
GAZİ ÜNİVERSİTESİ
Gazi
1926

İNSANSIZ ARAÇLARDA KOLEKTİF ZEKA UYGULAMASI

Fatih AYDEMİR

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ŞUBAT 2023

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza
Fatih AYDEMİR
...../……/……

İNSANSIZ HAVA ARAÇLARINDA KOLEKTİF ZEKA UYGULAMASI

(Doktora Tezi)

Fatih AYDEMİR

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2023

ÖZET

Sınırlı donanım kabiliyetleri ve merkezi olmayan karar verme sürecinin zorluğu nedeniyle, insansız hava aracı (İHA) sistemleri kullanılarak dinamik kapsama önemli araştırma konularından birisidir. Bilinmeyen bir ortamda bir grup İHA'nın işbirlikçi davranışı, çözülmesi gereken bir başka sorundur. Bu tez çalışmasında, iki farklı dinamik kapsama probleminin çözümüne yönelik merkezi olmayan yürütme şemasına sahip bir grup İHA'nın kullanıldığı iki farklı yöntem önerilmiştir. Önerilen yöntemlerde her bir İHA pekiştirmeli öğrenme ajanı olarak modellenmiş ve kısmi gözlemlenebilir dinamik ortamda işbirlikçi davranışlar üretmesi amaçlanmıştır. Ajanlar, kendi gözlemlerini ve iletişim kurabildiği ajanların gözlemlerini kullanarak ortam hakkında bilgi sahibi olmaya çalışırlar. Deneme/yanılma yaklaşımı temel alınarak geliştirilen bu yöntemde ajanlar, grup başarısını artırdıkları oranda olumlu ödül; başarıyı düşürdükleri oranda ise olumsuz ödül puanı alırlar. Belirli bir iletişim mesafesine sahip olan ajanlar, doğrudan veya dolaylı olarak birbirleri ile etkileşimde olabilirler. Bağlı ajanlar olarak adlandırılan bu yaklaşım, grup başarısını artırmaya yönelik politikalar üretmeyi kolaylaştırarak kapsam kalitesini artırmaya yardımcı olur. Birinci yöntem, hedef alanın düşük enerji tüketimi ile en yüksek seviyede kapsama yapılmasına yöneliktir. İkinci yöntem ise yüksek adillik indisi gözetilerek hedef alan üzerinde bulunan nesnelerin düşük enerji tüketimi ile en yüksek seviyede kapsanmasına yöneliktir. Önerilen yöntemlerin etkinliği ve verimliliği çok ajanlı aktör-eleştirmen bağlamında çalışan bir benzetim ortamında test edilmiştir. Elde edilen sonuçlar bu tez çalışması kapsamında detaylı olarak incelenerek sonuçları sunulmuştur. Sonuçlar, sınırlı iletişim mesafesine sahip İHA'ların hedef bölgede grup başarısı için hareket edebildiğini ve merkezi yönlendirmeye ihtiyaç duymadan hedefleri başarılı bir şekilde kapsayabildiğini göstermektedir.

Bilim Kodu : 92432

Anahtar Kelimeler : Dinamik ortamlar, çok ajanlı pekiştirmeli öğrenme, dinamik alan kapsama, ilgi noktası kapsama

Sayfa Adedi : 71

Danışman : Prof. Dr. Aydın ÇETİN

COLLECTIVE INTELLIGENCE APPLICATION IN UNMANNED VEHICLES

(Ph. D. Thesis)

Fatih AYDEMİR

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

February 2023

ABSTRACT

Dynamic coverage using unmanned aerial vehicle (UAV) systems is one of the important research topics due to the limited hardware capabilities and the difficulty of decentralized decision making. The cooperative behavior of a group of UAVs in an unknown environment is another problem that needs to be resolved. In this thesis, two different methods using a group of UAVs with a decentralized execution scheme are proposed for solving two different dynamic coverage problems. In the proposed methods, each UAV is modeled as a reinforcement learning agent, and it is aimed to produce collaborative behaviors in a partially observable dynamic environment. Agents try to have information about the environment by using their own observations and the observations of the agents with whom they can communicate. In this method, which was developed based on the trial-and-error approach, agents receive positive rewards as much as they increase group success, but if they reduce success, they get negative reward points. Agents with a certain communication distance can interact with each other directly or indirectly. This approach, called connected agents, helps improve the quality of coverage by facilitating the creation of policies to increase group success. The first method is for maximum coverage of the target area with low energy consumption. The second method, on the other hand, is for the objects located on the target area to be covered at the highest level with low energy consumption, taking into account the high fairness index. The effectiveness and efficiency of the proposed methods are tested in a simulation environment operating in a multi-agent actor-critic context. The results obtained were examined in detail within the scope of this thesis and the results were presented. The results show that UAVs with limited communication distance can act for group success in the target area and successfully cover targets without the need for central guidance.

Science Code	:	92432
Key Words	:	Dynamic environments, multi-agent reinforcement learning, dynamic area coverage, PoI coverage
Page Numbers	:	71
Supervisor	:	Prof. Dr. Aydın ÇETİN

TEŞEKKÜR

Doktora tezi olarak sunduğum bu çalışmanın, planlanıp yürütülmesinde tecrübe, bilgi ve deneyimleriyle bana yol gösteren çok değerli danışmanım Prof. Dr. Aydın ÇETİN'e önemli tavsiye ve değerli bilgilerini paylaşarak beni yönlendiren tez izleme komitesindeki hocalarım Prof. Dr. Necaattin BARIŞÇI ve Prof. Dr. Ömer DEPERLİOĞLU'na ayrıca bu zorlu süreçte gösterdiği sabır ve destekleri için sevgili eşim Merve Bengü AYDEMİR'e manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme sonsuz teşekkür, sevgi ve saygılarımı sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
ÇADDPG	Çok ajanlı derin deterministik politika gradyanı
ÇADPÖ	Çok ajanlı derin pekiştirmeli öğrenme
ÇAMKS	Çok ajanlı Markov karar süreci
ÇAPÖ	Çok ajanlı pekiştirmeli öğrenme
ÇAS	Çok ajanlı sistem
DAKMOKV	Dinamik Alan Kapsama Merkezi Olmayan Karar Verme
DDPG	Derin deterministik politika gradyanı
DÖ	Derin öğrenme
DÖA	Derinlik öncelikli arama
İHA	İnsansız hava aracı
IM	Izgara merkezi
İN	İlgi noktası
KGMKS	Kısmi gözlemlenebilir Markov Karar Süreci
MKS	Markov karar süreci
MO	Markov oyunu
PÖ	Pekiştirmeli öğrenme
PG	Politika gradyanı

1. GİRİŞ

Otonom mobil araçlar, alan kapsama, hedef takibi ve sınıflandırma, sınır güvenliği, arama ve kurtarma, harita oluşturma gibi birçok uygulama tarafından gözlem amacıyla kullanılmıştır [1-4]. Bu araçlar, çevresinden veriler alan ve belirli hedeflere ulaşmak için dış uyaranları davranışlarıyla nasıl ilişkilendireceğine kendi başına karar veren hesaplamalı bir sistemi ifade eder. Görev ortamından aldığı farklı uyaranlara yanıt vererek, farklı davranış kalıpları seçebilir ve sergileyebilirler. Belirsiz ya da beklenmedik olaylarla karşılaştığında dahi insan müdahalesi olmaksızın hedeflerini yerine getirmeye çalışırlar. Davranış kalıpları araç tarafından, öğrenme ve adaptasyon mekanizmalarına dayalı olarak önceden tanımlanabilir veya dinamik olarak üretilebilir. Uygulama ayrıntılarına bakılmaksızın, basit otonom veya yalnızca öngörülebilen sorunları halledebilen otomatik sistemler olmaktan öte belirsiz sorunlara daha "yaratıcı" çözümler üretebilirler. Personele yönelik riskleri sınırlamak, işgücü gereksinimlerini azaltmak, maliyeti düşürmek ve verimliliği artırmak gibi katkılar bu tür araçların kullanımı ile elde edilen avantajların bazılarıdır. Genel olarak, otonom araçların gücü, öngörülemeyen, dinamik olarak değişen ortamlarla başa çıkma becerilerinde yatmaktadır. Bununla birlikte, tek bir otonom araç ile karmaşık ortamlarda uzun dönemli ve güvenilir sonuçlar elde etmek oldukça zordur [5]. Aracın herhangi bir bileşeninde oluşabilecek bir hata, tüm sistemin işlev göremez hale gelmesine sebep olabilir. Gerçek dünya görevleri için yapısal esneklik, güvenilirlik, uyarlanabilirlik ve yeniden yapılandırılabilirlik elde etmek amacıyla çok araçlı otonom sistemler kullanılabilir. Bir problemin birden fazla araç tarafından çözülmeye çalışılması daha kalıcı çözümler üretmeye yardımcı olacaktır. Algılama, veri işleme, donanım boyutlarındaki küçülme ve kablosuz iletişim yeteneklerindeki son gelişmeler, çok sayıda otonom ve mobil aracın bir takım olarak çalışmasına olanak sağlamaktadır.

Arka Plan, Tanım ve Motivasyon

Basit otonom araçların düşük bilgi işlem gücü ve sınırlı algılama gibi dar kapsamlı yetenekleri vardır. Bu sebeple karmaşık bir problemin çözümü ancak çok ajanlı koordinasyon algoritmaları yardımıyla ölçeklenebilir, güvenilir ve sağlam bir şekilde gerçekleştirebilir. Takım üyeleri arasındaki koordinasyon çok ajanlı sistemlerin (ÇAS) başarısına yönelik önemli bir anahtardır. Ajanlar arasındaki koordinasyon, bilgi paylaşımı,

bilgi birleştirilmesi ve ortak karar verme anlamına gelir [6]. Koordinasyon ajanların sınırlı algılama, iletişim, veri işleme ve pil (ömür) gibi kaynaklarını yönetmelerine ve gözetim performansını artırmalarına yardımcı olur. Bilgi paylaşımı, ajanların çevresel bilgisini artırır. Ortak karar verme ise ajan eylemlerinin ortak akıl kapsamında organize edilmesini sağlar [7]. Ortak karar vermenin önemli bir yönü, algılama ve veri işlemenin takım üyelerine dağıtılmasıdır. Ajanların bireysel eylemlerinin bir bütün olarak takıma faydalı olmasını ve takım düzeyindeki amaca katkıda bulunmasını sağlar. Başarılı koordinasyon sonucu olarak ajanlar bilginin türünü, zamanını ve birleştirme yöntemini belirler. Ek olarak, bir eylemin zamanını ve bir sonraki eylemini belirlerler. Koordine edilmemiş bir davranış, hedefin verimli bir şekilde kapsanamamasına ve kaynak israfına neden olacaktır. Koordinasyon, uygulamaya ve mevcut kaynaklara bağlı olarak, merkezi bir noktadan veya dağıtılmış bir şekilde gerçekleştirilebilir.

İnsansız Hava Araçları (İHA), 3 boyutlu düzlemde hareket edebilen özel robot türleridir. Yüksek manevra kabiliyetleri ve görece düşük maliyetleri, onları büyük ölçekli bir ortamın havadan gözetlenmesi ve ulaşılması zor yerlere erişim için uygun kılar. Teknolojideki gelişmeler, daha düşük maliyetli hava araçlarının koordinasyon yetenekleri ile daha karmaşık problemlerin çözümü için fırsatlar sunmaktadır. Bir ortamın gözetleme yolu ile kapsanabilmesi çok sayıda otonom İHA'nın koordinasyonu ile gerçekleştirilebilir. Bununla birlikte, böyle bir yeteneğin elde edilebilmesi için sağlam, güvenilir ve ölçeklenebilir bir hareket modeline ihtiyaç vardır. Bu bağlamda, İHA'ları mobil bir düğüm olarak kullanmak, kurulumu kolay, zaman açısından verimli ve esnek iletişim sistemleri oluşturmanın bir yoludur. Mobil düğüm olarak kullanılan çoklu İHA uygulamaları için, İHA'ların dağıtım stratejileri, minimum enerji tüketimi ile optimum kapsama alanı sağlamalıdır [8]. Optimum alan kapsamı için kurulum stratejileri iki kategoriye ayrılabilir: (1) statik stratejiler, (2) dinamik stratejiler. Statik stratejilerden farklı olarak, dinamik stratejiler çevreye uyum sağlar [9]. Bu nedenle, dinamik stratejiler araştırmacılar tarafından optimum alan kapsama için otonom faaliyet gösteren İHA'lara rehberlik etmek için tercih edilmektedir [10].

Problem Tanımı

Mobil ajanların uygulama gereksinimleri ve fiziksel parametrelerinin çok çeşitli olması nedeniyle, kapsama ve bağlı ajan stratejisi içeren problemler oldukça çeşitlilik gösterir.

Farklı uygulama amaçlarına göre kapsama, ilgi noktası kapsama, bariyer kapsama ve alan kapsama olarak sınıflandırılabilir. Ayrıca, merkezi veya dağıtık bir algoritmanın gerekip gerekmediği, kullanılan algılama ve iletişim modelinin kullanılabilirliği ve doğruluğu gibi varsayımlara göre, problemin her biri farklı açılardan ele alınabilir. Bir grup mobil ajan, kapsama problemini koordineli, etkileşimli, bağımlı veya bağımsız bir şekilde ele alan ÇAS olarak tasarlanabilir [11]. ÇAS mimarisine sahip İHA sistemleri, "kolektif zekâ" olarak adlandırılan ortak bir hedefin gerçekleştirilmesine yönelik davranış geliştiren ajanlar yığını olarak tanımlanabilir [12]. Merkezi algoritmalara dayalı kolektif zeka ile geliştirilen yöntemler, dinamik ortamlarda hesaplama açısından pahalı ve esnek olmayabilir. Kapsama problemleri ile başa çıkmak için dağıtık çok ajanlı derin pekiştirmeli öğrenmenin (ÇADPÖ) kullanılmasına yönelik artan bir ilgi vardır [13]. ÇADPÖ'de her ajan, yerel gözlemlerine ve çevreyle olan etkileşimlerine dayanarak bağımsız karar vermeyi öğrenir. Bu yaklaşım daha ölçeklenebilir ve uyarlanabilir çözümler üretmeye yardımcı olur.

Alan Kapsama Problemi

Alan kapsama problemlerinde amaç tüm alanı kapsamaktır. Uygulama gereksinimlerine göre tam veya kısmi kapsama hedeflenebilir. Bununla birlikte ajan sayısı yeterli değilse tam kapsama sağlanamaz ve dolayısıyla amaç kapsama oranını en üst seviyeye çıkarmak olur.

Tam kapsama kendi içerisinde 2 sınıfta incelebilir; basit kapsama, çoklu kapsama. Basit kapsama, minimum sayıda ajan kullanarak kapsanacak hedef alanın tam kapsanmasını hedefler. Bu yaklaşımda kapsam ve bağlantı sağlanırken dağıtık sayıda çalışan düğüm sayısı en aza indirilir. Çoklu kapsama, basit kapsamanın uzantısı olarak kabul edilir; ancak bu yaklaşımda düğüm sayısının artırılarak hedef alanın çok fazla sayıda ajan ile kapsanması amaçlanır. Ajanlarda meydana gelebilecek problemler önemli verilerin kaybolmasına veya bozulmasına neden olabileceğinden, yüksek hata toleransı sağlamak ve doğru elde etmek bu yaklaşımın odak noktasıdır.

Bazı uygulamalarda, belirli bir alanın tam olarak kaplanması gerekli değildir. Bu durumda belirli bir kapsama derecesi sağlayan kısmi kaplama yeterli kabul edilir. Genel olarak, ortam izleme uygulamaları yalnızca kısmi kapsama gerektirir. Kısmi kapsama, ajanların enerji tüketimini azaltmanın ve görev devamlılığı sağlamanın bir yoludur; çünkü

konuşlandırılan ajan sayısı, dikkate alınan alanı tamamen kaplamak için gereken sayıdan azdır.

Bu tez çalışmasında, alan kapsama problemlerinden kısmi alan kapsama problemi ele alınmış ve çözüm yöntemi olarak ÇADPÖ'nün temel alındığı bir yöntem geliştirilmiştir. Yöntemde, her ajanın yalnızca yerel bilgileri ve yerel politikaları olmasına rağmen, tüm ajanların bilgilerini gözden geçiren ve onlara politikalarını nasıl güncelleyecekleri konusunda tavsiyelerde bulunan bir eleştirmen vardır. Ajanlar, global bilgiler içeren bir modül yardımıyla öğrenir ve en yüksek seviyede alan kapsayacak şekilde konumlanmaya çalışır. Alan üzerindeki uygun dağılım, sınırlı kaynakların etkin şekilde kullanılmasına yardımcı olabilir. Bununla birlikte, takım içerisindeki aktif olarak görev yapan ajan sayısındaki değişim dağılım sürecini zorlaştırır. Buradaki zorluk, kolektif zeka yoluyla en uygun konumların bulunması ve düşük enerji tüketimi ile ajanların alan üzerinde dağılımının ortam değişiklikleri sebebiyle yeniden yapılması olarak özetlenebilir. Bir İHA mevcut konumundan hareket ettiğinde, iletişim mesafesinden çıkabilir ve kapsama süreci sekteye uğrayabilir. Bu nedenle, takım başarısını artıracak hızlı ve sağlam bir çözüm ile ajanların yeniden konumlandırılmasını sağlayan hareket modeli oluşturmak oldukça önemlidir. Ek olarak, hedef alanın uzaklığı, merkezi birimde oluşabilecek hataların etkinlik alanının sınırlandırılması, bilinmeyen dinamik ortamda görev icrası gibi gereksinimler, dağıtık mimari kullanımını zorunlu hale getirir.

İlgi Noktası Kapsama Problemi

İlgi noktası (İN) kapsama, bir grup ajanın bilinmeyen bir ortamı keşfetme ve haritalama ve aynı zamanda mümkün olduğunca İN gözetleme ile görevlendirildiği bir kapsama problemidir. İN kapsamının amacı, bir ajan grubunun İN'ler olarak bilinen önceden tanımlanmış yerleri ziyaret etmesini ve/veya gözetlemesini sağlamaktır. Robotlar, dronlar ve hatta insanlar olabilen bu ajanlar, bir dizi İN'yi kapsamak için dinamik bir ortamda hareket etmelidir. İN'nin ne kadarının izlendiği, sürecin ne kadar hızlı yönetildiği ve hata toleransının ne kadar yüksek olduğu gibi konular kapsama kalitesini belirler.

Uygulama örnekleri, en az sayıda ajan kullanılarak bazı statik veya hareketli hedeflerin algılanmasını içerir. İN kapsama, statik İN ve hareketli İN kapsama olmak üzere iki kapsam altında incelenebilir.

Statik İN kapsama problemlerinde İN'ler sabit konuma sahiptir ve genellikle ajan konumlandırma algoritmasının temel fikri şu şekildedir: tüm ajanlar aynı algoritmayı çalıştırır ve hareket kararı her ajan tarafından ayrı ayrı alınır. Ajanlar, İN veya İN'lerin ağırlık merkezi olabilecek önceden tanımlanmış bir noktaya doğru hareket eder. Ajanlar arası bağlantı kurma ihtiyacının olduğu durumlarda, ajanların hareket mesafesi sınırlandırılır ve böylece iletişim mesafesinin dışına çıkılmaması sağlanabilir.

Hareketli İN kapsama problemlerinde, İN'ler devamlı olarak yer değiştirir ve ajanlar yüksek yoğunluklu İN bölgelerine doğru hareket eder. İN'lerin hareketli olması durumu, en uygun çözüm bulma sürecini ve uygun politika üretme sürecini sekteye uğratabilir. Bu sebeple hareketli İN kapsama problemleri genellikle çoklu hedef takibi problemleri temelinde ele alınır.

Bu tez çalışmasında, İN kapsama problemlerinden statik İN kapsama problemi ele alınmış ve çözüm yöntemi olarak ÇADPÖ'nün temel alındığı bir yöntem geliştirilmiştir. İN kapsama için ÇADPÖ kullanmanın temel zorluklarından biri, ortamın dinamik doğasıdır. ÇADPÖ algoritmaları ortam değişikliklerine karşı duyarlı olacak şekilde geliştirilebilir, yani ajan davranışlarını ortamın mevcut durumuna göre uyarlayabilirler. Bununla birlikte ajanlar arasında etkin koordinasyon gerekliliği başka bir zorluk ortaya çıkarır. Ajan eylemlerinin etkili ve verimli koordinasyonu için karmaşık iletişim algoritmaları ve hareket modelleri tasarlamak gerekir.

Çok Ajanlı Pekiştirme Öğrenmenin Sınırları

ÇAPÖ algoritmaları, ortam tarafından verilen ödülün türüne bağlı olarak üç gruba ayrılabilir: tamamen işbirlikçi, tam rekabetçi ve karma işbirlikçi-rekabetçi. İşbirlikçi ortamlarda tüm ajanlar, ortak bir ödülü en üst düzeye çıkarmak için işbirliği yapar. Bu uygulamanın bir örneği, trafik akışını ve yakıt verimliliğini artırırken çarpışmaları önlemek için ajanların işbirliği yapması gereken otonom sürüş uygulamalarıdır. Rekabetçi ortamlarda, tüm ajanların ödülü sıfıra eşittir. Satranç, Go ve poker dahil olmak üzere çeşitli masa ve kart oyunları bu rekabetçi ortamlara örnektir. Karma uygulamalar, yukarıda belirtilen özellikleri birleştirir ve genel toplamlı bir ödül sunar. Bunun tipik bir örneği, ajanların rakip takımlarla rekabet ederken kendi takım arkadaşlarıyla işbirliği yapmak zorunda olduğu takım oyunlarıdır.

Durağan Olmama

Çok ajanlı sistemlerde ortam, ajanların eylemleriyle değiştirilebilir; böylece, tek ajan perspektifinden, çevre durağan olmaz. Dolayısıyla durağan olmayan bir ortamda oluşturulan politikaların geçerlilik süresi çok uzun olamaz. Durağan olmamanın üstesinden gelmek için kullanılan yöntemlerden birisi ortak eylem öğrenicilerinin kullanılmasıdır. Bu, tek ajanlı bir PÖ kullanır; ancak değer fonksiyonlarını hesaplamak için kullanılan yerel eylem yerine ortak eylem kullanır. Bu yaklaşım, durağan olmama sorununu tamamen ortadan kaldırır. Bununla birlikte, eylem alanı boyutu ajan sayısının artışından çok fazla etkilendiğinden dolayı hesaplama açısından verimli değildir. Bu tür yaklaşımlar ölçeklendirmeyi zorlaştırır. Ek olarak, ajanların diğer ajan eylemlerini göz önünde bulundurabilmesi için merkezi bir denetleyici veya iletişim ağı gereklidir [14].

