ha, K., Chen, Z., Hu, W., Richter, W., Pillai, P., and Satyanarayanan, M. 2014. Towards Wearable Cognitive Assistance. *MobiSys 2014 - Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*.

- Hasan 2018. İki-sınıflı mobil kenar bilişim sistemleri için uyarlanır bir aktarma karar yöntemi. Yüksek Lisans Tezi. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- He T. Y., Zhao N., and Yin H. 2017. Integrated networking, caching and computing for connected vehicles: A deep reinforcement learning approach. *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.
- Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., and Young, V. 2015. Mobile Edge Computing A key technology towards 5G, ETSI whitepaper No:11, 2015.
- Jain, A., and Jat, D. S. 2020. An Edge Computing Paradigm for Time-Sensitive Applications. *Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 798-803.
- Jha, D. N., Alwasel, K., Alshoshan, A., Huang, X., Naha, R., Battula, S., Garg, S., Puthal, D., James, P., Zomaya, A., Dustdar, S., Ranjan, R. 2020. IoTSim-Edge: A simulation framework for modeling the behavior of Internet of Things and edge computing environments. *Software: Practice and Experience*.
- Kleinrock, L. 1967. Time-shared Systems: a theoretical treatment. *J. ACM* 14, 2 (April 1967), 242–261.
- Kumar, K. and Lu, Y. 2010. Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?. *Computer*, vol. 43, no. 4, pp. 51-56
- Lam K. and Leung H. 2007. An Adaptive Strategy for Resource Allocation Modeled as Minority Game. *First International Conference on Self-Adaptive and SelfOrganizing Systems (SASO 2007)*, Cambridge, MA, 2007, pp. 193-204.
- Li, A., Yang, X., Kandula, S., and Zhang, M. 2010. CloudCmp: Comparing Public Cloud Providers, *Proc. 10th ACM SIGCOMM Conf. Internet Measurement (IMC 10)*, 2010, pp. 1–14.
- Lin, X., Wang, Y., Xie, Q., and Pedram, M. 2014. Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment, *IEEE Transactions on Services Computing* 8, 2 (2014), 175–186
- Mao, Y., Zhang, J. and Letaief, K.B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices, *IEEE Journal on Selected Areas in Communications*, 34(12), 3590–3605.
- Melendez, S. and McGarry, M. P. 2017. Computation offloading decisions for reducing completion time. *14th IEEE Annual Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 2017, pp. 160-164.
- Miettinen, A. P. and Nurminen, J. K. 2010. Energy efficiency of mobile clients in cloud computing. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10)*. 24
- Nash, J., 1950, Equilibrium points in n-person games, *Proceedings of the national academy of sciences of the United States of America*, vol. 36. No:1.

- Özyurt C. Ö. 2018. Yeraltı üretim yöntemi seçiminde yapay sinir ağları ve oyun teorisinin kullanılabilirliğini araştırılması. Doktora Tezi. İstanbul Üniversitesi Fen Bilimleri Enstitüsü Maden Mühendisliği Ana Bilim Dalı, İstanbul
- Paczkowski, J., 2009. Iphone Owners Would Like to Replace Battery, All Things Digital, 21.08.2009; <http://digitaldaily.allthingsd.com/20090821/iphoneownerswould-like-to-replace-battery-att>.
- Ranadheera, S., Maghsudi, S., and Hossain, E. 2017. Mobile Edge Computation Offloading Using Game Theory and Reinforcement Learning. ArXiv, abs/1711.09012.
- Reents, G., Metzler, R., Kinzel W. 2001. A stochastic strategy for the minority game. *Physica A: Statistical Mechanics and its Applications*. 299. 253-261.
- Reznik A. An update on our Multi-Access Edge Computing ISG The Standard News From ETSI Issue 2, 2017
- Roth, A. and Erev, I. 1995. Learning in Extensive Form Games: Experimental Data and Simple Dynamic Models in the Intermediate Run. *Games and Economic Behavior*. 8. 164-212.
- Sasidevan, V. 2016. Effect of detailed information in the minority game: optimality of 2-day memory and enhanced efficiency due to random exogenous data. *Journal of Statistical Mechanics: Theory and Experiment* vol. 7, no. 7, p. 073405, 2016.
- Sasidevan, V., Dhar, D. 2014 Strategy switches and co-action equilibria in a minority game. *Physica A Statistical Mechanics and its Applications*, vol. 402, pp. 306–317, 2014.
- Sastry, P., Phansalkar, V., and Thathachar, M., 1994. Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information. *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 5, pp. 769- 777, May 1994.
- Satyanarayanan, M. 2017. The Emergence of Edge Computing. in *Computer*, vol. 50, no. 1, pp. 30-39, Jan. 2017, doi: 10.1109/MC.2017.9.
- Sönmez, C., Özgövde, A. and Ersoy, C. 2017. EdgeCloudSim: An environment for performance evaluation of Edge Computing systems. 14th IEEE Annual Consumer Second International Conference on Fog and Mobile Edge Computing (FMEC), Valencia, 2017, pp. 39-44.
- Statista, 2020. <https://www.statista.com/statistics/330695/number-of-smartphoneusers-worldwide/>. Erişim Tarihi: 04.05.2020.
- Taleb T., Samdanis K., Mada B., Flinck H., Dutta S. and Sabella D. 2017. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657-1681, thirdquarter 2017
- Tekin C. and van der Schaar M. 2014. An experts learning approach to mobile service offloading,” in 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Sept 2014, pp. 643–650.

