TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PEKİŞTİRMELİ ÖĞRENME YÖNTEMLERİ İLE İHA BAZ İSTASYONU İÇİN VERİ İLETİM HIZI TABANLI OPTİMAL GÜZERGAH BELİRLENMESİ

YÜKSEK LİSANS TEZİ

Melih Doğanay SAZAK

Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Ali Murat DEMİRTAŞ

ARALIK 2022

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Melih Doğanay SAZAK

ÖZET

Yüksek Lisans Tezi

PEKİŞTİRMELİ ÖĞRENME YÖNTEMLERİ İLE İHA BAZ İSTASYONU İÇİN VERİ İLETİM HIZI TABANLI OPTİMAL GÜZERGAH BELİRLENMESİ

Melih Doğanay SAZAK

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Ali Murat DEMİRTAŞ

Tarih: ARALIK 2022

Bu çalışmada, kullanıcılara verilen hizmeti artırmak için İnsansız Hava Aracı (İHA) üzerine bağlı baz istasyonu (Bİ) ile optimal güzergâh planlaması yapılmıştır. Çalışma iki parçada incelenmiştir:
İlk çalışmada İHABİ, 3-boyutta hareket kabiliyeti ile farklı hizmet kalitesi gereksinimlerine sahip hareketsiz kullanıcılara hizmet etmektedir. Güzergâh planlaması yapılırken İHABİ'nin kapsama alanı ve İHABİ ile statik yer baz istasyonu arasındaki ana ağa iletim kapasitesi sınırlandırılmıştır. Bu kısıtlamalar altında amaç, İHABİ için pekiştirmeli öğrenme (ing. Reinforcement Learning, RL) kullanılarak uçuş sırasında kullanıcılara sağlanan toplam veri hızını en üst düzeye çıkaran bir güzergâh bulmaktır. Problemimizde Q-Öğrenme (ing. Q-Learning, QL) uygulaması ile İHABİ, istenilen amaca ulaşmak için doğru aksiyonları almayı öğrenmektedir. Farklı öğrenme parametreleri ile deneme yanılma süreçleri sonucunda uygun parametreler belirlenmiş ve RL modeli bu parametrelerle eğitilmiştir. Kısıtlamaların etkilerini analiz etmek için farklı iletişim senaryoları karşılaştırılmıştır. Bahsedilen kısıtların ve heterojen hizmet kalitesi taleplerinin etkilerine göre İHABİ'nin güzergâh tercihleri ve toplam iletim hızı değişimleri incelenmiştir. Öne çıkan üç sonuç, kapsama, ana ağa iletim ve heterojen hizmet kalitesinin etkilerini göstermiştir. İHABİ, kapsama alanı kısıtlaması arttıkça irtifasını artırma eğilimindedir. Ayrıca, ana ağa iletim kısıtlaması, İHABİ'nin yörüngesini statik yer baz istasyonuna yaklaştırmaya zorlamaktadır. Son olarak İHABİ, kullanıcıların farklı hizmet kalitesi gereksinimlerini mümkün olduğunca dikkate almaktadır.

İHABİ, bu kısıtlamaları karşılamak için en uygun güzergâhı belirleyerek toplam iletim hızını maksimize etmektedir.
Çalışmanın ikinci kısmında İHABİ, sabit yükseklikte harekete ederek hareketli kullanıcılara hizmet sağlamaktadır. Kullanıcılar belirli bir örüntüyü takip ederek hareketini gerçekleştirmektedir. Bu koşullar altında İHABİ için minimumun maksimizasyonu ve maksimizasyon problemleri ele alınarak, ilgili problem için İHABİ'ye uygun güzergâh aranmaktadır. Problemimizde değişen ağ topolojisi sebebiyle Derin Q-Öğrenmesi (ing. Deep Q-Learning, DQN) algoritmasından yararlanılmıştır. Simülasyon sonuçları, minimumun maksimizasyonu probleminde İHABİ'nin kullanıcılara olan mesafesini dengeleyerek adil bir hizmet sağlamaya çalıştığını, maksimizasyon probleminde ise İHABİ'nin kullanıcıların fazla olduğu yerlere uğrayarak toplam hizmet miktarını maksimize etmeye çalıştığını göstermektedir.

Anahtar Kelimeler: Hava baz istasyonu, Sınırlı kapsama alanı, Ana ağa iletim kapasitesi, Heterojen hizmet kalitesi, İHA, Hareketli kullanıcı, Kablosuz iletişim, Pekiştirmeli öğrenme, Derin pekiştirmeli öğrenme.

ABSTRACT

Master of Science

DATA TRANSMISSION RATE BASED OPTIMAL TRAJECTORY DETERMINATION FOR UAV BASE STATION USING REINFORCEMENT LEARNING METHODS

Melih Doğanay SAZAK

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Dr. Ali Murat DEMİRTAŞ

Date: DECEMBER 2022

In this study, optimal trajectory planning is made with the base station (BS) connected to the Unmanned Aerial Vehicle (UAV) in order to increase the service provided to the users. The study is analyzed in two parts:
In the first part of the study, UAV-BS serves immobile users with different quality of service (QoS) requirements with its 3-dimensional mobility. While planning the trajectory, the coverage area of UAV-BS and the backhaul capacity between UAV-BS and the Ground Base Station (GBS) are limited. Under these constraints, the goal is to find a trajectory for the UAV-BS that maximizes the total data rate available to users during the flight using reinforcement learning. UAV-BS learns to take the right actions to achieve the desired goal using Q-Learning algorithm in our problem. As a result of trial and error processes with different learning parameters, appropriate parameters are determined and the reinforcement learning model is trained with these parameters. Different communication scenarios are compared to analyze the effects of constraints. Trajectory preferences and total transmission rate changes of UAV-BS are examined according to the effects of the mentioned constraints and heterogeneous QoS demands. Three prominent results demonstrate the effects of coverage, backhaul, and heterogeneous QoS. The UAV-BS tends to increase in altitude as the coverage increases. In addition, the backhaul constraint forces the trajectory of the UAV-BS to approach the GBS. Finally, UAV-BS takes into account the different QoS requirements of users as much as possible. UAV-BS maximizes the total transmission rate by determining the most suitable trajectory to meet these constraints.

In the second part of the study, UAV-BS provides services to mobile users by moving at a fixed altitude. Users perform their movement by following a certain pattern. Under these conditions, the maximization of the minimum and maximization problems for UAV-BS are handled, and a suitable trajectory for UAV-BS is sought for the related problem. Deep Q-Learning algorithm is used in our problem due to the changing network topology. The simulation results show that in the maximizing of minimum problem, UAV-BS tries to provide a fair service by balancing its distance to the users, and in the maximization problem, UAV-BS tries to maximize the total amount of service by visiting the places where the number of users are high.

Keywords: Air base station, Limited coverage, Backhaul capacity, Heterogeneous quality of service, UAV, Mobile user, Wireless communication, Reinforcement learning, Deep reinforcement learning.

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Dr. Ali Murat DEMİRTAŞ'a, kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümü öğretim üyelerine ve destekleriyle her zaman yanımda olan aileme, babam Tamer SAZAK'a, annem Yasemin SAZAK'a, kardeşim Çağatay SAZAK'a ve arkadaşlarıma çok TEŞEKKÜR ederim.

İÇİNDEKİLER

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

KISALTMALAR

AI : Yapay Zeka
BCD : Blok Koordinasyon İnişi
Bİ : Baz İstasyonu
DDPG : Derin Deterministik Politika Gradyanı
DQN : Derin Q-Öğrenme
ES : Kapsamlı Arama
FDMA : Frekans Bölmeli Çoklu Erişim
İHA : İnsansız Hava Aracı
LoS : Görüş Hattı
MDP : Markov Karar Süreci
ML : Makine Öğrenmesi
NN : Sinir Ağı
OP : Kesinti Olasılığı
RL : Pekiştirmeli Öğrenme
SINR : Sinyal Girişim Artı Gürültü Oranı
SLSQP : Sıralı En Düşük Kare Programlaması
SNR : Sinyal Gürültü Oranı
TDMA : Zaman Bölmeli Çoklu Erişim
QL : Q-Öğrenme
QoE : Deneyim Kalitesi
QoS : Hizmet Kalitesi

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
a_t	t zamanında alınan aksiyon
A	Aksiyon uzayı
B_{gb}	İHABİ'ye tahsis edilen bant genişliği
B_{ua}	Kullanıcıya tahsis edilen bant genişliği
c	Işık hızı (m/s)
$C(\Psi,\sigma)$	Ana ağa iletim kapasitesi (bit/s)
$d(\Psi,\Lambda)$	Kullanıcı ile İHABİ arasındaki mesafe (m)
$d(\Psi,\sigma)$	İHABİ ile statik yer baz istasyonu arasındaki mesafe (m)
f_c	Taşıyıcı frekansı (Hz)
$h(\Psi,\Lambda)$	Kullanıcı ile İHABİ arasındaki dikey uzaklık (m)
$L_{ua}(\Psi,\sigma)$	İHABİ ile statik yer baz istasyonu arasındaki yol kaybı
$L_{us}(\Psi,\Lambda)$	Kullanıcı ile İHABİ arasındaki yol kaybı
N	Bölüm sayısı
N_s	Adım sayısı
p	Geçiş olasılığı
P_{gb}	Statik yer baz istasyonu iletim gücü
P_{LoS}	Görüş hattında olma olasılığı
P_{NLoS}	Görüş hattında olmama olasılığı
P_{ua}	İHABİ iletim gücü
r	Ödül fonksiyonu
$r(\Psi,\Lambda)$	Kullanıcı ile İHABİ arasındaki yatay uzaklık (m)
$R(\Psi,\Lambda,B_{ua})$	İHABİ tarafından kullanıcıya ayrılan veri iletim hızı (bit/s)
R_i^{req}	Kullanıcı i için gerekli hizmet kalitesi miktarı
s_t	t zamanında bulunulan durum
S	Durum uzayı
SNR_{req}	İHABİ kapsama alanının belirlenmesi için gerekli SNR değeri
$S_{ua}(\Psi,\sigma)$	İHABİ'nin SNR değerinin alıcı tarafta 10'a bölümü (dB)
$S_{us}(\Psi,\Lambda,B_{ua})$	Kullanıcının SNR değerinin alıcı tarafta 10'a bölümü (dB)
T	Toplam uçuş süresi
$Q(s,a)$	Q-fonksiyonu
$Q^*(s,a)$	Optimal Q-fonksiyonu
$\theta(\Psi,\Lambda)$	Kullanıcı ile İHABİ arasındaki yükseliş açısı
α	Öğrenme hızı

Simgeler	**Açıklama**
γ	İndirgeme faktörü
Ψ	İHABİ koordinatları
σ	Statik yer baz istasyonu koordinatları
Λ_i	Kullanıcı i'nin koordinatları
ω_n	Gürültü figürü
π	İzlenen politika
π^*	Optimal politika
η	Yol kaybı
μ_{LoS}	P_{LoS} olasılığıyla aşırı yol kaybı (dB)
μ_{NLoS}	P_{NLoS} olasılığıyla aşırı yol kaybı (dB)

1. GİRİŞ

İnsansız Hava araçları, 1800'lü yılların ortasından günümüze kadar uzanan bir tarihi olan [10], bir operatör yardımıyla uzaktan komuta edilebilen teknolojik bir hava aracıdır. Geçmişten günümüze bakıldığında, teknolojideki ciddi gelişmeler sayesinde bu araçlar hem mekanik hem de donanımsal anlamda büyük yol kat etmiş ve yetenekleri artmıştır. Son yıllarda popülaritesi iyice artan yapay zekâ teknolojisi uygulamalarının İHA'ları da kapsaması, İHA'ların daha akıllı hale gelmesini ve çözümü daha kompleks olan problemlerde kullanılabilmesini sağlamıştır. İHA'ların göze en çarpan özellikleri, maliyet bakımından karşılanabilir ve yüksek çeviklik kapasitesine sahip olmalarıdır. Bu özellikleri, İHA'ların çeşitli kullanım alanlarında tercih edilen popüler bir araç olmasını sağlamıştır. Özellikle askeri operasyonlarda sıkça kendine yer bulan İHA'lar, bu alan dışında da kullanılmaktadır. Bu alanlara örnek olarak fotoğraflama, arama kurtarma, taşımacılık, yangın söndürme ve iletişim ağına destek örnek olarak verilebilir. Bu tez çalışmasında İnsansız Hava Aracı Baz İstasyonu (İHABİ) kullanılarak iletişim ağına ek desteğin sağlanması için İHABİ'ye en uygun hareket güzergâhı hesaplaması ele alınmıştır.

İHABİ, İHA'nın hareketinde belirli kısıtlamalara sebebiyet vermeyen bir alıcı-verici sistemin entegre edilmesiyle, İHA'ya baz istasyonu (Bİ) kabiliyetinin kazandırıldığı sistemdir. Özellikle bir bölge için kullanıcıların aşırı yoğun olduğu yerlerde (konser, futbol müsabakası, festival vb.) ya da doğal afetler yüzünden iletişim ağının zarar gördüğü yerlerde, kablosuz haberleşme ağına destek olması açısından sıkça tercih edilmektedir. Buradaki temel amaç, kullanıcılara sağlanan hizmet kalitesindeki yetersizliği ortadan kaldırmak ya da tamamen kesilen iletişimi yeniden canlandırmaktır.

Statik yer baz istasyonlarının kurulumunda tercih edilecek pozisyon için yapılan hesaplamalar zaman alıcı olabilmektedir [14]. Öte yandan İHABİ'lerin hareket kabiliyeti sayesinde yaşanabilecek problemlere karşı hızlıca aksiyon alınabilmektedir. İHABİ'ler ayrıca maliyet açısından statik yer baz istasyonlarına kıyasla çok daha avantajlı bir alternatiftir. İHABİ'lerin bir diğer avantajı ise, yüksek mertebelerde konumlanabildikleri için iletişimde yüksek görüş hattı (ing. Line of Sight, LoS) haberleşme bağlantısı olasılığına sahip olmasıdır. Bu sebeplerden dolayı İHABİ'ler, haberleşme ağına entegre edilmektedir. Tez temel olarak yedi kısımdan oluşmaktadır. Bu bölümün devamında tezin amacı ve kapsamı ile literatür araştırmasına değinilmiştir.

Bölüm 2'de Metodoloji başlığı altında makine öğrenmesi (ing. Machine Learning, ML) kavramı, kısaca genel tanıtımı ve alt kategorileri anlatılacak, alt kategorilerden çalışmamızda yararlandığımız pekiştirmeli öğrenme ve onun alt başlıklarından Q-Öğrenme ve Derin Q-Öğrenmesi detaylıca bahsedilecektir. Bölüm 3'de çalışmamızın ilk bölümünde kullanmış olduğumuz genel sistem ve onun içerisinde yer alan dinamikler, problemin açıklanması ile formülasyonu ve ilgili problem için önerilen algoritma ile probleme uyarlanması ele alınacaktır. Bölüm 4'de çalışmamızın ilk bölümünde gerçekleştirilen simülasyonların nasıl gerçekleştirildiği ve bunların sonuçları tartışılacaktır. Bölüm 5'de çalışmamızın ikinci bölümü incelenecek, Bölüm 3'de yer alan aynı başlıklar ikinci bölüm için ele alınacaktır. Bölüm 6'da çalışmamızın ikinci bölümü için simülasyonlar ve sonuçları değerlendirilecektir. Bölüm 7'de tez çalışması özetlenecek, elde edilen önemli sonuçlardan bahsedilecektir.

1.1 Tezin Kapsamı ve Amacı

İHABİ'ler, farklı haberleşme problemlerine etkili çözümler sunabilmektedir. Sahip olduğu özellikler sayesinde ani gelişen olaylara hızlıca aksiyon alabilmektedir. Bu özellikleri ile her ne kadar güven verici bir seçenek olsa da, İHABİ'yi en uygun şekilde konumlandırma, İHABİ'ye optimal güzergâh belirleme vb. gibi çözümü kolay olmayan problemler ortaya çıkabilmektedir. Bu problemlerin çözümü, kullanıcılara sağlanan hizmet kalitesini iyileştirmek açısından kritik önem taşımaktadır.

Çalışmamızın ilk bölümünde belirli bir ortama rastgele biçimde yerleştirilmiş belirli sayıdaki hareketsiz kullanıcılara sağlanan hizmet kalitesi miktarı, 3-boyutta hareket kabiliyetine sahip olan İHABİ ile arttırılacaktır. Ortamda bir adet statik yer baz istasyonu vardır ve İHABİ ile arasındaki ana ağa iletim kapasitesi (ing. backhaul capacity) sınırlandırılmıştır. Kullanıcıların İHABİ'den talep ettikleri veri iletim hızları farklı olabilmektedir. Çalışmanın bu kısmındaki amaç, İHABİ'nin uçuş süresi boyunca kullanıcılara sağlayacağı toplam veri iletiminin bu koşullar altında maksimize edilmesidir. Sağlanan bu toplam iletim miktarının en yüksek hale getirilmesi için İHABİ'ye en uygun güzergâh belirlenecektir. Bu güzergâhın belirlenmesi işlemi, optimizasyon çözücülerinden farklı olarak RL'deki bir metot olan QL ile yapılacaktır. Problemi açıklayan bir RL modeli ile herhangi çevresel bir ön bilgi olmadan İHABİ için uygun çözüm aranacaktır. Çalışmamızın ikinci bölümünde belirli bir ortama yerleştirilmiş, üç farklı hareketli kullanıcıya sağlanan hizmet kalitesi, 2-boyutta hareket kabiliyetine sahip olan İHABİ ile sağlanacaktır. Kullanıcıların hareketi tamamen rastgele değildir ve belirli bir örüntüyü takip etmektedir.

Bu senaryo için iki farklı tip optimizasyon problemi (minimum değerin maksimizasyonu ve maksimizasyon) çözülerek İHABİ'nin bu farklı problem tipleri doğrultusunda oluşturduğu güzergâhlar incelenecek ve sonuçlar yorumlanacaktır. İki tip problemde de ele alınacak ana parametre veri iletim hızıdır. Minimumun maksimizasyonu probleminde uçuş esnasında İHABİ, hareketli kullanıcılara adil bir veri iletimi sağlamaya çalışacaktır. Maksimizasyon probleminde ise İHABİ, uçuş esnasında hareketli kullanıcılara sağladığı toplam veri iletimini maksimize edecektir. Ortam hareketli kullanıcılardan dolayı dinamik olarak değişmektedir ve bu durumdan dolayı algoritma olarak DQN kullanılacaktır. İki farklı optimizasyon probleminin İHABİ'de meydana getirdiği güzergâhtaki farklılaşma incelenecek ve yorumlanacaktır.

1.2 Literatür Araştırması

İHABİ'ler son zamanlarda oldukça popüler hale geldiğinden, birçok gerçek dünya senaryosunda yaşanan iletişim sorunlarına pratik bir çözüm olarak önerilmektedir. Maliyet açısından uygun olması ve hareketliliği sayesinde kablosuz iletişim ağlarında dikkate değer ve önemli bir bileşen olarak karşımıza çıkmaktadır. Geleneksel karasal ağların konumlandırılması, uzun vadeli trafik tahminleri dikkate alınarak yapıldığı için zaman alan bir süreçtir [14]. Öte yandan İHABİ'ler, anlık talepleri karşılamak için kısa sürede dinamik olarak konumlandırılabilmektedir. İHABİ'ler statik yer baz istasyonlarına göre daha yüksek irtifalarda yer alabildikleri için LoS haberleşme bağlantısı açısından da avantaj sağlamaktadır [28]. Bu özellikleri ile İHABİ'ler, her ne kadar hayatı kolaylaştırsa da İHABİ'leri en uygun şekilde kullanmak da bir o kadar zor bir problemdir. Bu problemlere optimal konum bulma, optimal güzergâh belirleme, enerji tüketimini minimize etme, kapsama alanını maksimize etme ve görev süresini minimize etme örnek olarak verilebilir. Karşılaşılan bu problemler ve yüksek potansiyelleri sayesinde İHABİ'ler, birçok araştırmacının araştırma konusu olmuştur.

Al-Hourani, Kandeepan ve Lardner'in yapmış oldukları çalışma [1]'de, alçak irtifa hava platformlarının belirli bir bölgedeki maksimum kapsama alanını en iyi hale getirmek için yükseklik optimizasyonu yapılmıştır. Ayrıca, hava aracı ile yerdeki bir alıcı arasındaki LoS olasılığı için formül geliştirilmiştir. Çalışmada şehir ortamı ele alınarak, havadan karaya haberleşme hattı boş alan yol kaybı (ing. free space path loss) ile aşırı yol kaybının (ing. excessive path loss) birleşimi olarak modellenmiştir. Ortamda sadece LoS'nin olduğu ve olmadığı şeklinde iki farklı yayılım tipi olduğu varsayılmıştır.

Bu varsayımlar altında ekip, sigmoid fonksiyonuna benzer bir yapı kullanarak, havadan karaya haberleşme hattında LoS olasılığı için matematiksel formül geliştirmişlerdir. Bu formül, ortam parametrelerine ve yerdeki alıcı ile hava platformu arasındaki yükseliş açısına bağlıdır. Çalışmanın devamında ekip, formüle ettikleri LoS olasılığı denklemini kullanarak kabul edilebilecek maksimum hat kaybını matematiksel olarak göstermişlerdir. Bu hat kaybı denkleminden optimal yükseliş açısı, bu açıya ve belirlenen maksimum hat kaybı eşik değerine göre maksimum kapsama alanı yarıçapı hesaplanmıştır. Son olarak, maksimum kapsama alanı yarıçapı ve optimal yükseliş açısı değerlerinden optimal hava aracı yüksekliği bulunmuştur.