Durağan olmama, geçmiş verilerle ilgili durum-eylem çiftlerinin ödül bilgilerini geçersiz kılar ve bu etki zaman ilerledikçe artar. Bazı yaklaşımlar, eğitimi en verimli ortak politikaya yönlendirmek amacıyla değişen öğrenme oranları belirleyerek bu zorluğun üstesinden gelmeye çalışır. Bu yöntem, performansı iyileştirerek, ajan eylemlerinin neden olduğu negatif ödülleri göz ardı edebilen "iyimser" ajanları ortaya çıkarır. Gerçekleştirilen bir eylem için gelecekteki ödül, diğer ajanlar tarafından seçilen bir dizi farklı eylem kümesindeki maksimum ödül olarak değerlendirilebilir [15].

Ölçeklenebilirlik

Ajan sayısı arttıkça eylem boyutu da artar. Bu nedenle, bir gözlemcinin tüm ajanların eylem-durum bilgisini aldıktan sonra yapılacak eylemleri belirlediği merkezi yaklaşımlar büyük miktarda hesaplama kaynağı ve bellek gerektirir. ÇAPÖ'deki boyut artışı problemine yönelik bir çözüm, bağımsız öğrenicileri kullanmaktır; ancak bu yaklaşım durağan olmayan ortamda tutarlı sonuçlar etmeyi zorlaştırır. Bağlı ajan stratejisi, merkezi yönlendirmeden bağımsız bir modeldir. Bu tür uygulamalarda, her ajan çevreyle etkileşime girebilir ve diğer ajanlarla bilgi alışverişinde bulunarak zamanla değişen bir iletişim ağı oluşturabilir. Böylelikle, çok sayıda ajan ile ölçeklenebilir bir sistem elde edilebilir. Çünkü merkezi bir denetleyicinin olmaması ve iletişim bağlantılarındaki belirsizlik, birçok uygulamanın tipik gereksinimleridir.

Tez Çalışmasının Literatüre Katkısı

Literatürde, alan kapsama problemleri için birçok çalışma olsa dahi, çevresel değişimler, ÇAS'lardaki gelişmeler, yöntem ve teknolojideki iyileşmeler gibi sebepler yeni araştırma alanları ortaya çıkarmaktadır. Alan kapsama üzerine çalışmalar yapıldığında İHA'ların çevresel koşullar veya donanımsal arızalar nedeniyle çalışamaz hale gelebileceği durumu ilk akla gelen problemlerden bir tanesidir. Aktif görev yapan ajan sayısındaki değişimler ÇAS'ların hedef alana uygun dağılımı için yeni bir konumlanma stratejisine ihtiyacı ortaya çıkarır. Ayrıca ajanlar ve/veya komuta merkezi arasındaki iletişim kesintisi, beklenmeyen donanım ve yazılım arızaları gibi durumlar, uygun bir çözümün daha sofistike şekilde ele alınması ile mümkün olabilir. Bu tür aksaklıklar, uygulamalarda sıklıkla ortaya çıkabilir. Bu bağlamda, alan kapsama sürecini iyileştirebilmek adına üç konu incelenmiştir. Bunlar;

- Bilinmeyen bir ortama uyum sağlamak için modelsiz politika,
- Sağlamlığı artırmak için merkezi olmayan yürütme,
- Güvenilir alan kapsama süreci için ajan sayısından bağımsız bağlı ajan stratejisi.

Bir ajan, eylemleri karşılığında ortamdan bir ödül puanı alarak kendi kendine öğrenme gerçekleştirebilir. Ödül puanını artırmayı amaçlayan deneme-yanılma tabanlı bu yönteme pekiştirmeli öğrenme (PÖ) adı verilmektedir [15]. Bu tez çalışmasında, dinamik ortamda etkili alan kapsama için dağıtık mimariye sahip çok ajanlı bir yöntem önerilmiştir. Önerilen yöntem, eğitim zamanında merkezi bir eleştirmenle öğrenmeye dayanan, yürütme sırasında ise öğrenilen politikaları uygulayan PÖ temelli İHA ajanlarından oluşur. Merkezi olmayan karar verme yetenekleri ile modellenen İHA'lar, en yüksek seviyede kapsama alanı elde etmek için iletişim mesafelerini aşmadan birbirleri arasındaki mesafeyi artırır. Bu sayede kapsanmayan alan en aza indirilerek kaynaklar verimli şekilde kullanılır. İHA ajanlarının birbirleri arasında kurdukları ağ, İHA'ları düğüm olarak içeren yönsüz bağlı graf yapısındadır. İHA düğümleri, komşularına hangi düğümlerle komşu olduklarını sorar. Bu işlem, düğüm kalmayıncaya veya aynı düğümlere döngüler oluşana kadar özyinelemeli olarak devam eder. Böylelikle, her bir İHA düğümü erişilebilir İHA'ların listesine sahip olur. Her adımda İHA, erişilebilen her İHA'nın kapsama alanlarını birleştirir ve ardından hedef alanla kesiştirir. Sonrasında ise hedef alanın yüzde kaç oranında kapsandığını hesaplar. Bu yöntemdeki ana fikir, bağlı İHA yapısını koruyarak, yönsüz grafı hedef alana konumlandırmaktır. Konumlandırma kalitesi, dinamik ortamda görev icra eden araçların

bireysel öğrenme kabiliyetlerini, takım başarısı adına kullandıkları oranda artar. Ajanlar kendi gözlemleri ile erişilebilir ajanların gözlemlerini birleştirerek çevre hakkında daha fazla bilgi sahibi olur. Bu bilgi bütünü, ortamda oluşabilecek değişikliklere yüksek seviyede tolerans sağlanmasına yardımcı olur. Bununla birlikte, ajanların veri setine bağımlı olmadan kendi kendilerine öğrenmelerine olanak sağlar. Model bağımsız olarak oluşturulan bu ÇAS hareket modeli ile bir İHA düğümü, erişilebilir düğüm sayısını artırmayı öğrenmenin yanı sıra, kolektif zeka ile, bağlı düğümlerin en yüksek seviyede kapsama yapabileceği konumları da tespit edebilir.

Önerilen ilk yöntemin literatüre katkıları şu şekilde özetlenebilir:

(1) Bu çalışmada, en yüksek seviyede alan kapsama elde etmek için ÇADPÖ temelli, model bağımsız politikaya sahip bir yöntem önerilmiştir. Model bağımsız politika, değer işlevinden bağımsız ayrı bellek kullanmaktadır. Ayrı bellek kullanımı ajanların en düşük seviyede hesaplama maliyetiyle zaman verimli eylem seçmesine olanak verir.
(2) Önerilen yöntemde her ajan, erişilebilir ajanların eylemlerini göz önünde bulundurarak hedeflenen işbirlikçi davranışı ödüllendirir, hatalı eylemleri ise cezalandırır. Bu eğitim süreci çevrimiçi gerçekleştirilir; ancak yürütme süreci eylem-ödül çifti üzerinden işletilir. Yani önerilen yöntem, merkezi eğitim, merkezi olmayan yürütme şemasına sahiptir.
(3) Sağlam, sürdürülebilir ve hata toleransı yüksek alan kapsama için hedef alanda aktif görev alan ajan sayısına dayalı bağlı ajan stratejisi önerilmiş ve kullanılmıştır.
(4) Kapsam kalitesini artırmak için bireysel ajan ve bağlı ajanların kümülatif ödülüne dayalı ödül yapısı önerilmiştir.

Bu tez çalışmasında alan kapsama problemleri için bir yöntem önermenin yanı sıra, hedef alanda en yüksek seviyede İN kapsamak için her bir İHA'nın bir ajan olarak temsil edildiği ÇADPÖ tabanlı bir yöntem de önerilmiştir. Önerilen bu yöntem, ilk yöntemde olduğu gibi, bir aktör-eleştirmen ağı kullanarak durum ve eylem uzayını kolektif bir zeka temelinde ele alır. Bu yaklaşım, çarpışmaların önlenmesine ve ajanların mümkün olan en kısa sürede en uygun konumlara yönlendirilmesine yardımcı olur. Böylece, bağlı ajanlar ile düzenli-düzensiz şekilli alanlarda en yüksek seviyede İN kapsanabilir. Ajanların homojen ve dairesel kapsamaya sahip olduğu varsayılmıştır. Ajanlar, kapsanacak İN'lerin bulunduğu hedef alanın koordinat değerlerini kullanarak soyut bir dikdörtgen düzlem oluşturur ve bu

alanı ızgaralara böler. Ardından, en fazla sayıda İN'ye sahip en yakındaki ızgara için yol planlaması yapar. Yol planlaması, ortak kapsanan İN'lerin sayısını azaltarak yüksek adillik indisi elde etmeye yardımcı olur. Enerji tüketiminin en aza indirilmesi yol planlaması ile ele alınırken, ızgara ayrıştırma ile düzensiz şekilli alanların kapsama sürecine dahil edilmesi sağlanır.

Tez Çalışmasının Bölümleri

Tez çalışmasının ikinci bölümünde, kapsama problemlerinde kullanılan çalışmalara yer verilmiştir. Üçüncü bölümde ise PÖ'nün temel bileşenleri ve önerilen yöntemlerin detayları açıklanmıştır. Dördüncü bölümde, deneysel çalışmalara yer verilerek önerilen yöntemlerin elde ettiği sonuçlar sunulmuştur. Son bölümde ise bu tez çalışmasının literatür açısından değeri, limitleri ve kapsamlarının belirtildiği sonuç bölümü yer almaktadır.

2. LİTERATÜR ÖZETİ

Izgara tabanlı yaklaşımlar [17-26], dinamik kapsama problemlerini çözmek için hedef alanı alt alanlara ayrıştırır. Izgara ayrıştırma yöntemleri, merkezi bir denetleyici ile koordine edilen bir sisteme uygulanabilir. Bunun yanında her bir İHA'nın diğer İHA'lardan bilgi alması veya gözlemlerine doğrudan erişim sağlaması ile çoklu İHA sistemlerine de uygulanabilir. Choset, bir alanı ileri-geri hareketlerle ızgaralara bölen bir strateji sunmuştur [17]. [18–20]'deki çalışmalar, alanı alt alanlara ve ardından her bir alt alana bir ajan atama temeline dayanır. Cho ve arkadaşları, bir alanı kapsamak için arama yolu planı oluşturmayı hedefleyen bir yöntem önermiştir [21]. Düğümlere en kısa sürede ulaşabilmek için karma tamsayılı doğrusal programlama (KTDP) modeli kullanılmıştır. Optimum alan kapsama görevi için Voronoi bölmelerinin kullanıldığı çalışmalar mevcuttur [22–24]. [25]'te, alan kapsama problemi, ızgaralara ayrıştırılmış bir bölgeye İHA'ların atanarak modellendiği çoklu gezgin satıcı (ÇGS) temelli bir yöntem önerilmiştir. Izgara tabanlı yöntemlerde İHA atama aşamasında, İHA ile ızgara arasındaki mesafe, kapsanacak alan ve enerji tüketimi dikkate alınmalıdır [26].

Kapsama problemini bir optimizasyon problemi olarak ele alan bazı çalışmalarda [27-31], sürü zekasını temel alan yöntemler çözüm olarak önerilmiştir. [27,28]'da karınca olarak modellenen robot ajanlar, hedef alanı kapsamak için feromonların üzerinde hareket eder. Tao ve arkadaşları, yönlü duyargaların optimum algılama yönlerini hesaplayarak alan kapsama sürecini optimize etmeyi amaçlayan yapay balık sürüsüne (YBS) dayalı bir algoritma önermişlerdir [29]. [30]'de, ağın kapsama alanı artırılmaya çalışılarak daha iyi performans elde etmek için mobil duyarga ağlarının dinamik konuşlandırılmasında yapay arı kolonisi algoritması (YAK) uygulanmıştır. Başka bir çalışmada çok amaçlı parçacık optimizasyonu (ÇAPO) algoritması kullanılarak enerji verimli konum belirleme ile ağ kapsama oranının artırılması amaçlanmıştır [31]. Bir lider tarafından yönlendirilen İHA'ların alan üzerinde dağılımı için ardışık kuadratik programlama (AKP) algoritmasının kullanıldığı araştırma mevcuttur [32]. Sanal kuvvet (SK) algoritmaları ve teknikleri, mobil duyargaların oluşturulması ve yerleştirilmesi için kullanılmıştır [33,34]. Geleneksel optimizasyon yöntemleri ve öğrenme tabanlı akıllı optimizasyon algoritmaları (ÖTAO) [35], çok ajanlı sistemdeki her bir İHA'nın eylemlerini optimize eden dağıtık sistemlere uygulanabilir. Ancak, bu yöntemlerin davranışının kesin bir sonucu yoktur [36]. Bu

nedenle, dinamik alanda en yüksek seviyede kapsama için İHA'ları optimum konuma yerleştirmek zor olabilir.

En yüksek seviyede İN kapsama ve alan gözetimi gibi problemleri çözmek için konumlandırma stratejileri temelli çalışmalar yapılmıştır. Mozaffari ve arkadaşları, çoklu insansız hava araçlarının (İHA'lar) baz istasyonu olarak kullanılabileceği senaryolar üzerine çalışmışlardır [37]. Bu bağlamda yazarlar, uzun ömürlü ağ ve yüksek seviyede alan kapsaması için daire paketleme teorisine (DPT) dayalı 3 boyutlu (3B) bir dağıtım stratejisi önermiştir. [38]'te, duyarga düğümleri arası bağlantı kalitesini artırmak amacıyla YAK algoritmasına dayalı bir algoritma uygulanmıştır. Deneysel sonuçlar rastgele dağılım ve genetik algoritma (GA) ile karşılaştırılmıştır. Elde edilen sonuçlara göre, önerilen algoritma diğerlerinden daha uzun iletişim ömrüne ve daha yüksek kapsama oranına sahiptir. Gupta ve arkadaşları, tüm hedef noktaların k-kapsadığı ve duyarga düğümlerinin m-bağlı olduğu GA tabanlı bir yaklaşım ile en uygun duyarga düğüm yerleşimini araştırmışlardır [39]. Kalantari ve arkadaşları, alan kapsama sürecinde, sistem gereksinimlerini ve ortam kısıtlamalarını göz önünde bulundurarak ihtiyaç duyulan İHA baz istasyonu sayısını bulmaya çalışmıştır [40]. Araştırmacılar, baz istasyonlarını 3B bir düzlemde konumlandırabilmek için parçacık sürü optimizasyonuna (PSO) dayalı sezgisel bir algoritma önermiştir [41]. Njoya ve arkadaşları, duyarga düğümleri arasındaki bağlantıyı göz önünde bulundurarak duyarga düğümlerini etkili bir şekilde yaymayı amaçlayan melez bir yöntem önermiştir [42]. Jagtap ve Gomathi, hedef kapsama ve ağ bağlantısını içeren mobil duyarga konuşlandırma problemini öklid yayılma ağacı modeli (ÖYAM) yöntemi ile çözmeye çalışmıştır [43]. Ek olarak, araştırmacılar, mobil duyargaların az hareket-çok kapsama yapabilmesi için bir optimizasyon yöntemi kullanmıştır. [44]'de, ağ bağlantısı korurken kablosuz duyarga ağların kapsamını ve ömrünü artırmak için yeni bir yöntem önermişlerdir. Voronoi diyagramına (VD) ve geometrik merkeze (GM) dayanan bu yöntem VDGM, mobil duyargaların ön tanımlı bir mesafe içerisinde hareket etmesine izin verir. Bu hareket stratejisi, düğümlerin bağlanabilirliğini garanti ederken düşük enerji tüketimi ile alan kapsamını en üst düzeye çıkarmaya yardımcı olur. [45]'te, sabit kanatlı İHA'lar, nesnelerin interneti cihazlarından veri toplamak için kullanılmıştır. Nesnelerin interneti cihazlarının İHA’lara atanması ilkesini temel alan bir dağıtım stratejisi önerilmiştir. Her bir İHA'nın, kendisi ile eşleştirilmiş cihazların üzerinde dairesel bir yol izlediği varsayılmıştır. Cihaz ilişkilendirme problemi, İHA'ların servis kapasitelerinin yanı sıra cihazların yük isteklerini

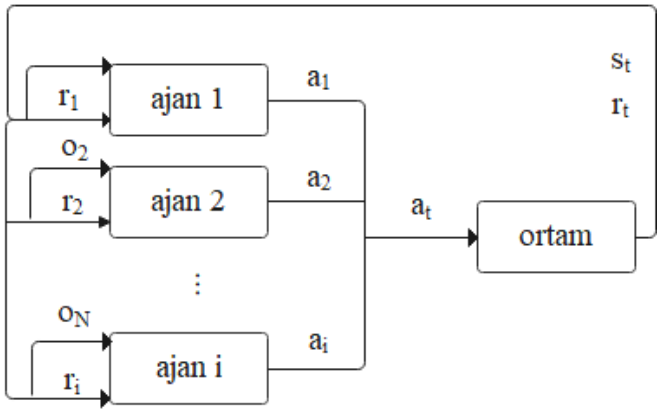
de dikkate alan çoklu sırt çantası problemi olarak tasarlanmıştır. Ganganath ve arkadaşları, engelli ve engelsiz ortamlarda mobil duyarga ağları ile dinamik kapsama için antifloklamaya dayalı iki farklı yöntem üzerinde çalışmışlardır [46].

Otomatik tasarım yöntemleri, çok ajanlı sistemler tarafından başarıyla uygulanmaktadır [47,48]. Otomatik tasarım yöntemleriyle uygulanan çok ajanlı sistemler iki alt kategoriye ayrılır: pekiştirmeli öğrenme (PÖ) [49] ve evrimsel algoritmalar [50]. Evrimsel strateji (ES), evrimsel hesaplama tekniklerini tek ve çok ajanlı sistemlere uygulayan otomatik tasarım yöntemidir [51]. ES yönteminde, bireysel davranış popülasyonu rastgele oluşturulur [52]. Her yinelemede, her bir bireysel davranış için bir dizi deney gerçekleştirilir. Aynı bireysel davranış, deneydeki tüm ajanlar tarafından kullanılır. Her deneyde, ajanların bu bireysel davranıştan kaynaklanan işbirlikçi davranışını değerlendirmek için bir uygunluk fonksiyonu kullanılır. Bu noktada, çaprazlama ve mutasyon gibi en yüksek puanı alan bireysel davranışın seçimi, genetik operatörler tarafından yapılır ve sonraki iterasyonlar tarafından kullanılır. PÖ'de, bir ajan, bulunduğu ortamla etkileşime girer ve her eylemi için olumlu veya olumsuz geri bildirim alır. Deneme-yanılma olarak da adlandırılan bu yöntem PÖ temelli sistemlerin öğrenme stratejisinin temelini oluşturur. Çok ajanlı pekiştirmeli öğrenme (ÇAPÖ) ile geliştirilen sistemler için, görev kolektif düzeyde ele alınır, ancak öğrenme genellikle bireysel düzeyde gerçekleştirilir. Genellikle, bir PÖ ajanı negatif veya yetersiz çözüm bilgisi kullanılarak (evrimsel algoritmalarda göz ardı edilir) hem pozitif hem de negatif eylemleri öğrenir [53]. Bunun yanında ÇAPÖ yaklaşımları çok ajanlı sistemlerin dinamik ortama uyum sağlamasına yardımcı olur. PÖ birçok alanda yaygın olarak kullanılmasına rağmen, çok ajanlı sistemlerle dinamik alan kapsama problemlerini içeren çok fazla çalışma yoktur. Bunun temel nedeni, dağıtık sistemlerde bireysel öğrenme yeteneğinin, işbirlikçi öğrenmeye uyum sağlamasının zor olmasıdır. Ortak bir öğrenme stratejisi kullanmak, farklı parçalardaki verileri kullanan sistemlerin performansını ve güvenilirliğini artırmaya yardımcı olur [54]. Çözüm olarak, Xiao ve arkadaşları, dinamik alan kapsama problemi için Q-öğrenmeye dayalı bir yol planlama algoritması önermiştir [55]. [56]'de, yörünge planlaması yapılabilmesi için yine Q-öğrenme temelli bir yaklaşım önerilmiştir. Q-öğrenme algoritması birçok çalışma tarafından kullanılmasına rağmen yapısı itibariyle dezavantajları vardır. Q-öğrenmede bir keşif stratejisi belirtilmemiştir; bu sebeple ajanlar herhangi bir durum için tüm olası eylemleri denemek zorundadır [57]. Tahmini değerde küçük değişiklikler olsa dahi, politika büyük ölçüde etkilenir [58]. [59]'de, ÇAPÖ ile İN

kapsama stratejisi, Laplace matrisinin yönlendirilmemiş graf oluşturmak için kullanıldığı bir yapı üzerine kurgulanmıştır. Liu ve arkadaşları, bağlantı kısıtlaması olan alan kapsama problemleri için, derin pekiştirmeli öğrenme (DPÖ) algoritması temelli, düşük enerji tüketimini hedefleyen bir İHA kontrol algoritması önermiştir [60]. Önerilen algoritma, derin deterministik politika gradyanına (DDPG) dayalı bir tür aktör-eleştirmen PÖ yöntemidir. DRL-EC3 olarak adlandırılan algoritma, düşük enerji tüketimi ile İHA'lar arasındaki bağlantıyı korurken yüksek adillik indisi elde etmeyi hedeflemektedir. Liu ve arkadaşları başka bir çalışmada, merkezi olmayan yönlendirmeden bağımsız bir grup İHA'nın kapsama alanını artırmak için bir dağıtık yapıda DPÖ yöntemi önermiştir [61]. Yöntem, bağlı İHA'lar ile minimum enerji tüketirken hedef İN'lerin yüksek adillik indisi ile kapsanmasını amaçlamaktadır. Eylem alanı, durum, ödül ve gözleme göre DNN oluşturmuş her bir İHA'yı bir ajan olarak modellemişlerdir. [62]'te hedef alana İHA'ların yerleştirilmesi için kullanılan merkezi olmayan karar algoritması, kapsama puanını yüksek adillik indisi ve enerji verimli olacak şekilde en üst düzeye çıkarmak için kullanılmıştır. Aktör-eleştirmen durum tabanlı oyun (SBG-AC) olarak adlandırılan algoritma, bir grup İHA'ya minimum etkileşimle karar verme yeteneği sağlamayı amaçlar. Pham ve arkadaşları, aynı eylem ve durum uzaylarına sahip homojen İHA grubu ile tam kapsama elde etmek için Q-öğrenmeye dayalı bir algoritma önermiştir [63]. Q-işlevi, Sabit Seyrek Temsil (FSR) ve Radyal Temel Fonksiyon (RBF) yaklaşım teknikleri kullanılarak ayrıştırılır. Kullanılan ayrıştırma tekniğinin yapısı nedeniyle dağıtık davranış stratejileri oluşturulması oldukça zordur.