- Toprak, S. 2019. Oyun teorisi ve parçacık sürü optimizasyonu kullanılarak yatırım fonu getirilerinin analizi. Yüksek Lisans Tezi. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Ana Bilim Dalı, İstanbul
- Tran, T. X., Hajisami, A., Pandey P. and Pompili, D. 2017a. Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges. in IEEE Communications Magazine, vol. 55, no. 4, pp. 54-61, April 2017
- Tran, T. X. , Hosseini, M. - P., and Pompili, D., 2017b. Mobile Edge Computing: Recent Efforts and Five Key Research Directions, IEEE COMSOC MMTC Commun.-Frontiers, pp. 29-34, 2017.
- Von Neumann, J., Morgenstern, O. 1944. Theory of Games and Economic Behaviour, Princeton University.
- Watkins C. J. C. H. 1989. Learning From Delayed Rewards. Doktora Tezi, King's College, 1989.
- Watkins C. J. C. H. and Dayan, P. 1992. Q-Learning. Machine Learning, 8, 279-292.
- Whitehead, D. 2008. The El Farol Bar problem revisited: reinforcement learning in a potential game. ESE Discussion Papers 186, University of Edinburgh.2008
- Xu J., Chen L., and Ren S. 2017a. Online learning for offloading and autoscaling in energy harvesting mobile edge computing,” IEEE Transactions on Cognitive Communications and Networking, vol. 3, no. 3, pp. 361–373, Sept 2017.
- Xu J., Palanisamy B., Ludwig H., and Wang Q. 2017b. Zenith: Utility-aware resource allocation for edge computing. 2017 IEEE International Conference on Edge Computing (EDGE), June 2017, pp. 47–54.
- Xu, Z., Liu, X., Jiang, G., Tang, B. 2019. A time-efficient data offloading method with privacy preservation for intelligent sensors in edge computing. J Wireless Com Network 2019, 236
- Yi, S., Hao, Z., Qin, Z., and Li, Q., 2015. Fog computing: Platform and applications. in Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb), Washington, DC, USA, 2015, pp. 73–78.
- Yang, K., Ou S. and Chen H., 2008, On effective offloading services for resource-constrained mobile devices running heavier mobile Internet applications. in IEEE Communications Magazine, vol. 46, no. 1, pp. 56-63, January 2008.
- Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., Pan, L., Maharjan, S., Zhang, Y. 2016. Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks. IEEE Access, vol. 4, pp. 5896-5907, 2016,
- Zheng, J., Cai, Y., Lu, N., Xu, Y., and Shen, X. 2015. Stochastic game-theoretic spectrum access in distributed and dynamic environment. IEEE Trans. Veh. Tech., vol. 64, no. 10, pp. 4807-4820, Oct. 2015.
- Zheng, J., Cai, Y., Wu, Y., and Shen, X. 2016. Stochastic computation offloading game for mobile cloud computing. IEEE/CIC International Conference on Communications in China (ICCC), 2016 pp. 1-6.

Zheng K., Mao Y., Leng S., Maharjan S., and Zhang Y. 2017. Optimal delay constrained offloading for vehicular edge computing networks. 2017 IEEE International Conference on Communications (ICC), May 2017,