Alzenad ve ekibi, çalışma [2] ile baz istasyonu takılı İHA için yer kullanıcılarını maksimum şekilde kapsayacak olan 3-boyutlu yerleştirme problemini ele almışlardır. Araştırılan problemde İHABİ'nin iletim gücü mümkün olduğunca azaltılmaya çalışılmıştır. Çalışmada kullanıcıların kapsanıp kapsanmaması, ilgili kullanıcıya olan hat kaybının belirli bir eşik değere göre kıyaslanmasıyla belirlenmiştir. Belirlenen değerden yüksek olan kayıplar için kullanıcının kapsama dışında kaldığı kabul edilmiştir. Problemin çözümünün kolaylaşması adına iki adımlı bir yol izlenmiştir. İlk adım, İHABİ'nin maksimum kapsama alanı için uygun yüksekliğin bulunmasıdır. Bunun için belirlenen bir hat kaybı eşik değeri ve çevreye göre değişen yükselme açısı yardımıyla bu koşullar için maksimum kapsama alanının yarıçapı hesaplanmıştır. Hesaplanan yarıçap üzerinden yükseklik hesaplaması çalışma [1]'e dayanılarak gerçekleştirilmiştir. İkinci adım, İHABİ için yatay düzlemde 2-boyutta optimal koordinatların belirlenmesidir. Bunun için formüle edilen İHABİ yerleştirme problemi ile İHABİ'nin kapsama alanı içerisinde maksimum kullanıcının yer alması hedeflenmiştir. Bu hesaplamanın çıktıları olan İHABİ'nin yatay düzlemdeki koordinatları ile kapsama alanına giren kullanıcılar, kullanıcıların minimum hizmet kalitesini sağlamak koşuluyla İHABİ'nin iletim gücünü minimize etmek için formüle edilen diğer probleme girdi olarak verilmiştir. İletim gücü minimize edilirken kapsama alanındaki kullanıcı sayısının korunması hedeflenmiştir. Bu problem ile çembersel kapsama alanının merkezi ve kapsama alanının yarıçapı yeniden ayarlanmıştır. Son olarak, yeni çembersel kapsama alanının yarıçapından yükseklik bilgisi hesaplanarak İHABİ için 3-boyutlu yerleştirme problemi çözülmüştür. 2 farklı çevre koşulu için oluşturulan simülasyon ortamına göre kullanıcıların dağılımlarının heterojenlik miktarı, ortalama iletim gücünün miktarını önemli ölçüde değiştirmiştir. Simülasyonların sonuçlarında kullanıcıların birbirlerine daha yakın durdukları (kümelendikleri) dağılım senaryosunda İHABİ'nin kapsama alanında artış ve ortalama iletim gücünde düşüş gözlemlenmiştir.

Lai ve çalışma arkadaşları yapmış oldukları çalışma [27]'de kullanıcıların talep ettiği minimum veri iletim hızını sağlamak koşuluyla, İHABİ için maksimum kullanıcıyı kapsayan 3-boyutlu bir optimal konum aramıştır. İHABİ'nin amacı, aşırı yoğunluk sebebiyle hizmette aksaklıklar yaşayan yer baz istasyonuna destek olmaktır. Çalışmada, İHABİ ile yer baz istasyonunun farklı spektrumlarda çalışmasından dolayı aralarında girişimin olmadığı varsayılmıştır. NP-tam sırt çantası problemi (ing. knapsack problem) olarak modellenen İHABİ ile yerleştirme probleminin kompleksliği, önerilen çözüm yöntemi ile polinomsal zamanda çözülebilecek hale getirilmiştir. Algoritmanın çıktısı, İHABİ için yatay düzlemde x ve y koordinatları ile kapsama alanının yarıçapıdır. Yükseklik ile kapsama alanının yarıçapı arasındaki ilişki, çalışma [1] ile hesaplanmıştır. Sonuçlar incelendiğinde, önerilen algoritmanın farklı hizmet kaliteleri için yapılan simülasyonların hepsinde İHABİ'nin istenen hizmet kalitesinin üzerinde hizmet sağladığı görülmüştür. Ayrıca, çalışma [2]'deki yöntem ile kıyaslandığında, önerilen yöntem sayesinde İHABİ'nin iletim gücünün efektif olarak kontrol edilmesi ile iletim gücünün %29'dan fazla arttığı gözlemlenmiştir. Bunun sebebinin çalışma [2]'de yüksek kullanıcı yoğunluğu durumunun ele alınmamasından kaynaklandığı düşünülmüştür.

Zhang ve ekibi yapmış oldukları çalışma [56]'de, İHA'yı iletişim ağında röle (ing. relay) olarak kullanmışlardır. İHA'nın amacı, yer baz istasyonunun kendi başına erişemediği kullanıcıya köprü görevi görerek yer baz istasyonunun ulaşmasını sağlamaktır. Çalışmadaki hedef, İHA'nın güzergâhının ve iletim gücünün ayarlanarak kesinti olasılığının (ing. outage probability, OP) minimize edilmesidir. OP, sinyal gürültü oranının (ing. signal to noise ratio, SNR) belirlenen eşik değerinin altına düşme olasılığı olarak açıklanmıştır. Çalışmadaki senaryoda bir adet kullanıcı, bir adet yer baz istasyonu ve bir adet İHA yer almıştır. Haberleşme ağı, yarı çift yönlü (ing. half duplex link) iletişime dayanmaktadır. Optimal şekilde İHA güzergâhının ve iletim gücünün ayarlanması iki alt problem altında gerçekleştirilmiştir. İlk optimizasyon problemi, İHA'nın iletim gücü bilgisini alarak yeni güzergâh belirlemiştir. İkinci optimizasyon problemi ise hesaplanan yeni güzergâh bilgisi ile yeni iletim gücünü çıktı olarak vermiştir. Bu iki işlem, OP belirli bir değere yakınsayana kadar yinelemeli olarak devam etmiştir. Önerilen algoritma, iki farklı şema ile kıyaslanarak performansı incelenmiştir. Yazarların önerdikleri yöntemde, güç ve güzergâh optimize edilmiştir. Şema 1'de güç optimize edilmiştir fakat güzergâh daireseldir. Şema 2'de, İHA sabit bir noktadadır ve iletilen güç de sabittir. Elde edilen sonuçlar doğrultusunda en yüksek OP Şema 2'de çıkmıştır. Önerilen yöntem ile Şema 1 başta benzer OP'ye sahip olsa da, belirli bir süre sonra önerilen yöntemin sonucundaki OP %23 oranında azalmıştır.

Kapsamlı arama (ing. exhaustive search, ES) sonucunda elde edilen optimal güzergâh ve iletim gücü hesaplaması ile önerilen algoritma sonucunda elde edilen simülasyon sonuçları kıyaslandığında, iki yöntem arasındaki OP miktarı %5'den az çıkmıştır.

Zeng ve çalışma arkadaşları yapmış oldukları çalışmada [54], İHABİ tabanlı bir iletişim ağında farklı hizmet kalitesi (ing. Quality of Experience, QoE) talebinde bulunan kullanıcıların bulunduğu bir senaryoda, kullanıcı iletişim planlamasını, İHABİ güzergâhını, İHABİ iletim gücünü ve bant genişliği tahsisini birlikte optimize ederek kullanıcıların talep ettiği QoE'yi sağlamayı ve İHABİ'nin enerji verimliliğini maksimize etmeyi hedeflemişlerdir. Problem, karışık konveks ve konkav olmadığı (ing. mixed integer non-convex and non-concave) için çözüm adımlara bölünmüştür. Bu çalışmada statik yer baz istasyonu yer almamıştır, İHABİ tek kaynak olarak kullanılmıştır. Kullanıcılar bulundukları bölgede rastgele olarak dağıtılmıştır ve kullanıcıların beklediği deneyim miktarı birbirlerine göre farklılık gösterebilmektedir. İHABİ'nin kullanıcılarla haberleşmesinde zaman bölmeli çoklu erişim (ing. time divide multi-access, TDMA) ile frekans bölmeli çoklu erişim (ing. frequency divide multi-access, FDMA) tekniği birleştirilmiştir. Kullanıcılar hareket etmemektedir ve yapılan haberleşme planlamasına göre kullanıcılara deneyim sağlanmaktadır. İHABİ tarafından kullanıcılara sağlanan toplam bant genişliği ve toplam iletim gücü sınırlıdır. Dolayısıyla kullanıcılara sağlanan veri iletim hızını güzergâh, iletim gücü, bant genişliği ve haberleşme planlaması belirlemiştir. İHABİ eğer kullanıcının talep ettiği minimum iletim hızını karşılayamazsa, o kullanıcıya deneyimin sağlanmadığı varsayılmıştır. Diğer çalışmalardan farklı olarak, bu çalışmada gecikme kavramı (toplam hizmet süresi) da değerlendirilmiştir. Kullanıcıların gerçek zamanlı olup olmamasına bağlı olarak ilgili kullanıcıya verilecek toplam hizmet süresi ayarlanmıştır. İHABİ'nin enerji tüketimi iki parçadan oluşmaktadır. 3-boyutlu uçuş halindeki İHABİ için enerji tüketim modeli ile iletişim tüketim modeli toplanarak toplam enerji harcaması hesaplanmıştır. Enerji verimliliği, birim harcanan toplam enerji başına iletilen toplam veri miktarı şeklinde tanımlanarak bu ifadenin maksimizasyonu hedeflenmiştir. Çalışmada verilmiş koşullar altında oluşturulan optimizasyon probleminin çözümü üç kademede gerçekleştirilmiştir. İlk etapta amaç fonksiyonu kesirli gösterimden fark gösterimine dönüştürülmüştür. Sonrasında orijinal problem, Blok Koordinasyon İnişi (ing. Block Coordination Descent, BCD) algoritmasıyla dört alt probleme (kullanıcı iletişim planlaması optimizasyonu, İHABİ güzergâh optimizasyonu, bant genişliği tahsisi optimizasyonu ve iletim gücü tahsisi optimizasyonu) ayrılarak optimizasyon çözücüsüyle çözülmüştür. Son olarak Dinkelbach-based yinelemeli algoritma ile İHABİ enerji verimliliği maksimize edilmiştir.

Önerilen yöntemin performansı için simülasyonlar iki parçada gerçekleştirilmiştir. Yeterli kaynakların mevcut olduğu durumda İHABİ sabit yükseklikte hareket etmiştir. Tam tersi koşulda ise İHABİ'ye yüksekliğini ayarlama kabiliyeti verilmiştir. Çalışmada önerilen yöntemin dışında karşılaştırma amacıyla diğer şemalar olan sadece veri iletimi maksimizasyonuna odaklı İHABİ, sadece enerji minimizasyonuna odaklı İHABİ, statik İHABİ, kullanıcıların gereksinimlerini karşılamaya önem vermeyen İHA ele alınmıştır. Simülasyonların sonucu göstermiştir ki önerilen yöntem diğer şemalara göre çok daha yüksek enerji verimliliği sağlamıştır.

Chowdhury ve ekibi yapmış oldukları çalışma [13]'de 3-boyutlu anten çizgesi (ing. antenna radiation pattern) ve ana ağa iletim kapasitesi kısıtını ele alarak baz istasyonu taşıyan İHA için belirli bir başlangıç noktasından bitiş noktasına götüren optimal güzergâh araştırmışlardır. İHABİ, sistemde röle olarak görev almıştır. Amaç, kullanıcıların hücresel bağlantı kalitesini arttırmaktır. Çalışmada İHABİ, 3-boyutta hareket kabiliyetine sahiptir ve banliyö (ing. suburban) ortamında uçmaktadır. Uçuş süresi sınırlandırılmıştır ve bitiş noktasına kadar bu sürede varması hedeflenmiştir. Kullanıcılar ve İHABİ, çok yönlü antene sahiptir. Kullanıcılar, çevresindeki yer baz istasyonları ve İHABİ'ler arasından en yüksek sinyal seviyesine sahip olana bağlanmıştır. Sistemde üç farklı hat kaybı modeli vardır. Farklı 3 modelin seçilmesinin sebebi, sistemi gerçek dünya hücresel ağlara benzetmektir. Anten çizgesi ve ana ağa iletim kapasitesinin kısıtı, optimal İHABİ güzergâhını etkilemektedir. Anten çizgesi modeli olarak 3GPP seçilmiştir ve İHA ile statik baz istasyonu arasındaki mesafeye bağlı olarak anten kazancı hesaplaması yapılmıştır. Kullanıcıdaki baştan sona (ing. end-to-end) sinyalin girişim artı gürültü oranı (ing. signal to interference ratio, SINR), kullanıcıya sağlanan anlık kapasite hesabında kullanılmıştır. Bu oranı etkileyen parametrelerde, yer baz istasyonun anten çizgesi ve 3GPP hat kayıp modeli doğrudan etkilidir. Çalışmada performans metriği olarak kullanılan spektral verimlilik, belirli bir kullanıcının anlık kapasitesinin hücredeki toplam kullanıcı sayısına bölünmesi şeklinde tanımlanmıştır. Bu verimlilik uçuş süresi boyunca kümülatif olarak toplanmış ve bu değeri maksimize eden güzergâh, dinamik programlama kullanılarak elde edilmiştir. Farklı senaryolar için farklı analizler yapılarak yöntemin performansı çok yönden ele alınmıştır. Hesaplamalar yapılırken hesaplama yükünü hafifletmek için zaman ve çevre bilgileri ayrıklaştırılmıştır. İncelenen konular şu şekildedir:

- Optimal Güzergâh
- Spektral Verimlilik Karşılaştırması

- Kesinti Olasılığı Karşılaştırması
- Çalışma Zamanı Kıyaslaması

Optimal güzergâh incelendiğinde İHABİ, statik yer istasyonuna yakın fakat çok yakın konumlanmamıştır. Bunun sebebi, İHABİ'nin kullanıcılara sağlanan hizmet kalitesini arttırmaya çalışırken kullanıcılara olan uzaklığını dengelemeye çalışmasından kaynaklanmıştır. Ayrıca İHABİ, uçuş süresi boyunca vaktinin çoğunluğunu optimal noktada salınarak geçirmiştir. Uçuş süresinin arttırılması, beklendiği gibi toplam hizmet kalitesini de arttırmıştır. Spektral verimlilik kıyaslaması yapılırken İHABİ için 3-boyutlu ve 2-boyutlu hareket senaryoları incelenmiştir. 2-boyutlu senaryolarda İHABİ farklı sabit yüksekliklerde uçmuştur. Sonuçlar İHABİ'nin olmadığı senaryo sonucuna göre yüzdesel olarak elde edilmiştir. İHABİ'nin olduğu tüm koşullar, olmadığı koşula göre daha iyi çıkmıştır. En iyi sonuca, tahmin edildiği gibi İHABİ'ye 3-boyutta hareket yeteneği sağlandığında ulaşılmıştır. Yer baz istasyonu sayısındaki artış, İHABİ tarafından sağlanan spektral verimliliği her senaryoda azaltmıştır. Simülasyonların diğer kısmında, İHABİ'nin olmadığı senaryo ile birlikte 2-boyut ve 3-boyutta hareketli İHABİ'nin bulunduğu senaryolar incelenmiştir. Sonuçlara göre 3-boyutun OP'ye etkisi 2-boyuta göre kısmen daha yüksek çıkmıştır. İHABİ'nin olmadığı durumda ise performansta gözle görülür bir düşüş olmuştur. Çalışma zamanı kıyaslaması yapılırken toplam uçuş süresi ve yatay yer düzlemindeki çözünürlük değerleri değiştirilmiştir. Beklendiği gibi çözünürlük artınca hesaplama süresi de artmıştır fakat daha iyi optimal İHABİ konumları bulunmuştur. Ayrıca İHABİ'ye ait alternatif irtifa değerleri miktarı da arttırılarak süre hesabı yapılmıştır. Alternatif irtifa değerlerinin arttırılması daha fazla tercih anlamına geldiği için bu durum hesaplama süresini arttırmıştır.

Buraya kadar anlatılan çalışmalarda problemler, optimizasyon çözücüleri ile çözülmüştür. Son gelişmeler ve artan popülerliği ile RL, dışbükey olmayan optimizasyon problemini çözmek için umut verici bir alternatif haline gelmiştir.

Bayerlein ve arkadaşları çalışma [6]'da RL kullanarak İHABİ için uçuş zamanı boyunca kullanıcılara sağladığı toplam iletim hızını maksimize edecek optimal bir güzergâh belirlemişlerdir. RL ile herhangi bir çevresel ön bilgiye gerek kalmadan çevre öğrenilerek problem çözülmüştür. Çalışmada kullanmış oldukları İHABİ modeline göre İHABİ, sabit yükseklik ve sabit hız ile hareket etmektedir. Sınırlı uçuş süresine sahip olup, uçuşa başladığı yere tekrar geri dönmesi hedeflenmiştir. Haberleşme kanalı log-mesafe hat kaybı (ing. log-distance path loss) modeli olarak tanımlanmıştır ve ortamda yer alan engeller için sabit zayıflama faktörü kullanılmıştır.

Bu koşullar altında maksimizasyon problemi oluşturulmuştur. Ekip, yapmış oldukları çalışmada QL ve DQN algoritmalarını kullanmışlardır. Bu iki algoritma için durum değişkenleri (ing. states), İHABİ'nin x koordinat değeri, y koordinat değeri ve zaman bilgisinden oluşmaktadır. Aksiyon değişkenleri (ing. actions), yukarı, aşağı, sağ ve sol olacak şekilde 4 farklı hareket yönünden oluşmuştur. Ödül fonksiyonu 3 parçadan meydana gelmiştir. Bunlar, belirli bir t anındaki toplam iletim hızı, İHABİ'nin tanımlanan alanın çıkması durumunda verilen negatif ceza ve uçuş süresinin aşılması durumunda bitiş noktasına gelene kadar verilen negatif cezadır. Simülasyon sonuçlarına göre, çalışmada kullanılan iki teknik ile İHABİ, kısıtlı bir simülasyon ortamı için çalışmada istenen hedefi karşılayan en kısa ve verimli olan güzergâhı hesaplamıştır. Sonuçlar incelendiğinde, QL algoritmasının optimal çözümü bulmak için gerçekleştirdiği yineleme miktarı, DQN algoritmasına kıyasla çok daha fazla olmuştur. QL, 800.000 bölüm (ing. episode) tekrarlayarak, DQN ise sadece 27.000 bölüm tekrarlayarak güzergâhı optimize etmiştir. Bu da göstermiştir ki simülasyon ortamının arttırılması ile QL algoritması birtakım problemler yaşamaya başlamaktadır fakat DQN algoritması bu problemin üstesinden gelebilmektedir.

Zhang, ve ekibi, İHABİ için 3-boyutlu konumlandırma ve güç bölüştürme problemini ele alarak, sistem verimliliğini maksimize etmek için çalışma [55]'yı gerçekleştirmişlerdir. Oluşturdukları konveks olmayan problemin çözümü için derin deterministik politika gradyanı (ing. Deep Deterministic Policy Gradient, DDPG) algoritmasını kullanmışlardır. Çalışmalarında, çalışma [1]'deki havadan karaya kanal modelinden yararlanmışlardır. Amaç, belirli bir ortamda rastgele dağılmış hareketsiz kullanıcılara, İHABİ tarafından sağlanan hizmet kalitesini maksimize etmektir. Oluşturulan optimizasyon problemine göre İHABİ, tanımlanan bölgede kalacak ve yükseklik değerlerini aşmayacak, aynı şekilde sahip olduğu kısıtlı toplam güç miktarını da geçmeyecektir. Kullanıcılara ulaşan alıcı sinyal gücü belirli bir değerin altında kalıyorsa, toplam hizmet miktarı hesaplamasında yer almayacaktır. Tercih edilen DDPG algoritmasına göre durum değişkenleri, İHABİ'nin 3-boyutlu koordinatları, aksiyon değişkenleri ise İHABİ'nin bir sonraki durum değişkenine geçişini sağlayan 3-boyutlu hareket değişimidir. Aksiyon uzayının ayrıca güç ayırma probleminden dolayı daha yüksek değişken sayısından oluşmasının önüne geçmek için su doldurma (ing. Water-filling) algoritması, Markov karar sürecine (ing. Markov Decision Process, MDP) entegre edilmiştir. Son olarak şekillendirilen uygun ödül fonksiyonu ile RL modeli tamamlanmış ve problemin çözümünde kullanılmıştır. Simülasyonlar, 3 farklı senaryo için gerçekleştirilmiş ve önerdikleri algoritma, DQN algoritması ve genetik algoritma ile kıyaslanarak performansı incelenmiştir.

- Senaryo 1'de sistem verimini maksimize etmek için İHABİ'ye optimal 3-boyutlu konumlandırma ve güç bölüştürme optimizasyonu birlikte yapılmıştır.
- Senaryo 2'de sistem verimini maksimize için İHABİ'ye optimal 3-boyutlu konum aranmıştır. Senaryo 1'den farklı olarak toplam güç kullanıcılara eşit olarak bölüştürülmüştür.
- Senaryo 3'de İHABİ'nin yatay düzlemdeki koordinatları tüm kullanıcıların orta noktası olarak seçilmiş ve sadece yükseklik için optimizasyon yapılmıştır. Kullanıcılara sağlanan güç eşit olarak dağıtılmıştır.

Sonuçlar göstermiştir ki, senaryo 1'de önerilen algoritma en iyi performansı göstermiştir. Sistem performansı diğer senaryolarla da kıyaslandığında en yüksek burada görülmüştür. Senaryo 2 ve senaryo 3 incelendiğinde genetik algoritmanın önerilen yönteme göre küçük bir farkla daha iyi sonuçlar verdiği gözlemlenmiştir. Bunun sebebinin daha az sayıda değişken ile sistemin tanımlanabilmesinden kaynaklandığı kararına varılmıştır. DQN, herhangi bir senaryoda en iyi sonucu gösterememiştir. Bunun sebebi, aksiyon uzayının kuantalanmasından (ing. quantization) kaynaklanmaktadır. Diğer bir sonuç olarak, kullanıcı dağılımındaki heterojenliğin artması ile senaryo 1'deki sonuçların senaryo 2'deki sonuçlara kıyasla aradaki farkın açılacağı belirtilmiştir.