3. SİSTEM MODELİ VE METODOLOJİSİ

Bu bölümde, önerilen modeller ve modellerin temelini oluşturan ÇADPÖ'nün rolü hakkında bilgi verilmektedir. ÇADPÖ oluşturmak için gerekli aşamalar sırasıyla gözden geçirilmiş ve özet bilgilere yer verilmiştir. Bu bağlamda ilk aşamada, tek ajanlı PÖ'nın temel yapısı hakkında bilgi verilmiştir. Sonrasında ise ÇAPÖ'ye giriş yapılmış ve temel bilgilere yer verilmiştir. ÇAPÖ tek ajanlı PÖ'nün, bir grup ajanın çevre ve birbirleriyle etkileşimlerini kullanarak en uygun politikaları öğrenmesini sağlayan genelleştirilmiş halidir (Şekil 3.1). Yani ÇAPÖ, öğrenme sürecinde diğer ajanların varlığını göz ardı etmez.



Şekil 3.1. ÇAPÖ çalışma prensibi

Birden fazla ajanın kullanıldığı durumlarda çeşitli zorluklarla karşılaşılabilir. Bunlar;

- Ajanların aynı anda davranışlarını değiştirmesinden dolayı sistemin durağan olmaması,
- Ortak eylem uzayı ajan sayısıyla katlanarak büyüdüğü için ölçeklenebilirlik sorunu,
- Ajanların sistemin yalnızca kısmi bilgilerine erişiminin olduğu gerçek dünya uygulamalarında sıklıkla ortaya çıkan kısmi gözlemlenebilirlik sorunu,
- Mahremiyet ve güvenlik (paylaşılan bilgiler)

ÇAPÖ genellikle bir Markov Oyunu (MO) olarak formüle edilir; Stokastik Oyun (SO) olarak da adlandırılır. MO, oyun teorisi literatüründe tek ajanlı PÖ problemlerini ve tekrarlanan oyunları modellemek için kullanılan Markov Karar Süreç'lerini (MKS) kullanır. Tekrarlanan oyunlarda, aynı oyuncular sahne oyunu adı verilen belirli bir oyunu

tekrar tekrar oynarlar. Bu nedenle, tekrarlanan oyunlar, durum bilgisi olmayan statik bir ortam olarak kabul edilir ve ajanların, yalnızca ajanların arasındaki etkileşimlerden etkilenir. MO, ajanların ödüllerini etkileyen dinamik bir ortamı göz önünde bulundurarak bu eksikliği giderir. MO'ler tamamen işbirlikçi, tamamen rekabetçi veya karma olarak üç sınıfa ayrılabilir. Tamamen işbirlikçi senaryolar, ajanların aynı fayda veya ödül işlevine sahip olduğunu varsayarken, tamamen rekabetçi ortamlarda, genellikle sıfır toplamlı oyunlar olarak bilinen karşıt hedeflere sahip ajanlar kullanılır. Karma ayar, ödüller üzerinde herhangi bir kısıtlamanın dikkate alınmadığı genel durumu kapsar. Buna genel toplamlı oyunlar da denir.

Bu bölümde, önerilen yöntemleri inşası için gerekli süreçler hakkında bilgi verilmiştir. PÖ ve ÇAPÖ hakkında bilgi verildikten sonra sırasıyla derin öğrenme, Çok Ajanlı Derin Deterministik Politika Gradyanı (ÇADDPG) [64] ve önerilen yöntemler detaylı olarak açıklanmıştır. [65]'ye göre derin öğrenme, farklı soyutlama düzeylerine karşılık gelen birden çok temsil düzeyini öğrenen bir makine öğrenme yaklaşımıdır. Basit bir durumda biri giriş sinyali alan, diğeri çıkış sinyali gönderen iki nöron grubu içerir. Girdi katmanı bir girdi aldığında, girdinin işlenmiş halini bir sonraki katmana iletir. Derin bir ağda, giriş ve çıkış arasında çoklu doğrusal ve çoklu işlem katmanlarından oluşan birçok katman vardır [69]. ÇADPÖ ise, karmaşık görevleri çözmek amacıyla rekabet eden veya iş birliği yapan PÖ ajanlarından oluşan çok ajanlı bir sistemdir. Tek ajanlı sistemlerde, ajan yalnızca kendi eylemlerinin sonucuyla ilgilenir. Çok ajanlı bir sistemde ise bir ajan yalnızca kendi eyleminin sonuçlarını değil, aynı zamanda diğer ajanların davranışlarını da inceler. Öğrenme süreci karmaşıktır; çünkü tüm ajanlar birbirleriyle etkileşime girerek aynı anda deneme-yanılma yoluyla öğrenir. ÇADDPG, her ajanın aktör-eleştirmen olmak üzere 2 farklı ağ içerdiği bir ÇADPÖ yaklaşımıdır. Aktör ağı, ajanın bulunduğu duruma göre yürütülecek eylemi hesaplarken, eleştirmen ağ aktör ağının performansını iyileştirmek için eylemin sonuçlarını değerlendirir. Bu tez çalışması kapsamında önerilen her iki yöntem de benzer bir alt yapıya sahiptir. Her iki yöntem de ÇADDPG yöntemini güncelleyerek farklı problemlere farklı yaklaşımlar sunmaktadır.

3.1. Tek Ajanlı Pekiştirmeli Öğrenme

Tek ajanlı PÖ'de, ortam genellikle bir MKS olarak modellenir. MKS ayrık zamanlı bir stokastik kontrol sürecidir.

3.1.1. Markov Karar Süreci

PÖ'de, bir ajan sıralı bir karar verme problemini çözmek için çevre ile etkileşime girer. Tamamen gözlemlenebilir ortamlar (S, A, P, R, γ) olarak tanımlanan MKS'ler olarak modellenir. S ve A sırasıyla durum ve eylem uzaylarını tanımlar.

- $P := S \times A \mapsto [0,1]$, bir a eylemini gerçekleştirdikten sonra s durumundan s' durumuna geçiş olasılığını belirtir;
- $R : S \times A \times S \mapsto \mathbb{R}$, ajanın a eylemini s durumunda gerçekleştirmesi ve s''ye geçişle sonuçlanması için aldığı anlık ödülü tanımlayan ödül işlevidir;
- $\gamma \in [0,1]$, anlık ve gelecek ödülleri değiştiren indirim faktörüdür.

MKS'lerin tam gözlemlenebilirlik varsayımı, ajanın her t adımında sisteminin mevcut durumuna s erişmesini sağlar. Ajan, sistemi s durumundayken $P(.|s,a)$ olasılık dağılımından örneklenen yeni bir s' durumuna geçirmek için a eylemini gerçekleştirmeye karar verir. Ajan, anlık olarak $R(s,a,s')$ ile ödüllendirilir. Böylece ajan ödülü $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s,a,s') \mid a \sim \pi(.|s), s_0\right]$ olarak ifade edilir. Buna sonsuz ufuklu indirimli ödül denir. Diğer bir popüler formülasyon, indirimsiz sonlu ufuklu ödül $\mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(s,a,s') \mid a \sim \pi(.|s), s_0\right]$'dir. Burada ödül, sonlu bir ufuk H üzerinden hesaplanır. Bu yaklaşım, epizodik süreçlerde (yani, sonu olan görevler için) yaygın olarak kullanılır.

Ajan, beklenen ödülü en yüksek seviyeye çıkarmaya çalışan, durum ve eylemlerle ilişkili en uygun politikayı π^* bulmayı amaçlar. Bir politika veya strateji, ajanın her t adımındaki davranışını tanımlar. Kararlı bir politika, algılanan her durumda yapılacak eylemleri döndürür. Öte yandan, stokastik bir politika, eylemler üzerinde bir dağılım sağlar. Belirli

bir π politikası altında, s_t durumu veya (s_t, a_t) için değer işlevi (Q_işlevi) aşağıdaki gibi tanımlanabilir:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s, a, s_{t+1} \mid a_t \sim \pi(. \mid s_t), s_0 = s\right] \quad (3.1)$$

$$Q^{\pi}(s, a) \ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s, a, s_{t+1} \mid a_t \sim \pi(. \mid s_t), s_0 = s, a_0 = a\right] \quad (3.2)$$

MKS ile elde edilen en iyi çözüm, ajanın seçim yapabileceği tüm olasılıkların en iyi alternatifidir. Bir MKS modellenirken temelde 3 yöntem baz alınır.

Değer İterasyonu

MKS'de amaç, bir sonraki duruma geçiş yaparken, en iyi durum-eylem ikilisi üretmektir. Değer iterasyonu, eylemin anlık ödülüne bir sonraki adımda seçilecek eylemin değerinin eklenmesi temeline dayanır. Ufuk uzunluğu arttıkça, aşağıdan yukarıya doğru eylemler iteratif olarak sürdürülerek en üst seviyedeki değer hesaplanır.

Plan İterasyonu

Plan iterasyonu, değer iterasyonundan farklı olarak, bir başlangıç değeri değil, herhangi rastgele bir planı girdi olarak alır. Bu plan üzerinde ilk olarak, değer iterasyonunda yapıldığı gibi, plan değerlendirmesi yapılır. Fakat plan iterasyonunda bu aşamadan sonra, plan geliştirme aşaması vardır. Plan iterasyonunda, plan geliştirme aşaması olması sebebiyle değer iterasyonuna göre daha fazla işlem gerektirir.

Doğrusal Programlama

MKS modelinin başka bir çözüm yöntemi, doğrusal programlama yöntemi kullanılarak bir plan için hesaplanan değerin en büyüklenmesi yaklaşımıdır.

3.1.2. Kısmen gözlemlenebilir Markov Karar Süreci

Kısmen gözlemlenebilir Markov Karar Süreci (KGMKS), gözlemlenemeyen sistem durumlarını olasılıksal olarak gözlemlere bağlayan ve dinamik sistem modellemek için kullanılan bir tür MKS'dir.

Ajan, sistem durumuna ve ajanın gelecekteki eylemlerine bağlı olarak gelecekte beklenen ödülleri en üst düzeye çıkarmak amacıyla sistemi etkileyen (yani sistem durumunun değişmesine neden olabilecek) eylemler gerçekleştirebilir. Amaç, ajanın eylemlerine rehberlik eden en uygun politikayı bulmaktır. MKS'lerden farklı olarak, KGMKS 'lerdeki bir ajan tüm sistem durumunu doğrudan gözlemleyemez; ancak duruma bağlı gözlemler yapar. Ajan, sistemin mevcut durum bilgisini oluşturmak için bu gözlemleri kullanır. Bu yaklaşım, tüm olası durumlar üzerinden bir olasılık dağılımı olarak ifade edilir. KGMKS'nin çözümü, bir durumda hangi eylemin gerçekleştirileceğini belirleyen bir politikadır. Durumlarının sürekli olduğu, sonsuz bir durum kümesiyle sonuçlandığını ve bunun da MKS'lere kıyasla KGMKS'leri çözmeyi çok daha zor hale getirdiği unutulmamalıdır.

Klasik MKS'lerde, ajanın tüm durum bilgilerine erişimi olduğu varsayılmıştır; ancak, bu varsayım gerçek dünya uygulamaları için uygulanabilir değildir. KGMKS'nin yapısı, çeşitli gerçek dünya problemlerini modellemek için yeterince geneldir. Uygulamalar, yönlendirme problemleri, makine bakımı gibi genel olarak belirsizlik altındaki planlamaları içerir. Örneğin nesnelerin interneti cihazları, duyargaları kullanarak ortam hakkında bilgi edinir. Duyarga ölçümleri gürültülü ve sınırlıdır, dolayısıyla ajan bulunduğu çevrenin yalnızca kısmi bilgilerine sahip olabilir. Bu bağlamda KGMKS, durum hakkındaki belirsizliği dikkate almak için MKS süreçlerini iyileştirmeye çalışır.

KGMKS, 7-öğeli bir $(S, A, T, R, \Omega, O, \gamma)$ ile tanımlanır:

- $S = \{s_1, s_2, ..., s_n\}$, kısmen gözlemlenebilir durumların kümesidir;
- $A = \{a_1, a_2, ..., a_m\}$, eylemlerin kümesidir;
- T, eyleme bağlı olarak $s \to s'$durum geçişi için koşullu geçiş olasılığı $T(s' \mid s, a)$ kümesidir;

- $R : S \times A \rightarrow \mathbb{R}$, ödül işlevidir;
- $\Omega = \{o_1, o_2, ..., o_k\}$, gözlemlerin kümesidir;
- O, ulaşılan duruma ve alınan eyleme bağlı $O(o \mid s', a)$ gözlem olasılıkları kümesidir,
- $\gamma \in [0,1]$, indirim faktörüdür.

Her zaman periyodunda, çevre bilinmeyen bir $s \in S$ durumundadır. Ajan, ortamın $T(s' \mid s, a)$ olasılıkla $s' \in S$ durumuna geçmesine neden olan bir $a \in A$ eylemini seçer. Aynı zamanda, ajan, $O(o \mid s', a)$ olasılıkla ortamın yeni durumuna bağlı olan bir $o \in \Omega$ gözlemi elde eder. Bunların karşılığında ajan bir ödül $R(s, a)$ alır. Ardından süreç tekrar eder. Ajanın hedefi, her t anında seçtiği eylemler ile gelecekteki ödüllerin toplamını en üst düzeye çıkaran politika oluşturmaktır.

$$\max E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \tag{3.3}$$

Sonlu zaman ufku için, yalnızca zaman ufkuna kadar olan toplam kullanılır.

3.2. Çok Ajanlı Pekiştirmeli Öğrenme

Çok ajanlı pekiştirmeli öğrenme, pekiştirmeli öğrenmenin bir alt alanıdır. Paylaşılan bir ortamda bir arada bulunan çok sayıda öğrenme ajanının davranışını incelemeye odaklanır. Her ajan kendi ödülleriyle öğrenme gerçekleştirir ve kendi ödül puanını artırmak için eylemlerde bulunur. Bazı durumlarda bu bir ajanın yüksek ödül almak için yaptığı eylem, karmaşık grup dinamikleri sebepleriyle diğer ajanların çıkarlarına zıttır. ÇAPÖ algoritmalarında ajan sayısındaki artışın sebep olduğu problemlerin çözümüne yönelik birçok öğrenme şeması ve stratejisi önerilmiştir.

3.2.1. Öğrenme şeması

Boyutsallık, kısmi gözlemlenebilirlik ve durağan olmama durumu, ÇAPÖ için üç kritik zorluğu temsil eder. Bu bölümde, ÇAPÖ'nün merkezi veya dağıtık senaryolar için öğrenme ve yürütme süreçlerinde kullandığı öğrenme şemaları anlatılmaktadır.

Merkezileştirme ve yerelleştirme

Öğrenme algoritmalarında, bir ajan bir politikayı eğitim aşamasında öğrenir ve yürütme aşamasında onu uygular. ÇAPÖ algoritmalarında bu aşamalar, merkezi veya merkezi olmayan yaklaşımlarla ele alınabilir. Merkezileştirilmiş yaklaşımlarda ajanlar politikalarını geliştirmek için bilgi paylaşırken, dağıtık yaklaşımlarda bağımsız olarak öğrenirler. Eğitim ve yürütme aşamalarının merkezi veya dağıtık olmasına bağlı olarak üç ana öğrenme planı bulunmaktadır.

Merkezi eğitim merkezi yürütme: Merkezi eğitim merkezi yürütme (MEMY) şemasında, merkezi bir yönlendirici, kısmi gözlemlenebilirlik ve durağan olmama sorunlarını azaltan ortak bir politika üretebilmek için ajanlardan bilgi toplar. Bununla birlikte, MEMY'nin tüm ajan bilgilerini kullanarak politika üretme zorunluluğundan dolayı boyutsallık problemleri ortaya çıkabilir. Ayrıca, birbiriyle çelişen hedeflere sahip ajanlar birbirlerinin politikalarını bozarak öğrenmeyi zorlaştırabilir. Tek ajanlı PÖ algoritmaları yeterli olabilir; çünkü MEMY'de merkezi yönlendirme göz önünde bulundurulmaz. Böylece, MEMY'nin aksine, tamamen merkezi olmayan bir plan uygulanır.

Merkezi olmayan eğitim merkezi olmayan yürütme: Merkezi olmayan eğitim merkezi olmayan yürütme (MOEMOY) şeması, her ajanın ek bilgi alışverişi yapmadan bağımsız olarak öğrenmesine olanak tanır. Ajanlar birbirlerinin varlığından habersizdir ve dolayısıyla ortam ajan için durağan değildir. Dağıtık yürütme süreci ölçeklenebilirlik açısından fayda sağlar. Tamamen merkezi ve tamamen merkezi olmayan yaklaşımların sınırlamaları göz önüne alındığında, ara çözüm olarak bir şema daha bulunmaktadır.

Merkezi eğitim merkezi olmayan yürütme: Tamamen merkezi ve tamamen merkezi olmayan yaklaşımların eksikliklerinin üstesinden gelmek amacıyla merkezi eğitim merkezi olmayan yürütme (MEMOY) yöntemini önerilmiştir [64]. Eğitim aşamasında ajanlar, durağan olmama ve kısmi gözlemlenebilirliği azaltabilmek için ek bilgileri paylaşır ve ardından yürütme aşamasında bu bilgileri kullanır. MEMOY şeması, ajanların doğasına bağlı olarak kullanılabilecek iki popüler strateji içerir.

Parametre paylaşımı: Parametre paylaşımı (PP), homojen yapıdaki ajan grubunun iş birliği yaptığı büyük ölçekli ortamlarda kullanılan bir yaklaşımdır. PP, tüm ajanların eğitim

aşamasında tek bir sinir ağı kullanarak aynı anda öğrenmesine olanak tanır ve böylece boyutsallığın neden olduğu problemleri en aza indirgemeye yardımcı olur.

Merkezi eleştirmen, merkezi olmayan aktör: Ajanların heterojen olduğu durumlar, merkezi eleştirmen, merkezi olmayan aktör kullanımı için daha uygundur. Aktör-eleştirmen mimarisini temel alır. Eleştirmen, aktörü değerlendirmeye odaklandığından, yürütme aşamasında kullanılmaz.

Öğrenme stratejileri

Bu bölümde, insan bilişsel mekanizmalarından ilham alan bazı PÖ öğrenme stratejilerini sunulmaktadır.

Bellek: Bellek, ajanların ortam dinamiklerini analiz etmesine yardımcı olan bir mekanizmadır [65]. PÖ yaklaşımları genellikle sıralı problemler için kullanılır. Dolayısıyla, ajanlara bellek eklemek, ajanların ortam dinamiklerini anlama yeteneklerini güçlendirir.

Maskeleme: Maskeleme, ajanların istenmeyen eylemler gerçekleştirmesini engelleyerek ortamı daha güvenli hale getirir ve karar almayı kolaylaştırır [65]. Bir araştırmacı, bir eylemin ters etki yaratacağını önceden bildiğinde, ajanların bu eylemi yapmasını engelleyebilir. Maskeleme, eğitimi hızlandırır ve eylem alanını daraltarak yüksek boyut probleminin yan etkilerini azaltır. Öğrenmeyi kolaylaştırmanın bir başka yolu da keşfi azaltmaktır.

Müfreadat öğrenme: Müfredat öğrenme [66] zorluğu kademeli olarak artıran bir öğrenme yöntemini ifade eder. Örneğin, insanlar araba kullanmayı öğrendiklerinde genellikle trafiğin az olduğu alanlarda başlarlar ve ustalaştıklarında daha yoğun alanlara geçerler. ÇAPÖ'de ajanlar genellikle durağan olmama nedeniyle pratik politikaları öğrenemezler. Müfredat öğrenimi ile ajanlar durağan ortamlarda öğrenmeye başlar ve bu durağanlığı kademeli olarak kaldırarak görevi daha zor hale getirir. Öğrenmeyi kolaylaştırmanın bir başka yolu da hiyerarşik öğrenmedir.

Hiyerarşik pekiştirmeli öğrenme: Hiyerarşik pekiştirmeli öğrenme "böl ve fethet" algoritmalarıdır [67]. Ana politika alt politikalara bölünür ve sonrasında bu alt politikaların yeniden kullanımı sağlanır. Alt politikalar genellikle ana politikalardan daha az kaynak yoğundur; çünkü daraltılmış bir durum-eylem uzayında çalışırlar. Böylece boyutsallığın yan etkileri azaltılır.

3.2.2. Markov/Rastsal oyunlar

MO'ler veya SO'ler [25], ajanlar arasındaki ilişkiyi göz önünde bulundurarak MKS yapısını çok ajanlı sistemlere uyarlar. $N>1$ ajan sayısı için; S durum uzayı ve A_i, i ajanının eylem uzayını ifade ettiği durumda tüm ajanların ortak eylem uzayı $A := A_1 \times A_2 \times ... \times A_N$ şeklinde gösterilir.

s durumunda, her ajan i bir A_i eylemi seçer ve $a = [a_i]_{i \in N}$ ortak eylemi yürütülür. s durumundan yeni s' durumuna geçiş $P : S \times A \times S \mapsto 0, 1$ olasılık fonksiyonu tarafından yönetilir. Her i ajanı, R_i ödül işlevi tarafından tanımlanan bir anlık ödül r_i alır, $S \times A \times S \mapsto \mathbb{R}$. Böylece, MO, γ indirim faktörü için, $(N, S, (A_i)_{i \in N}), P, (R_i)_{i \in N}, \gamma)$ şeklinde tanımlanır. MO'deki geçiş ve ödül işlevlerinin ortak eylem uzayının A'ya bağlı olduğu unutulmamalıdır. Her i ajanı, uzun vadeli ödülünü en yüksek düzeye çıkaracak olan en uygun politikayı $\pi_i^* : S \mapsto A_i$ bulmaya çalışır. Tüm ajanların ortak $\pi(a \mid s) = \coprod_{i \in N} \pi_i(a^i \mid s)$ olarak tanımlanır. Dolayısıyla, ajan i'nin değer fonksiyonu aşağıdaki gibi tanımlanır:

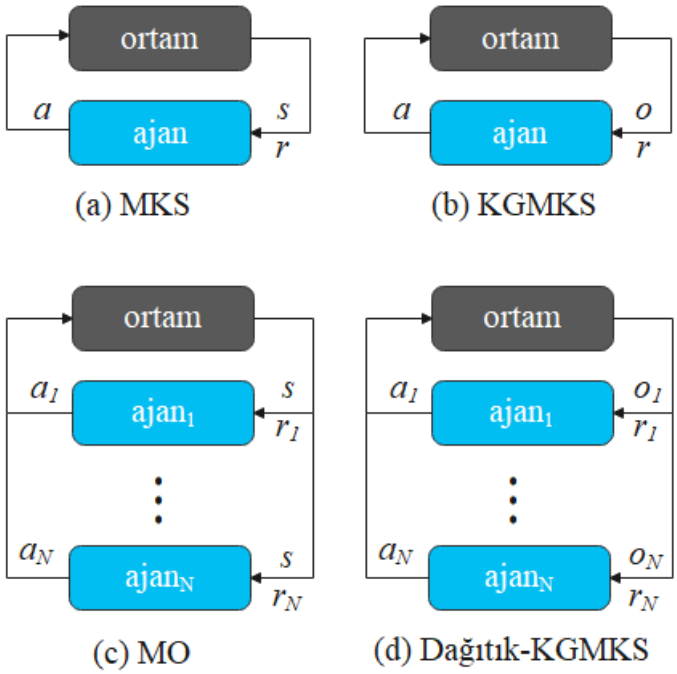
$$V \frac{\pi}{i}(s) = \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_i(s_t, a_t, s_{t+1}) \mid (a_t \sim \pi(s_t), s_0 = s) \right] \tag{3.4}$$

ÇAPÖ sistemlerinin karmaşıklığının temel sebebi ajan politikalarının durağan olmaması ve öğrenme anında değişikliğe uğramasıdır. ÇAPÖ problemleri 3 konsept altında ele alınır; tam işbirlikçi, tam rekabetçi ve karma. Tam işbirlikçi ortamlarda, tüm ajanlar aynı ödül fonksiyonuna $R_i = R$ sahiptir ve dolayısıyla aynı değer veya durum-eylem fonksiyonuna sahiptir. Tam işbirlikçi MO'ler çok ajanlı MKS (ÇAMKS) olarak da adlandırılır. ÇAMKS'ler problemleri basit düzeyde alır; tek ajanlı PÖ algoritmalarını merkezi bir

kontrol ünitesi ile çok ajanlı PÖ'lere uygulama temeline dayanır. Öte yandan, tam rekabetçi MO'ler $(\sum_i R_i = 0)$ ve genel toplam MO'lar $(\sum_i R_i \in \mathbb{R})$ bir Nash dengesi aranarak ele alınır.

3.2.3. Ağa bağlı Markov oyunları

İşbirlikçi MO'ler veya Dağıtık-KGMKS 'ler, aynı ödül sinyalini $(R_1 = ... = R_N = R)$ paylaştıklarından dolayı yalnızca homojen işbirlikçi ajanlar için uygundur. Bununla birlikte, gerçek dünya uygulamalarının çoğu, farklı tercihleri ve hedefleri olan heterojen ajanları içerir. Ek olarak, aynı ödül işlevinin paylaşılması, tüm ajanların küresel bir değer tahmini için karmaşık bir durum-eylem işlevi gerektirir. Şekil 3.2.'te MKS tiplerinin temel farklılıkları gösterilmiştir.



Şekil 3.2. MKS türleri

Bu eksikliklerin üstesinden gelmek için Ağa Bağlı MO, bir iletişim ağı yoluyla paylaşılan bilgilerden yararlanarak işbirlikçi ajanları farklı ödül işlevleriyle modellenen MO'lar, $G_t = (N, \mathcal{E}_t)$'ün N düğümü t zamanında bir dizi kenar $\mathcal{E}_t$ ile birbirine bağlayan, zamanla değişen bir iletişim ağı olduğu $(N, S, (A_i)_{i \in N}), P, (R_i)_{i \in N}, (G_t)_{t \geq 0})$ olarak tanımlanır. Bir düğüm $(i, y) \in \mathcal{E}_t, \forall i, j$, hem i hem de j ajanlarının t anında iletişim kurabileceği ve

karşılıklı olarak bilgi paylaşabileceği anlamına gelir. Böylelikle, ajanlar yerel ve komşu bilgilerini bilirler. Herhangi bir $(s,a,s') \in S \times A \times S$ için takımın ortalama ödülü aşağıdaki şekilde ifade edilir:

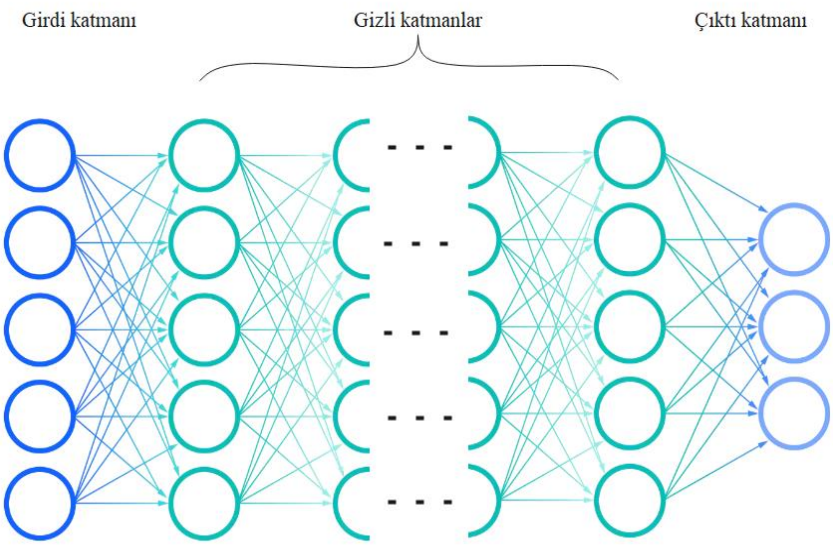
$$\bar{R}(s,a,s') = \frac{1}{N-1} \sum_{i \in N} R_i(s,a,s') \tag{3.5}$$

Ajanlar Eş.3.5. ile ödülü en üst düzeye çıkararak en uygun ortak politikayı öğrenmeye çalışırlar. Özetlemek gerekirse, Ağa Bağlı MO'ların klasik MO'lara kıyasla avantajları şunlardır: (i) farklı ödül fonksiyonlarına sahip heterojen ajanları modelleyebilmesi; (ii) merkezi olmayan ÇAPÖ algoritmalarının tasarımını kolaylaştıran komşular arası iletişimi dikkate alarak koordinasyon maliyetinin azaltması.

3.3. Derin Öğrenme

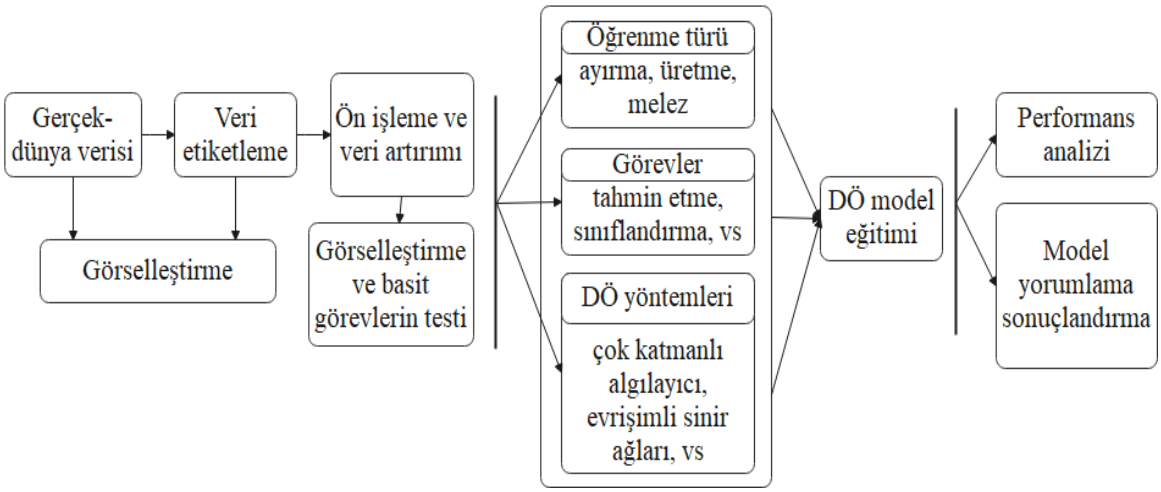
Derin öğrenme (DÖ) bir makine öğrenmesi yaklaşımıdır ve sinir ağları insan beyninin bir kopyası olarak kullanılır. İnsan beyninin en temel birimi olan nörona dayanmaktadır. Derin öğrenme, nöronların bir sinir ağı modeli oluşturmak için birlikte nasıl çalıştıklarını açıklamak için kullanılan bir terimdir. Derin öğrenme modeli, bir sinir ağının nihai ürünüdür. Çoğu zaman, derin öğrenmede, derin öğrenme modelinin veriler üzerinde tekrar tekrar eğitim yaptığı yapılandırılmamış veriler kullanılır.

Tipik bir derin sinir ağı, giriş ve çıkış katmanları dahil olmak üzere birden çok gizli katman içerir. Şekil 3.3., derin bir sinir ağının (gizli katman=N ve $N \geq 2$) genel yapısını göstermektedir. Bununla birlikte, DÖ teknikleri şu şekilde sınıflandırılabilir; (i) Denetimli: etiketli eğitim verilerini kullanan göreve dayalı bir yaklaşım, (ii) Denetimsiz: etiketlenmemiş veri kümeleri analiz eden veriye dayalı bir süreç, (iii) Yarı denetimli: hem denetimli hem de denetimsiz yöntemlerin birlikte kullanıldığı yaklaşım ve (iv) Pekiştirme: önerilen yöntemlerin temel aldığı, ortam odaklı bir yaklaşım.



Şekil 3.3. Derin sinir ağı

Bir DÖ modeli, genellikle makine öğrenimi modellemesiyle aynı işleme aşamalarını takip eder. Şekil 3.4.'te, gerçek dünya sorunlarını çözmek için veri anlama ve ön işleme, DÖ model oluşturma ve eğitim ile doğrulama/yorumlama gibi üç adımından oluşan iş akışı gösterilmiştir [70].



Şekil 3.4. Derin öğrenme iş akışı

Makine öğrenmesi modellemesinden farklı olarak, DÖ modelinde özellik çıkarımı manuel değil otomatiktir. K-en yakın komşu, destek vektör makineleri, karar ağacı, rastgele orman, lineer regresyon, birliktelik kuralları, k-ortalama kümeleme gibi yöntemler çeşitli uygulama alanlarında yaygın olarak kullanılan makine öğrenme tekniklerine örneklerdir [71].

DÖ’nün özellikleri aşağıdaki gibi özetlenebilir:

- Veri Bağımlılıkları: Veriye dayalı bir model oluşturmak için genellikle büyük miktarda veriye bağlıdır. Bunun nedeni, veri hacmi küçük olduğunda, derin öğrenme algoritmalarının genellikle düşük performans göstermesidir [72].
- Donanım Bağımlılıkları: DÖ algoritmaları, büyük veri kümelerine sahip bir modeli eğitirken büyük hesaplama işlemleri gerektirir. Hesaplamalar ne kadar büyük olursa, bir GPU'nun bir CPU'ya göre avantajı o kadar fazla olur; GPU çoğunlukla işlemleri verimli bir şekilde optimize etmek için kullanılır [73].
- Özellik Mühendisliği Süreci: Özellik mühendisliği, etki alanı bilgisini kullanarak ham verilerden özelliklerin (özellikler ve nitelikler) çıkarılması işlemidir. DÖ ile diğer makine öğrenimi teknikleri arasındaki temel ayrım, yüksek düzey özellikleri doğrudan verilerden çıkarma girişimidir [74].
- Model Eğitimi ve Yürütme süresi: Genel olarak, algoritma eğitimi, DÖ algoritmasındaki çok sayıda parametre nedeniyle uzun zaman alır. Bu nedenle, model eğitim süreci daha uzun sürer [75].
- Kara Kutu Algısı ve Yorumlanabilirlik: DÖ ile MÖ'yü karşılaştırırken önemli bir faktördür. Bir derin öğrenme sonucunun, yani “kara kutu”nun nasıl elde edildiğini açıklamak zordur. Öte yandan, makine öğrenimi algoritmaları, özellikle kural tabanlı makine öğrenimi teknikleri, insanlar için kolayca yorumlanabilen kararlar vermek için açık mantık kuralları sağlar [76].

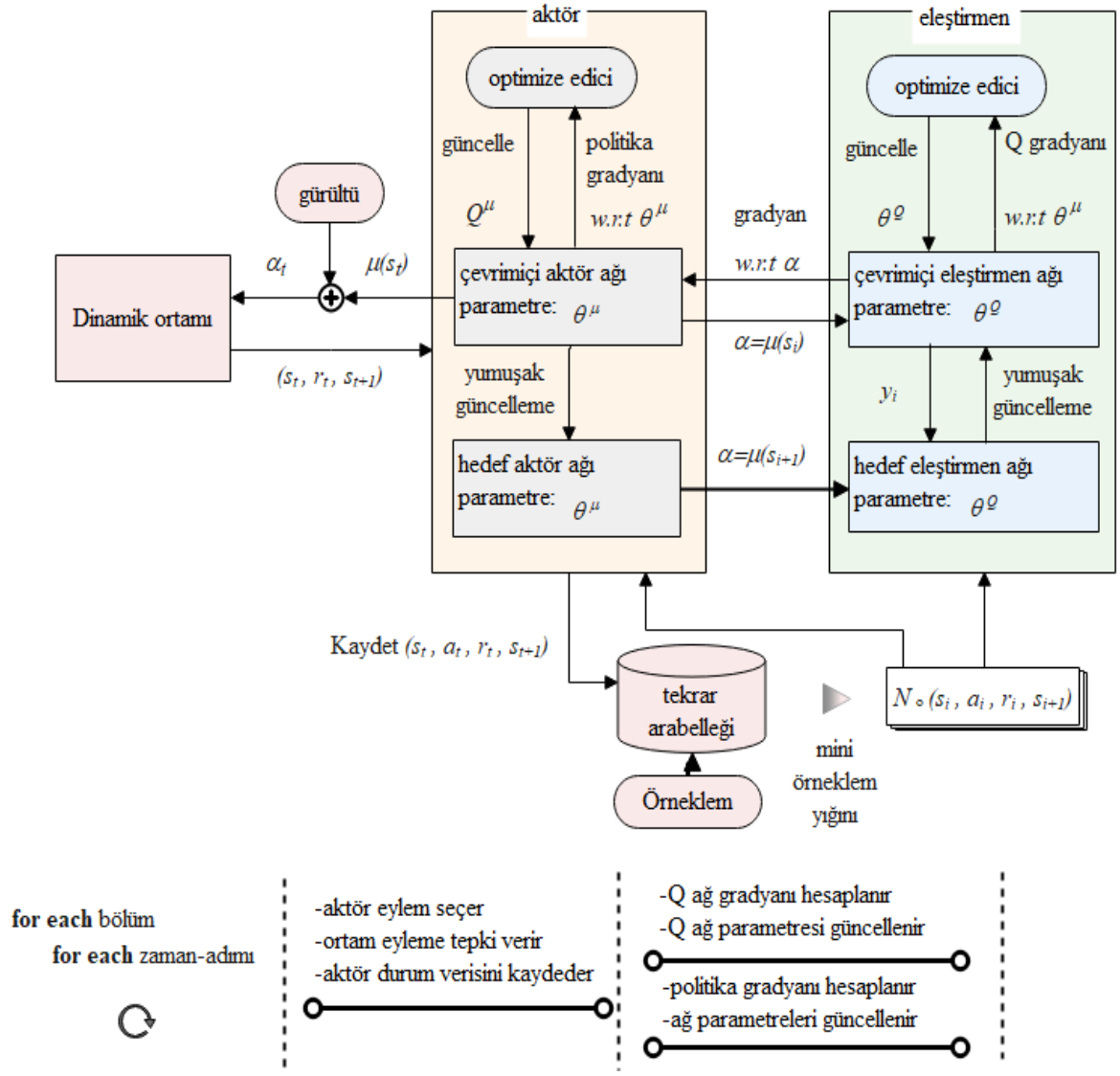
3.4. Çok Ajanlı Derin Deterministik Politika Gradyanı

ÇADPÖ algoritmaları, ortak bir ortamda etkileşim halinde olan çok sayıda ajandan (robotlar, makineler, arabalar vb.) oluşan sistemlerle ilgilenir. Her ajan, her zaman adımında bir karar verir ve önceden belirlenmiş bireysel bir hedefe ulaşmak için diğer ajanlarla birlikte çalışır. ÇADPÖ algoritmalarının amacı, tüm ajanların ortak amaca ulaşması için her ajanın bir politika öğrenmesidir. Yani ajanlar, çevre ile etkileşim yoluyla uzun vadeli kümülatif indirimli ödülü en üst düzeye çıkarmak için en uygun politikayı öğrenmeyi amaçlayan öğrenebilir birimlerdir. Ortamların karmaşıklığı veya problemin karmaşık doğası nedeniyle, ajanları eğitmek zorlu bir görevdir ve ÇADPÖ 'nün bunlarla ilgilendiği bazı problemler belirlenimsiz problemler olarak kategorize edilir [77].

DDPG, derin Q-öğrenmenin [78] temel başarısını sürekli eylem alanına uyarlayan bir aktör-eleştirmen yöntemidir. ÇADDPG algoritması, ajanların yalnızca yerel bilgilere erişebildiği ve diğer ajanlarla politikalarını paylaştığı aktör-eleştirmen politika gradyan yöntemlerinin bir uzantısı olarak önerilmiştir. Politika gradyanı (PG) yaklaşımındaki ana fikir, verilen gradyan yönünde adımlar atarak belirli bir hedefi en yükseğe çıkarmak için politika parametresi ayarlamaktır. Sistem içerisinde bir eleştirmen kullanmak, ortamın dinamik durumunu ele almak için yaygın bir çözümdür. Bu nedenle, bu merkezi eleştirmen, yerel gözlemlere sahip ajanların esnekliğini artırmak için güvenilir bir rehber olarak kullanılabilir. ÇADDPG'de her ajanın iki ağı vardır: bir aktör ağı ve bir eleştirmen ağ. Aktör ağı, ajanın bulunduğu duruma göre yürütülecek eylemi hesaplarken, eleştirmen ağ aktör ağının performansını iyileştirmek için eylemin sonuçlarını değerlendirir. Eleştirmen ağ güncellemesi için kullanılan deneyim tekrar arabelleği, eğitim verilerindeki korelasyonları kırmaya ve eğitimi daha kararlı hale getirmeye yardımcı olur. Eğitim aşamasında her ajan, aktörün yerel gözlemlere erişiminin olduğu bir DDPG algoritması tarafından eğitilir. Merkezileştirilmiş eleştirmen ise girdi olarak tüm durum-eylemleri birleştirir ve buna karşılık gelen Q-değerini elde etmek için yerel ödül fonksiyonunu kullanır. Yürütme aşamasında, eleştirmen ağ kaldırılır ve ajanlar sadece aktör ağı kullanır. Bu, yürütmenin merkezi olmadığı anlamına gelir. Aslında ÇADDPG, DDPG'nin çok ajanlı versiyonu olarak düşünülebilir. Temel amaç yürütmeyi merkezden uzaklaştırmaktır. Q-öğrenme ve DDPG, gruptaki diğer ajanların bilgilerini kullanmadıkları için çok ajanlı ortamlarda düşük performans gösterir. ÇADDPG yaklaşımı, tüm ajanların gözlemlerini ve eylemlerini kullanarak bu zorluğun üstesinden gelir.

3.5. Önerilen Yöntemler

Bu tez çalışması kapsamında Şekil 3.5.'te görüldüğü gibi dinamik ortamda kapsama problemlerine yönelik bir alt yapı geliştirilmiştir. Geliştirilen alt yapı temel problem gereksinimlerinden soyutlanarak hem alan kapsama hem de İN kapsama problem için kullanılabilir şekilde tasarlanmıştır.



Şekil 3.5. Önerilen yöntemler için tasarlanan alt yapı

Bu bölümün devamında kapsama problemlerine yönelik önerilen yöntemler ve detayları hakkında bilgi verilmiştir.

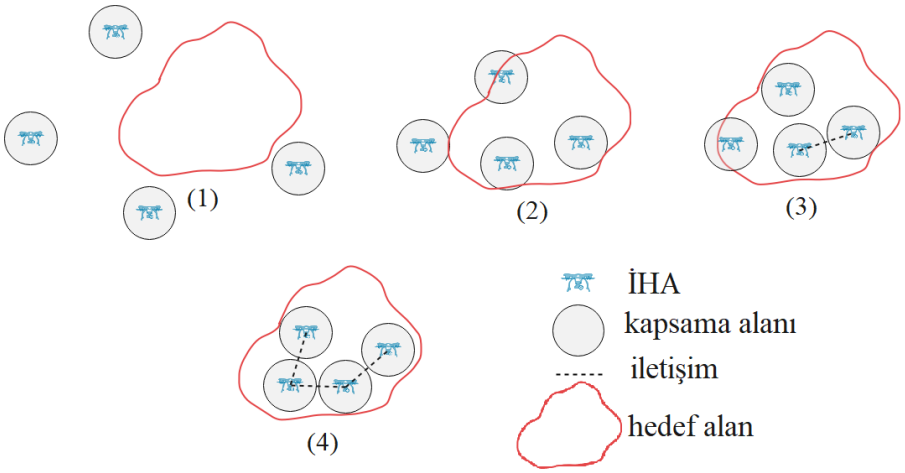
3.5.1. Alan kapsama

Alan kapsama problemi ÇADDPG 'ye dayalı çok ajanlı sistem modeli üzerine kurgulanmıştır. Önerilen bu yönteme Dinamik Alan Kapsama Merkezi Olmayan Karar Verme (DAKMOKV) [79] yöntemi adı verilmiştir. Bu bölümde, DAKMOKV'un genel sistem mimarisinin açıklanmasının ardından İHA düğümleri ile bağlı graf oluşturma stratejisine ayrıntılı şekilde değinilmiştir. Son olarak, genel bakış ve DAKMOKV'un sözde kodu sunulmuştur.

Problem Formülasyonu

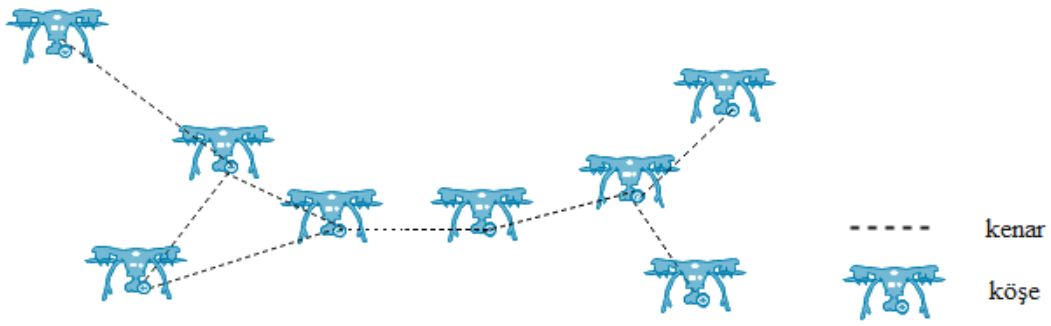
ÇAS'ta her İHA bir ajan tarafından temsil edilir. Ajanlar, beş olası eylemden birini gerçekleştirebilir: kuzey, batı, güney, doğu veya mevcut durumda kalma.

Şekil 3.6.'da gösterildiği gibi, bir ajan her zaman adımında hareket ettikten sonra grubun etki alanına girebilir veya çıkabilir. Önerilen yöntem, hareketli ajanlarda ağ iletişiminin sürekliliğini garanti etmektedir.



Şekil 3.6. ÇAS hareket modeli

Mobil ve sınırlı kapsama kabiliyetine sahip ajanlar, konumlarını ortam koşullarına ve diğer ajanların konumlarına göre ayarlar. Bu kavram, Şekil 3.7. gösterildiği gibi bir yönlendirilmemiş ajan grafı oluşturmamıza yardımcı olur.