Literatürde birçok çalışmada kullanıcıların hareket etmediği varsayılmıştır fakat gerçek dünyada bu durum daha farklıdır. Kullanıcılar farklı zaman dilimlerinde farklı konumlarda bulunabilir.

Ghanavi ve arkadaşları yapmış oldukları çalışma [18]'de hareketli kullanıcıların olduğu bir sisteme İHABİ entegre ederek, kullanıcıların hareketlerinden dolayı hizmet kalitesinde meydana gelen düşüşleri dengelemeyi amaçlamışlardır. Adil bir karşılaştırma için, İHABİ içermeyen geleneksel, sadece statik baz istasyonlarından oluşan senaryodaki toplam baz istasyon sayısı ile bir adet İHABİ ve statik baz istasyonlarından oluşan senaryodaki toplam baz istasyon sayısı eşit tutulmuştur. Bu çalışmadaki problemin çözümü için sezgisel algoritmalar (ing. heuristic algorithms) tercih edilmemiştir. Bunun sebebi, kullanıcıların hareketlerinin düzenli olarak değişmesinden dolayı ağ topolojisinin de değişime uğramasıdır. Bunun sonucu olarak da her değişimde algoritma başa sarılmalı ve her topoloji için tekrar çalıştırılmalıdır. Bu durum çok zaman alıcı bir yöntem olduğu için öğrenme tabanlı bir algoritma olan RL yöntemlerinden QL, problemin çözümünde tercih edilmiştir. Belirli bir alana dağıtılan kullanıcılar, rastgele hareket modeline (ing. random walk model) göre hareket etmektedir ve hareketleri yüzünden hizmet kalitesinde düşüşler meydana gelebilmektedir.

Önceden belirlenen eşik hizmet kalitesi miktarının altında sağlanan hizmetler istenmeyen durumdur. Çalışmada sunulan algoritma ile kullanıcıların pozisyonları tahmin edilerek İHABİ, uygun konuma yönlendirilmiş ve bu sorunun gerçekleşme olasılığı düşürülmüştür. Elde edilen simülasyon sonuçları iki farklı senaryodan oluşmuştur. Birinci senaryoda 19 adet statik baz istasyonu, diğer senaryoda 18 adet statik baz istasyonu ve 1 adet hareketli İHABİ kullanılmıştır. Sonuçlar göstermiştir ki İHABİ içeren senaryo, İHABİ içermeyen senaryoya göre ortalama spektral verimlilikte belirgin bir fark oluşturmuştur. Özellikle 100 saniye sonra, İHABİ içeren sistemdeki verimlilik, geleneksel sisteme göre %100'den daha fazla verimli çalışmıştır.

Chen ve arkadaşları, ortamdaki hareketli kullanıcıların konum değişimlerine uyum sağlayan İHABİ'yi haberleşme ağına ekleyerek kullanıcılara sağlanan hizmet kalitesini arttırmayı çalışma [12]'de ele almışlardır. Bu problemin çözümü için aktör-kritik tabanlı öğrenme algoritması kullanarak İHABİ'nin konumlandırılmasında optimale yakın sonuçlar bulmayı hedeflemişlerdir. Kendilerinden önce yapılan İHABİ konumlandırma çalışmalarının statik kullanıcılar için gerçekleştirildiğini belirten ekip, problemi hareketli kullanıcılar için ele almıştır. Ortam, hareketli kullanıcılar, hareketli İHABİ'ler ve statik baz istasyonundan meydana gelmiştir. İHABİ'ler arası girişim olabilmektedir fakat statik baz istasyonu ile İHABİ'ler farklı frekanslarda çalıştıkları için aralarında girişim söz konusu değildir. İHABİ tarafından kullanıcılara sağlanan bant genişlikleri eşit olarak dağıtılmıştır ve her bir zaman diliminde bir kullanıcıya sadece tek bir İHABİ atanmıştır. İHABİ için uçuş yüksekliği sabit tutulmuştur, dolayısıyla problem 2-boyutlu konum arama problemine dönüştürülmüştür. Çalışmada, [1]'de verilen havadan karaya kanal modeli kullanılmıştır. Bu bilgiler ışığında İHABİ için oluşturulan konum optimizasyonu problemi, karışık tam sayılı konveks olmayan programlama (ing. Mixed-integer non convex programming) haline dönüştürülmüştür. Bu çalışmada İHABİ ve kullanıcıların pozisyonları sürekli değerler almaktadır. Çalışma [18]'de ise önerilen çözümde QL kullanıldığı için değerler ayrıktır ve çalışma kapsamı farklılaşmaktadır. Durum ve aksiyon uzayının sürekli değerler alması sebebiyle aktör-kritik tabanlı algoritma tercih edilmiştir. Durum uzayı İHABİ'lerin ve kullanıcıların konum ve İHABİ'ler ile kullanıcı ilişkilendirilmesi bilgilerinden oluşmaktadır. Aksiyon uzayı, İHABİ'nin bir zaman adımında yatay düzlemdeki koordinat değişimi bilgisinden oluşmaktadır. Son olarak ödül fonksiyonu, mevcut andaki toplam hizmet kalitesi ile bir önceki zamandaki toplam hizmet kalitesi arasındaki farktır. Aktör kritik tabanlı İHABİ yerleştirme problemi iki aşamadan oluşmuştur. İlk aşama olan eğitim sürecinde İHABİ için aktör-kritik tabanlı algoritma koşturularak modellenen sinir ağı (ing. neural network, NN) her yinelemede ağırlıklarını optimal çözüme yakınsayacak şekilde güncellemiştir.

Sinir ağının öğrendiği politika böylelikle her adımda geliştirilmiştir. İkinci aşama ise karar sürecidir. Bu aşamada eğitimi tamamlanmış sinir ağı ile İHABİ, bir bölüm boyunca kullanıcıların hareketlerine göre optimal/optimale yakın aksiyon kararları alarak arzulanan konuma konumlanmıştır. Simülasyonlar, önerilen yöntem dışında karşılaştırma amacıyla üç farklı yöntem ile denenmiş ve sonuçlar incelenmiştir. Bu yöntemler sırasıyla sıralı en düşük kare programlaması (ing. sequential least-squares programming, SLSQP), sezgisel metod ve hareketsiz İHABİ şeklindedir. Elde edilen sonuçlar doğrultusunda önerilen yöntem, İHABİ'nin toplam uçuş süresinin %78'lik kısmında sezgisel yönteme göre, %84'lük kısmında ise SLSQP algoritmasına göre daha yüksek hizmet verimliliği sağlamıştır. Hareketsiz duran İHABİ'ler ise hiçbir zaman diliminde önerilen yöntemden yüksek sonuç vermemiştir. Ayrıca çalışmada, kullanıcılar iki farklı dağılım şekli ile dağıtılarak dört farklı yöntemin performansları incelenmiştir. Hem gaus (ing. gaussian) hem de tekdüze (ing. uniform) dağılım şekilleri için önerilen yöntem diğer yöntemlerden daha iyi performans göstermiştir.

Fotouhi ve arkadaşları yapmış oldukları çalışma [16]'da haberleşme ağının performansını arttırmak için İHABİ kullanmışlardır. Amaç, hareketli kullanıcılara ve dolayısıyla değişen ağın topolojisine uyum sağlayarak kullanıcıya sağlanan hizmet kalitesini maksimize etmektir. Ortam, kapalı kare bir çevreden oluşmaktadır. Ortamda bir adet İHABİ, bir adet hareketli kullanıcı ve bir adet de statik baz istasyonu yer almaktadır. İHABİ, belirlenen sabit bir yükseklik ve sabit hızda uçarak hizmet sağlamaktadır. Kullanıcı, rastgele yol noktası (ing. random way point) hareket modeline göre hareketini gerçekleştirmektedir. Yer baz istasyonu ana ağa iletim kapasitesi limitini sağlamak için konumlandırılmıştır. İHABİ ile kullanıcı arasındaki iletişim yer baz istasyonuna yüklenmiştir. İHABİ ile yer baz istasyonu sabit güçte çalışmaktadır ve çalışma frekansları farklı olduğu için aralarında girişim oluşmamaktadır. Bu koşullar altında İHABİ, otonom olarak kullanıcının hareketlerine adapte olarak RL algoritması ile ağ kapasitesini arttıracaktır. Bir varsayım olarak İHABİ, kullanıcının ve kendisinin pozisyon bilgisini her adımda bilecektir. Önerilen RL modellerinin durum uzayı, İHABİ ve kullanıcının iki boyutlu konumlarından oluşturulmuştur. Aksiyon uzayı, İHABİ'nin sekiz farklı hareketinden meydana gelmiştir. MDP'yi tamamlayan son parça olan ödül fonksiyonu ise iki kısımdan oluşturulmuştur. İlk parça İHABİ'nin o andaki toplam hizmet miktarının kendisidir. İkinci parça ise İHABİ'nin tanımlanan bölgenin dışına çıkması üzerine verilen negatif cezayı temsil etmektedir. Çalışmada QL ve DQN algoritmaları tercih edilmiştir. QL için durum ve aksiyon uzayı kuantalanmıştır. Simülasyon sonuçları çeşitli konuları ele alacak şekilde kapsamlı olarak incelenmiştir. Gözlemlenen sonuçlardan bir tanesi, DQN eğitimi esnasında ceza alınan yineleme sayısının, QL ile eğitilen senaryoya göre daha fazla çıkması olmuştur.

QL'nin istenmeyen durum olan, İHABİ'nin tanımlanan bölgenin dışına çıkması sorununu daha erken kavradığını göstermiştir. Öte yandan sistemdeki toplam hizmet miktarları kıyaslandığında DQN algoritmasının sonucunun, QL algoritmasına kıyasla daha yüksek olduğu görülmüştür. Bunun ana sebebi, DQN algoritmasının ayrık değerler yerine sürekli değerler için çözüm üretmesi olmuştur. Çalışmada incelenen diğer bir parametre ise İHABİ'nin kullanıcılara olan ortalama yakınlık miktarıdır. DQN ile elde edilen yakınlık miktarı, QL ile elde edilenden %20 daha az çıkmıştır. Bu sonuç ilk etapta QL'nin daha iyi iş çıkardığını gösterse de sistem performansını erişim bağlantısı (ing. access link) ve ana ağa iletim bağlantısı ortak olarak belirlemektedir. Dolayısıyla DQN hem erişim bağlantısı hem de ana ağa iletim bağlantısı kısıtları ile İHABİ için daha iyi konumlar belirleyerek toplam hizmet kalitesini daha iyi noktaya taşımıştır. Çalışmada ayrıca, öğrenme tabanlı algoritmalar dışında iki farklı yöntem daha ele alınarak karşılaştırma yapılmıştır. Bunlardan bir tanesi tek atlamalı açgözlü (ing. 1-hop greedy), diğeri ise çift atlamalı açgözlü (ing. 2-hop greedy) modellerdir. Tek atlamalı açgözlü senaryoda yer baz istasyonunun bulunmadığı durum ele alınıp hizmet kalitesi maksimize edilmeye çalışılmıştır. Çift atlamalı açgözlü senaryo, öğrenme algoritmaları için oluşturulan senaryo ile aynıdır. Karşılaştırılan yöntemlerin sonucu olarak DQN algoritması en iyi sistem performansını sağlamıştır.

Çalışmamızın ilk bölümünde ([43]), bir mobil İHABİ, bir statik yer baz istasyonu ve rastgele dağıtılmış kullanıcıları içeren bir senaryoyu inceliyoruz. Diğer çalışmalardan ([13], [16] ve [6]) farklı olarak, her kullanıcının farklı QoS talep ettiği durumu ele alıyoruz. Ayrıca İHABİ farklı irtifalarda uçabilmektedir. İrtifa değişikliği yapabilme özelliği sayesinde sınırlı kapsama alanında değişiklik yaparak hizmet verdiği kullanıcı sayısını ayarlayabilmektedir. [54]'deki kısıtlamalara ek olarak, gerçek dünya koşullarını taklit etmek için İHABİ ve statik yer baz istasyonu arasındaki ana ağa iletim kapasitesini de dikkate alıyoruz. Bildiğimiz kadarıyla bahsi geçen bütün bu kısıtların birlikte ele alındığı çalışmayı ilk biz yapıyoruz. Bu koşullar altında İHABİ'nin uçuş sırasında kullanıcılara sağladığı veri hızının toplamını maksimize etmeyi hedefliyoruz. İHABİ, güncellenen Q-tablosu ile kullanıcıların bulunduğu bölgeyi keşfedecek ve en uygun güzergahı belirleyecektir.

Çalışmamızın ikinci bölümünde, hareketli kullanıcıların bulunduğu ortamda hareket eden İHABİ, maksimizasyon ve minimumun maksimizasyonu problemlerine göre kullanıcılara sağlanan veri iletim hızını uygun güzergâh tercihi ile istenen şekilde sağlamaktadır. Bildiğimiz kadarıyla, hareketli kullanıcıların bulunduğu senaryoda İHABİ için veri iletim hızındaki adilliği sağlayan minimumun maksimizasyonu problemine literatürde değinilmemiştir.

Bu problem, maksimizasyon problemi ile kıyaslanarak İHABİ'deki güzergâh değişimleri incelenmiş ve karşılaştırılmıştır. Çalışmamızın bu bölümünde DQN kullanılarak İHABİ'nin değişen ortam koşullarına adapte olması ve böylelikle uygun güzergâhları hesaplaması hedeflenmiştir.

2. METODOLOJİ

2.1 Makine Öğrenmesi

Makinelere düşünme özelliği kazandırmak kolay bir problem değildir. Bu özelliği kazandırmak ve insanlar gibi düşünebilmesini sağlamak ML'den geçmektedir. "Makine Öğrenmesi" terimini ilk kez Arthur Samuel 1959 yılında ortaya atmıştır [42]. ML, yapay zekanın (ing. Artificial Intelligence, AI) bir alt alanıdır ve geçmişteki tecrübelerden yola çıkarak, herhangi bir programlama gerektirmeden verilerden öğrenen bir yapıdan oluşur. Öğrenilen veriler arasındaki örüntüleri anlayarak gelecek ile ilgili tahminlerde bulunur. Hesaplama yeteneği oldukça kuvvetlidir ve normal bir insandan çok daha hızlı hesaplamalarda bulunur. Çalışma mantığı ise basittir:

ML için oluşturulan modele girdi olarak ulaşılmak istenen hedef ve bu hedef ile ilişkili öz nitelik (ing. feature) verilir. Öğrenme modeli birtakım yineleme sonucunda öz nitelik ile hedef arasında ilişki kurar ve öğrenme tamamlanır. Bu yöntem geleneksel programlama yöntemlerinden biraz farklıdır. Geleneksel programlama yönteminde ise sistemin çalışma şekli ve sisteme girdi, kullanıcı tarafından sağlanır ve istenen çıktı elde edilir. ML'deki öğrenme kavramı, girdi çıktı arasındaki ilişkinin bilgisayar tarafından kullanıcı müdahalesi olmaksızın kendiliğinden kavranmasını ifade etmektedir ve bu yönüyle geleneksel programlamadan farklıdır.

ML'nin geçmişi 1940'lı yıllara dayanmaktadır. Bu alanda atılan ilk adım Walter Pitts ve Warren McCulloch'un yapmış olduğu çalışma [32]'de sinir ağlarının matematiksel modelinin çıkarılmasıyla başlamıştır. Daha sonra 1950 yılında Alan Turing yayınladığı çalışma [50] ile makinenin de insanlar gibi tepkiler vererek bir insanı kandırmasını ele almıştır. Diğer bir gelişme ise 1957 yılında Frank Rosenblatt'ın [39] algılayıcıyı (ing. perceptron) tasarlamasıyla yaşanmıştır. Bu alandaki en büyük atılımlar 1900'lerin sonları ve 2000'li yılların başında meydana gelmiştir. 1986 yılında Geoffrey Hinton ve ekibinin geri yayılım (ing. backpropagation) algoritmasını [40] ele almasıyla ML modelleri bu algoritmayla öğrenimini gerçekleştirmiştir. Yine 2006 yılında Geoffrey Hinton ve arkadaşlarının yaptığı çalışma [21] ile bilgisayarlara görüntülerdeki objeleri ayırabilme yeteneği kazandırılmıştır.

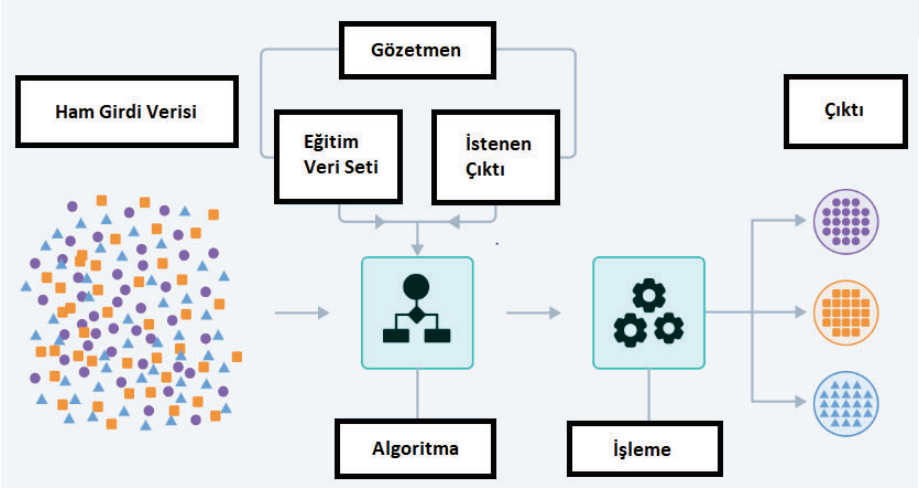
ML, hayatımızda birçok yerde karşımıza çıkmaktadır. Dolandırıcılık tespiti, yüz tanıma, hastalık tanısı, otonom araçlar kullanılan alanlardan sadece birkaçıdır.

Çeşitli alanlar için karşılaşılan problemler de çeşitli olabilmektedir. Dolayısıyla kullanılan algoritmalar da farklı tiplerde olmaktadır. Denetimli öğrenme (ing. supervised learning), denetimsiz öğrenme (ing. unsupervised learning), yarı denetimli öğrenme (ing. semi-supervised learning) ve pekiştirmeli öğrenme ML'de kullanılan algoritmaların yer aldığı başlıca kategorilerdir.

2.2 Denetimli Öğrenme

Denetimli Öğrenme, eşleştirilmiş bir girdi-çıktı örneklerine dayalı olarak bir sistemin girdi-çıktı ilişkisi bilgilerini elde etmeye yönelik bir ML tekniğidir [31]. Bu teknik ile bir dizi girdi değişkeni ile bir çıktı değişkeni arasındaki ilişki öğrenilir ve bu ilişki görünmeyen veriler için çıktıları tahmin etmede kullanılır [15].

Denetimli öğrenmede kullanılan algoritma, belirli bir girdi için elde edilen çıktının etiketlenen çıktıyı vermesine kadar yinelemeli olarak devam eder. Öğrenme esnasındaki bu hata miktarının sıfırlanması hedeflenir. Öğrenme tamamlandığında algoritmanın performansı, eğitim esnasında kullanılmayan bir veri seti ile test edilir. İstenen başarı oranı kıstası sağlanmadıysa farklı parametrelerle ve veri seti ile eğitim tekrar yapılabilir. Şekil 2.1'de [5] denetimli öğrenme döngüsü gösterilmiştir.



Şekil 2.1: Denetimli öğrenme döngüsü.

Denetimli öğrenme algoritmaları, hedeflenen çıktının tipine göre iki ana kategoride incelenebilir. Bunlar "Regresyon" [38] ve "Sınıflandırma" [26] problemleridir.

Regresyon problemleri, bağımlı bir değişken ile bir ya da daha fazla bağımsız değişken arasındaki ilişkiyi belirlemek ve bilinmeyen ya da gelecekteki durumlarda kestirim yapmak için tercih edilir. Sahip olduğu bağımsız değişken sayısına göre şekillenir.

Amaç, eldeki veri setinin dağılımına en uygun eğriyi bularak minimum kare hatasını minimize etmektir. Bu problemlerdeki performans metriği olarak R-kareler (ing. R-squared) metodu kullanılır. R-kareler metodu, eğrinin verilere ne kadar iyi uyum sağladığını gösteren bir ölçüttür.

Sınıflandırma problemleri, yeni gözlemlerin kategorisini belirlemek için kullanılır. Verilen bir veri kümesinden veya gözlemlerden öğrenim gerçekleştirilir ve ardından yeni gözlem bir dizi gruba sınıflandırılır. Problemin çıktı sayısına göre ikili sınıflandırma veya çoklu sınıflı sınıflandırma olarak ikiye ayrılır. Amaç, eldeki veri setini ayıran en uygun eğrileri bularak verileri sınıflara bölmektir. Bu problemlerde performans metriği olarak karışıklık matrisi (ing. confusion matrix) tercih edilir. Karışıklık matrisi, doğru tahmin ve yanlış tahmin sayılarını matris formatında içerir.

Gerçek dünyadaki birçok problemlerle başa çıkmada denetimli öğrenme, güçlü bir araç olarak ortaya çıkmıştır fakat denetimli öğrenmede karşılaşılan birtakım zorluklar vardır. Verilerin insan hatasından kaynaklı olarak yanlış oluşturulması ve bu sebepten dolayı algoritmanın yanlış öğrenmesi, verilerin çok büyük olduğunda sınıflandırmanın zorlaşması ve modellerin eğitiminin uzun zaman alabilmesi bunlara örnek verilebilir. Elbette ki bu zorluklar, doğru tekniklerle azaltılabilmektedir.

Denetimli öğrenme kendine farklı konularda kullanım alanı bulmuştur. İstenmeyen e-posta algılama, konuşma tanıma, sağlıkta hastalık tanısı ve ilaç sınıflandırma bunlara örnek olarak verilebilir.