Şekil 3.7. Bağlı yönsüz ajan grafı

Varsayım 1: Her ajanın kendi konumunu bildiği varsayılır. Hedef alana ulaşabilmek, hedef

alandaki diğer ajanları keşfedebilmek ve bilgi paylaşmak için iletişim mesafesi içerisindeki ajanlar ile bağlı-ajan yapısı oluşturmak için coğrafi yaklaşım kullanılır.

ÇAS'taki bir ajan, tüm $i = 1,..., N$ için $A_i \in A$ olacak şekilde A_i ile temsil edilir. t anında, ajanın konumu $(x_i^A(t), y_i^A(t))$ olarak tanımlanır. G^N tüm ajanların kümesi için:

$$G^N = \{(x_1^A(t), y_1^A(t)), (x_2^A(t), y_2^A(t)), ..., (x_m^A(t), y_m^A(t))\} \quad (3.6)$$

şeklinde gösterilir.

Varsayım 2: Homojen ajanların 2 boyutlu (2B) ilgi alanına konumlandığı varsayılmıştır. Yükseklik kısıtlamaları dikkate alınmamıştır. A_i ve A_j arasındaki mesafe $\left|A_i A_j\right|$, d_{ij} ile temsil edilmiştir. G_i^C bağlı ajanların listesidir, $G_i^C \subseteq G^N$, ve A_i ve A_j 'in bağlı olduğunu göstermek için Eş. 3.7. kullanılır.

$$\left|(x_i^A(t), y_i^A(t)), (x_j^A(t), y_j^A(t))\right| \leq d_{ij} \quad (3.7)$$

Varsayım 3: Her ajanın iletişim mesafesini kullanarak aynı büyüklükte bir dairesel algılama bölgesi oluşturduğu varsayılmıştır. Ajanları birbirine bağlamak için Derinlik Öncelikli Arama (DÖA) algoritması kullanılmıştır [80]. DÖA 'de, erişilebilir ajanların bulunabilmesi için her kenar üzerinden geçilmektedir. DÖA için sözde kod, Çizelge 3.1.'de verilmiştir.

Çizelge 3.1. DÖA algoritması

```
if geçerli tepe noktası  keşfedilmemiş kenarlara sahipse, then
    Keşfedilmemiş kenarı geç
if ulaşılan tepe noktası ziyaret edilmişse then
    return önceki köşeye dön
else
    if geçerli tepe noktası   başlangıç tepe noktası değilse, then
        ilk kez   'ya ulaşmak için kullanılan kenarı geç
```

Ajanlar, her adımda bağlandıkları erişilebilir ajanların kapsadığı alanın birleşimini alırlar. π işleminin hesaplanması için formül şu şekilde tanımlanır:

$$\pi[i] = x \in \bigcup M \quad \Leftrightarrow \exists U \in M, x \in U \tag{3.8}$$

$$M = \{c(A_1), c(A_2), ..., c(A_M)\}, \quad A \in G_i^C \tag{3.9}$$

Burada $c(A_i)$, hedef alanda A_i ajanı tarafından kapsanan alanı belirtir. Ayrıca her adımda her ajan, erişebileceği tüm ajanların bilgisini (kapsadığı alanı) paylaşır. Bu nedenle süreç, tüm $A_L \in G_i^C$ için $L \in \{1, ..., N\}$ olacak şekilde $\pi[i...L] = \{\pi[1], \pi[2], ..., \pi[L]\}$ alt sürecinin bir dizisi haline gelir. Son olarak, ajan için toplam alan kapsamını şu şekilde tanımlanır:

$$\pi_i^L = \bigcup \pi[i...L] \tag{3.10}$$

DAKMOKV 'ın amacı, hedef alan ile ÇAS'ın kapsadığı alanın kesişimini en üst düzeye çıkarmaktır:

$$z = T \bigcap \pi \tag{3.11}$$

Burada T hedef alanı belirtir. Ek olarak, çarpışmaları önlemek için her ajanın bir sonraki konumunu düzenlemesi gerekir. Bir ajan ile farklı bir ajan arasındaki mesafe, yarıçaplarının (f) toplamından daha az ise ajanlar daha az ödül alırlar.

$$p_i = \prod_j w(ij), \qquad j \in \{1, ..., N\} \text{ tüm} \qquad A_j \in G_i^C \tag{3.12}$$

$$w(ij) = \begin{cases} 1, & f_i + f_j \le d_{ij} \\ \gamma_r, & f_i + f_j > d_{ij} \end{cases} \tag{3.13}$$

γ_r çarpışma için alan kapsamı indirim faktörünü ifade eder. t anındaki kapsanan alan için $\mu(t)$ kullanılır ve böylece ödül fonksiyonu aşağıdaki gibi tanımlanır:

$$r_i = \frac{1}{\mu(T)} p_i * \mu(z) * 100 \tag{3.14}$$

ÇADDPG'ye Dayalı ÇAS Modeli

İHA'lar arasındaki işbirliği karmaşıklığını çözmek için ÇAS modeli oluşturulmuştur. Bu modelde her ajan, gerçek bir İHA ile aynı hareket modeline sahip bir İHA'yı temsil etmektedir.

Ajan politikası μ ile politika parametresi ise θ ile temsil edilir. N ajanlı ÇAS modelinde, durum geçişleri için politikalar $\mu = \{\mu_1, \mu_2, ..., \mu_N\}$, parametreler ise $\theta = \{\theta_1, \theta_2, ...\theta_N\}$ ile ifade edilir. Ayrıca, model, politika bağımsız eğitim için bir deneyim tekrar arabelleği kullanır. Ajan, her bir adımda ortak durumu, sonraki ortak durumu, ortak eylemi ve ajanların her biri tarafından alınan ödülleri gösteren bilgileri $(x, x', a_1, ..., a_N, r_1, ..., r_N)$ şeklinde saklar. Ardından, ajanı eğitmek için tekrar arabelleğinden bir örnekleme yapar. Örnekleme alınmış bilgiler kullanılarak ajanın eleştirmeni güncellenir. Böylelikle eleştirmen için örneklenmiş zamansal fark hatası kullanan kayıp fonksiyonu şu şekilde tanımlanır:

$$y = r_i + \gamma Q_i^{\mu'}(x', a'_1, ..., a'_N)\big|_{a'_j = \mu'_j(o_j)} \tag{3.15}$$

$$L(\theta_i) = \frac{1}{S}\sum_j (y^j - Q_i^{\mu}(x^j, a_1^j, ..., a_N^j))^2 \tag{3.16}$$

Yukarıdaki eşitlikte a', γ, S, o ve Q_i^{μ} sırasıyla; sonraki ortak eylemi, indirim faktörünü, tekrar arabelleğinden rastgele seçilen örneklemenin boyutunu, ortamın kısmi gözlemlerini ve merkezi eylem-değer fonksiyonunu gösterir. Aktörün güncellenmesi için örneklenen politika gradyanı aşağıdaki gibi tanımlanır:

$$\nabla_{\theta_i} J \approx \frac{1}{S} \sum_j \nabla_{\theta_i} \mu_i(o_j^i) \nabla_{a_i} Q_i^{\mu}(x^j, a_j^i, ..., a_N^j)\big|_{a_i = \mu_i(o_i^j)} \quad (3.17)$$

İHA Düğümlü ÇAS

Optimum konumları bulmak ve çarpışmaları önlemek için ödül yapısı tasarlanmıştır. Bir ajan, iletişim mesafesindeki herhangi bir ajana bağlamaya çalışan hareketli bir tepe noktasını temsil eder. Her ajan, her adımdan sonra bağlanan ajanların konum ve alan kapsama bilgilerini gözlemler. Böylece her ajan dolaylı olarak erişebileceği bir ajan listesine sahip olur. Erişilebilir ajanların alan kapsamı hedef alanla ne kadar örtüşürse, o kadar büyük ödül alır. Ödül sözde kodu Çizelge 3.2.'de verilmiştir.

Çizelge 3.2. Ödül yapısı

Başlat: ödül
Getir: Algoritma 1 kullanarak ajanın konum ve alan kapsama bilgileri
Hesapla: Eş.3.10'u kullanarak erişilebilir ajanların kapsadığı alanın birleşimi
Hesapla: Eş.3.11'i kullanarak hedef alan ile kapsama kesişim
Hesapla: Eş.3.12'yi kullanarak çarpışma cezası ödülünü
Hesapla: Eş.3.14'u'u kullanarak ödül
return

Bağlı ajan sayısını artırmanın toplam kapsamı artırmak için yeterli olmayabileceği unutulmamalıdır. Birbirine yakın ajanların kapsadığı birçok ortak alan olabilir.

DAKMOKV'un İnşası

DAKMOKV, eğitim katmanı ve yürütme katmanından oluşan bir dinamik alan kapsama yöntemidir. Eğitim katmanında, eylemlere göre politika eğitimi için ödül yapısı kullanılır. Her ajan yalnızca yerel gözlemlere sahiptir ve durumunu paylaşmak için iletişim mesafesindeki diğer ajanlarla iletişim kurabilir. Her ajan merkezi olmayan bir şekilde optimum konum belirlemek için bir politika öğrenir. Ayrıca, paylaşılan durumları kullanarak yeni eylemlerde bulunabilir ve konumunu değiştirebilir.

[81]'den esinlenerek, alan kapsama sürecinde davranış stratejileri üretmek için “çekme” ve “kaçınma” yaklaşımları kullanılmıştır. Bir ajan, hedef alandaki kapsanan alana yönlenirken ve kapsadığı alanı artırırken olumlu bir ödül alır. Öte yandan, bir ajan başka bir ajana çok yakın olduğunda negatif bir ödül alınır. Böylece her bir ajan, diğerlerine istenilen mesafede bulunarak olumlu bir ödül almayı amaçlar ve çok yakın olduğu durumda olumsuz bir ödül almaktan kaçınmayı öğrenir.

Çizelge. 3.2.'de gösterildiği gibi tasarlanan ödül yapısı, kolektif davranışı zorunlu kılar ve bu nedenle, açık bir ödülün paylaşılmasına gerek kalmaz. Bir ajanın kapsamasını bağlı ajan kapsamaları ile birleştirmek, sırasıyla ajana özgü ve gruba özgü ödülü temsil eder. Ajanların hedef bölgeye konumlandırılması bu grup ödülüne bağlıdır. Bu yaklaşım, tüm ajan ödüllerinin ortalamasını almak gibi ortak ödülü dikkate alma ihtiyacını ortadan kaldırır. Böylelikle ajanlar, grup ödülü sonuçlarından doğrudan etkilenirler.

Önerilen yöntem, model bağımsız tasarımı nedeniyle tekrar arabelleğini kullanarak öğrenme sürecini işletir. Merkezi modül, ajanlara eğitim süresi boyunca politikalarını nasıl güncelleyecekleri konusunda rehberlik eder. Eğitim süreci bölümlere ayrılmıştır ve her bölümün başında ajanların durumu yeniden belirlenir. Eğitim sürecinde ajanlar, tekrar oynatma belleğinde eylem, durum ve ödülden oluşan bilgiyi saklar. Her zaman adımında, tekrar oynatma arabelleğinden örneklem yapılır. Ardından Eş.3.17. kullanılarak aktörlerin politika gradyanı güncellenirken, eleştirmenler Eş.3.16. kullanılarak güncellenir. Her ajan, kapsadığı alanı artırmak için optimum ortak eylemi belirlemelidir. Ardından ajanlar, performanslarını ödül yapısına göre değerlendirir. DAKMOKV’un sözde kodu, Çizelge 3.3.'te verilmiştir.

Çizelge 3.3. DAKMOKV

Başla: Öğrenme ve ödül indirim faktörü
for $bölüm = 1$'den M 'ye kadar do
  Başla: N ve hedef alan
  Getir: x ilk durumu
    for $t = 1$'den maks_adım_sayısı'na kadar do
    her ajan i için, rasgele eklenmiş gürültü ve seçilen eylem N_t , $a_i = \mu_{\theta_i}(o_i) + N_t$
mevcut politika ve gözlem
    Yürüt: eylem kümesi $a = (a_1, ..., a_N)$
    Hesapla: Çizelge 3.2. ve yeni x' kullanarak ödül r
    Sakla: (x, a, r, x') 'yi yeniden yürütme belleği D 'ye
    $x \leftarrow x'$
    for ajan $i = 1$'den N 'ye kadar do
      Örneklem: D 'den mini-yığın S
      Güncelle: Eş.3.16.'yı kullanarak eleştirmen
      Güncelle: Eş.3.17.'yi ve örneklem alınmış politika gradyanı ile eleştirmen
    end for
  Güncelle: Tüm ajanların hedef ağ parametreleri
  end for
end for

Yürütme süresi boyunca merkezi modül kaldırılır. Yürütme sürecinde, aktör ağı, yerel gözlem yetenekleri ile eylem seçimi için kullanılır. Ajanlar, politika eğitimine rehberlik edecek yeterli bilginin eğitim sürecinde alınmış olmasını amaçlar.

3.5.2. İlgi noktası kapsama

Hedef alanda etkili kapsama için İHA'lar bulundukları ortamlara uygun davranışlar sergilemelidir. Bununla birlikte, hedef alanda İHA'lar arasındaki bağlantıyı korumak ve enerji tasarrufu sağlamak için İHA hareketleri azaltılmalıdır. Bu minvalde yapılan bu çalışmada, bir ajan grubundaki her bir İHA aynı hareket modeline sahip bir ajan olarak

modellenmiştir. Birden fazla ajanın ortaklaşa çalışması ile aşağıdaki hedeflerin karşılanması amaçlanmıştır:

1. Öğrenme yetenekleri ile donatılmış dağıtık bir sistem inşa etmek;
2. En fazla sayıda İN kapsamak;
3. Ajan hareketlerinden kaynaklanan enerji tüketimini en aza indirmek;
4. Hedef bölge sınırlarını aşmadan ajanlar arasındaki bağlantıyı sağlamak;
5. Ajan hareketlerini optimize etmek ve ajanlar arasındaki çarpışmaları önlemek;
6. Bilinmeyen dinamik ortamlarda görev devamlılığı sağlamak.

Ajanlar öğrenme yetenekleri sayesinde hedef alandaki değişikliklere tepki verebilir ve kolektif başarı elde edebilir. Ayrıca dağıtık sistem mimarisine uygun tasarım ile de merkezi kontrolden bağımsız karar alabilmeleri sağlanabilir. PÖ yaklaşımıyla geliştirilen yöntemde, merkezi kontrolden bağımsız kolektif başarı üretmeye çalışan mobil ajanlar elde edilmesi amaçlanmıştır.

Problem Formülasyonu

Bu çalışmanın temel amacı; dinamik alanda İN kapsama ile görevli bir grup İHA'nın, ÇADPÖ yaklaşımı ile modellendiği bir yöntem inşa etmektir. ÇADPÖ yaklaşımında İHA'lar, dinamik ortamda görev yapan mobil ajanlar olarak kabul edilir ve ortak bir hedef için etkileşime girer. Bu yolla, kapsama en üst düzeye çıkarılırken düşük enerji tüketimi ile yüksek adillik indisi sağlayan stratejiler üretilerek akıllı bir sistem elde edilebilir.

Ajan tabanlı kapsama ve enerji tüketimi konularını sadeleştirmek için hedef alan ızgaralara bölünmüş ve her ızgaranın merkezi ızgara merkezi (IM) olarak adlandırılmıştır. Her ajanın makul bir sürede bir IM'ye konumlanmış olması amaçlanmaktadır.

Varsayım 1: Ajan takımı içerisindeki tüm ajanların aynı özelliklere sahip olduğu ve 2B bir düzlemde hareket ettikleri varsayılmıştır. Takım içerisindeki her bir ajan A_i ile temsil edilir, $A_i \in A \mid i = 1,...,N$.

Varsayım 2: Her ajanın kendi konumunu bildiği varsayılmıştır. Önerilen yöntemde, hedef

alana ulaşmak, hedef alandaki diğer ajanları keşfetmek ve iletişim aralığındaki ajanlarla etkileşimde bulunabilmek için coğrafi yaklaşım kullanır, $(x_i^A(t), y_i^A(t))$. Ajanlar, takımda kaç ajan olduğu bilgisine sahiptir; ancak etkileşimde bulunamadıkları yani iletişim mesafesi içerisinde bulunmayan ajanların konum bilgisine sahip değildir. Ortamdaki tüm ajanların t anındaki konumları $A^N = \{(x_1^A(t), y_1^A(t)), (x_2^A(t), y_2^A(t)), ..., (x_m^A(t), y_m^A(t))\}$ ile gösterilir.

Varsayım 3: Her ajanın, dairesel bir şekle sahip olduğu (çap: $\varnothing_A$) ve dairesel bir algılama bölgesine sahip olduğu varsayılmaktadır. A_i ve A_j arasındaki mesafe $d_{ij} = |A_i A_j|$ şeklinde ifade edilir. A_i ve A_j ajanlarının etkileşime girebilmesi için d_{ij} eşitliğin sağlanması gerekmektedir (Bkz. Eş. 3.7.).

Her ajanın iletişim/algılama aralığı gibi bağlantı sınırlamaları vardır. Her bir öğrenme adımında her ajan, hedef alanı ızgaralara ayırır. İletişim mesafesi, ızgara ayırma mesafesinden daha kısa olduğunda ajanlar arası bağlantı kopacaktır. Bununla birlikte iletişimin yeniden sağlanabilmesi adına yapılabilecek herhangi bir ajan eylemi, diğer ajan konumlarının değişmesine sebep olabilir. Uygun çözüm bulunabilmesi amacıyla yapılan her bir eylem, enerji tüketimini artırır.

Bu bölümde, hedef alanda bağlı ajanların (bilgi alışverişi yapabilen) yüksek adil indisi şemsiyesi altında en az enerji tüketimi ile en fazla İN kapsama elde edilebilmesi amacıyla önerilen yöntemin detayları sunulmuştur.

Izgara ayrıştırma

Sistem içerisinde her bir İHA bir ajan tarafından temsil edilir. İki boyutlu bir ortamda ajanlar, hedef alanda ızgaralar oluşturarak merkezlerine konumlanmayı öğrenir. Hedef alan düzenli bir şekil olmayabilir. Düzenli veya düzenli olmayan hedef alan için bir soyut alan oluşturulur. Soyut alan, hedef alanı içeren en küçük düzenli dörtgen alandır. Ajanlar, ızgaralara ayrılmış soyut alan merkezlerine yönlenir. Ajanlar, kendilerine en yakın ve en çok İN içeren ızgaraya giderek en hızlı ve en uygun çözümü üretmeye çalışır. Aynı zamanda, hedef alanda ajanlar arası iletişimin sağlanması da amaçlanır. T hedef alanı ve

k^T hedef alanın köşeleri olmak üzere, hedef alanı içeren en küçük düzenli dörtgen alan $\left\{(x^T_{\min}, y^T_{\max}), (x^T_{\max}, y^T_{\max}), (x^T_{\max}, y^T_{\min}), (x^T_{\min}, y^T_{\min})\right\} | \left\{(x^T_i, y^T_i), ..., (x^T_n, y^T_n)\ \forall x^T, y^T \in k^T\right\}$ şeklinde ifade edilir. *IM* 'ler ise $j \in \{1, ..., N\}$ tüm $IM_j \subseteq T$ ile temsil edilir. Hedef alan içerisindeki tüm İN'lerin kümesi $\dot{I}N_T = \{\dot{I}N_1, ..., \dot{I}N_n\}$ şeklinde tanımlanır. Izgara ayrıştırma algoritması Çizelge. 3.4.'de verilmiştir.

Çizelge 3.4. Izgara Ayrıştırma

Başla: $L_I(i,k) \leftarrow$ (ızgara indeksi,ilgi noktası sayısı) – ızgalar için boş liste

Eşitle: ızgara ayrıştırma için mesafe m, $m \le A_m \,|\, A_m$ ajan iletişim mesafesi

Başla: $x_{temp} = x_{\min} + (m/2),\ \ y_{temp} = y_{\min} + (m/2)$

$\dot{I}N_T \leftarrow$ hedef alandaki ilgi çekici noktalar

for y_{temp} 'den $y_{\max}$ ' a kadar do

    $j = 0$

    for x_{temp} 'den $x_{\max}$ ' a kadar do

        $IM_j = (x_{temp}, y_{temp})$

        $x_{temp} = x_{temp} + m$

        $\dot{I}N_{temp} = IM_j.buffer(m/2) \bigcap \dot{I}N_T$

        $L_I.insert(j, IN_{temp}.size())$

        $j{+}{+}$

    end for

    $x_{temp} = x_{temp} + (m/2)$

    $y_{temp} = y_{temp} + m$

end for

return r_1

İlk olarak Çizelge. 3.4.'te görüldüğü üzere hedef alanı içeren en küçük düzenli dörtgen alanın koordinatları bulunur. Daha sonra belirlenen ayrışma mesafesi kullanılarak (genellikle ajanın iletişim/algılama mesafesi), ajanın konumlandırılacağı *IM* 'ler bulunur. Yani burada temel amaç ızgara oluşturup merkezine ajan koymak değil; hesaplanan

noktalara ajan yerleştirmek ve ajanların iletişim mesafesi kadar bir kapsama alanı oluşturmaktır. Oluşturulan kapsama alanları ile hedef alandaki İN'ler kesiştirilir (Şekil 3.8.). Böylelikle *IM* 'lerde kaç adet İN olduğu bulunur.

Şekil 3.8. IM-İN ikilisi

Son olarak *IM* 'nin indeksi ve içerdiği İN sayısını içeren ikili listeye atılır.

Ödül stratejisi

Seyir mesafesini en aza indirmenin zamandan, enerjiden ve donanım kullanım süresinden tasarruf etmek gibi birçok getirisi vardır. Çizelge 3.5.'te görüldüğü gibi, hedef ataması gerçekleştirmek ve seyahat mesafesini en aza indirmek amacıyla bir ödül yapısı tasarlanmıştır. Ödül, her bir ajanın her bir *IM* 'ye uzaklığı ile belirlenir. Her adımda, ajanların *IM* 'lere olan uzaklıkları hesaplanır. Fakat alınan ödülü en yüksek düzeye çıkarmak burada başarısız bir strateji olacaktır. Çünkü elde edilen ödülün en yüksek düzeye çıkarılması, ajanları hedef alandan uzaklaştıracak ve alanların kapsanmasını engelleyecektir. Bu sebeple elde edilen mesafe değerlerinden en küçük olanı seçilir. Sonrasında ise bu uzaklık değeri -1 ile çarpılarak negatif değeri alınır. Bu sayede ajanların ızgaralara ayrıştırılmış alanlara dağılım ödülü negatif değerden 0'a yakınsar. Böylelikle, PÖ'nün temelinde var olan ödül güdümlü öğrenme gerçekleştirilir.