2.3 Yarı Denetimli Öğrenme

Yarı denetimli öğrenme, sistemlerin hem etiketli hem de etiketsiz verilerin varlığında verilerden öğrenmeyi ele alır. Başka bir ifadeyle, denetimli ve denetimsiz öğrenme arasında yer alan bir tür makine öğrenimidir. Yarı denetimli öğrenmenin amacı, etiketli ve etiketsiz verilerin birleştirilmesinin öğrenme davranışını nasıl değiştirebileceğini anlamak ve bu doğrultuda algoritmalar geliştirmektir [57].

Yarı denetimli öğrenmenin çalışma mantığı şu şekildedir:

- Etiketli sınırlı veriler ile model eğitilir.
- Etiket içermeyen veriler eğitilen modelden geçirilerek sahte (ing. pseudo) etiketler elde edilir.

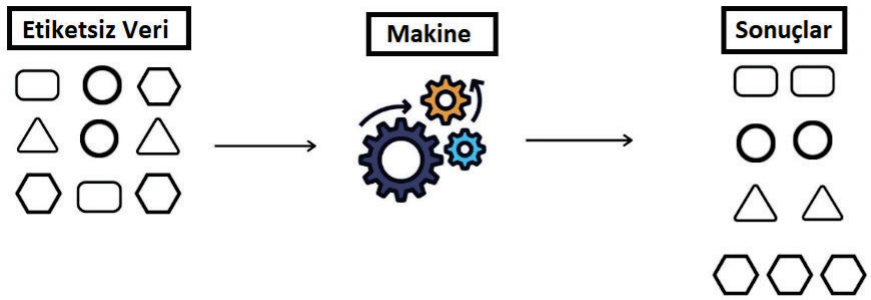
- Sahte etiketli verilerden güvenilir olanlar tercih edilir.
- Orijinal etiketli veriler ile güvenilir sahte etiketli veriler birleştirilerek yeni veri seti oluşturulur ve model tekrar eğitilerek performansı arttırılır.

Veri miktarının sürekli artması denetimli öğrenme için problem oluşturmaktadır, çünkü bu durum etiketleme için harcanan iş gücünü arttırmaktadır. Yarı denetimli öğrenme, veri sayısının artması durumunda ön plana çıkmaktadır.

Ses ve video analizi, internetteki içeriklerin sınıflandırılması, protein dizisi sınıflandırılması gibi çalışmalar yarı denetimli öğrenmenin kullanıldığı yerlerdir.

2.4 Denetimsiz Öğrenme

Denetimsiz öğrenmede veriler sadece girdilerden oluşur, girdilere ait etiketler bu öğrenme modelinde yer almaz [17]. Kullanılan algoritmalar, eldeki tek bilgi olan girdi verilerini inceleyerek bu veriler arasındaki gizli kalıpları veya grupları öğrenmeye çalışır. Şekil 2.2 [37] denetimsiz öğrenme döngüsünü göstermektedir.



Şekil 2.2: Denetimsiz öğrenme döngüsü.

Denetimsiz öğrenme modelleri kümeleme, boyut azaltma ve ilişki kurma gerektiren problemlerde tercih edilir. Anomali tespiti, tıbbi görüntüleme ve tavsiye motorları, denetimsiz öğrenme algoritmalarından yararlanan başlıca konulardır.

2.5 Pekiştirmeli Öğrenme

Pekiştirmeli Öğrenme, önceden çevresel bilgilere ihtiyaç duymadan kendi kendine öğrenmeye dayalı bir ML yöntemidir [47]. Öğrenme süreci, çevre ve ajan arasındaki periyodik etkileşimlere dayanır. Bu süreçte, ajan çevresiyle etkileşime girer ve kalan aksiyonlarla toplam ödülünü en üst düzeye çıkaran doğru aksiyonu almayı öğrenir [24].

MDP özelliğine sahip bir ortam için, gelecekteki durum ve beklenen ödül, mevcut durum ve alınan aksiyon tarafından tahmin edilebilir. Bu durum ajanın herhangi bir durumda en iyi aksiyonu almasına yardımcı olur.

Pekiştirmeli öğrenmenin gelişimi üç ana unsurdan meydan gelmiştir [47]:

- Deneme-yanılma yoluyla öğrenme
- Optimum kontrol problemi
- Zamansal fark (ing. Temporal difference) öğrenme yöntemleri

Deneme-yanılma yoluyla öğrenme üzerine yapılan çalışmaların temeli 1900'lü yılların başlarına dayanmaktadır. 1911 yılında Thorndike [49], belirli bir durum için verilen tepkilerde iyi sonuçların daha sık tekrarlanabileceğini, kötü sonuçlarda ise bu durumun tam tersinin gerçekleştiğini "Etki Yasası" olarak ifade etmiştir. Bu konsept için 1954 yılında Minsky'nin [34] bir sinir ağı tasarlamasıyla, bu konsept ML alanına taşınmıştır. Yine 1961 yılında Minsky [33], kredi atama problemini ele alarak RL'nin temellerini ele almıştır. Çalışmasında kredinin birçok seçenek arasında nasıl dağıtılacağını incelemiştir. 1963 yılında Andreae [4], STELLA sistemini tasarlayarak çevre ile etkileşim halinde bulunmayla öğrenmeyi gerçekleştirmiştir.

Optimum kontrol problemi üzerine yapılan çalışmalar 1800'lü yıllara kadar dayansa da bu alandaki önemli atılımlar Bellman'ın yaptığı çalışma [8] ile dinamik programlamanın tanıtılmasıyla başladı. Çalışmada optimal kontrol problemlerinin tanıtılan yöntem ile çözümü ele alındı. Buradaki önemli etmen, çözüm için oluşturulan fonksiyonun dinamik sistemin durumunu kullanarak optimal bir değer fonksiyonu (ing. value function) üretmesidir. Bu optimal değer fonksiyonu Bellman denklemi olarak geçer ve RL'nin temel hesaplamasında kullanılır. 1957 yılında Bellman [7], MDP'yi tanımlayarak yeni bir duruma geçişin yalnızca mevcut duruma ve bu durumda alınan aksiyona bağlı olduğunu ifade etmiştir. Howard [22], MDP'nin çözümü için politika iterasyonu (ing. policy iterasyon) metodunu önermiştir.

Zamansal fark öğrenme yöntemi, her adımda tahmin üretir. Minsky [33], 1961 yılında ele aldığı çalışmayla zamansal fark yönteminin RL'deki öneminden bahsetmiştir. 1972'de Klopf [25], deneme-yanılma öğrenme sürecini zamansal fark metodu ile birleştirmiştir.

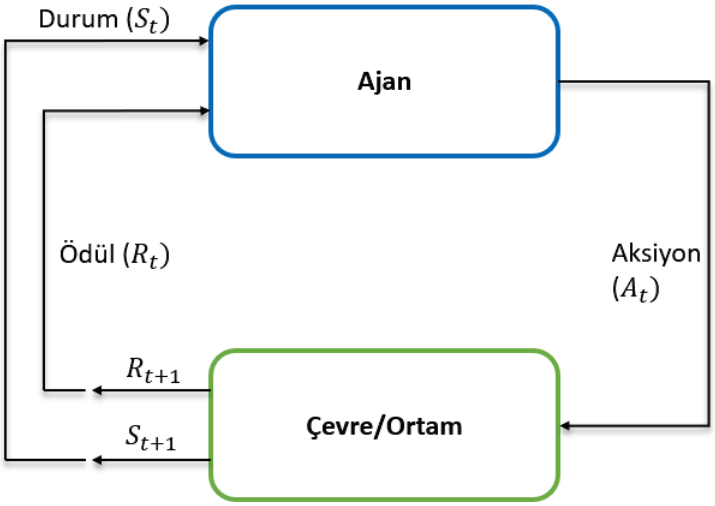
1992 yılında bu ana unsurları ele alan ve RL alanında önemli ses getiren, Watkins ve Dayan'ın yaptığı çalışma [52] ile MDP kullanılarak ve ayrık değerler içeren problemler için QL algoritması tanıtılarak günümüz RL'nin temelleri atılmıştır. QL'deki en büyük eksiklik, durum ve aksiyon uzayının büyük olduğu durumlarda hesaplama maliyetinin artması ve öğrenmedeki performansın düşmesidir. 1996 yılında Bertsekas and Tsitsiklis'in [9] yapmış olduğu çalışma, sinir ağları ile yaklaşık dinamik programlama yöntemini ele almış ve bu yöntemle QL algoritmasında yaşanan kısıtlamanın önüne geçilmiştir. 2015 yılında Google DeepMind'ın ele aldığı çalışma [35] ile derin sinir ağları tabanlı DQN mimarisi tanıtılarak dinamik programlama metodlarına yaklaşım sağlanmıştır. Bütün bu son gelişmeler, araştırmacıların bu alanda yaptığı çalışmaların artışını sağlamıştır.

Bir RL modelinde ajan, önceden tanımlanan durumlardan birinde olabilir. Ajanın her durumda ödülünü en üst düzeye çıkarmak için en iyi aksiyonu öğrenmesi gerekir [47]. Bunu başarmak için ajan, farklı aksiyonları denemekten ve bunlardan ders çıkarmaktan sorumludur. Bu aksiyonlar, ajan tarafından önceden bilinmeyen bir ortamda gerçekleştirilir.

Eğer bir RL modeli, bir parametre dizisi ile ifade edilirse $(< S, A, p, r, \gamma >)$:

- S (Durum): S, durum uzayını gösterir ve ajan t zamanında $s_t \in S$ durumundadır.
- A (Aksiyon): A, aksiyon uzayını gösterir. Ajan t zamanında s_t durumundayken $a_t \in A$ aksiyonunu yapar.
- p (Olasılık): Geçiş olasılığı (ing. transition probability) dağılımıdır. Mevcut $s_t \in S$ durumunda bulunan ajanın $a_t \in A$ aksiyonunu yaptığında $s_{t+1} \in S$ durumuna geçiş olasılığını temsil eder. Matematiksel olarak $p(s'|s,a) = P[s_{t+1} = s'|s_t = s, a_t = a]$ şeklinde ifade edilir. Çalışmamızdaki problem deterministtik olarak ele alındığı için $p(s'|s,a) = 1$' dir.
- r (Ödül Fonksiyonu): Ödül fonksiyonudur. Mevcut $s_t \in S$ durumunda bulunan ajanın $a_t \in A$ aksiyonunu yaparak $s_{t+1} \in S$ durumuna geçtiğinde alacağı ödül miktarını belirler.
- γ (İndirim Faktörü): İndirim faktörüdür. Bu faktör, anlık ödüller ile gelecekteki ödüller arasındaki önem dengesini kontrol eder. $\gamma \in [0,1)$ aralığında sürekli değerler alır. Bu değerin 0'a doğru gitmesi gelecekteki ödüllere verilen önemi azaltırken 1'e doğru gitmesi gelecekte karşılaşılacak ödüllerin daha çok önemsendiğini belirtir.

Şekil 2.3'de RL modeli çalışma döngüsü yer almaktadır:



Şekil 2.3: Pekiştirmeli öğrenme modeli çalışma döngüsü.

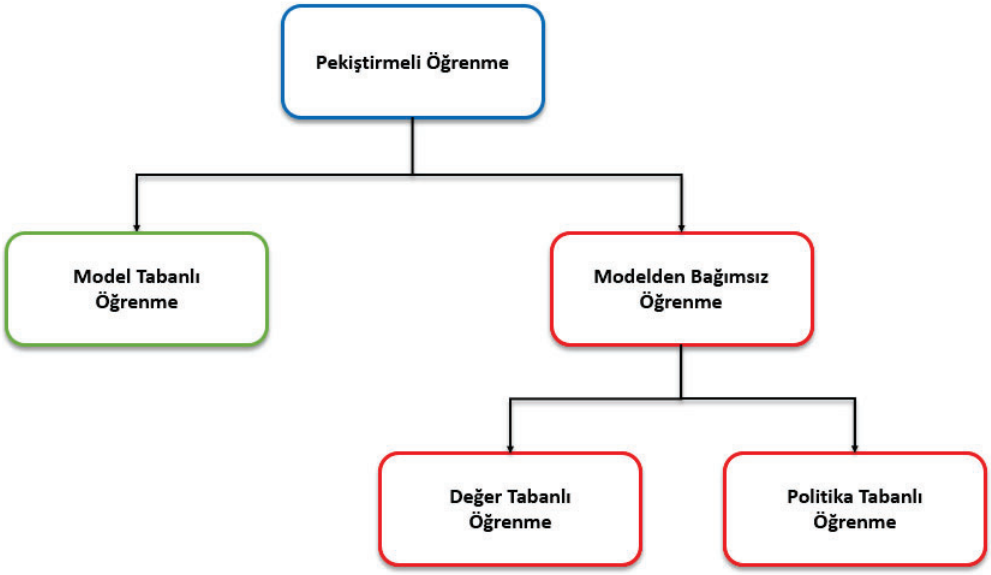
Bir RL'deki çalışma akışı şu şekildedir:

- Çevre/Ortam hazırlanır. Ortam, gerçek bir fiziksel sistemi veya benzetilmiş bir ortamı temsil edebilir.
- Uygun ödül fonksiyonu tanımlanır. Ajan için bir performans metriğidir ve ajanın aldığı aksiyonların kalitesini, ulaşılmak istenen hedefe göre değerlendirir. Ajanın aldığı aksiyonların doğruluğu/kalitesi, sağlanan uygun ödüllere doğrudan bağlıdır. Bu fonksiyon, RL'deki en önemli unsurlardan biridir.
- Uygun ajan, problem tipine göre seçilir. RL için oluşturulan problemler durum ve aksiyon uzayının sürekli ve ayrık olmasına göre farklılık gösterebilir. Bu farklı problemler için farklı algoritmalar mevcuttur. Uygun algoritma ile optimal politikayı takip etmesi hedeflenen ajan tanımlanır.
- Tanımlanan ajan, tanımlanan ödül fonksiyonu ve ortam eşliğinde eğitilir. Ajanın takip edeceği politikanın zamanla geliştirilmesi ve optimale yakınsaması için eğitim gerçekleştirilir ve ajanın öğrendiği politikanın performansı test edilir.
- Test edilen modelin beklentileri sağlayıp sağlamadığı değerlendirilir. Eğer model bu konuda yetersiz kalıyorsa birtakım parametre gözden geçirilir. Bu parametrelere ödül fonksiyonu, model parametreleri, çevresel değişkenler örnek verilebilir.

RL, denetimli öğrenmeye benzer şekilde girdi ile çıktı arasında bir ilişki kurar fakat bu ilişkiyi kurma şekli denetimli öğrenmeye göre farklılık gösterir. Hazır olarak eşleştirilmiş girdi çıktı birlikteliğinden farklı olarak RL'de, ortamla kurulan etkileşimler ile hesaplanan doğru çıktı, etkileşim esnasında alınan ödül ve cezalara göre belirlenir. Ayrıca kararlar belirli bir sıra ile verildiğinden dolayı, RL'de girdiler arasında bir ilişki vardır.

RL, otonom araçlar, robotik, video oyunları, finans sektörü dahil olmak üzere birçok gerçek hayat senaryosu ve uygulaması için faydalıdır. Kendi kendine oynamayı öğrenerek GO oyununda dünya şampiyonunu yenen AlphaGo Zero [45], endüstriyel otomasyon ve robotik sektörü için robotların istenen şekilde kontrol edilmesi [19], finans portföyünün optimal şekilde yönetilmesi [23], RL alanında yapılmış çalışmalardan sadece birkaçıdır.

RL algoritma sınıfları Şekil 2.4' deki gibi üç kategoride incelenir:



Şekil 2.4: Pekiştirmeli öğrenme algoritma sınıfları.

- Politika tabanlı (ing. Policy based): Politika tabanlı yaklaşımda, herhangi bir değer bilgisi kullanılmadan alınan ödül miktarını maksimize edecek politika, doğrudan optimize edilmektedir [36]. Diğer bir deyişle, eğitim için kullanılan algoritma belirli bir durum için optimal aksiyon çıktısı verir (Optimal çıktı için optimal politikanın öğrenilmesi gerekir). Politikalar rastlantısal (ing. stochastic) ve deterministtik olmak üzere ikiye ayrılır:

  - Rastlantısal politikalarda belli bir $s_t \in S$ durumu için alınan $a_t \in A$ aksiyon kararı belirli bir olasılıkla gerçekleşir.
  - Deterministtik politikalarda belli bir $s_t \in S$ durumu için alınan $a_t \in A$ aksiyon kararı değişmez.
  - REINFORCE [53] algoritması politika tabanlı sınıfta yer alır.

- Değer tabanlı (ing. Value based): Değer tabanlı yaklaşımda yinelemeler ile, maksimum değer politikası altında optimal değer fonksiyonu aranır. Optimal değer fonksiyonu, karar vericinin deneyiminden kademeli olarak oluşturulur [48]. QL [52], DQN [35] ve SARSA [41] algoritmaları bu sınıfta yer alır.

- Model tabanlı (ing. Model based): Model tabanlı yaklaşımda, eğitilecek olan ajan, sanal bir modelden oluşan ortam/çevre ile etkileşim halinde bulunarak öğrenimini gerçekleştirir. Bu yaklaşımda dikkat edilmesi gereken nokta, modellemede yapılan hatanın ajanın öğrenimini de etkilemesidir [51].

Son zamanlarda modelden bağımsız öğrenme sınıfı için geliştirilen algoritmalar hem politika tabanlı yaklaşımı hem de değer tabanlı yaklaşımı birlikte ele almaktadır. Aktör-kritik metotları buna örnektir ve Kritik ve Aktör olarak iki ana bileşenden oluşur.

- Kritik NN, değer fonksiyonunu tahmin eder. Bu ağ, ajan tarafından alınan aksiyonunun kalitesini belirler.

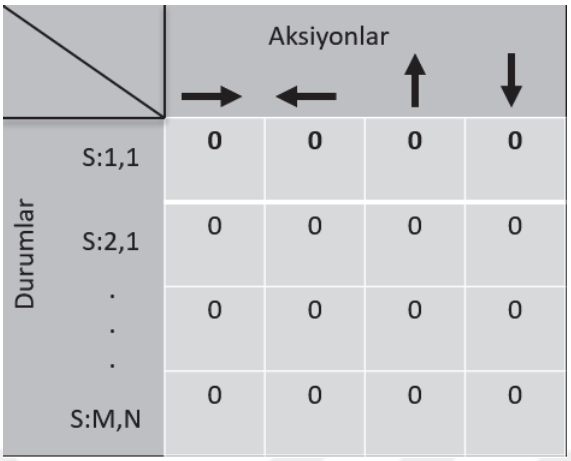
- Aktör NN, ajanın nasıl hareket edeceğini belirler. Bu ağ, kritik ağından gelen değer çıktısına göre politikasını iyileştirecek şekilde günceller.

Hibrit yapıyı kullanan bu algoritmalara DDPG [30] ve SAC [20] örnek olarak verilebilir.

2.5.1 Q-Öğrenme

QL algoritması, politika dışı RL algoritması olup, zamansal fark yöntemini kullanarak genel optimum (ing. Global optimum) çözümünü arayan bir metottur. Politika dışı kategorisine girmesinin sebebi, ajanın eğitimi için takip edilen politika ile ajanın aksiyon almak için kullandığı politikanın farklılık göstermesidir. Zamansal fark yöntemini kullanarak her adımda tahminler üretmektedir. QL modeli, durum, aksiyon, ödül ve Q-değerinden oluşur.

QL, çevre ile etkileşime giren ajanın davranışını (politika), zaman içinde aldığı toplam ödül miktarını artıracak şekilde optimize etmeyi amaçlar. Bu amaç için bir matris ya da tablo yapısından yararlanır [46]. Örnek bir Q-tablosu şekil 2.5'de verilmiştir:



Şekil 2.5: Q-Tablosu.

Q-Tablosu, oluşturulan QL modelindeki durumların türüne göre iki veya çok boyutlu olabilir. Şekil 2.5, iki boyutlu QL modeline ait bir Q-Tablosunu göstermektedir. Tabloda bir durum değişkeni için ajanın bulunabileceği M sayıda alternatif ve ajanın uygulayabileceği N adet aksiyon bulunmaktadır. Q-tablosundaki her bir değer, optimal politikayla o durumda o aksiyon gerçekleştirildiğinde gelecekteki beklenen maksimum ödül miktarını ifade eder. Her bir zaman adımında tablodaki değerler güncellenerek mevcut politika iyileştirilir.

Bir t anında, her bir (s_t, a_t) çifti için bir durum-aksiyon değeri fonksiyonu veya Q-fonksiyonu $Q(s_t, a_t)$ tanımlanır. Belirli bir durum için ajan, o durumdaki tüm aksiyonların Q-fonksiyonu değerlerini karşılaştırarak hangi aksiyonun gerçekleştirileceğini belirler. Belirli bir durum-aksiyon çifti için beklenen gelecekteki ödül, $t+n$ zamanında bölümün sonuna kadar denklem 2.1'deki π politikası kullanılarak hesaplanır [47]:

$$Q^{\pi}(s,a) = E_{\pi}[\sum_{k=0}^{n-1} \gamma^k r_{k+t+1} | s_t = s, a_t = a] \tag{2.1}$$

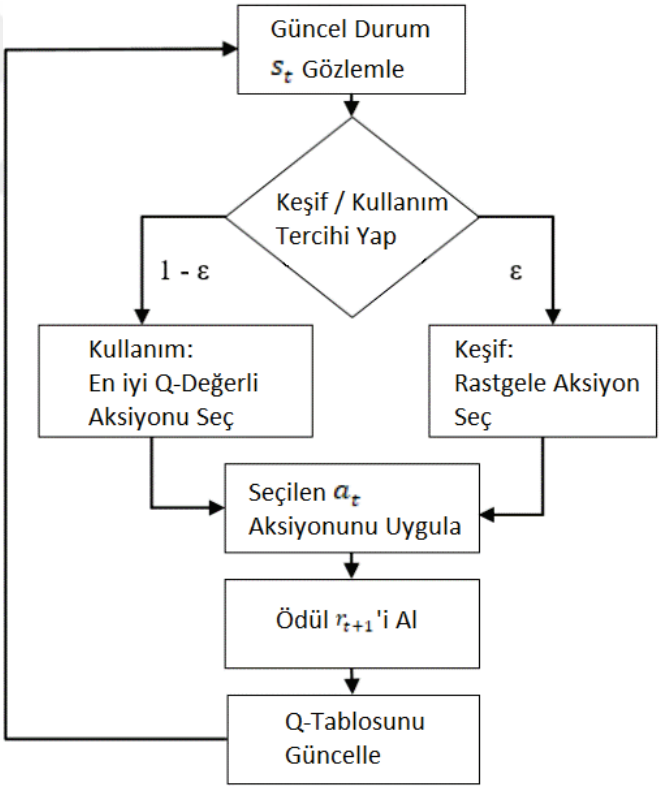
Optimal sonuçlar $Q^{\pi^*}(s,a)$'ları üreten Q-fonksiyonu verildiğinde, ajan tarafından alınan aksiyonlar açgözlü bir şekilde seçilerek en uygun politika (π^*) türetilir [6]:

$$\pi^*(a|s) = \underset{a}{\operatorname{argmax}}\, Q^{\pi^*} \tag{2.2}$$

Q-fonksiyonunun en uygun değerleri yinelemeli olarak hesaplanarak en uygun politika bulunur. Bellman denklemine göre, Q-fonksiyonu öğrenme sürecindeki her yinelemede şu şekilde güncellenir:

$$Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha[r_t + \gamma \max_{a'} Q^{\pi}(s_{t+1}, a') - Q^{\pi}(s_t, a_t)] \tag{2.3}$$

Burada α öğrenme oranıdır (ing. learning rate) ve $\alpha \in [0, 1]$ aralığında değerler alır. Ortamı tanımayan ajan, önce her bir durum-aksiyon çiftini rastgele deneyerek ortamı keşfeder ve çıkarımlarda bulunur. Her keşif adımında, Q-fonksiyonunu günceller. Ajan, bir süre sonra güncellenen Q-fonksiyonunu kullanarak aksiyon almaya başlar. Belirli sayıda yinelemeden sonra QL algoritması, her bir durum-aksiyon çiftinin yeterli sayıda ziyaret edildiği varsayımı altında en uygun politikaya yakınsar ve öğrenme bu noktada sona erer. Algoritmanın akışı Şekil 2.6'da [11] verilmiştir.

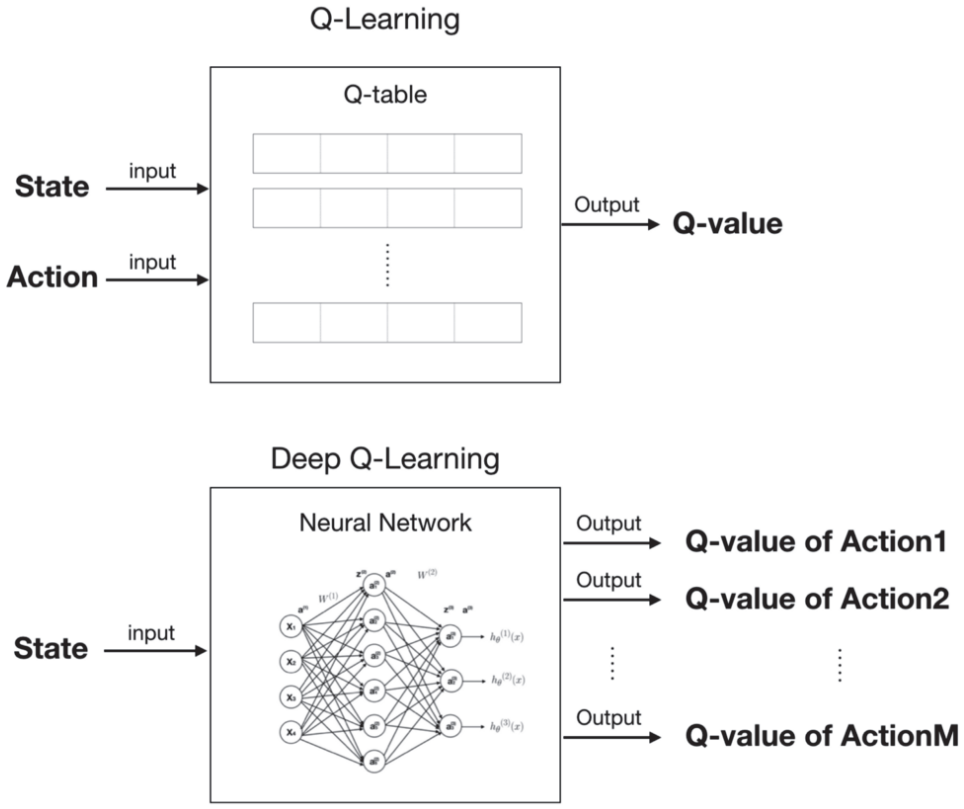


Şekil 2.6: Q-Öğrenme algoritma akışı.

2.5.2 Derin Q-Öğrenme

QL, oldukça kullanışlı bir RL algoritmasıdır ve literatürdeki birçok çalışmada kendine yer bulmuştur. Her ne kadar pratik çözümler üretse de, kullanım alanı belirli bir noktaya kadar sınırlıdır. Şekil 2.5'deki Q-Tablosundan da görülebileceği gibi, oluşturulan tablonun boyutları sınırlıdır. Bu da problemdeki durum ile aksiyon uzayının ayrık ve sınırlı değerler almasını gerektirir.

DQN, QL'nin yetersiz kaldığı noktalarda üstün performans gösteren bir algoritmadır. RL'nin derin sinir ağları ile birleştirilerek bir üst seviyeye çıkarılmış, bu ağlar üzerinden tahminlerde bulunan bir yapısı olarak göze çarpar [35]. Şekil 2.7'de [3] bu görülebilir.



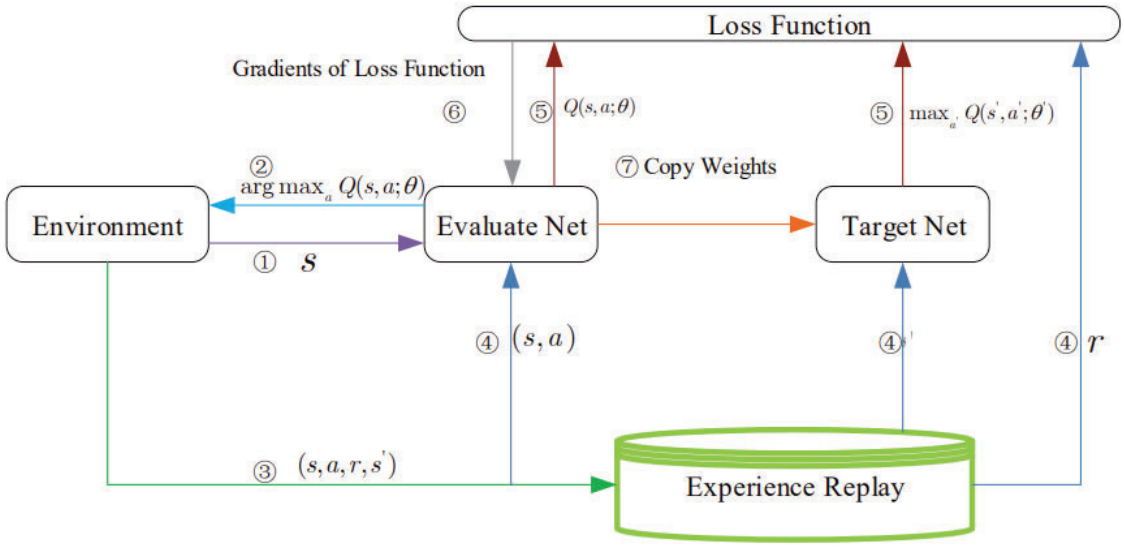
Şekil 2.7: Q-Öğrenme ile Derin Q-Öğrenme model tahmin yapısı.

QL'den farklı olarak DQN, model girdisi olarak sadece durum bilgisini alır ve modelin ağırlık parametrelerini güncelleyerek en uygun politikaya yakınsamaya çalışır. Modelin çıktısı tek bir Q-değerinden oluşmaz, verilen durum girdisi için tanımlanan bütün aksiyonlara Q-değerleri üretir. En yüksek Q-değerine sahip olan aksiyon ajanın tercihi olur. Göze çarpan diğer bir fark ise DQN'ye özgü eklenen iki yeni yapıdır [35]:

- Deneyim Tekrarı (ing. Experience Replay)
- Hedef Ağ (ing. Target Network)

Deneyim Tekrarı, ajanın çevre ile etkileşim halindeyken bulunduğu durumların, aldığı aksiyonların ve bunlar karşılığında aldığı ödüllerin ve geçiş yaptığı yeni durumların bilgisinin bir hafızaya kaydedilmesini kapsar. Bu bilgiler belirlenen kapasitedeki bir hafızada <Durum, Aksiyon, Ödül, Sonraki Durum> formatında kaydedilir ve hafıza doldukça en eski veriler hafızadan silinir. Eğitim esnasında kaydedilen bu örnekler rastgele ve eşit olasılıkla ya da belirli bir öncelik sırasıyla [44] parçalar halinde seçilerek sinir ağı modelinin eğitimi sağlanır. Buradaki en önemli nokta, örnekler arasındaki ardışık yapının bozularak korelasyonun düşürülmesi ve daha iyi bir örnekleme verimliliğinin sağlanmasıdır.

Hedef Ağ, DQN algoritması için önerilmiş ikinci bir hesaplayıcı sinir ağıdır. Algoritmanın her örnek için ulaşılmak istenen değer ile tahmin değerini aynı sinir ağını kullanarak hesaplaması halinde bir istikrar problemi oluşur [35]. Bu problemin önüne geçmek için ikinci bir sinir ağı olarak "Hedef Ağ" algoritmada yer alır. Böylelikle tahmin değeri hesabı ve hedef değer hesabı farklı sinir ağlarında hesaplanarak eğitim daha istikrarlı bir hale gelir. Şekil 2.8'de [29] DQN algoritması özetlenmiştir.



Şekil 2.8: Derin Q-Öğrenme algoritma şeması.

Bu algoritmada gerçekleşen adımlar sırasıyla şu şekildedir:

1. Ajanın bulunduğu çevreden $s_t \in S$ durum bilgisi alınır.

2. Tahmin Ağı çevreden alınan durum bilgisine karşılık bir Q-değeri üretir. Bu değer mevcut durum içerisinde alınabilecek aksiyonlar arasından en iyi Q-değerine sahip olana aittir.

3. Ajan, en iyi Q-değerine sahip aksiyonu alındıktan sonra çevreden ödül ve bir sonraki durum bilgisi elde edilir. Bu parametreler bir arada deneyim hafızasına kaydedilir.

4. Deneyim hafızasından rastgele örnekler seçilerek tahmin ağına ve hedef ağına girdi olarak verilir. Buradaki örnekler, iki sinir ağının eğitimi için kullanılmaktadır.

5. Tahmin Ağı ile Hedef Ağının sonuçları arasındaki fark kayıp fonksiyonunu oluşturur. Hedef ağı, bir sonraki durum ve ona ait en iyi aksiyon çiftini üretir.

6. Tahmin Ağı, belirtilen kayıp fonksiyonunun (algoritma 2) gradyanına göre kendi ağırlık parametrelerini günceller.

7. Hedef Ağı kendini eğitmez, sadece belirli periyotlarla tahmin ağının mevcut ağırlık değerlerini kendine kopyalar.

3. İHABİ SİSTEM MİMARİSİ VE PEKİŞTİRMELİ ÖĞRENME MODELİ

Bu bölümde İHABİ için veri iletim bazlı maksimizasyon problemimizin içeriği detaylandırılacaktır. İHABİ'nin özellikleri, kullanılan haberleşme modeli ve problemin matematiksel formülü anlatılacak, devamında kullanılan QL algoritması ile İHABİ için hesaplanan güzergâhlardan bahsedilecektir.

3.1 Genel Sistem

Modelimizde bir adet statik yer baz istasyonu, bir adet İHABİ ve bir bölgede rastgele dağılmış kullanıcılar ele alınmıştır. Beklenmedik durumlarda, örneğin aşırı nüfuslu bir alanda, statik yer baz istasyonu tarafından kullanıcılara sağlanan hizmet kalitesi yeterli olmayabilir. Ayrıca her kullanıcı aynı hizmet kalitesi isteğine sahip olmayabilir. Bu nedenle, bazı kullanıcılar için zayıf hizmet kalitesine sahip olma olasılığı daha da yüksek olabilir. Bu sorunu aşmak için çalışmamızda statik yer baz istasyonunun yükünü alan, sabit iletim gücü sağlayan bir İHABİ kullanılmıştır. İHABİ sınırlı bir kapsama alanına sahiptir ve bu alanda kullanıcılara hizmet verebilmektedir. İHABİ hizmet sunarken, ana ağa iletim kısıtlaması ve heterojen kullanıcı hizmet kalitesi koşullarını dikkate alarak güzergâhını belirler. Uçuş süresi boyunca toplam veri hızını maksimize eden güzergâhı seçer [43].

Zorluklar:

- Sınırlı Kapsama Alanı
- Sınırlı Ana Ağa İletim Kapasitesi
- Heterojen Hizmet Kalitesi Dağılımı
- Sınırlı Uçuş Süresi

Önerilen Çözüm:

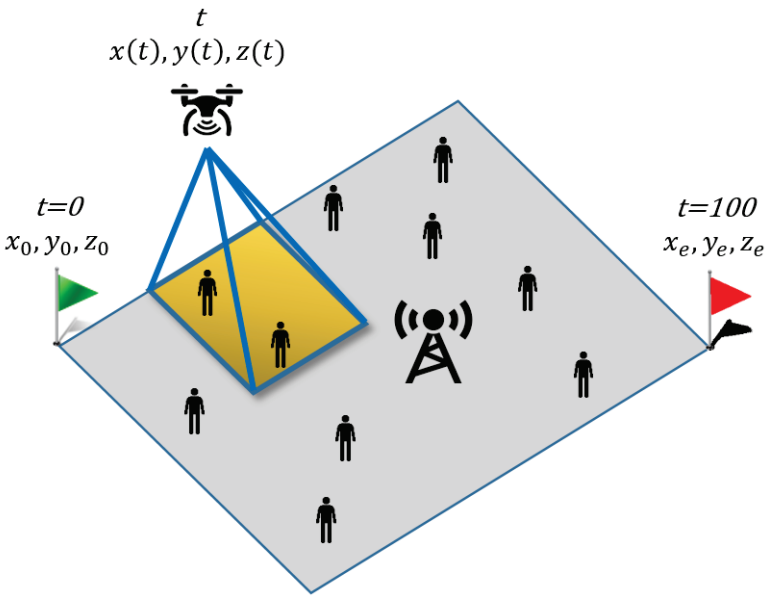
- QL ile İHABİ hareket politikası eğitimi

3.2 İHABİ Modeli

İHABİ, sınırlandırılan bir ortamda farklı yüksekliklerde uçma kabiliyeti ile rastgele olarak dağıtılan farklı hizmet kalitesi talebinde bulunan kullanıcılara sınırlı bir uçuş süresince hizmet verecektir.

- İHABİ için izin verilen toplam uçuş süresi $T = 100$'dür.
- İHABİ'nin hızı ve iletim gücü sabittir.
- İHABİ birden fazla kullanıcıya hizmet verebilmektedir.
- İHABİ ile statik yer baz istasyonu arasında sinyal girişiminin oluşmadığı varsayılmıştır.
- t anındaki İHABİ koordinatları $(x(t), y(t), z(t))$ olarak ifade edilir.
- İHABİ'nin uçuş öncesi başlangıç koordinatları (x_0, y_0, z_0), bitiş koordinatları ise (x_e, y_e, z_e)'dir.
- İHABİ, bulunduğu bölgede sınırlı sayıda kullanıcıları kapsayabilir.

İHABİ Modeli Şekil 3.1'de görselleştirilmiştir:



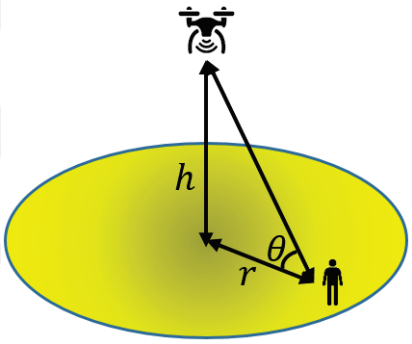
Şekil 3.1: İHABİ modeli.

3.3 Hava Haberleşme Kanalı Modeli ve Özellikleri

İHABİ, belirli bir noktadan uçuşuna başlar ve belirli bir noktada uçuşunu sonlandırır. Uçuş sırasında İHABİ'nin konumu $\Psi \in R^3$ olarak ifade edilir. Toplamda $I = 1,2,...,n$ adet kullanıcı vardır ve bu kullanıcıların konumları $\Lambda_i \in R^3$ ile belirtilmiştir. Statik yer baz istasyonunun konumu ise $\sigma \in R^3$ olarak ifade edilir [43].

Bu çalışmada, havadan yere iletişim için kanal modeli [1] kullanılmıştır. İHABİ konumunun, belirli bir kullanıcının konumunun ve statik yer baz istasyonunun konumunun sırasıyla $\psi = (x_{ua}, y_{ua}, z_{ua})$, $\Lambda = (x_{us}, y_{us}, z_{us})$ ve $\sigma = (x_{gb}, y_{gb}, z_{gb})$ olduğu varsayılarak kanal modeli denklemleri açıklanmıştır [43].

Belirli bir kullanıcı ile İHABİ arasındaki LoS olasılığı, aralarındaki yatay yol $r(\Psi,\Lambda)$ ve dikey yola $h(\Psi,\Lambda)$ bağlıdır. Yükseliş açısı Şekil 3.2'de görüldüğü üzere şu şekilde hesaplanır [43]:



Şekil 3.2: Yükseliş açısı.

$$\theta(\Psi,\Lambda) = (180/\pi)\arctan\left(h(\Psi,\Lambda)/r(\Psi,\Lambda)\right) \tag{3.1}$$

Kullanıcı ile İHABİ arasındaki LoS olasılığı şu şekilde bulunur:

$$P_{LoS}(\Psi,\Lambda) = \frac{1}{1+\alpha e^{-\beta(\theta(\Psi,\Lambda)-\alpha)}} \tag{3.2}$$

Buradaki eşitlikte yer alan α ve β parametreleri banliyö ortamına özgün sabit parametrelerdir. Daha sonra kullanıcı ile İHABİ arasındaki yol kaybı hesabı şu şekilde yapılır [43]:

$$L_{us}(\Psi,\Lambda) = 10\eta log_{10}(\frac{4\pi f_c}{c}) + \mu_{NLoS} + \\ 10\eta log_{10}(d(\Psi,\Lambda)) + P_{LoS}(\Psi,\Lambda)(\mu_{LoS} - \mu_{NLoS}) \tag{3.3}$$

Eşitlikteki $d(\Psi,\Lambda)$ ifadesi, İHABİ ile kullanıcı arasındaki mesafedir ve şu şekilde hesaplanır [43]:

$$d(\Psi,\Lambda) = \sqrt{h(\Psi,\Lambda)^2 + r(\Psi,\Lambda)^2} \tag{3.4}$$

f_c, c ve η sırasıyla taşıyıcı frekansını, ışık hızını ve yol kaybı bileşenlerini temsil eder. μ_{LoS} ve μ_{NLoS}, dB cinsinden ilişkili olarak sırasıyla P_{LoS} ve P_{NLoS} olasılıklarıyla aşırı yol kaybını belirtir. İHABİ ile statik yer baz istasyonu arasındaki ana ağa iletim bağlantısının her zaman bir LoS bağlantısına sahip olduğu varsayılır. Böylece İHABİ ile statik yer baz istasyonu arasındaki yol kaybı hesabı şu şekilde bulunur [43]:

$$L_{ua}(\Psi,\sigma) = 10\eta log_{10}(\frac{4\pi f_c}{c}) + \mu_{NLoS} + \\ 10\eta log_{10}(d(\Psi,\sigma)) + (\mu_{LoS} - \mu_{NLoS}) \tag{3.5}$$

Çalışmanın bu bölümündeki temel amaç, İHABİ tarafından kullanıcılara sağlanan toplam veri iletim hızını uçuş süresi boyunca maksimize etmektir. İHABİ tarafından kullanıcıya ayrılan veri iletim hızı şu şekilde hesaplanır [43]:

$$R(\Psi,\Lambda,B_{ua}) = B_{ua} log_2(1 + 10^{S_{us}(\Psi,\Lambda,B_{ua})}) \tag{3.6}$$

Bu hesaplamada yer alan parametre olan B_{ua}, İHABİ'den kullanıcıya tahsis edilen bant genişliğini temsil eder. Diğer parametre olan S_{us}, kullanıcının SNR değerinin (dB cinsinden) alıcı uçta 10 ile bölünmesidir ve şu şekilde bulunur [43]:

$$S_{us}(\Psi,\Lambda,B_{ua}) = (P_{ua} - L_{us}(\Psi,\Lambda) - 10log_{10}B_{ua} - \omega_n)/10 \tag{3.7}$$

S_{us} ifadesinde yer alan parametrelerden P_{ua}, İHABİ'nin iletim gücünü ve ω_n ise gürültü figürünü (ing. noise figure) ifade eder.