Çizelge 3.5. Ödül İşlevi 1

Başla: $r_1 = 0$
for ızgara 0'dan IM^N 'e kadar do
ajanların ızgaraya mesafesini (d) hesapla
$r_1 = r_1 + (-\min(d))$
end for
return r_1

IM 'lere doğru hareket eden ajanlar birbirleriyle çarpışırlarsa cezalandırılır. Bu nedenle, ajanlar çarpışmalardan kaçınırken hedef alanları kapsamayı öğrenmelidir. Çarpışma olup olmadığının belirlenebilmesi için her adımda ajanlar arasındaki mesafe hesaplanır. Ajanların daire şeklinde modellendiği düşünüldüğünde mesafe, ajanların yarıçap toplamlarından daha küçük ise çarpışma gerçekleşmiş demektir.

Çizelge 3.6. Ödül İşlevi 2

Başla: $r_2 = 1$, $\varnothing_A$
for ajan 0 dan E^N 'e kadar do
if $d_{A_i A_j} < \varnothing_A$ then
$r_2 = \gamma_r * r_2$
end if
end for
return r_2

Çizelge. 3.6.'te sözde kodu verilen çarpışma hesaplama yönteminde γ_r çarpışmadan kaçınma için indirim faktörünü temsil eder:

$$p_i = \prod_j w(ij), \quad j \in \{1,...,N\} \text{ tüm } A_j \in E_i^C \tag{3.18}$$

$$w(ij) = \begin{cases} 1\ , & (\varnothing_i + \varnothing_j)/2 \le d_{ij} \\ \gamma_r, & (\varnothing_i + \varnothing_j)/2 > d_{ij} \end{cases} \tag{3.19}$$

Ajanların, İN sayısı fazla olan IM 'lere doğru yönlenmesini sağlayan ödül işlevi Çizelge. 3.7.'de verilmiştir:

Çizelge 3.7. Ödül İşlevi 3

Başla: $r_3 = 0$
for ajan 0 dan E^N 'e kadar do
Bul: A_i 'nin erişebildiği ajanları
Bul: Erişilebilir ajanların (A^R) kapsadığı IM 'leri
Topla: $r_3 \leftarrow L_I(i,k)$ kullanarak kapsanan İN sayıları (c),
$r_3 = \sum_i^n c_i \ \forall i,n \ \{A_i,...,A_n\} \in A^R s$
Hesapla: Kapsanan İN'lerin tüm İN'lere göre yüzdesi
$r_3 = r_3 * 100 / \dot{I}N^N \ \forall i,n \ \{\dot{I}N_i,...,\dot{I}N_n\} \in \dot{I}N_T$
end for
return r_3

En kısa yol, çarpışmadan kaçınma ve kapsanan İN'ler için tasarlanan ödül işlevleri kullanılarak elde edilen kümülatif ödül işlevi aşağıdaki gibidir:

$$r_i = r_1 * r_2 * r_3 \tag{3.20}$$

Bu çalışmada önerilen yöntem, ödül güdümlü çalışma yaklaşımına sahiptir ve ajanlar Eş. 3.21. kullanarak öğrenirler.

ÇADDPG'ye dayalı ÇAS modeli

Ajanlar arasındaki iş birliği karmaşıklığının çözümü için bir ÇAS modeli oluşturulmuştur. 2B bir düzlem üzerinde hareket eden ajanların A_i, t zamanındaki konumu $(x_i^A(t), y_i^A(t))$ ile ifade edilmiştir. Hedef alan T , ızgaralara ayrıştırma işlemi sonrasında gidilecek ızgara alanları ise $j \in \{1,...,N\}$ tüm $IM_j \subseteq T$ ile temsil edilir.

Önerilen bu ÇAS modelinde aşağıdaki kısıtlamaların karşılanması beklenmektedir.

- Ajanlar, kapsanan İN sayısı maksimum olacak şekilde IM 'lere dağılmalıdır.
- Her IM yalnızca bir ajan tarafından kapsanmalıdır. Örneğin $\forall i, j, IM_i^{'} \neq IM_j^{'}$ $i, j \in \{1, ..., IM^T\}$
- Hedef alan içerisindeki IM 'lere konumlanmış ajanlar, birbirleriyle iletişim sağlayabilecek mesafede olmalıdır.
- Ajanlar çarpışmadan kaçınmalıdır yani herhangi iki ajan herhangi bir t anında aynı konumda bulunamaz. Örneğin $\forall i, j, (x_i^A(t), y_i^A(t)) \neq (x_j^A(t), y_j^A(t))$

Çok ajanlı sistem modeli, Bölüm 3.5.1.'de ÇADDPG'ye Dayalı ÇAS Modeli adı altında sunulan model ile aynı hareket, eğitim ve yürütme modeline sahiptir.

Önerilen Yöntemin İnşası

Önerilen yöntem model bağımsız öğrenme yapısına sahiptir; tekrar arabelleğini kullanarak öğrenme sürecini işletir. Merkezileştirilmiş modül, ajanlara eğitim süresi boyunca politikalarını nasıl güncelleyecekleri konusunda rehberlik eder. Eğitim süreci bölümlere ayrılmıştır ve her bölüm koşulmadan önce ajanların konumu rasgele belirlenir. Ajanlar, eğitim sürecinde senaryoyu yeniden koşmak için tekrar arabelleğine eylem, durum ve ödülden oluşan bilgi grubunu depolar. Her zaman adımında, tekrar arabelleğinden örnekleme yapılır. Eleştirmenler Eş. 3.16. ile güncellenirken, Eş. 3.17. ile ajanların politika gradyanı güncellenir. Her ajan, kapsama puanını artırmak için en uygun ortak eylemi belirler. Yürütme sürecinde ise merkezi modül kaldırılır ve yerel gözlemler için aktör ağı kullanılır. Önerilen yöntemin sözde kodu Çizelge 3.8.'te verilmiştir.

Çizelge 3.8. Önerilen Yöntem

Başla: öğrenme ve ödül indirim faktörü
for $bölüm = 1$'den M 'ye kadar do
  Başla: N ve hedef alan T
  Bul: Çizelge 3.4. ile IM^T
  Al: x'in ilk durumu
    for $t = 1$'den maks_adım_sayısı'na kadar do
    her bir ajan i için, seçilen eylem ve gürültü: N_t , $a_i = \mu_{\theta_i}(o_i) + N_t$ politika ve gözlem: $w.r.t$
    Eylemleri uygula: $a = (a_1, ..., a_N)$
    Hesapla: Çizelge 3.7. kullanarak ödül r
    Belirle: yeni durum x'
    Sakla: (x, a, r, x') bilgisini tekrar oynama belleğine D
    $x \leftarrow x'$
    for ajan $i = 1$'den N 'ye kadar do
      Örneklem al: tekrar oynatma arabelleğinden D mini-yığın olarak S
      Güncelle: Eş. 3.16'yı kullanarak eleştirmen
      Güncelle: Örneklem alınmış politika gradyanı Eş.3.17.'yi kullanarak aktör
    end for
  Güncelle: her ajan için hedef ağ parametreleri
  end for
end for

Her ajanın kendine özgü aktör ve eleştirmen ağları vardır. Daha önce açıklandığı gibi, yöntem, tekrar arabelleğinde depolanan deneyimlerden (yani eylem, durum ve ödül) öğrenir. Diğer bir deyişle, öğrenme süreci boyunca her t zaman diliminde, ağdaki tüm ajanlar için aktörler ve eleştirmenler, rastgele örneklem alınan mini-yığın kullanımıyla deneyimlerden güncellenir.

Çizelge 3.8.'te eğitim sürecindeki öğrenme yaklaşımı sözde kodu verilmiştir. Önerilen yöntemde, ajanların başlangıç konumlarından hedef alan üzerindeki konumlanmalarına kadar geçen süreç bir bölüm olarak ifade edilir. Eğitim için her bölüm t zaman

dilimlerinden oluşur. Eğitim döngüsünde, sistem s_1 başlangıç durumunu alır ve ortamın başlangıç koşulları oluşturulur. Her ajan i, Q_i gözlemi ile aktör μ_{θ_i}'ye göre bir eylem seçer. Ajanın yerel olarak en uygun politika seçmesini ve daha fazla keşif gerçekleştirmesini önlemek için, seçilen eyleme gürültü eklenir. Seçilen eylemi gerçekleştiren ajanlar bir ödül değeri r_t ve yeni bir s_{t+1} durumu elde edecektir. Seçilen eylem, ajanı hedef bölge dışına çıkmaya veya diğer ajanlarla çarpışmaya zorlarsa Eş. 3.20.'e göre cezalandırılır. Dolayısıyla ajan bu eylemden kaçınmayı ve yeni konumu seçmemeyi öğrenir. Daha sonra (s_t, a_t, r_t, s_{t+1}) 'in son değerleri tekrar oynatma arabelleğinde saklanır. Eğitim sürecinin sonunda, t zaman aralığındaki her ajan, tekrar oynatma arabelleğinden D mini-yığın S rastgele seçer ve ardından Eş. 3.16'yı kullanarak eleştirmeni günceller. Bundan adımdan sonra, aktör Eş. 3.17. ile güncellenir. En son aşamada hedef ağ, kayıp işlevi ve öğrenme oranı ile yavaş yavaş güncellenir.

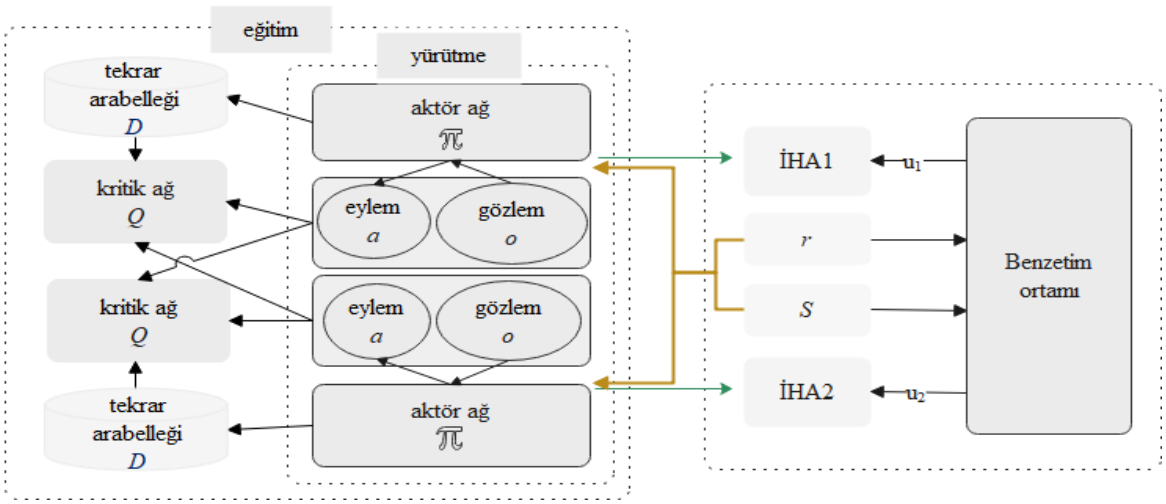
4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Bu tez çalışması kapsamında 2 farklı kapsama problemi için 2 farklı yöntem önerilmiştir. Bu bölümde önerilen yöntemler ayrı ayrı ele alınarak elde ettiği benzetim sonuçları sunulmuştur. Her iki yöntem için de aynı benzetim ortamı farklı deneysel ayarlar ile kullanılmıştır.

Çizelge 4.1. Önerilen yöntemlerde kullanılan semboller

Semboller	Açıklama
$s \in S$	Durumlar
$a \in A$	Eylemler
$r \in R$	Ödüller
γ	İndirim faktörü: $0 < \gamma \leq 1$
$\pi(a \mid s)$	Stokastik politika
$\pi_\theta(.)$	θ ile belirlenmiş politika
$a = \mu(s)$	μ deterministik politika: $S \rightarrow A$
$Q(s,a)$	(s,a) durum-eylem çifti için değer fonksiyonu

Yöntemlerin değerlendirilebilmesi amacıyla, OpenAI [64] ekibi tarafından geliştirilen çok ajanlı aktör-eleştirmen platformunu kullanarak Şekil 4.1.'te gözüktüğü gibi bir benzetim ortamı tasarlanmıştır.



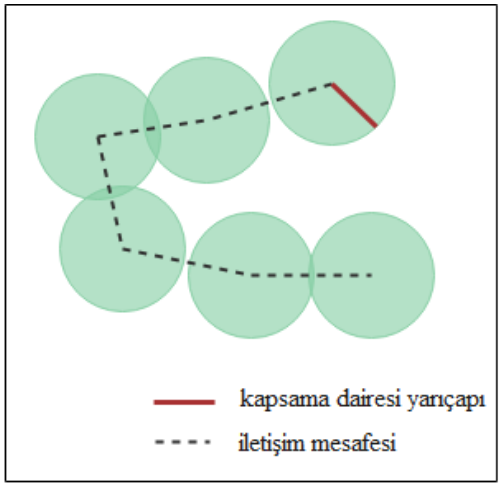
Şekil 4.1. Deney platformu

4.1. Alan Kapsama

Benzetim ortamı, geometrik merkezin orijin olduğu, her kadranın 1 birim olarak ayarlandığı bir koordinat düzlemidir. Düzlem, ajanlardan ve eğitim ayarlarına göre oluşturulmuş bir hedef alandan oluşur. Ajanın etrafındaki daire, alan kapsamını temsil etmek için kullanılır. Ayrıca benzetim ortamında bir ajan düğüm ile temsil edilmektedir. İletişim mesafesi, kapsama dairesi yarıçapı ve hedef alanın şekli, eğitim bölümünün başında belirlenir. Ajan, iletişim mesafesindeki diğer herhangi ajanlarla iletişim kurarken, sadece yerel ortamı gözlemleyebilir. Hedef alan herhangi bir şekil olabilir. Her eğitim bölümünde, ajanların konumları ve hedef alanın konumu rastgele oluşturulur.

4.1.1. Alan kapsama eğitim süreci

DDMDAC'ta işbirlikçi ödül ajanlar tarafından paylaşılır. Hedef bölgede erişilebilir ajanlar için ödül çarpışmadan olumsuz, kapsamadan olumlu etkilenir. Yeşil alanın kapsanan alanı temsil ettiği durumda işbirlikçi ödül elde etmek için kesişimler birleştirilir (Şekil 4.2.). Ajanların benzer özelliklere sahip olduğu varsayılmaktadır.



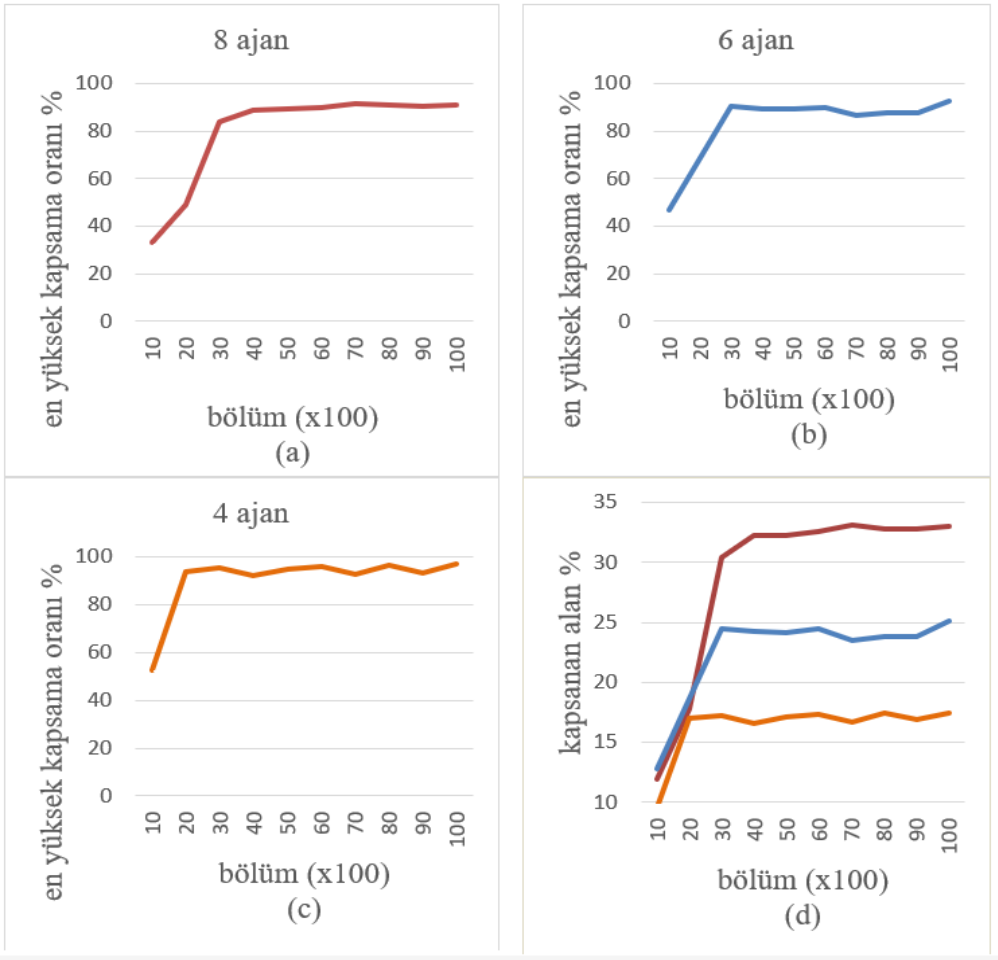
Şekil 4.2. Kolektif ödül için birleştiriltirilen kesişimler

Eğitim için bölüm sayısı 10000 olarak belirlenmiştir ve her bölümde bir ajan 150 eylem yapabilir. Her bölümdeki eğitimin ortalama sonuçları performansı değerlendirmek için kullanılır. Yığın boyutu 1024 olarak tanımlanmıştır. Diğer parametreler ise şu şekilde tanımlanmıştır: çok katmanlı algılayıcıdaki birim sayısı 64, indirim faktörü 0,95 ve öğrenme oranı 0.01'dir.

4.1.2. Alan kapsama deneysel sonuçlar

Deney 1'de, ajan sayısının 4 ila 8 arasında değiştiği durum için alan kapsama sonuçları incelenmiştir. Her bir ajanın kapsama dairesi yarıçapı ve iletişim aralığı sırasıyla 0,12 ve 0,5 birim iken hedef alan 0,5 × 1 birim olarak ayarlanmıştır.

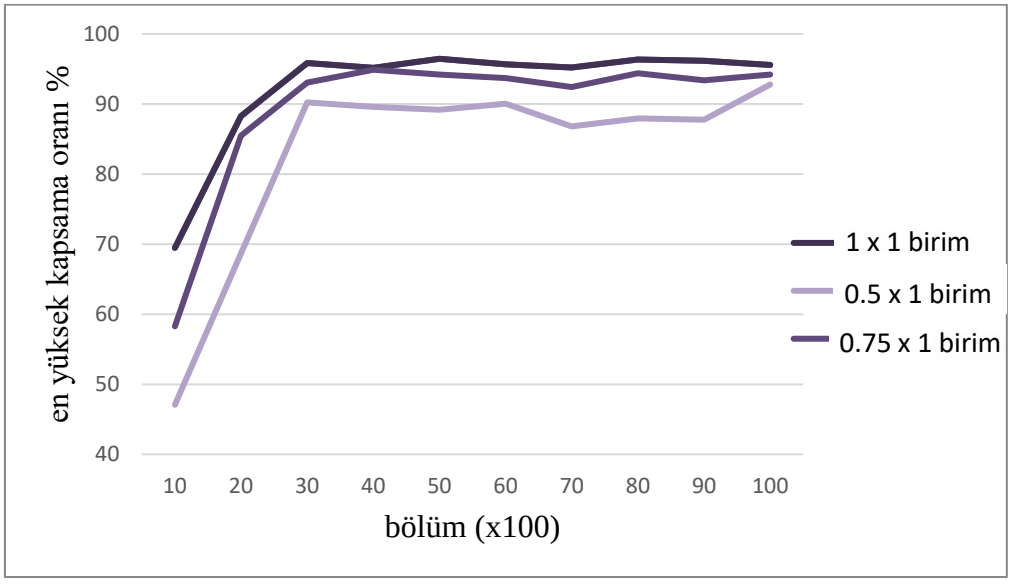
Etkileşimlerin sayısındaki ve çeşitliliğindeki değişiklikler, deneyimin kalitesini etkileyebilir. Deney 1 sonuçlarına göre beklendiği gibi ajan sayısı arttıkça ortam ayarları aynı olsa dahi kapsama performansının arttığı gözlemlenmiştir (Şekil 4.3.). 8 ajan, sırasıyla 6 ajana ve 4 ajana kıyasla yaklaşık %30,9 ve %88,3 ortalama artışla daha iyi kapsama oranı elde etmiştir. Örneğin, bölüm sayısı 5000 olduğu durumda 8 ajandan oluşan grup %32,3'e, 6 ajandan oluşan grup %24,2'ye ve 4 ajandan oluşan grup ise %17,1 kapsama oranına ulaşmıştır.



Şekil 4.3. Ajan sayısındaki değişimin alan kapsamasına etkisi

İkinci deneyde 6 ajan, hedef alan boyutunun kapsama performansı üzerindeki etkisini analiz etmek için kullanılmıştır. Ajanlar arasındaki mesafe 0,5 birimden küçük veya eşitse, ajanlar iletişim kurabilmektedir.

Kapsanacak alanın boyutunun artması, bağlı ajanlar için farklı bağlanma tiplerini beraberinde getirir. Ajanlar, merkezi bir yönlendirmeye ihtiyaç duymadan farklı uygun konumlar bulduğu gözlemlenmiştir. Ajanların, ortak kapsanan alanlardan kaçınmak ve kapsanmayan alanları azaltmak için çeşitli alternatif eylemleri oluşmuştur. Ajan sayısının değişmediği ancak hedef alanın büyüdüğü durumlarda, ajanlar iletişim mesafesini aşmadan aralarındaki mesafeyi daha kolay artırabilirler. 1x1 birimli hedef alanın kullanılması, 0,75x1 birimli hedef alana ve 0,5x1 birimli hedef alana kıyasla sırasıyla yaklaşık %1,4 ve %3'lük ortalama artışla daha iyi kapsama oranı elde edilmesini sağlamıştır (Şekil 4.4.). Örneğin, bölüm sayısı 3000 olduğunda, 0,5x1 birimlik hedef alanın en yüksek kapsama oranı %90,3'e eşitken, 0,75x1 birimlik hedef alanın en yüksek kapsama oranı %93,1'e, 1x1 birimlik hedef alanın en yüksek kapsama oranı %95,9'a eşittir. Bu sonuçlar, hedef bölgenin büyüklüğüne göre kullanılacak optimal ajan sayısını belirlemede yardımcı olabilecek bilgiler sunmaktadır.