İHABİ ile statik yer baz istasyonu arasındaki ana ağa iletim kapasitesi de optimizasyon probleminde dikkate alınır ve aşağıdaki gibi hesaplanır [43]:

$$C(\Psi,\sigma) = B_{gb} log_2(1 + 10^{S_{ua}(\Psi,\sigma)}) \tag{3.8}$$

Bu kapasite eşitliğindeki S_{ua}, İHABİ'nin SNR değerinin (dB cinsinden) 10 ile bölünmesidir ve şu şekilde hesaplanır [43]:

$$S_{ua}(\Psi,\sigma) = (P_{gb} - L_{ua}(\Psi,\sigma) - 10log_{10}B_{gb} - \omega_n)/10 \tag{3.9}$$

S_{ua} ifadesinde yer alan parametrelerden P_{gb}, İHABİ'nin iletim gücünü, B_{gb} ise statik yer baz istasyonunun İHABİ'ye ayırabileceği bant genişliği miktarını temsil eder.

3.4 Maksimizasyon Problemi Formülasyonu

t anında, İHABİ'nin konumu, i kullanıcısına sağlanan veri hızı ve ana ağa iletim kapasitesi sırasıyla $\Psi(t)$, $R_i(t)$ ve $C(t)$ olarak ifade edilmiştir. Ayrıca, İHABİ kapsama alanının belirlenmesi için gerekli SNR değeri ve kullanıcı i için gerekli hizmet kalitesi miktarı sırasıyla SNR_{req} ve R_i^{req} olarak gösterilmiştir. Aşağıdaki adımlar sırasıyla takip edilerek problem formüle edilmiştir [43]:

- Açıklanan model kullanılarak İHABİ'nin kapsama alanına giren kullanıcılar, eşik SNR değerine göre belirlenir [43]:

$$SNR_i \geq SNR^{req} \tag{3.10}$$

- Bu koşulu sağlayarak İHABİ'nin kapsama alanında bulunan kullanıcılar, toplama işlemine dahil edilmeden önce minimum hizmet kalitesi ister kriterleri açısından kontrol edilir [43]:

$$\zeta_i = \begin{cases} 1, & R_i \geq R_i^{req} \\ 0, & Otherwise \end{cases} \tag{3.11}$$

- Denklem 3.10 ile denklem 3.11'yi sağlayan kullanıcılar üzerinden uçuş süresi boyunca toplam iletim hesabı yapılır [43]:

$$\int_{t=0}^{T} \sum_{i=1}^{n} \zeta_i R_i(t)\, dt \tag{3.12}$$

- Son olarak sınırlı ana ağa iletim kapasitesini de göz önünde bulundurarak, maksimizasyon problemi aşağıdaki gibi güncellenir [43]:

$$\max_{\Psi(t)} \int_{t=0}^{T} \min(\sum_{i=1}^{n} \zeta_i R_i(t), C(t))\, dt \tag{3.13}$$

Bu probleme göre İHABİ, uçuş esnasındaki her zaman adımında kapsadığı ve hizmet kalitesi isterlerini sağlayan kullanıcılar için sağladığı toplam iletimi ana ağa iletim kapasitesini de aşmayacak şekilde maksimize edecektir ve bunu güzergâhını ayarlayarak yapacaktır [43].

3.5 Q-Öğrenme ile Güzergâh Optimizasyonu

Bu çalışmada çevre, ızgaralar şeklinde 3-Boyutlu sınırlı bir alandan oluşmaktadır. Her ızgara, İHABİ'nin koordinatlarını temsil eder. İHABİ, bu ızgaralar üzerinde bulunur ve altı farklı yönde hareket edebilir. Q-değerlerini içeren Q-tablosu, eğitim süreci başlamadan önce 0'larla doldurulur. İHABİ, başlangıç pozisyonuna $(x_{ua_0}, y_{ua_0}, z_{ua_0})$ yerleştirilir ve ε-açgözlü (ing. ε-greedy) denilen bir politika ile nereye hareket edeceğini belirler. Bu politikaya göre İHABİ, ilk hareketini $\varepsilon \in [0,1]$ olasılıkla rastgele yaparak çevreyi keşfetmeye başlar. ε zamanla azaldıkça, keşif (ing. exploration) süreci kendini kullanım (ing. exploitation) süreciyle değiştirmeye başlar. Böylece İHABİ, yapacağı aksiyonu rastgele seçmek yerine, Q-fonksiyonunu maksimize eden değere göre hangi aksiyonu yapacağını belirlemeye başlar. Bu nedenle, öğrenme sürecinde keşif ve kullanım arasında en uygun denge sağlanmalıdır [43].

Önerilen algoritma, Algoritma 1'de açıklanmıştır:

Her bölüm için aşağıdaki prosedür gerçekleştirilir:

- Her bölüm, İHABİ'nin bir başlangıç koordinatına rastgele yerleştirilmesiyle başlar.
- İHABİ, kendini s_t'den s_{t+1}'e taşımak için π politikasına göre bir aksiyonda bulunur.

Algorithm 1 İHABİ için Q-Öğrenme Algoritması

Input: Bölüm Sayısı: N, Adım Sayısı: N_s, Öğrenme Hızı: α, İndirim Faktörü: γ
Initialize: $Q_0(s,a) \leftarrow 0, \forall s_0 \in S, \forall a_0 \in A$
for $Bolum = 1 : N$ **do**
    İHABİ rastgele başlangıç noktalarına yerleştirilir
    for $k = 1 : N_s$ **do**
        ε-açgözlü politikasına göre a aksiyonunu al
        Denklem (3.10)'e göre kapsama alanındaki kullanıcıları belirle
        Denklem (3.11)'e göre kendi QoS kriterlerini sağlayan kullanıcıları belirle
        Denklem (3.12)'e göre kullanıcılara sağlanan toplam veri iletimini hesapla
        Denklem (3.8)'e göre ana ağa iletim kapasitesini hesapla
        Denklem (3.13)'e göre ödül miktarını belirle
        Bir sonraki durumu elde et (s')
        Q-Değerini güncelle:
        $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
        Mevcut durumu güncelle:
        $s \leftarrow s'$
    end
end

- İHABİ, denklem 3.10'a göre kapsadığı alanda kullanıcılara hizmet verebildiğinden, bu kullanıcılar belirlenir ve bu kullanıcıların minimum hizmet kalitesi ister gereksinimlerinin karşılanıp karşılanmadığı denklem 3.11 ile kontrol edilir.
- İHABİ tarafından hizmet kalitesi ister kriterlerini karşılayan kullanıcılara sağlanan veri hızlarının toplamı denklem 3.12 kullanılarak hesaplanır.
- Ayrıca, İHABİ ve statik yer baz istasyonu arasındaki ana ağa iletim kapasitesi denklem 3.8 kullanılarak hesaplanır.
- Bu kapasite hesaplamalarına dayanılarak ödül, denklem 3.13 kullanılarak bulunur.
- Mevcut durum-aksiyon çiftinin Q-değeri bu ödülle güncellenir.
- Bu adımlar her bölüm için periyodik olarak devam eder.
- Bölüm, önceden belirlenen zaman adımı değerine ulaştığında veya İHABİ bitiş noktasına ulaştığında sona erer.

4. PEKİŞTİRMELİ ÖĞRENME MODELİNİN SİMÜLASYONU VE SONUÇLARI

Bu bölümde İHABİ'nin yer aldığı benzetim ortamının özellikleri detaylı bir şekilde açıklanmış ve benzetim ortamındaki İHABİ'nin farklı koşullar altındaki davranışları incelenerek elde edilen sonuçlar analiz edilmiştir. Simülasyonlar ve sonuçlar önemli noktalarıyla anlatılmıştır. Simülasyonlar ve algoritma için kullanılan yazılım ve kütüphaneler Çizelge 4.1'de verilmiştir.

Çizelge 4.1: RL için kullanılan yazılım/kütüphaneler.

İsim	**Versiyon**	**Açıklama**
Python	3.8.2	Yazılım dili
Numpy	1.21.1	Python hesaplama kütüphanesi
Pandas	1.3.1	Güçlü veri yapıları kütüphanesi
OpenCV-Python	4.5.1.48	Bilgisayar görüsü uygulamaları kütüphanesi

4.1 Simülasyon Ortamının Oluşturulması

Simülasyon ortamı, küp şeklindeki bir ortamın birim boyutlu ızgaralara bölünmesine dayanmaktadır. RL modelimizin durumları, bu ızgaraların konumları ve zaman bilgisi olarak (x,y,z,t) ile temsil edilir. A = {İleri, Geri, Sağ, Sol, Yukarı, Aşağı} olarak tanımlanan durumlar arasında geçiş yapılırken altı farklı aksiyon gerçekleştirilebilir. İHABİ, yaptığı her hareket için bir birim hareket eder, dolayısıyla hareket hızı sabittir. Yapılan her aksiyondan elde edilen Q-değerleri, Q-tablosunda tutulur [43].

Ortam 18x18x3 birim kare ızgaralardan oluşturulmuştur. Her ızgara 50x50 metre boyutundadır. 30 adet statik kullanıcı ve statik baz istasyonu yere rastgele yerleştirilmiştir. Heterojenlik oluşturmak için kullanıcılara farklı minimum hizmet kalitesi ister gereksinimleri verilmiştir. İHABİ'nin kapsama alanındaki çeşitlilik, eşik SNR değeri (denklem 3.10) değiştirilerek sağlanmıştır. 2.0 Mbit, 1.6 Mbit ve 1.2 Mbit olmak üzere üç farklı hizmet kalitesi ister gereksinimi kullanıcılara rastgele verilmiştir. İHABİ, başlangıç konumundan $(x_{ua_0}, y_{ua_0}, z_{ua_0})$ hareketine başlar ve toplam veri hızının maksimum olduğu konumları ziyaret etmeye çalışır. Bitiş noktasında uçuşunu sonlandırır. İHABİ'nin uçuş süresi $T = 100$'dür [43].

Simülasyon parametreleri Çizelge 4.2'deki gibidir:

Çizelge 4.2: Simülasyon parametreleri

Parametre	**Değer**
η	2.5
α	9.61
β	0.16
μ_{LoS}	1 dB
μ_{NLoS}	20 dB
ω_n	6 dB
f_c	2 GHz
P_{gb}	42 dBm
P_{ua}	36 dBm
B_{gb}	1 MHz
B_{ua}	0.1 MHz

Uçuş sırasında meydana gelen olaylar Algoritma 1'deki gibidir. Her adımdaki hesaplamalardan sonra ödül hesaplanır ve Q-tablosunun güncelleme işlemi için kullanılır. Ödül sinyali üç bileşenden oluşur:

- Birinci bileşen, kullanıcılara sağlanan toplam veri hızıdır ve İHABİ'nin optimal aksiyonları ile her adımda iyileştirilmektedir.
- Ortam sınırlı olduğu için İHABİ, durum uzayının dışına çıkmamalıdır. Ajan için eğitim sırasında sınırlı ortamın dışına çıkması durumunda negatif ödül verilir. Bu negatif ödül, ödül sinyalinin ikinci bileşenini oluşturur.
- Ödülün son kısmı, İHABİ'yi uçuş süresi içerisinde bitiş noktasına uçmaya teşvik eder. İHABİ bu süre içerisinde bunu yapmazsa yaptığı her hareket için ceza alır ve bu cezalar ya bitiş noktasına ulaşana kadar ya da bölüm bitene kadar devam eder.

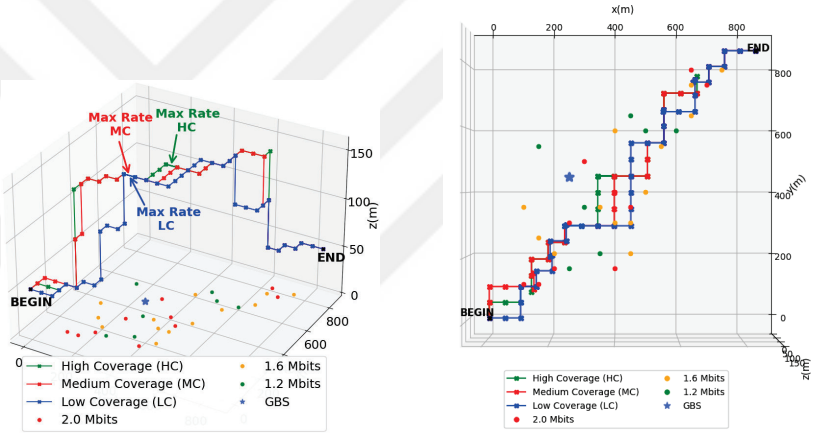
İHABİ, ε-açgözlülük stratejisine göre hareket eder. ε parametresi zamanla üssel olarak azaldığı için keşfe öncelik veren İHABİ, bir süre sonra kullanıma önem vermeye başlar ve Q-tablosuna göre aksiyon alır [43].

Deneme yanılma yoluyla kazanılan deneyime göre, öğrenme oranı $\alpha = 0.1$, toplam bölüm sayısı $N = 10^6$, her bölümdeki zaman adımı sayısı $N_s = 150$ ve indirgeme faktörü $\gamma = 0.99$ seçilerek en uygun çözüm elde edilmiştir [43].

4.2 Simülasyon Sonuçları

İHABİ'nin farklı haberleşme senaryolarına yönelik eğitimleri sonucunda öğrendiği güzergâhlar Şekil 4.1, 4.2 ve 4.3'de verilmiştir. Eğitim sırasında İHABİ'nin bölüm başına toplam veri iletim hızı ödülündeki değişim Şekil 4.4'de gösterilmiştir.

Şekil 4.1, İHABİ'nin farklı kapsama alanları için güzergâh davranışını göstermektedir. İHABİ, kapsama alanı arttıkça irtifasını uçuşun erken anlarında arttırıp, bitişine doğru düşürerek mümkün olduğunca çok kullanıcıya ulaşmaya çalışmaktadır. Bu durum, kapsama alanındaki kullanıcılara sağlanan yüksek iletim hızının irtifa artışına bağlı olarak düşmesine ve yüksek hizmet kalitesi isteri talepleri olan kullanıcıların kaybedilmesine neden olmaktadır. Bunun karşılığında ise kapsama alanı arttıkça, daha düşük hizmet kalitesi isteri taleplerine sahip daha fazla kullanıcı kazanılmakta ve toplam hizmet kalitesi artmaktadır [43].



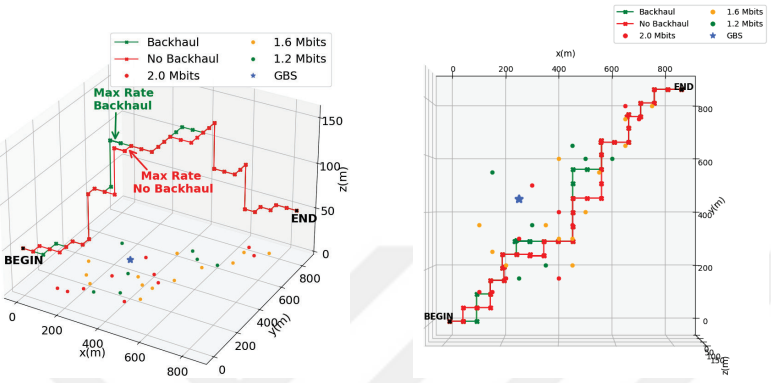
(a) 3B (b) 2B

Şekil 4.1: Kapsama alanı bazlı güzergâh.

En az kapsama alanına sahip senaryo için İHABİ, hizmet kalitesi isteri talebi yüksek olan kullanıcılara odaklanarak toplam hizmet kalitesini artırmaya çalışmaktadır. Bu durum, İHABİ için daha düşük irtifalarda daha uzun uçuş süresi ile sonuçlanmıştır. Ayrıca, İHABİ'nin aksiyon tanımları durmayı içermediğinden, İHABİ'nin uçuş süresinin çoğunu alt şekillerde oklarla gösterilen iki nokta arasındaki yolda gel-git yaparak geçirdiğini de belirtmek gerekir. Diğer bir çarpıcı sonuç ise bu yolların statik yer baz istasyonuna yakın ve kullanıcıların yoğun olduğu yerlerde olmasıdır [43].

Şekil 4.2, ana ağa iletim kapasitesi kısıtlamasının güzergâh üzerindeki etkisini ortaya koymaktadır. Şekilde iki farklı güzergâh bulunmaktadır. Yeşil güzergâh, ağda bir ana ağa iletim kapasitesi sınırlaması olduğunda oluşturulmuştur.

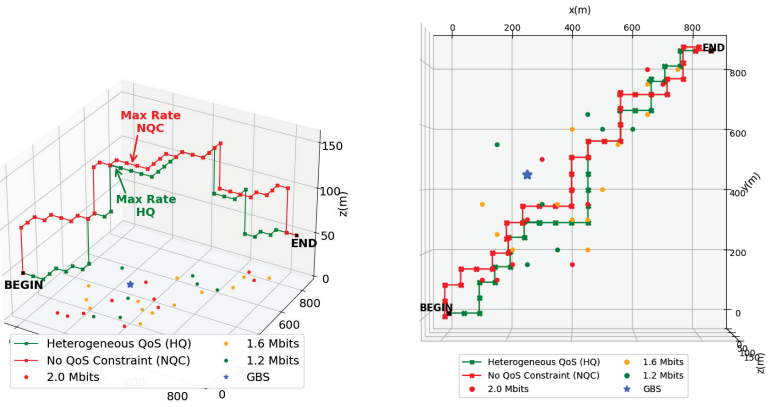
Kırmızı olan için ağda ana ağa iletim kısıtlaması yoktur. Bu güzergâhlar incelendiğinde, küçük bir farkın meydana geldiği görülmektedir. Buradaki önemli olan nokta, veri iletim hızının maksimum olduğu ve burada salındığı konumlarıdır. Ana ağa iletim kapasitesi varken, İHABİ hem kullanıcılara hem de statik yer baz istasyonuna mümkün olduğunca yakın kalmaya çalışmaktadır. İki senaryodaki İHABİ'nin gel-git yaptığı yerler arasında çok fazla mesafe olmamasının ana sebebinin statik yer baz istasyonu civarındaki kullanıcı sayısının en fazla olmasından kaynaklandığı belirtilmelidir [43].



(a) 3B (b) 2B

Şekil 4.2: Ana ağa iletim kapasitesi bazlı güzergâh.

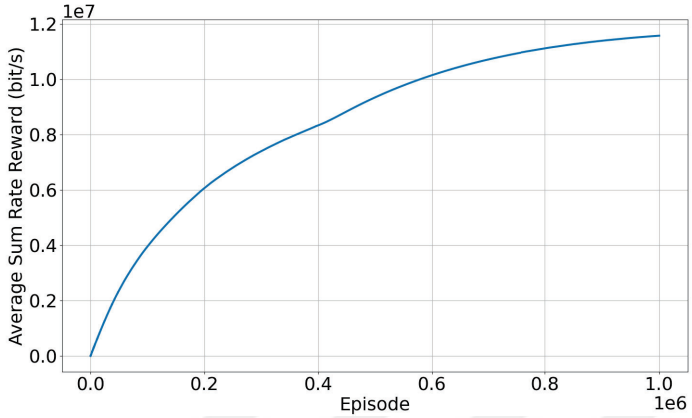
Şekil 4.3, güzergâh üzerindeki heterojen hizmet kalitesi isteri durumunun etkisini göstermektedir. Heterojen hizmet kalitesi koşulunun olduğu senaryoda İHABİ, kullanıcıların ihtiyaçlarını mümkün olduğunca karşılayacak bir güzergâh belirler. Özellikle başlangıç ve bitiş noktalarına yakın, orta ve yüksek miktarda hizmet kalitesi isteri taleplerinde bulunan kullanıcılara hizmet vermek için daha alçak irtifalarda uçma eğilimindedir. Öte yandan, hizmet kalitesi isteri kısıtlamasının olmaması durumu, İHABİ'nin çoğunlukla daha yüksek irtifalarda uçmayı tercih ettiğini göstermektedir [43].



(a) 3B (b) 2B

Şekil 4.3: Heterojen QoS bazlı güzergâh.

Şekil 4.4, eğitim sırasında her bölüm için toplam veri hızı ödülündeki değişikliği göstermektedir. İHABİ, kademeli olarak aldığı toplam ödülü artırır. Bölüm, maksimum değerine yaklaştığında öğrenme aşaması neredeyse tamamlanır ve toplam veri iletim hızı maksimum değerine yaklaşır, bu durum Şekil 4.4'de görülebilir.



Şekil 4.4: İHABİ ile kullanıcılar arasındaki eğitim sırasında toplam veri hızı ödülündeki değişim bölümler boyunca gösterilmektedir.

Ana ağa iletim ve heterojen hizmet kalitesi koşullarına sahip farklı kapsama alanları için eğitimin son bölümündeki ortalama veri iletim hızları Çizelge 4.3'de verilmiştir. Buradaki sonuçlar sırasıyla sabit irtifada 2-Boyutta ve 3-Boyutta hareket eden İHABİ içindir. Çizelge 4.3'ye göre, İHABİ'nin hareket kabiliyetinin olduğu boyutta artış ile ortalama veri iletim hızı artmaktadır. Bunun ana sebebi İHABİ'deki hareket kabiliyeti sınırlamasının azaltılmasıdır. Ayrıca, İHABİ'nin kapsama alanındaki artış ile beklendiği gibi ortalama iletim hızında da artış beraberinde gelmektedir [43].

Çizelge 4.3: Farklı senaryolar için son bölümdeki ortalama veri iletim hızı (Mbps).

Boyut	**Yüksek Kapsama**	**Orta Kapsama**	**Düşük Kapsama**
2D (z = 50 m)	9.49	9.39	9.30
2D (z = 100 m)	10.74	10.50	10.20
2D (z = 150 m)	11.54	11.28	10.87
3D	**12.51**	**12.09**	**11.77**

5. İHABİ SİSTEM MİMARİSİ VE DERİN PEKİŞTİRMELİ ÖĞRENME MODELİ

Bu bölümde İHABİ için veri iletim bazlı minimumun maksimizasyonu ve maksimizasyon problemlerinin içeriği detaylandırılacaktır. İHABİ'nin özellikleri, kullanılan haberleşme modeli ve problemin matematiksel formülü anlatılacak, devamında kullanılan DQN algoritması ile İHABİ için hesaplanan güzergâhlardan bahsedilecektir.

5.1 Genel Sistem

Modelimizde bir adet İHABİ ve bir bölgede dağılmış hareketli kullanıcılar ele alınmıştır. Kullanıcılar, İHABİ yardımı ile hizmet sahibi olmaktadır. İHABİ, sabit iletim gücüne sahiptir, ilk bölümdeki çalışmadan farklı olarak sabit yükselikte uçmaktadır. Bu modelde ilk bölümde yer alan kısıtlamalar yer almamaktadır. İHABİ hizmet sunarken kullanıcıların hareketini de hesaba katmak durumundadır. Sistemdeki bu hareketlilik her zaman diliminde ağ topolojisinin değişimine sebep olmaktadır. Sürekli değişen çevre, çalışmamızda bizi DQN algoritmasına yönlendirmiştir. DQN algoritması ile İHABİ, sürekli değişen ağ topolojilerini kavrayarak güzergâh tayini yapar. Minimumun maksimizasyonu probleminde kullanıcılara adil hizmet dağıtımına, maksimizasyon probleminde ise kullanıcılara sağlanan toplam hizmet miktarına önem verir.

Zorluklar:

- Hareketli Kullanıcılar
- Sürekli Değişen Ağ Topolojisi

Önerilen Çözüm:

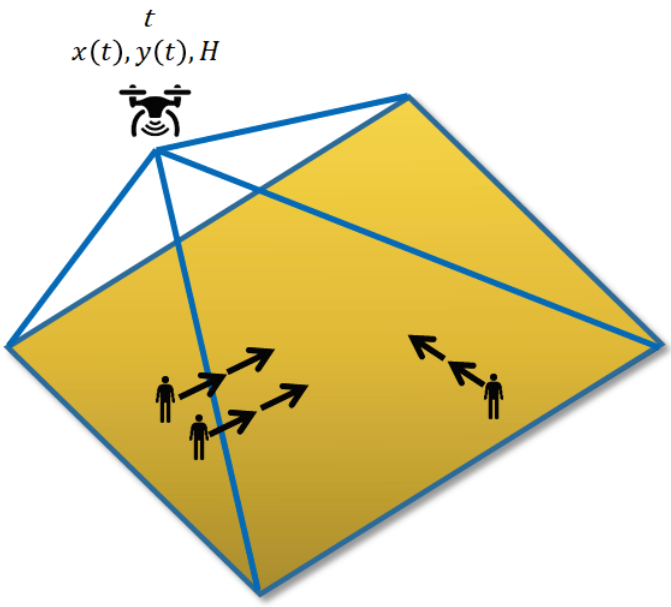
- DQN ile İHABİ hareket politikası eğitimi

5.2 İHABİ Modeli

İHABİ, sınırlandırılan bir ortamda sabit yükseklikte uçma kabiliyeti ile rastgele olarak dağıtılan, minimum hizmet kalitesi talebi kısıtlamasında bulunmayan kullanıcılara sınırsız uçuş süresince hizmet verecektir.

- İHABİ için uçuş süresi kısıtlaması bulunmamaktadır.
- İHABİ'nin hızı ve iletim gücü sabittir.
- İHABİ birden fazla kullanıcıya hizmet verebilmektedir.
- t anındaki İHABİ koordinatları $(x(t), y(t), H)$ olarak ifade edilir.
- İHABİ, bulunduğu bölgedeki tüm kullanıcıları kapsayabilir.

İHABİ Modeli Şekil 5.1'de görselleştirilmiştir:



Şekil 5.1: İHABİ modeli.

5.3 Hava Haberleşme Kanalı Modeli ve Özellikleri

İHABİ, belirli bir noktadan uçuşuna başlar ve süresiz olarak uçuşuna devam eder. Uçuş sırasında İHABİ'nin konumu $\Psi \in R^3$ olarak ifade edilir. Toplamda $I = 1, 2, ..., n$ adet kullanıcı vardır ve bu kullanıcıların konumları $\Lambda_i \in R^3$ ile belirtilmiştir [43].

Bu çalışmada, havadan yere iletişim için kanal modeli [1] kullanılmıştır. İHABİ'nin konumunun ve belirli bir kullanıcının konumunun sırasıyla $\psi = (x_{ua}, y_{ua}, z_{ua})$, $\Lambda = (x_{us}, y_{us}, z_{us})$ olduğu varsayılarak kanal modelinin denklemleri açıklanmıştır.

Belirli bir kullanıcı ile İHABİ arasındaki LoS olasılığı, aralarındaki yatay yol $r(\Psi,\Lambda)$ ve dikey yola $h(\Psi,\Lambda)$ bağlıdır. Yükseliş açısı şu şekilde hesaplanır [43]:

$$\theta(\Psi,\Lambda) = (180/\pi)\arctan\left(h(\Psi,\Lambda)/r(\Psi,\Lambda)\right) \tag{5.1}$$

Kullanıcı ile İHABİ arasındaki LoS olasılığı şu şekilde bulunur [43]:

$$P_{LoS}(\Psi,\Lambda) = \frac{1}{1+\alpha e^{-\beta(\theta(\Psi,\Lambda)-\alpha)}} \tag{5.2}$$

Buradaki eşitlikte yer alan α ve β parametreleri banliyö ortamına özgün sabit parametrelerdir. Daha sonra kullanıcı ile İHABİ arasındaki yol kaybı hesabı şu şekilde yapılır [43]:

$$\begin{aligned} L_{us}(\Psi,\Lambda) = 10\eta log_{10}(\frac{4\pi f_c}{c}) + \mu_{NLoS} + \\ 10\eta log_{10}(d(\Psi,\Lambda)) + P_{LoS}(\Psi,\Lambda)(\mu_{LoS} - \mu_{NLoS}) \end{aligned} \tag{5.3}$$

Eşitlikteki $d(\Psi,\Lambda)$ ifadesi, İHABİ ile kullanıcı arasındaki mesafedir ve şu şekilde hesaplanır [43]:

$$d(\Psi,\Lambda) = \sqrt{h(\Psi,\Lambda)^2 + r(\Psi,\Lambda)^2} \tag{5.4}$$

f_c, c ve η Sırasıyla taşıyıcı frekansını, ışık hızını ve yol kaybı bileşenlerini temsil eder. μ_{LoS} ve μ_{NLoS}, dB cinsinden ilişkili olarak sırasıyla P_{LoS} ve P_{NLoS} olasılıklarıyla aşırı yol kaybını belirtir [43].

Çalışmadaki temel amaç, İHABİ tarafından kullanıcılara sağlanan toplam veri iletim hızını uçuş süresi boyunca maksimize etmektir. İHABİ tarafından kullanıcıya ayrılan veri iletim hızı şu şekilde hesaplanır [43]:

$$R(\Psi,\Lambda,B_{ua}) = B_{ua} log_2(1 + 10^{S_{us}(\Psi,\Lambda,B_{ua})}) \tag{5.5}$$

Bu hesaplamada yer alan parametre olan B_{ua}, İHABİ'den kullanıcıya tahsis edilen bant genişliğini temsil eder. Diğer parametre olan S_{us}, kullanıcının SNR değerinin (dB cinsinden) alıcı uçta 10 ile bölünmesidir ve şu şekilde bulunur [43]:

$$S_{us}(\Psi,\Lambda,B_{ua}) = (P_{ua} - L_{us}(\Psi,\Lambda) - 10log_{10}B_{ua} - \omega_n)/10 \tag{5.6}$$

S_{us} ifadesinde yer alan parametrelerden P_{ua}, İHABİ'nin iletim gücünü ve ω_n ise gürültü figürünü (ing. noise figure) ifade eder.

5.4 Maksimizasyon ve Minimumun Maksimizasyonu Problemleri Formülasyonu

t anında, İHABİ'nin konumu ve i kullanıcısına sağlanan veri hızı sırasıyla $\Psi(t)$ ve $R_i(t)$ olarak ifade edilmiştir. Maksimizasyon problemi şu şekilde formüle edilir:

$$\max_{\Psi(t)} \int_{t=0}^{T} \sum_{i=1}^{3} R_i(t)\,dt \tag{5.7}$$

Bu formülasyona göre İHABİ, uçuş esnasındaki her zaman adımında tüm kullanıcılar için sağladığı toplam iletimi maksimize edecektir ve bunu güzergâhını ayarlayarak yapacaktır.

Minimumun Maksimizasyonu problemi şu şekilde formüle edilir:

$$\max_{\Psi(t)} \int_{t=0}^{T} \min(R_1(t), R_2(t), R_3(t))\,dt \tag{5.8}$$

Bu formülasyona göre İHABİ, uçuş esnasındaki her zaman adımında tüm kullanıcılar için sağladığı veri iletim hızlarından en düşük olan kullanıcıyı arttıracak şekilde güzergâh ayarlaması yapacaktır.

5.5 Derin Q-Öğrenme ile Güzergâh Optimizasyonu

Bu çalışmada çevre, ızgaralar şeklinde 3-Boyutlu sınırlı bir alandan oluşmaktadır. Her ızgara, İHABİ'nin koordinatlarını temsil etmektedir. İHABİ, bu ızgaralar üzerinde bulunur ve dört farklı yönde (İleri, Geri, Sağ, Sol) hareket edebilir. Oluşturulan Derin Q-Ağlarının ağırlıkları eğitim öncesi rastgele değerler alır.

İHABİ, başlangıç pozisyonuna (x_{ua_0}, y_{ua_0}, H) yerleştirilir ve ε-açgözlü politikası ile aksiyonu nasıl alacağını belirler. Kullanıcıların hareketli olması, sinir ağı modeline girdi olarak verilen durumların farklı bir yapıda verilmesini gerektirmektedir. Bunun sebebi algılama örtüşmesi (ing. perceptual aliasing) problemidir. Bu probleme göre belirli bir durumda olan ajanın koşullara göre farklı aksiyon alması gerekebilir. Deterministtik algoritmalarda çıktı, belirli bir durum için değişmediğinden ajan, hep aynı aksiyonu alır ve algılama örtüşmesinin oluştuğu durumlar için istenmeyen sonuçlar doğurabilir. Çalışmamızda bu durumun önüne geçmek için çalışma [35]'den esinlenerek ajanın aldığı ardışık zamanlı durumlar istiflenmiştir. İstiflenme sayesinde sinir ağları, kullanıcıların hangi yönde hareket ettiğine dair bir fikir üretebilmektedir. Bu pratik yöntem ile oluşturulan durum ve diğer durum bilgileri ile birlikte aksiyon ve ödül bilgileri bir hafızaya kaydedilir ve eğitim bu hafızadaki tecrübeler üzerinden gerçekleştirilir.

Önerilen algoritma, Algoritma 2'de açıklanmıştır.

Algorithm 2 İHABİ için Derin Q-Öğrenme Algoritması

Input: Bölüm sayısı: N, Adım sayısı: N_s, İstifleme Sayısı: N_i
Initialize: Deneyim Hafızası D, N_d kapasitesiyle hazırlanır
Tahmin ağı Q, θ rastgele ağırlıklarıyla hazırlanır
Hedef ağı $\hat{Q}$, $\theta^- = \theta$ rastgele ağırlıklarıyla hazırlanır
for $Bolum = 1 : N$ **do**
İHABİ başlangıç koordinatına yerleştirilir
İHABİ ve kullanıcıların başlangıç durumları N_i kadar istiflenir ve bir araya getirilerek durum dizisi $S_N(t)$ oluşturulur.
for $k = 1 : N_s$ **do**
ε-açgözlü politikasına göre rastgele a_t ya da $t = \underset{a}{\operatorname{argmax}}\, Q(s_t, a; \theta)$ aksiyonu seç
İHABİ için seçilen aksiyon a_t'yi uygula ve yeni durum s_{t+1} ve ödül r_t'yi al
İHABİ'nin yeni durumu s_{t+1}'i ve kullanıcıların yeni durumlarını birleştirerek durum dizisine sondan ekle, dizinin başındaki durumu at
$(S_N(t), a_t, r_t, S_N(t+1))$ örneğini Deneyim Hafızası D'ye kaydet
Deneyim hafızasından rastgele örnekler al
Hedef değer $y_j = \begin{cases} r_j & j+1 \text{ adımında biterse} \\ r_j + \max\limits_{a'} \hat{Q}(s_{j+1}, a'; \theta^-) & \text{Diğer} \end{cases}$
Gradyan inişini, $(y_j - Q(s, a_j; \theta))^2$ hata fonksiyonundaki tahmin ağının parametresi θ 'ya göre uygula
Periyodik olarak tahmin ağının ağırlık parametrelerini hedef ağın parametrelerine ata, $\hat{Q} = Q$
end
end

Her bölüm için aşağıdaki prosedür gerçekleştirilir:

- Her bölüm, İHABİ'nin bir başlangıç koordinatına yerleştirilmesiyle başlar. Başlangıç koordinatı istiflenerek durum dizisi oluşturulur.
- İHABİ, kendini s_t'den s_{t+1}'e taşımak için π politikasına göre bir aksiyonda bulunur.
- Seçilen aksiyon ile bir sonraki durum ve ödül alınır. Alınan ödül problem tipine göre denklem 5.8 ya da denklem 5.7 ile hesaplanır.
- Seçilen aksiyon sonucunda varılan İHABİ'nin yeni durumu ve kullanıcıların yeni durumu birleştirilip durum dizisine eklenerek dizi güncellenir.
- Bir önceki durum dizisi, aldığı aksiyon, ödül ve şu andaki durum dizisi hafızaya kaydedilir.
- Eğitim için hafızadan eşit olasılıkla ve rastgele örnekler alınarak tahmin ağı (Q) eğitilir.
- Hedef ağ kullanılarak bir hedef değer üretilir.
- Tahmin ağı ve hedef ağı arasındaki farkın karesi hata fonksiyonu olarak tanımlanır ve bu fonksiyonun ters yöndeki gradyanı tahmin ağının parametresine göre alınarak bu fark minimize edilmeye çalışılır.
- Belirli periyotlarla hedef ağının parametreleri tahmin ağının parametre değerlerini alır.
- Bu adımlar her bölüm için periyodik olarak devam eder.
- Bölüm, önceden belirlenen zaman adımı değerine ulaştığında sona erer.

6. DERİN PEKİŞTİRMELİ ÖĞRENME MODELİNİN SİMÜLASYONU VE SONUÇLARI

Bu bölümde İHABİ'nin yer aldığı benzetim ortamının özellikleri detaylı bir şekilde açıklanmış ve benzetim ortamındaki İHABİ'nin farklı koşullar altındaki davranışları incelenerek elde edilen sonuçlar analiz edilmiştir. Simülasyonlar ve sonuçlar önemli noktalarıyla anlatılmıştır. Simülasyonlar ve algoritma için kullanılan yazılım ve kütüphaneler Çizelge 6.1'de verilmiştir.

Çizelge 6.1: DRL için kullanılan yazılım/kütüphaneler.

İsim	**Versiyon**	**Açıklama**
Python	3.8.2	Yazılım dili
Tensorflow	2.7.1	Makine öğrenimi kütüphanesi
Numpy	1.21.1	Python hesaplama kütüphanesi
Pandas	1.3.1	Güçlü veri yapıları kütüphanesi
OpenCV-Python	4.5.1.48	Bilgisayar görüsü uygulamaları kütüphanesi

6.1 Simülasyon Ortamının Oluşturulması

Simülasyon ortamı, küp şeklindeki bir ortamın birim boyutlu ızgaralara bölünmesiyle oluşturulmuştur. RL modelimizin durumları, bu ızgaralarda bulunan İHABİ ve tüm kullanıcıların 2-boyutlu konum bilgilerinden oluşmaktadır. İHABİ'nin konumu $\Psi = (x_{ua}, y_{ua})$ ve i adet kullanıcının konumları $\Lambda_i = (x_{us_i}, y_{us_i})$ ile ifade edilirse, oluşturduğumuz modelimizin durum gösterimi $(\Psi, \Psi, \Lambda_1, \Lambda_1, \Lambda_2, \Lambda_2, \Lambda_3, \Lambda_3)$ şeklindedir. Çalışmadaki istiflenme katsayısı 2 olarak seçildiği için ortamda yer alan her elemanın kendi durumundan bir tane daha vardır. A = {İleri, Geri, Sağ, Sol} olarak tanımlanan durumlar arasında geçiş yapan İHABİ, toplamda dört farklı aksiyon alabilir. İHABİ'nin hareket hızı sabittir.

Ortam 17x17x3 birim kare ızgaralardan oluşturulmuştur. Her ızgara 50x50 metre boyutundadır. 3 adet hareketli kullanıcı ortama yerleştirilmiştir. Kullanıcı hareketi belirli bir örüntüyü takip etmektedir. İHABİ, başlangıç konumundan (x_{ua_0}, y_{ua_0}, H) hareketine başlar ve incelenen problem tipine göre veri iletim hızını baz alarak uygun yerlerde uçuşunu gerçekleştirmeye çalışır. Uçuş (400, 400) noktasından başlar ve İHABİ ilk etapta koordinat eksenindeki x değerleri için 0 ile 400 değerleri arasında dolanır. Bu, uçuş için ilk periyodu oluşturur.

İkinci periyotta ise İHABİ'nin x değerleri 400 ile 800 aralığında değerler alır. Bu bilgilendirme, Şekil 6.4'ün yorumlanmasında kullanılmıştır.

Simülasyon parametreleri Çizelge 4.2'deki gibidir. (Statik yer baz istasyonu ortamda bulunmadığı için baz istasyonuna ait parametreler kullanılmamaktadır)

Bu bölümde kullanılan ödül sinyali iki bileşenden oluşur:

- Birinci bileşen, minimumun maksimizasyonu problemi için denklem 5.8, max problemi için denklem 5.7'den oluşur.
- Ortam sınırlı olduğu için İHABİ, durum uzayının dışına çıkmamalıdır. İHA eğitim sırasında sınırlı ortamın dışına çıkarsa negatif ödül verilir. Bu negatiflik, ödül sinyalinin ikinci bileşenini oluşturur.

İHABİ, ε-açgözlülük stratejisine göre hareket eder. ε parametresi zamanla üssel olarak azaldığı için keşfe öncelik veren İHABİ, bir süre sonra kullanıma önem vermeye başlar ve eğitilen tahmin ağının ürettiği çıktılara göre hareket eder.

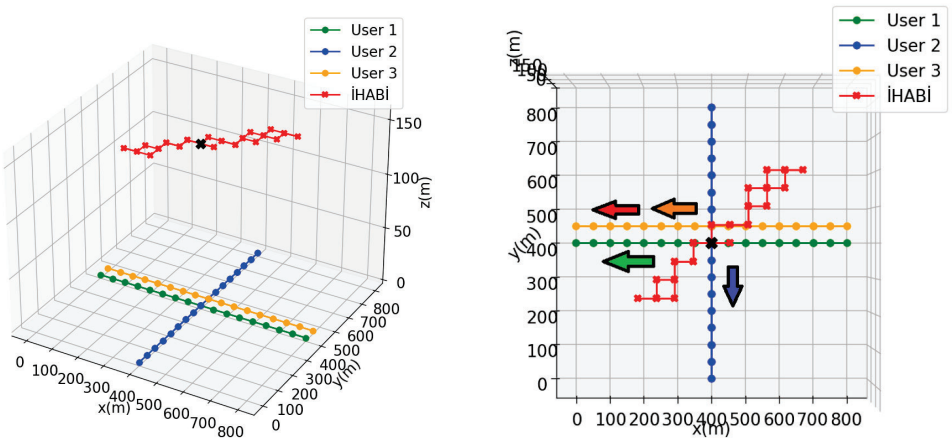
Deneme yanılma yoluyla kazanılan deneyime göre kullanılan eğitim modeli parametreleri Çizelge 6.2'de yer almaktadır:

Çizelge 6.2: Derin sinir ağı modelleri için kullanılan parametreler.

Parametre Adı	**Değer**
Bölüm Sayısı N	250
Adım Sayısı N_s	750
Q Öğrenme Hızı	0.0005
$\hat{Q}$ Öğrenme Hızı	0.0005
İndirgeme Faktörü γ	0.9
Bölüt Miktarı	128
Hafıza Kapasitesi	300000
Periyodik Parametre Kopyalama	20

6.2 Simülasyon Sonuçları

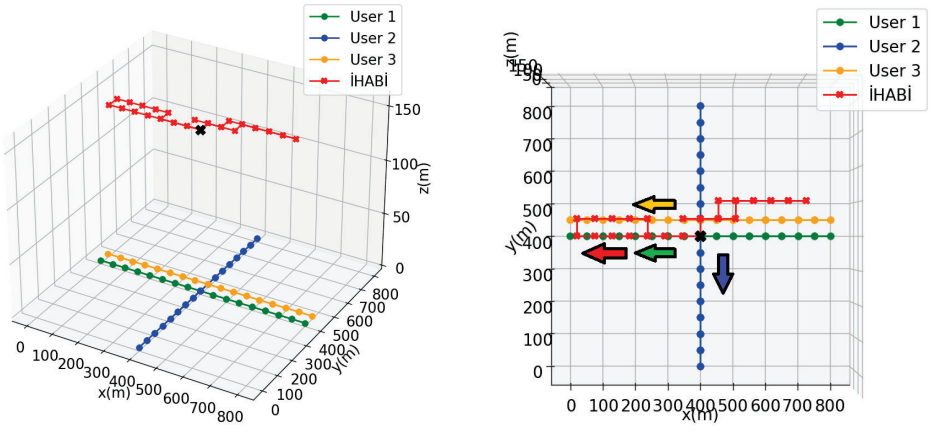
İHABİ'nin farklı optimizasyon problemlerine yönelik eğitimleri sonucunda öğrendiği güzergâhlar Şekil 6.1 ve 6.2'de verilmiştir. Eğitim sırasında İHABİ'nin bölüm başına toplam veri iletim hızı ödülündeki değişimler, ortalama ve kümülatif olarak Şekil 6.3'de gösterilmiştir.



(a) 3B (b) 2B

Şekil 6.1: Maks-Min problemi için güzergâh.