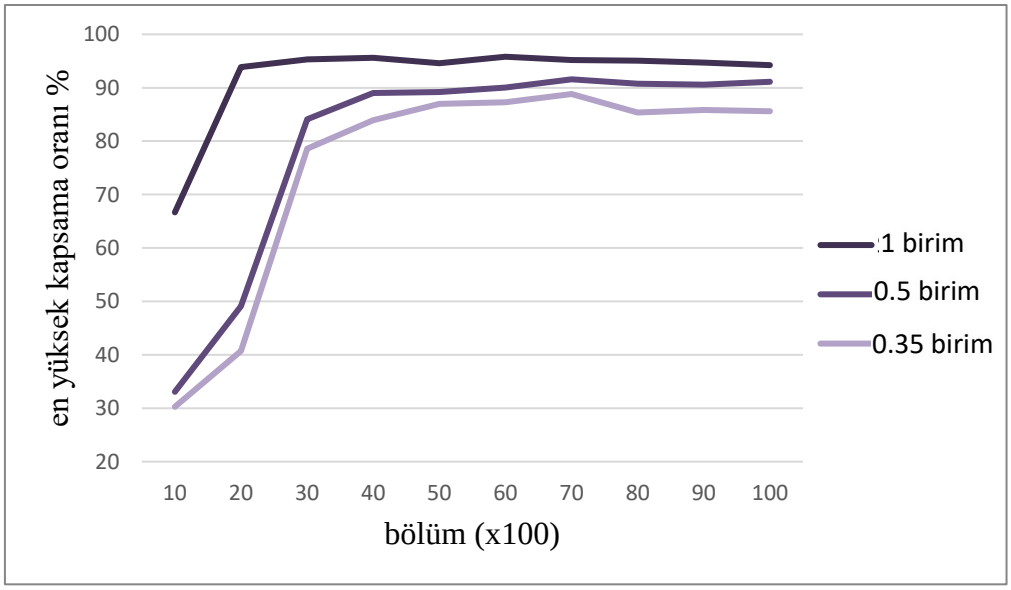


Şekil 4.4. Hedef alan büyüklük değişiminin alan kapsamasına etkisi

Deney 3'te, ajanların iletişim mesafesindeki değişikliğin kapsama performansına etkisi analiz edilmiştir. Hedef alan 0,5×1 birim olarak belirlenmiş olup, sistemdeki ajan sayısı

8'dir. Deney sonuçlarına göre iletişim mesafesi kapsama performansı üzerinde önemli bir etkiye sahiptir.

İletişim mesafesi yeterince uzun olduğunda, hedef bölgeye ulaşan ajanların bağlanmak için çok yakın olması beklenmez. Ajanlar, olası eylem-durum uzayı arttıkça bağlı grafın bir üyesi olmak için daha az eylem gerçekleştirirler. 1 birimlik iletişim mesafesinin kullanılması, 0,5 birimlik iletişim mesafesi kullanılmasına ve 0,35 birimlik iletişim mesafesi kullanılmasına göre sırasıyla yaklaşık %3,4 ve %10,1 ortalama artışla daha iyi kapsama oranı sağlamıştır. 2000 bölüm söz konusu olduğunda, 1 birim kullanılan senaryonun en yüksek kapsama oranı %93,9'a eşitken, 0,5 birim kullanılan senaryonun en yüksek kapsama oranı %49,1'e eşit ve 0,35 birim kullanılan senaryonun en yüksek kapsama oranı %40,7'ye eşittir. Ek olarak, ajanların merkezi bir nokta ile iletişim kurmak yerine doğrudan diğer ajanlarla iletişim kurması, erişilebilir ajanları bulmak için yapılan hesaplama karmaşıklığının azalmasına olanak sağlar.



Şekil 4.5. İletişim mesafesindeki değişikliğin alan kapsaması üzerine etkisi

Deneysel sonuçlara göre, DDMDAC algoritmasının bilinmeyen koşullarda dinamik kapsama görevini verimli bir şekilde tamamlayabildiği görülmüştür. Bu sonuçların elde edilebilmesi için önerilen yöntem, konum tabanlı stratejiler kullanan PÖ tabanlı birçok araştırmanın aksine, yerel gözlemler kullanan model bağımsız politika gradyan tarzında modellenmiştir. Benzetim çalışmalarının özeti Çizelge 4.2.'de sunulmuştur.

Çizelge 4.2. Alan Kapsama için önerilen yöntem benzetim sonuçları özeti

Bölüm sayısı (x100)	Ajan sayısındaki değişikliğin kapsama üzerindeki etkisi			Hedef alan boyut değişikliğin kapsama üzerindeki etkisi			İletişim mesafesi değişikliğin kapsama üzerindeki etkisi		
	4 ajan	6 ajan	8 ajan	0,5x1 birim	0,75x1 birim	1x1 birim	0,35 birim	0,5 birim	1 birim
10	9,52	12,76	11,96	12,76	15,80	18,84	10,94	11,96	24,11
20	16,98	18,63	17,75	18,63	23,19	23,95	14,71	17,75	33,96
30	17,23	24,48	30,41	24,48	25,24	26,00	28,42	30,41	34,46
40	16,60	24,30	32,19	24,30	25,74	25,81	30,35	32,19	34,59
50	17,14	24,19	32,25	24,19	25,56	26,16	31,46	32,25	34,22
60	17,29	24,43	32,57	24,43	25,41	25,94	31,56	32,57	34,65
70	16,71	23,55	33,12	23,55	25,07	25,83	32,12	33,12	34,42
80	17,42	23,85	32,82	23,85	25,60	26,13	30,87	32,82	34,38
90	16,88	23,81	32,76	23,81	25,33	26,09	31,04	32,76	34,26
100	17,49	25,17	32,95	25,17	25,55	25,93	30,96	32,95	34,09

Lokasyona dayalı stratejilerde kapsanan alan genellikle bir ızgaranın kapsanmasına dayanır. Kapsanan alan başka bir ajanla ortak kapsandığında ise tutarsız ödüllere ve hatalı Q-değerlerine neden olur. Bu nedenle ajanları optimuma yakın konumlara yerleştirebilmek için tamamen gözlemlenebilir bir ortama ihtiyaç vardır. Öte yandan önerilen yöntemde durum uzayı yerel gözlemlere bağlıdır ve bu durum büyüyen eylem-durum uzayını ortadan kaldırır. Bu strateji ÇAS'da eğitim sürecinde doğru politika üretmeye yardımcı olur. Gerçek uygulamalarda, ortam genellikle bilinmez ve ortamı tanımak için ajanların keşfine ihtiyaç duyulur. Önerilen yöntem, alan kapsama sorununu gerçek dünya formülasyonuna izin verir; çünkü ajanlar yalnızca gözlemleyebildikleri alana göre öğrenir ve hareket eder. Tasarlanan ödül yapısı, açık bir ödülün paylaşılmasına ihtiyaç duymadan kolektif davranışı zorlar. Böylelikle, ortamın öğrenilmesi ajanların bireysel gözlemlerine dayalı olarak merkezi olmayan bir şekilde ele alınabilir.

4.2. İlgi Noktası Kapsama

İN kapsama için önerilen yöntem, alan kapsama problemi için önerilen yöntemde olduğu gibi geometrik merkezin orijin olduğu ve her kadranın 1 birim olarak ayarlandığı bir koordinat düzleminde test edilmiştir.

4.2.1. İlgi noktası kapsama eğitim süreci

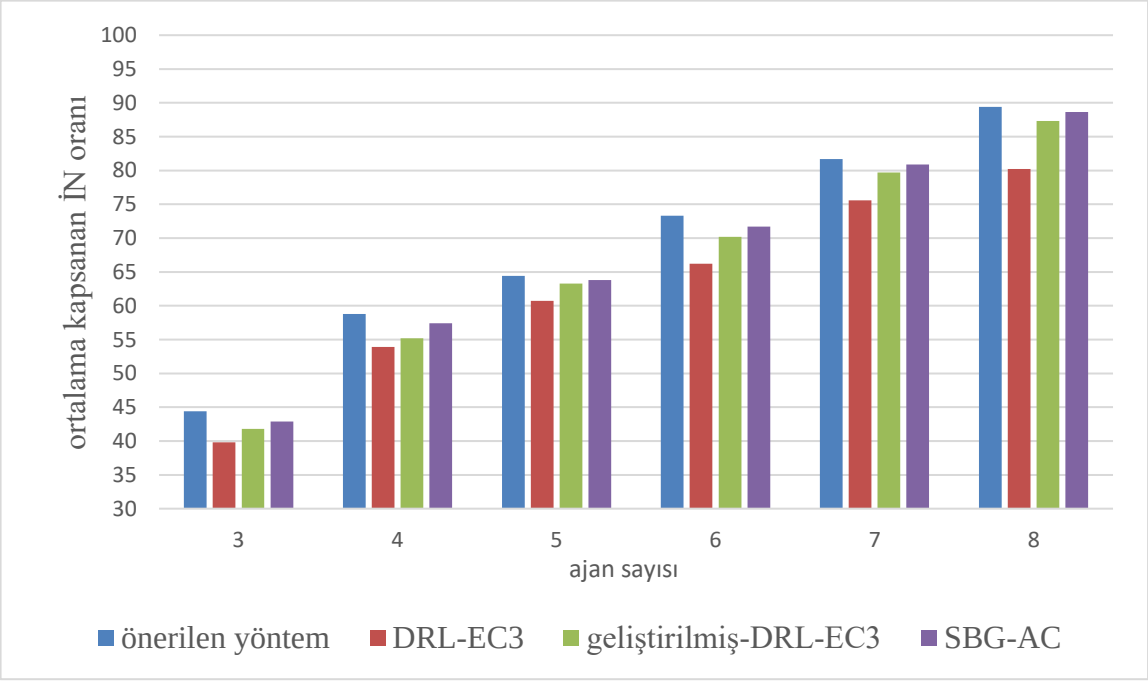
Eğitim bölümünün başlangıcında, ajanların konumları, hedef alanın şekli ve konumu, İN'lerin sayısı ve konumları ve ajanların iletişim mesafesi rastgele oluşturulur. Eğitimin bölüm sayısı 5000, bölümün adım sayısı ise 250 olarak belirlenmiştir. Diğer kalan parametreler şu şekilde belirlenmiştir; çok katmanlı algılayıcıdaki birim sayısı 64, öğrenme oranı 0.001, yığın boyutu 1024 ve indirim faktörü 0.99'dur. γ_r yani r_2 işlevinde kullanılacak öğrenme oranı 1 olarak seçilmiştir. Ajanların konumları, İN sayısı ve konumları gibi parametrelerin eğitim başlangıcında rastgele üretilmesi nedeniyle benzetim senaryoları 50 defa tekrarlanmıştır. Elde edilen metriklerin ortalaması alınarak önerilen yöntemin performansı değerlendirilmiştir. Önerilen model, aynı benzetim ayarları kullanılarak DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC ile karşılaştırılmıştır. Benzetim sonuçlarının karşılaştırılması ve doğrulanması için aşağıda belirtilen üç konu kullanılmıştır:

1. Ajan sayısı artışının kapsama üzerindeki etkisi: Sistemin kapsadığı ortalama İN puanıdır. Çizelge 3.7. kullanılarak hesaplanır.
2. Ajan sayısı artışının enerji kullanımı üzerindeki etkisi: Sistemin sahip olduğu ajan sayısı karşılığında kapsadığı İN sayısı enerji verimliliğini ifade eder. Kapsanan İN'lerin normalleştirilmiş halidir; yani Çizelge 3.7. ile elde edilen sonucun sistemdeki ajan sayısına oranıdır.
3. Kapsanan İN'ler için adillik indisidir: Bir İN'nin kaç ajan tarafından kapsandığı bilgisine ilişkin Jain [82] adillik indisidir. N ortamdaki İN sayısını temsil ederken $c_t(i)$ t anındaki i İN'sini kapsayan ajan sayısını ifade eder. $J=1$ olması ajanlar arasında mükemmel adillik olduğu durumu ifade eder.

$$J = \frac{(\sum_{i=1}^{N} c_t(i))^2}{N \sum_{i=1}^{N} c_t(i)^2} \quad (4.1)$$

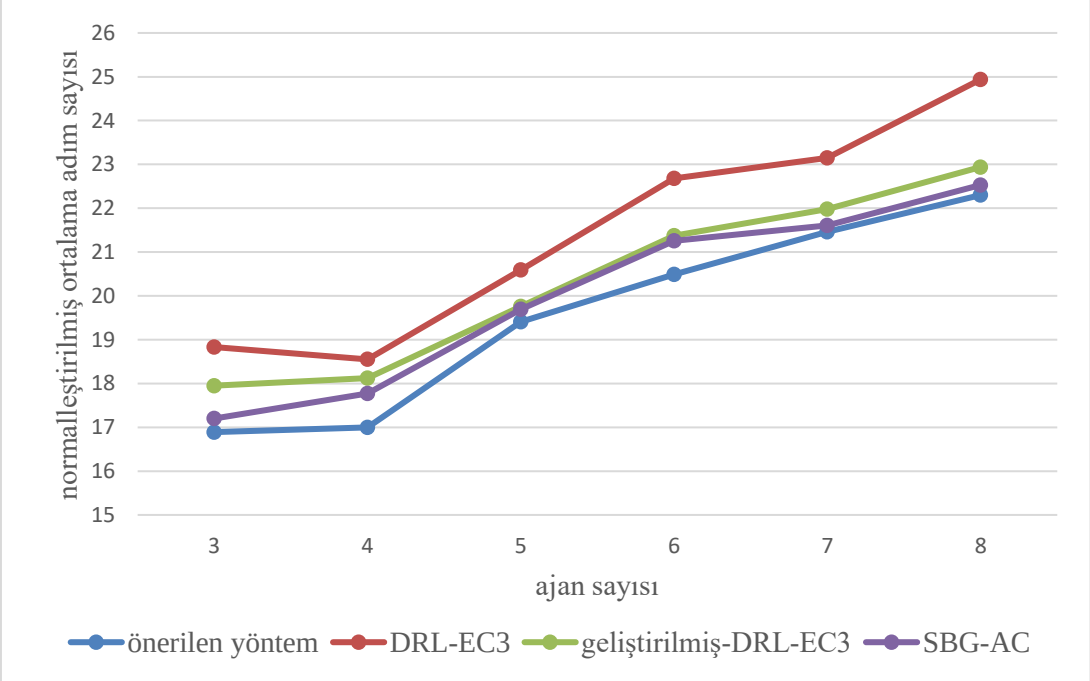
4.2.1. İlgi noktası kapsama deneysel sonuçlar

Birinci deneyde ajan sayısını artırmanın etkisi incelenmiş ve 4 modelin performansları karşılaştırılmıştır. Karşılaştırma için kullanılacak kapsama oranları, Çizelge 3.7.'de verilen yönteme göre hesaplanmıştır. Önerilen yöntem DRL-EC3'e göre yaklaşık %8,64, geliştirilmiş-DRL-EC3'e göre %3,51 ve SBG-AC'e göre %1,54 daha fazla kapsama elde etmiştir (Şekil 4.6.). Örneğin ajan sayısı 4 olduğunda, önerilen yöntem yaklaşık %58,8, DRL-EC3 yaklaşık %53,9, geliştirilmiş-DRL-EC3 yaklaşık %55,2 ve SBG-AC yaklaşık %57,4 kapsama oranı elde etmiştir. 8 ajan olduğu durumunda, önerilen yöntemle elde edilen kapsam yaklaşık %89,4, DRL-EC3 ile elde edilen %80,2, geliştirilmiş-DRL-EC3 ile elde edilen %87,3 ve SBG-AC ile elde edilen kapsam %88,'dir. Diğer senaryolar için de benzer bir eğilim vardır. Yani, önerilen yöntemle elde edilen ortalama kapsanan İN oranı düzenli olarak artmıştır. Ajan sayısındaki artış, ajanların İN kapsama sürecinde farklı desenlerde bağlantı kurmasına izin verdiği görülmüştür.



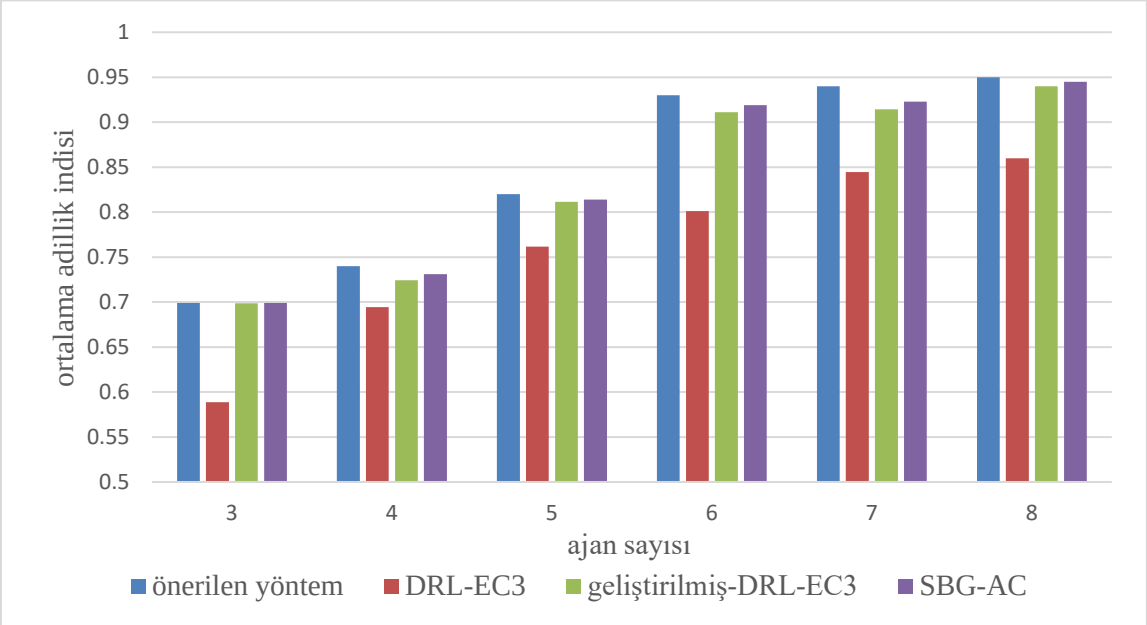
Şekil 4.6. Kapsanan İN-ajan ilişkisi

İkinci deneysel çalışmada ise 4 modelin enerji tüketim karşılaştırması yapılmıştır. Enerji tüketiminin hesaplanmasında ajanların gerçekleştirdiği eylem sayısı kullanılmıştır. Her bölümde, sistemin ulaştığı kapsama oranı kaydedilmiş, eğitim sonunda kapsama puanları toplanmış ve sonrasında eylem sayısına bölünmüştür. Elde edilen sonuç, sistemdeki ajan sayısına bölünerek, 1 eylem başına kapsanan İN sayısı bulunur. Son olarak, bir ajanın bir İN'yi kapsamak için ilk konumdan son konuma kadar kaç adım attığı hesaplanır. Bu sonuç, normalleştirilmiş ortalama enerji tüketimini temsil eder. DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC modelleri sırasıyla önerilen yönteme göre yaklaşık %8,7, %3,75 ve %2,1 daha fazla enerji tüketmiştir (Şekil 4.7.). Örneğin ajan sayısı 4 ve 8 olduğunda normalleştirilmiş ortalama adım büyüklüğü şu şekilde oluşmuştur; önerilen model 17 ve 22.3, DRL-EC3 18.55 ve 25, geliştirilmiş-DRL-EC3 21.37 ve 22.94, SBG-AC ise 17.7 ve 22.53. Bu sonuçlara göre ajan sayısı arttıkça enerji tüketim değerlerinde önemli bir değişimin olmadığı gözlenmiştir. Çok ajanlı sistemlerin rekabetçi ve işbirlikçi doğası nedeniyle, ajan sayısının politikalar üzerinde büyük bir etkisi yoktur. Önerilen yöntem daha az enerji tüketerek yüksek kapsama oranına ulaşmıştır. Bunun 2 nedeni olduğu düşünülmektedir; (i) yol planlaması için özel bir ödül stratejisi tasarlanmıştır; (ii) ödül, yalnızca ajanın kendi durumuna göre değil, erişebildiği her ajan durumuna göre hesaplanır.



Şekil 4.7. Eylem sayısı-ajan sayısı ilişkisi

Son olarak dört model, ajan sayısına göre adillik indisi açısından karşılaştırılmıştır. Önerilen modelin, sırasıyla yaklaşık %10,3, %1,4 ve %1'lik ortalama artışla DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC'den daha iyi bir adillik indisine sahip olduğu görülmüştür (Şekil 4.8.). Örneğin, ajan sayısı 3 olduğunda, önerilen model 0,699 adillik indisine, DRL-EC3 0,58 adillik indisine, geliştirilmiş-DRL-EC3 0,698 adillik indisine ve SBG-AC 0,698 adillik indisine ulaşmıştır. Ajan sayısı 6 olduğunda, önerilen model, DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC'nin adillik indisi sırasıyla 0.93, 0.80, 0.91 ve 0.92 olmuştur. Diğer senaryolar için de benzer bir eğilim gözlenmektedir. Ajan sayısı arttıkça, kapsanan ızgara sayısı da artar. Bu ilişki, adillik indisinde doğrudan iyileşmeye yol açar. Ayrıca önerilen yöntem diğer yöntemlere kıyasla, amaca dayalı ödüllendirme stratejileri kullanılması nedeniyle, daha az adımda benzer adillik indisi elde etmiştir.



Şekil 4.8. Adillik indisi-ajan sayısı ilişkisi

Bu bölümde, önerilen modelin enerji, kapsama ve adillik indeksi gibi konular karşısındaki davranışı incelenmiştir. Daha sonra bu üç metrik kullanılarak önerilen yöntem DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC modelleri ile karşılaştırılmıştır ve benzetim sonuçları özeti Çizelge 4.3.'te verilmiştir.