Şekil 6.1, minimumun maksimizasyonu problemi ele alındığında İHABİ'nin bu probleme karşı verdiği davranışı göstermektedir. İHABİ, ortamda yer alan kullanıcılara sağladığı hizmet kalitesinden en düşük olanını her adımda mümkün olduğunca arttırmaya çalışarak kullanıcılar arasında adil bir hizmet paylaşımı sağlamaya çalışacaktır. Bunu yapmak için hem x ekseninde, hem de y ekseninde hareket eden kullanıcılara, merdiven şeklini andıran güzergâhında hareket ederek iki eksendeki kullanıcılara yakınlığını sağlamaya çalışmaktadır. Şekil 6.1'in (b) bölümünde yer alan oklar, başlangıç noktasından hemen sonra kullanıcıların ve İHABİ'nin hangi yönde hareket etmeye başladıklarını göstermektedir. (400, 400) noktası, İHABİ'nin başlangıç koordinatıdır.



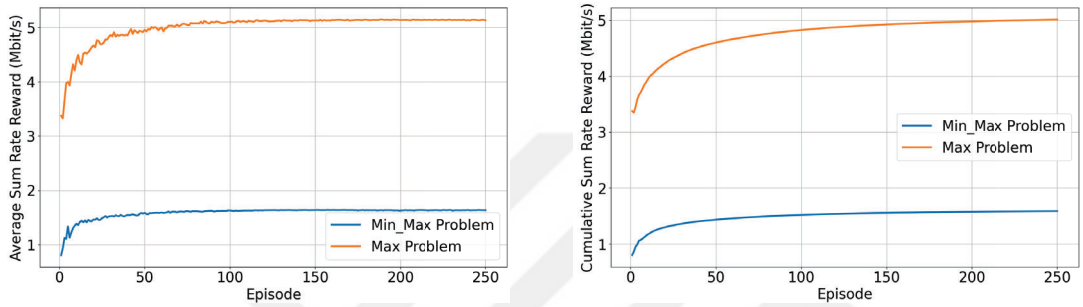
(a) 3B (b) 2B

Şekil 6.2: Maksimizasyon problemi için güzergâh.

Şekil 6.2, maksimizasyon problemi ele alındığında İHABİ'nin bu probleme karşı verdiği davranışı göstermektedir. İHABİ, ortamda yer alan kullanıcılara sağladığı hizmet kalitesini her adımda mümkün olduğunca arttırarak, kullanıcılar arası adil hizmet dağıtımı gözetmeksizin uçuş boyunca sağladığı toplam hizmeti en üst düzeye çıkaracaktır.

Bunu yapmak için büyük oranda x ekseni boyunca hareketini değiştiren iki adet kullanıcıya odaklanmaktadır. Kullanıcı sayısının fazla olması, toplam hizmet kalitesi açısından İHABİ'ye daha cazip gelen bir seçenektir ve bu sebepten dolayı İHABİ, mümkün olduğunca x eksenindeki koordinat değerlerini değiştirmeyi tercih etmektedir. Şekil 6.2'in (b) bölümünde yer alan oklar, başlangıç noktasından hemen sonra kullanıcıların ve İHABİ'nin hangi yönde hareket etmeye başladıklarını göstermektedir. (400, 400) noktası, İHABİ'nin başlangıç koordinatıdır.

Şekil 6.3, iki farklı optimizasyon problemindeki eğitim esnasında her bölüm için elde edilen ödüldeki değişimi göstermektedir. Şekilde iki farklı ödül grafiği yer almaktadır.

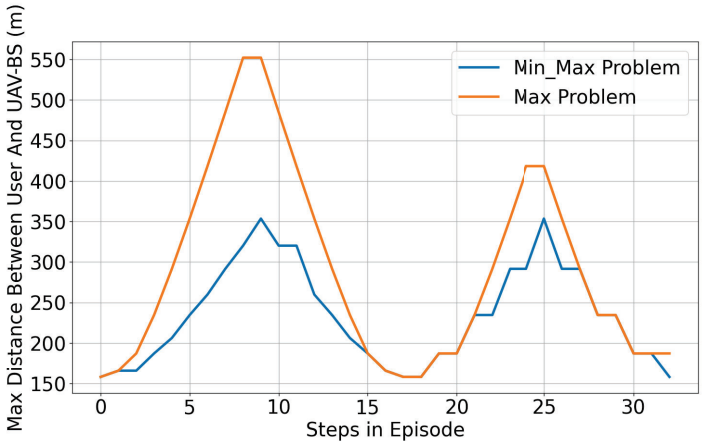


(a) Ödül1 (b) Ödül2

Şekil 6.3: İHABİ'nin eğitim esnasındaki aldığı ödül değişimleri.

Şekil 6.3 - (a), bölüm başına alınan ortalama ödül miktarını göstermektedir. Grafik incelendiğinde beklendiği gibi maksimizasyon problemi, minimumun maksimizasyonu problemine göre çok daha yüksek veri iletim hızının sağlandığını göstermektedir. Eğriler incelendiğinde genel trend artış yönündedir fakat belirli aralıklarla anlık düşüşler de yaşanmaktadır. Derin öğrenme bazlı RL metodlarında bu durum olağandır. Özellikle "Deneyim Tekrarı" tekniği ile yaşanan bu istikrarsızlık mümkün olduğunca azaltılmaktadır. Şekil 6.3 - (b), eğitim boyunca her bölümde alınan ödülün kümülatif olarak toplanarak o anki bölümün değerine bölünmesiyle elde edilen ödül değişimini yansıtmaktadır. Bu teknikle ödüldeki monoton düzenli artış, yani ajanın düzenli olarak aksiyonlarını iyileştirdiği daha rahat gözlemlenmektedir. Şekil 6.3 - (a) ve Şekil 6.3 - (b) incelendiğinde ödül miktarı maksimum değere yakınsamaktadır, bu da eğitimin tamamlandığını göstermektedir. İki optimizasyon problemindeki belirgin fark, İHABİ'nin kullanıcılara olan uzaklığıdır. Bu durum Şekil 6.4'de gösterilmişir.

Minimumun maksimizasyonu probleminde İHABİ kullanıcılara olan uzaklığını dengede tutmaya özen göstermiştir. Maksimizasyon probleminde ise kullanıcı çoğunluğunun olduğu bölgeye gitmeyi tercih etmiştir. Bu durum Şekil 6.4'de de açıkça görülmektedir.



Şekil 6.4: İHABİ ile kullanıcılar arasındaki maksimum uzaklık değişimi.

Burada dikkat çeken bir nokta, şekildeki eğrilerin periyodik bir trendi göstermesidir fakat maksimum probleminde yer alan eğri incelendiğinde, ilk periyotta yer alan maksimum uzaklık değeri ile ikinci periyottaki maksimum uzaklık değeri farklı çıkmıştır. Bu durum, derin öğrenme algoritmalarının genellikle genel optimuma yakın çözümler üretmesinden kaynaklanmaktadır.

Sonuçlar incelendiğinde İHABİ, istenmeyen durum olan sınırlandırılan bölgenin dışına çıkma durumunu kavrayarak bölge içerisinde hareket etmeyi öğrenmiştir. Aynı şekilde istenen optimizasyon problemine uygun güzergâhlar belirleyerek arzu edilen hedefi gerçekleştirmeye odaklanmıştır. Çalışmadaki dikkat çeken tek eksiklik, gerçekleştirilen hedefin en iyi sonuca ulaşamamasıdır. Bu problem, Çizelge 6.2'de yer alan parametreler değiştirilerek iyileştirilebilir.

7. SONUÇ

Bu tez çalışmasında iki farklı çalışma yapılarak ilk çalışmada uçuş sırasında kullanıcılara sağlanan toplam hızı maksimize etmek için kapsama, ana ağa iletim ve heterojen hizmet kalitesi koşullarını içeren bir simülasyon ortamında yol optimizasyonu problemi ele alınmıştır. İkinci çalışmada ise iki farklı optimizasyon problemi ele alınarak İHABİ'ye veri iletim hızı tabanlı güzergâhlar belirlenmiştir ve problemlerin İHABİ'de meydana getirdiği davranış farkı incelenmiştir.

Tez çalışmasının ilk kısmında, İHABİ için literatürde yer alan ve farklı problem tipleri içeren çalışmalar incelenmiştir. Bu çalışmalar genel olarak konumlandırma ve güzergâh belirleme problemini ele almıştır. Takip etme kolaylığı açısından bu çalışmalar, problemde kullanılan çözüm yöntemine göre sınıflandırılmıştır.

Tez çalışmasının devamında ML ve onun alt kategorisi olan RL açıklanmıştır. Çalışmada ele alınan problemin çözümünde kullanılan, RL'nin bir alt kategorisi olan QL algoritması detaylı olarak anlatılmıştır. İlk çalışmada ele alınan problem bir maksimizasyon problemidir. Bu problem, çözüm aşamasından önce uygun bir şekilde matematiksel olarak formüle edilmiştir. İHABİ'nin özellikleri, kullanılan haberleşme modeli ve problemin matematiksel formülasyonu anlatılmış, sonrasında ise kullanılan QL algoritması ile İHABİ için hesaplanan güzergâh ayarlamasından bahsedilmiştir.

İlk çalışmanın son aşamasında, farklı haberleşme senaryoları oluşturularak bu koşullar altındaki davranış ele alınmıştır. İHABİ'nin takip ettiği güzergâhlar görsel olarak oluşturularak okuyucunun inceleyip değerlendirmesi kolaylaştırılmıştır. Simülasyon sonuçları, öne çıkan üç sonuç olan kapsama alanı, ana ağa iletim ve heterojen hizmet kalitesi isterlerinin etkilerini göstermiştir. İHABİ, kapsama alanı kısıtlaması arttıkça irtifasını artırma eğiliminde olmuştur. Ayrıca, ana ağa iletim kısıtlaması, İHABİ'nin yörüngesini statik yer baz istasyonuna yaklaştırmaya zorlamıştır. Son olarak İHABİ, uçuş sırasında kullanıcıların farklı hizmet kalitesi gereksinimlerini mümkün olduğunca dikkate alarak güzergâh belirlemesi yapmıştır.

İkinci çalışmada kullanıcılara hareket yeteneği verildiği için ortam dinamik olarak değişmektedir ve bu durumla başa çıkabilmek için derin öğrenme tabanlı algoritmalara ihtiyaç duyulmaktadır. Bu sebepten dolayı, İHABİ tarafından değişen bu koşullara adapte olunabilmesi için DQN algoritması kullanılmıştır.

İHABİ'nin farklı problemler karşısında verdiği tepkiyi gözlemlemek adına maksimizasyon ve minimumun maksimizasyonu problemleri ele alınmıştır. Bu problemler, çözüm aşamasından önce uygun bir şekilde matematiksel olarak formüle edilmiştir. İHABİ'nin özellikleri, kullanılan haberleşme modeli ve problemlerin matematiksel formülasyonları anlatılmış, sonrasında ise kullanılan DQN algoritması ile İHABİ için hesaplanan güzergâh ayarlamalarından bahsedilmiştir.

İkinci çalışmanın son aşamasında, iki farklı optimizasyon problemi ayrı ayrı ele alınarak, hareketli kullanıcıların yer aldığı senaryolarda İHABİ'nin davranışı incelenmiştir. Simülasyon sonuçları, İHABİ'nin minimumun maksimizasyonu probleminde kullanıcılara olan uzaklığını mümkün olduğunca dengeleyerek adil bir hizmet vermeye özen gösterdiğini göstermiştir. Maksimizasyon probleminde ise İHABİ, kullanıcı sayısının fazla olduğu yerleri tercih ederek uçuş esnasındaki toplam veri iletim miktarını maksimuma çıkarmayı hedeflemiştir. Sonuçlar incelendiğinde genel en iyi çözüme yakın çözümler elde edilmiştir. Bu durumun derin öğrenme algoritmalarında sıkça karşılaşılan bir durum olduğu ve hiperparametre ayarlamalarıyla iyileştirilebileceği belirtilmiştir.

KAYNAKLAR

[1] **Al-Hourani, A., Kandeepan, S.,** AND **Lardner, S.** Optimal lap altitude for maximum coverage. *IEEE Wireless Communications Letters 3*, 6 (2014), 569–572.

[2] **Alzenad, M., El-Keyi, A., Lagum, F.,** AND **Yanikomeroglu, H.** 3-d placement of an unmanned aerial vehicle base station (uav-bs) for energy-efficient maximal coverage. *IEEE Wireless Communications Letters 6*, 4 (2017), 434– 437.

[3] **Amber.** (deep) q-learning, part1: basic introduction and implementation. https://medium.com/@qempsil0914/zero-to- one-deep-q-learning-part1-basic-introduction-and-implementation-bb7602b55a2c, 2019. Accessed: 2022-12-03.

[4] **Andreae, J. H.** Stella: A scheme for a learning machine. *IFAC Proceedings Volumes 1*, 2 (1963), 497–502.

[5] **Baheti, P.** Supervised and unsupervised learning [differences & examples]. https://www.v7labs.com/blog/supervised- vs-unsupervised-learning, 2022. Accessed: 2022-12-03.

[6] **Bayerlein, H., De Kerret, P.,** AND **Gesbert, D.** Trajectory optimization for autonomous flying base station via reinforcement learning. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (2018), IEEE, pp. 1–5.

[7] **Bellman, R**. A markovian decision process. *Journal of mathematics and mechanics* (1957), 679–684.

[8] **Bellman, R.** Dynamic programming. *Science 153*, 3731 (1966), 34–37.

[9] **Bertsekas, D.,** AND **Tsitsiklis, J. N.** *Neuro-dynamic programming*. Athena Scientific, 1996.

[10] **Buckley, J.** *Air power in the age of total war*. Routledge, 2006.

[11] **Chang, F., Chen, T., Su, W.,** AND **Alsafasfeh, Q.** Charging control of an electric vehicle battery based on reinforcement learning. In *2019 10th International Renewable Energy Congress (IREC)* (2019), IEEE, pp. 1–63.

[12] **Chen, Z., Zhong, Y., Ge, X.,** AND **Mia, Y.** An actor-critic-based uav-bss deployment method for dynamic environ- ments. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)* (2020), IEEE, pp. 1–6.

[13] **Chowdhury, M. M. U., Maeng, S. J., Bulut, E.,** AND **Güvenç, I.** 3-d trajectory optimization in uav-assisted cellular networks considering antenna radiation pattern and backhaul constraint. *IEEE Transactions on Aerospace and Electronic Systems 56*, 5 (2020), 3735–3750.

[14] **Çiçek, C. T., Gültekin, H., Tavlı, B.,** AND **Yanıkömeroğlu, H.** Backhaul-aware optimization of uav base station location and bandwidth allocation for profit maximization. *IEEE Access 8* (2020), 154573–154588.

[15] **Cunningham, P., Cord, M.,** AND **Delany, S. J.** Supervised learning. In *Machine learning techniques for multi- media*. Springer, 2008, pp. 21–49.

[16] **Fotouhi, A., Ding, M.,** AND **Hassan, M.** Deep q-learning for two-hop communications of drone base stations. *Sensors 21*, 6 (2021), 1960.

[17] **Ghahramani, Z.** Unsupervised learning. In *Summer school on machine learning* (2003), Springer, pp. 72–112.

[18] **Ghanavi, R., Kalantari, E., Sabbaghian, M., Yanıkömeroğlu, H.,** AND **Yongacoğlu, A.** Efficient 3d aerial base station placement considering users mobility by reinforcement learning. In *2018 IEEE Wireless Communi- cations and Networking Conference (WCNC)* (2018), IEEE, pp. 1–6.

[19] **Gu, S., Holly, E., Lillicrap, T.,** AND **Levine, S.** Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)* (2017), IEEE, pp. 3389–3396.

[20] **Haarnoja, T., Zhou, A., Abbeel, P.,** AND **Levine, S.** Soft actor-critic: Off-policy maximum entropy deep reinfor- cement learning with a stochastic actor. In *International conference on machine learning* (2018), PMLR, pp. 1861–1870.

[21] **Hinton, G. E., Osindero, S.,** AND **Teh, Y.-W.** A fast learning algorithm for deep belief nets. *Neural computation 18*, 7 (2006), 1527–1554.

[22] **Howard, R. A.** Dynamic programming and markov processes.

[23] **Hu, Y.-J.,** AND **Lin, S.-J.** Deep reinforcement learning for optimizing finance portfolio management. In *2019 Amity International Conference on Artificial Intelligence (AICAI)* (2019), IEEE, pp. 14–20.

[24] **Kaebling, L. P., Littman, M. L.,** AND **Moore, A. W.** Reinforcement learning: A survey. *Journal of artificial intelligence research 4* (1996), 237–285.

[25] **Klopf, A. H.** *Brain function and adaptive systems: a heterostatic theory*. No. 133. Air Force Cambridge Research Laboratories, Air Force Systems Command, United . . . , 1972.

[26] **Kotsiantis, S. B., Zaharakis, I., Pintelas, P., ET AL.** Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering 160*, 1 (2007), 3–24.

[27] **Lai, C.-C., Chen, C.-T.,** AND **Wang, L.-C.** On-demand density-aware uav base station 3d placement for arbitrarily distributed users with guaranteed data rates. *IEEE Wireless Communications Letters 8*, 3 (2019), 913–916.

[28] **Lee, W., Jeon, Y., Kim, T.,** AND **Kim, Y.-I.** Deep reinforcement learning for uav trajectory design considering mobile ground users. *Sensors 21*, 24 (2021), 8239.

[29] **Li, D., Xu, S.,** AND **Li, P.** Deep reinforcement learning-empowered resource allocation for mobile edge computing in cellular v2x networks. *Sensors 21*, 2 (2021), 372.

[30] **Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ...** & **Wierstra, D.** (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

[31] **Liu, B.** Supervised learning. In *Web data mining*. Springer, 2011, pp. 63–132.

[32] **McCulloch, W. S.,** AND **Pitts, W**. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 5*, 4 (1943), 115–133.

[33] **Minsky, M.** Steps toward artificial intelligence. *Proceedings of the IRE 49*, 1 (1961), 8–30.

[34] **Minsky, M. L.** *Theory of neural-analog reinforcement systems and its application to the brain-model problem*. Prin- ceton University, 1954.

[35] **Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ...** & **Hassabis, D.** (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.

[36] **Nachum, O., Norouzi, M., Xu, K.,** AND **Schuurmans, D.** Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems 30* (2017).

[37] **Raj, R.** Supervised, unsupervised and semi-supervised learning with real-life usecase. https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning. Accessed: 2022-12-03.

[38] **Rong, S.,** AND **Bao-Wen, Z.** The research of regression model in machine learning field. In *MATEC Web of Conferences* (2018), vol. 176, EDP Sciences, p. 01033.

[39] **Rosenblatt, F.** The perceptron: a probabilistic model for information storage and organization in the brain. *Psycho- logical review 65*, 6 (1958), 386.

[40] **Rumelhart, D. E., Hinton, G. E.,** AND **Williams, R. J.** Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[41] **Rummery, G. A.,** AND **Niranjan, M.** *On-line Q-learning using connectionist systems*, vol. 37. University of Camb- ridge, Department of Engineering Cambridge, UK, 1994.

[42] **Samuel, A. L.** Machine learning. *The Technology Review 62*, 1 (1959), 42–45.

[43] **Sazak, M. D.,** AND **Demirtaş, A. M.** Uav-bs trajectory optimization under coverage, backhaul and qos constraints using q-learning. In *2022 International Balkan Conference on Communications and Networking (BalkanCom)* (2022), pp. 157–161.

[44] **Schaul, T., Quan, J., Antonoglou, I.,** AND **Silver, D.** Prioritized experience replay. *arXiv preprint ar- Xiv:1511.05952* (2015).

[45] **Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., ET AL.** Mastering the game of go without human knowledge. *nature 550*, 7676 (2017), 354–359.

[46] **Spano, S., Cardarilli, G. C., Di Nuonzio, L., Fazzolari, R., Giardino, D., Matta, M., Nannarelli, A.,** AND **Re, M.** An efficient hardware implementation of reinforcement learning: The q-learning algorithm. *Ieee Access 7* (2019), 186340–186351.

[47] **Sutton, R. S.,** AND **Barto, A. G.** *Reinforcement learning: An introduction*. MIT press, 2018.

[48] **Szepesvari, C.,** AND **Littman, M. L.** A unified analysis of value-function-based reinforcement-learning algorithms. *Neural computation 11*, 8 (1999), 2017–2060.

[49] **Thorndike, L.,** AND **Bruce, D.** *Animal intelligence: Experimental studies*. Routledge, 2017.

[50] **Turing, A. M.** (2009). Computing machinery and intelligence. In *Parsing the turing test* (pp. 23-65). Springer, Dordrecht.

[51] **Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., Zhang, S., Zhang, G., Abbeel, P.,** AND **Ba, J.** Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057* (2019).

[52] **Watkins, C. J.,** & **Dayan, P.** (1992). Q-learning. *Machine learning*, *8*(3), 279-292.

[53] **Williams, R. J.** Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning 8*, 3 (1992), 229–256.

[54] **Zeng, F., Hu, Z., Xiao, Z., Jiang, H., Zhou, S., Liu, W.,** AND **Liu, D.** Resource allocation and trajectory optimization for qoe provisioning in energy-efficient uav-enabled wireless networks. *IEEE Transactions on Vehicular Technology 69*, 7 (2020), 7634–7647.

[55] **Zhang, M., Fu, S.,** AND **Fan, Q.** Joint 3d deployment and power allocation for uav-bs: A deep reinforcement learning approach. *IEEE Wireless Communications Letters 10*, 10 (2021), 2309–2312.

[56] **Zhang, S., Zhang, H., He, Q., Bian, K.,** AND **Song, L.** Joint trajectory and power optimization for uav relay networks. *IEEE Communications Letters 22*, 1 (2017), 161–164.

[57] **Zhu, X.,** AND **Goldberg, A. B.** Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning 3*, 1 (2009), 1–130.