Çizelge 4.3. İN Kapsama için önerilen yöntem benzetim sonuçları özeti

	Önerilen yöntem	DRL-EC3	Geliştirilmiş-DRL-EC3	SBG-AC
Yaklaşım	Çok ajanlı derin pekiştirmeli öğrenme	Çok ajanlı derin pekiştirmeli öğrenme	Çok ajanlı derin pekiştirmeli öğrenme	Çok ajanlı derin pekiştirmeli öğrenme
Algoritma	MADDPG	DDPG	MADDPG	MADDPG
Ödül stratejisi	Ajan ve grup bazlı	Ajan bazlı	Ajan bazlı	Grup bazlı
Ortalama kapsanan İN'lerin oranı	3 ajan: 44,4 4 ajan: 58,8 5 ajan: 64,4 6 ajan: 73,3 7 ajan: 81,7 8 ajan: 89,4	3 ajan: 39,8 4 ajan: 53,9 5 ajan: 60,7 6 ajan: 66,2 7 ajan: 75,6 8 ajan: 80,2	3 ajan: 41,8 4 ajan: 55,2 5 ajan: 63,3 6 ajan: 70,2 7 ajan: 79,7 8 ajan: 87,3	3 ajan: 42,9 4 ajan: 57,4 5 ajan: 63,8 6 ajan: 71,7 7 ajan: 80,9 8 ajan: 88,65
Normalleştirilmiş ortalama adım sayısı	3 ajan: 16,9 4 ajan: 17 5 ajan: 19,4 6 ajan: 20,5 7 ajan: 21,5 8 ajan: 22,3	3 ajan: 18,9 4 ajan: 18,6 5 ajan: 20,6 6 ajan: 22,7 7 ajan: 23,2 8 ajan: 25	3 ajan: 17,9 4 ajan: 18,1 5 ajan: 19,8 6 ajan: 21,4 7 ajan: 22 8 ajan: 22,9	3 ajan: 17,2 4 ajan: 17,7 5 ajan: 19,7 6 ajan: 21,2 7 ajan: 21,6 8 ajan: 22,5
Ortalama adillik indisi	3 ajan: 0,69 4 ajan: 0,74 5 ajan: 0,82 6 ajan: 0,93 7 ajan: 0,94 8 ajan: 0,95	3 ajan: 0,59 4 ajan: 0,69 5 ajan: 0,76 6 ajan: 0,80 7 ajan: 0,84 8 ajan: 0,86	3 ajan: 0,69 4 ajan: 0,72 5 ajan: 0,81 6 ajan: 0,91 7 ajan: 0,92 8 ajan: 0,94	3 ajan: 0,69 4 ajan: 0,73 5 ajan: 0,82 6 ajan: 0,92 7 ajan: 0,92 8 ajan: 0,94

Bu çalışmada, amaca yönelik 3 farklı ödül stratejisi tasarlanırken, DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC yöntemlerinde sadece bir ödül stratejisine yer verilmiştir. Kolektif ödül tasarımı ile düşük enerji tüketimi-yüksek kapsama oranı elde edilmeye çalışılmıştır.

3 farklı ödül stratejisi kullanımı, elde edilebilecek ödül boyutunu artırarak durum-eylem ikilileri arasındaki ilişkiyi belirginleştirir. Böylelikle öğrenme süresi düşerken öğrenme kalitesi artar. Bununla birlikte, hedef alandaki ızgaralar ile bağlı ajanlar arasındaki mesafenin ölçülmesi temelinde tasarlanan ödül stratejisi (Bkz. Çizelge 3.7.), öğrenmeye büyük katkı sağlamıştır. Bunun iki temel sebebi vardır; (i) sistemdeki ajanların bağlanarak haberleşme mesafesi dışına çıkmadan ızgaralara konumları, (ii) ajanların, ızgaralara dağılarak kesişimi azalttıkları oranda yüksek ödül almaları. DRL-EC3 ve geliştirilmiş-DRL-EC3, alınan ödülün yalnızca ajan-çevre etkileşiminden etkilendiği, yani sadece ajana özgü ödül stratejisi kullanır. Bununla birlikte, SBG-AC ve önerilen yöntem, tüm ajanların ortak bir ödülü paylaştığı ortamın mevcut durumuyla ilgilenmektedir. Önerilen yöntemin aksine DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC yöntemlerinde hedef atama ile ilgili bir ödül stratejisi bulunmamaktadır. Önerilen yöntemde ajanlar, bağlı ajanların bilgilerini kullanarak ödül puanlarını en yüksek düzeye çıkarmaya çalışırlar. Bu yaklaşım ajanların birbirleriyle bağlanmasını ve kolektif karar almalarını zorlamaktadır. Ayrıca önerilen yöntemde ajanlar, kendilerine en yakın mesafedeki İN yoğunluğu en yüksek IM'ye konumlanmaya çalıştıklarından dolayı enerji tüketimi açısından da olumlu sonuçlar elde edilmiştir. Bu yaklaşım ÇAS'larda, hedef atama ile yol planlama algoritmasını temsil etmektedir. Bu algoritma yardımıyla, ajanların kapsama alanlarındaki kesişimi en aza indirilerek adillik indisi yüksek sonuçlar elde edilmiştir.

5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında iki farklı kapsama probleminin giderilmesine yönelik iki farklı kapsama modeli önerilmektedir. Bu bağlamda; kapsama problemi kapsama konusunda karşılaşılan zorluklar, araştırmalar, geliştirilebilecek alanlar ve önerilen modellerin temelini oluşturan yapılar detaylı olarak bilgi verilmiştir. Sonrasında ise tasarlanan modellerin çalışma prensipleri ve benzetim çalışmaları ile elde edilen çıktılar sunulmuştur.
Tez çalışmasında ilk olarak, dinamik ortamda birden fazla ajan ile alan kapsama yapabilmek için dağıtık sistem mimarisine sahip derin pekiştirmeli öğrenme tabanlı bir model önerilmiştir. Önerilen yöntem, eğitim zamanında merkezi bir eleştirmenle öğrenmeye dayanan, yürütme sırasında ise öğrenilen politikaları uygulayan İHA ajanlarından oluşur. Kapsanmayan alanı en aza indirmeye çalışan ajanlar birbirleri arasında ağ kurarak yönsüz bağlı graf oluştururlar. Ajan düğümleri, ağ üzerindeki diğer ajanların gözlemlerini ve konum bilgilerini alır. Konumlandırma kalitesi, dinamik ortamda görev icra eden ajanların bireysel öğrenme kabiliyetlerini, takım başarısı adına kullandıkları oranda artar. Erişilebilir ajanların oluşturduğu bu bilgi bütünü, ortamda oluşabilecek değişikliklere yüksek seviyede tolerans sağlanmasına yardımcı olur. Bununla birlikte, ajanların veri setine bağımlı olmadan kendi kendilerine öğrenmelerine olanak sağlar. Model bağımsız olarak oluşturulan bu ÇAS hareket modeli ile bir İHA düğümü, erişilebilir düğüm sayısını artırmayı öğrenmenin yanı sıra, kolektif zeka ile, bağlı düğümlerin en yüksek seviyede kapsama yapabileceği konumları da tespit edebilir.

Tez çalışması kapsamında alan kapsama problemlerine ek olarak İN kapsama problemleri için de bir yöntem önerilmiştir. Önerilen ilk yöntemde olduğu gibi bu yöntem de her bir İHA'nın ajan olarak temsil edildiği aktör-eleştirmen stratejisi ile dağıtık olarak çalışan DPÖ ajanlarından oluşur. Ajanlar durum ve eylem uzayını kolektif bir zeka temelinde ele alarak, çarpışmaların önlenmesine ve ajanların mümkün olan en kısa sürede en uygun konumlara gidiş için yol planlaması yapmayı amaçlar. Hedef alanın ızgaralara ayrıştırılarak içerisinde kalan İN'leri kapsama temeline dayanan bu yöntemde, düşük enerji tüketimi ile yüksek adillik indisi şemsiyesi altında en yüksek seviyede İN kapsamak amaçlanmıştır.

Önerilen yöntemlerin kapsama performanslarının değerlendirilebilmesi için sık kullanılan

bir benzetim ortamı seçilmiştir. Önerilen ilk yöntem için 3 farklı senaryo belirlenmiş ve ilk deney aşamasında ajan sayısındaki değişikliklerin kapsama üzerindeki etkisi incelenmiştir. 8 ajanın olduğu durum, sırasıyla 6 ajana ve 4 ajana kıyasla yaklaşık %30,9 ve %88,3 ortalama artışla daha iyi kapsama oranı elde etmiştir. İkinci deneyde hedef alanın boyut değişikliklerinin alan kapsama üzerindeki etkisi analiz edilmiştir. Elde edilen sonuçlara göre ajanlar, merkezi bir yönlendirmeye ihtiyaç duymadan ortak kapsanan alanlardan kaçınmak ve kapsanmayan alanları azaltmak için çeşitli alternatif eylemler oluşmuştur. Ajan sayısının değişmediği ancak hedef alanın büyüdüğü durumlarda, ajanlar iletişim mesafesini aşmadan aralarındaki mesafeyi daha kolay artırabilmektedir. Deney sonuçları, hedef bölgenin büyüklüğüne göre kullanılacak optimal ajan sayısını belirlemeye yardımcı olabilecek niteliktedir. Önerilen ilk yöntemin son deney çalışmasında, iletişim mesafesindeki artışın kapsama oranına etkisi incelenmiştir. İletişim mesafesi yeterince uzun olduğunda, hedef bölgeye ulaşan ajanlar fazla bağlantı seçeneği elde edebilmiştir. Ajanlar, olası eylem-durum uzayı arttıkça bağlı grafın bir üyesi olmak için daha az eylem gerçekleştirmiştir. Literatürde benzer bir problem için benzer bir yöntem ile yapılan bir çalışma bulunmamaktadır. Dolayısı ile sonuçlar diğer çalışmalar ile kıyaslanmamış; PÖ temelinde gerçekleştirilebilecek senaryolar üzerinden tartışmalar yapılmıştır.

Önerilen ikinci yöntem için 3 farklı deney senaryosu belirlenmiş ve elde edilen sonuçlar literatürdeki benzer çalışmalar ile kıyaslanmıştır. İkinci yöntemin ilk deney çalışmasında ortamdaki ajan sayısı artırılarak ortamdaki İN'lerin kapsanan İN'lere oranları incelenmiştir. Önerilen yöntem DRL-EC3'e göre yaklaşık %8,64, geliştirilmiş-DRL-EC3'e göre %3,51 ve SBG-AC'e göre %1,54 daha fazla kapsama oranı elde etmiştir. İkinci deney çalışmasında modellerin enerji tüketimleri kıyaslanmıştır. Kıyaslama için ise ajanların 1 İN kapsamak için kaç adet eylemde bulunduğu konusu seçilmiştir. Elde edilen sonuçlara göre DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC modelleri sırasıyla önerilen yönteme göre yaklaşık %8,7, %3,75 ve %2,1 daha fazla enerji tüketmiştir. İkinci yöntemin son deney çalışması adillik indisi üzerine yapılmıştır. Önerilen modelin, sırasıyla yaklaşık %10,3, %1,4 ve %1'lik ortalama artışla DRL-EC3, geliştirilmiş-DRL-EC3 ve SBG-AC'den daha iyi bir adillik indisine sahip olduğu görülmüştür.

Deneysel sonuçlara göre önerilen yöntemler dinamik ortamda kapsama görevini başarıyla tamamlayabilmiştir. Ajanlar, test yani yürütme aşamasında, kendi aktör ağlarını kullanarak grubun ortak amacına yönelik dağıtık ancak kolektif eylemler oluşturmuştur. Model

bağımsız politika gradyanı tarzında modellenen yöntemler yerel gözlemler kullanarak büyüyen eylem-durum uzayı ortadan kaldırmıştır. Bu yaklaşım, doğru politikanın daha kısa sürede üretilmesine olanak sağlamıştır. Tasarlanan ödül yapıları, ödül paylaşımına gerek kalmadan kolektif davranışı zorlamıştır. Bu nedenle yürütme süreci merkezi olmayan bir şekilde ele alınabilmiştir. Merkezi kontrolden bağımsız yerel gözlemlere dayalı bu modeller, sahip olduğu ödül stratejileri ile gerçek uygulamalarda kapsama problemlerinin çözümlenmesine yardımcı olacak niteliktedir.

KAYNAKLAR

1. Valsan, A., Parvaty, B., Vismaya, D. G. H., Unnikrishnan, R. S., Reddy, P. K. and Vivek, A. (2020 15-17 Haziran). *Unmanned aerial vehicle for search and rescue mission*. 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, Hindistan, 684–687.

2. Zhang, W., Song, K., Rong, X. and Li, Y. (2018). Coarse-to-fine UAV target tracking with deep reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 16(4), 1522–1530.

3. Sanfourche, M., Le Saux, B., Plyer, A. and Le Besnerais, G. (2015, 30 Mart-1 Nisan). *Environment mapping & interpretation by drone*. 2015 Joint Urban Remote Sensing Event (JURSE), Lausanne, İsviçre, 1–4.

4. Bassoli, R., Sacchi, C., Granelli, F. and Ashkenazi, I. (2019, 2-9 Mart). *A Virtualized Border Control System based on UAVs: Design and Energy Efficiency Considerations.* 2019 IEEE Aerospace Conference, Big Sky, MT, ABD, 1-11.

5. Schwager, M., Julian, B. J., Angermann, M., and Rus, D. (2011) Eyes in the sky: decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9), 1541–1561.

6. Landgren, P., Srivastava V. and Leonard N. E. (2021). Distributed cooperative decision making in multi-agent multi-armed bandits, *Automatica*, 125.

7. Amirkhani, A. and Barshooi, A. H. (2022). Consensus in multi-agent systems: a review. *Artificial Intelligence Review*, 55, 3897–3935.

8. Tang, R., Qian, X. and Yu, X. (2019). On virtual-force algorithms for coverage-optimal node deployment in mobile sensor networks via the two-dimensional Yukawa Crystal, *International Journal of Distributed Sensor Networks*, 15(9), 155014771986488.

9. Omoniwa, B., Galkin, B. and Dusparic, I. (2022, 8-11 Haziran). *Energy-aware optimization of UAV base stations placement via decentralized multi-agent Q-learning*. 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, ABD, 216-224.

10. Wang, G., Cao, G. and La Porta, T. F. (2006). Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6), 640–652.

11. Dorri, A., Kanhere, S. S. and Jurdak, R. (2018). Multi-Agent Systems: A Survey. *IEEE Access*, 6, 28573-28593.

12. Woolley, A.W., Aggarwal, I., and Malone, T.W., (2015). Collective Intelligence and Group Performance. *Current Directions in Psychological Science*, 24(6), 420-424.

13. Kiumarsi, B., Vamvoudakis, K.G., Modares, H. and Lewis, F.L. Optimal and Autonomous Control Using Reinforcement Learning: A Survey. *IEEE Transactions On Neural Networks and Learning Systems*, 29, 2042-2062.

14. Qu, G., Wierman, A., and Li, N. (2020, 11-12 Haziran). *Scalable reinforcement learning of localized policies for multi-agent networked systems*. Learning for Dynamics and Control, Berkeley, CA, ABD, 256-266.

15. Padakandla, S., K. J., P. and Bhatnagar, S. Reinforcement learning algorithm for non-stationary environments. *Applied Intelligence*, 50(11), 3590-3606.

16. Kober, J., Bagnell, J. A. and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11), 1238-1274.

17. Choset, H. (2001). Coverage for robotics-a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1), 113–126.

18. Kang, Y. and Shi, D. (2018 24-27 Ağustos). *A research on area coverage algorithm for robotics.* 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Lanzhou, Çin, 6–13.

19. Nguyen, M. T., Maniu, C. S. and Olaru, S. (2015, 14-16 Ekim). *Control invariant partition for heterogeneous multi-agent dynamical systems.* 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romanya, 354-359.

20. Ann,S., Kim, Y. and Ahn, J. (2015). Area allocation algorithm for multiple UAVs area coverage based on clustering and graph method. *IFAC-PapersOnLine*, 48(9), 204–209.

21. Cho, S. W., Park, J. H., Park, H. J. and S. Kim. (2021). Multi-UAV coverage path planning based on hexagonal grid decomposition in maritime search and rescue. *Mathematics*, 10(1), 83–98.

22. Santos, M., Madhushani, U., Benevento, A. and Leonard, N. E. (2021 4-5 Kasım). *Multi-robot learning and coverage of unknown spatial fields*. 2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), Cambridge, İngiltere, 137-145.

23. Li, J., Wang, R., Huang, H. and Sun, L. (2009, 22-24 Mayıs). *Voronoi based area coverage optimization for directional sensor networks*. 2009 Second International Symposium on Electronic Commerce and Security, Nanchang, Çin, 488-493.

24. Portela, J. N. and Alencar, M. S. (2008). Cellular coverage map as a voronoi diagram. *Journal of Communication and Information Systems*, 23(1), 1–6.

25. Yang, C. and Szeto, K. Y. (2019, 10-13 Haziran). *Solving the traveling salesman problem with a multi-agent system.* 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, Yeni Zellanda, 158–165.

26. Almadhoun, R., Taha, T., Seneviratne, L. and Zweiri Y. (2019). A survey on multi-robot coverage path planning for model reconstruction and mapping. *SN Applied Sciences*, 1(8), 1–24.

27. Rosalie, M., Brust, M. R., Danoy, G., Chaumette, S. and Bouvry, P. (2017, 17-21 Temmuz). *Coverage optimization with connectivity preservation for UAV swarms applying chaotic Dynamics*. 2017 IEEE Int. Conf. on Autonomic Computing (ICAC), Columbus, OH, ABD, 113–118.

28. Antal, M., Tamas, I., Cioara, T., Anghl, I. and Salomie, I. (2013, 5-7 Eylül). *A swarm-based algorithm for optimal spatial coverage of an unknown region*. 2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romanya, 7–13.

29. Tao, D., Tang, S. and Liu, L. (2013, 14-16 Ekim). *Constrained artificial fish-swarm based area coverage optimization algorithm for directional sensor networks*. 2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems, Hangzhou, Çin, 304–309.

30. Öztürk, C., Karaboga, D. and Gorkemli, B. (2012) Artificial bee colony algorithm for dynamic deployment of wireless sensor networks. *Turkish Journal of Electrical Engineering and Computer Sciences*, 20(2), 255–262.

31. Akram, J., Javed, A., Khan, S., Akram, A. and Munawar, H. S. (2021, 22-26 Mayıs). *Swarm intelligence-based localization in wireless sensor networks*. SAC '21: Proceedings of the 36th Annual ACM Symposium on Applied Computing, Virtual Event Republic of Korea, 1906–1914.

32. Zhang, Z., Xu, X., Cui, J. and Meng, W. (2021). Multi-UAV area coverage based on relative localization: Algorithms and optimal UAV placement. *Sensors*, 21(7), 2400-2414.

33. Liang, K., Chung, C. Y. and Li, C. F. (2014, 27-29 Ağustos). *A virtual force based movement scheme for area coverage in directional sensor networks*. 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japonya, 718-722.

34. Yang, C. and Wen, J. (2013, 3-5 Temmuz). *A hybrid local virtual force algorithm for sensing deployment in wireless sensor network*. 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Taichung, Tayvan, 617–621.

35. Li, W., Wang, G.-G. and Gandomi, A. H. (2021). A survey of learning-based intelligent optimization algorithms. *Archives of Computational Methods in Engineering*, 28(5), 3781–3799.

36. Xu, D., Zhang, X., Zhu, Z., Chen, C. and Yang, P. (2014). Behavior-based formation control of swarm robots. *Mathematical Problems in Engineering*, 2014(1), 1-13.

37. Mozaffari, M., Saad, W., Bennis, M., Debbah, M. (2016). Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters*, 20(8), 1647–1650.

38. Yue, Y., Cao, L. and Luo, Z. (2019). Hybrid artificial bee colony algorithm for improving the coverage and connectivity of wireless sensor networks. *Wireless Personal Communications*, 108, 1719-1732.

39. Gupta, S.K., Kuila, P. and Jana, P.K. (2016). Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks. *Computers Electrical Engineering*, 544–556.

40. Kalantari, E., Yanikomeroglu, H. and Yongacoglu, A. (2016, 18-21 Eylül). *On the number and 3d placement of drone base stations in wireless cellular networks*. 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall) Montreal, QC, Kanada. 1-6.

41. Shi, W., Li, J., Xu, W., Zhou, H., Zhang, N., Zhang, S. and Shen, X. (2018). Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks. *IEEE Access 6*,12518–12529.

42. Njoya, N., A., Ari, A., Awa, M., Titouna, C., Labraoui, N., Effa, Y., Abdou, W. and Gueroui, A. (2020). Hybrid wireless sensors deployment scheme with connectivity and coverage maintaining in wireless sensor networks. *Wireless Personal Communications* 112, 1893-1917.

43. Jagtap, A.M. and Gomathi, N. (2019). Minimizing movement for network connectivity in mobile sensor networks: An adaptive approach. *Cluster Computing*, 22(1), 1373-1383.

44. Shu, T., Dsouza, K.B., Bhargava, V. and Silva, C. (2019, 5-8 Mayıs). *Using geometric centroid of voronoi diagram for coverage and lifetime optimization in mobile wireless sensor networks*. 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE). Edmonton, AB, Kanada, 1-5.

45. Kuo, Y.C., Chiu, J.H., Sheu, J.P. and Hong, Y.W.P. (2021). Uav deployment and iot device association for energy-efficient data-gathering in fixed-wing multi-uav networks. IEEE Transactions on Green Communications and Networking, 5(4), 1934–1946.

46. Ganganath, N., Cheng, C.T. and Tse, C.K. (2016). Distributed antiflocking algorithms for dynamic cover33 age of mobile sensor networks. *IEEE Transactions on Industrial Informatics*, 12(5), 1795–1805 .

47. Rossi, F., Bandyopadhyay, S., Wolf, M. and Pavone, M. (2018). Review of multi-agent algorithms for collective behavior: A structural taxonomy. *IFAC-PapersOnLine*, 51(12), 112–117.

48. Cao, Y., Yu, W., Ren, W. and Chen, G. (2012). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1), 427–438.

49. Vidyasagar, M. (2020 1-3 Temmuz). *Recent advances in reinforcement learning*. 2020 American Control Conference (ACC), Denver,CO, ABD, 4751–4756.

50. Vikhar, P. A. (2016 22-24 Aralık). *Evolutionary algorithms: a critical review and its future prospects*. 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, Hindistan, 261–265.

51. Rijn, S. V., H. Wang, Leeuwen M. and Bäck, T. (2016, 6-9 Aralık) *Evolving the structure of Evolution Strategies.* 2016 IEEE Symp. Series on Computational Intelligence (SSCI), Athens, Yunanistan, 1-8.

52. Zhang, Z., Sun, X., Hou, L., Chen, W. and Shi, Y. (2017, 5-8 Ekim). *A cooperative co-evolutionary multi-agent system for multi-objective layout optimization of satellite module*. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Kanada, 147-151.

53. Nowé, A., Vrancx, P. and De Hauwere, Y. M. (2012). Game theory and multi-agent reinforcement learning. *Adaptation, Learning, and Optimization*, 12, 441–470.

54. Sun, W., Zhang, G. C., Zhang, X. R., Zhang, X. and Ge, N. N. (2021). Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy. *Multimedia Tools and Applications*, 80(20), 30803–30816.

55. Xiao, J., Wang, G., Zhang, Y. and Cheng, L. (2020). A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning. *IEEE Access*, 8, 33511–33521.

56. Samir, M., Ebrahimi, D., Assi, C., Sharafeddine, S. and Ghrayeb, A. (2020). Trajectory planning of multiple dronecells in vehicular networks: A reinforcement learning approach. *IEEE Networking Letters*, 2(1), 14-18.

57. Jiang, S., Huang, Z. and Ji, Y. (2021). Adaptive UAV-assisted geographic routing with Q-Learning in VANET. *IEEE Communications Letters*, 25(4), 1358-1362.

58. Crites, R. H. and Barto, A. G. (1994, 1 Ocak). An actor/critic algorithm that is equivalent to Q-learning," NIPS'94: Proceedings of the 7th International Conference on Neural Information Processing Systems, Denver, CO, ABD, 401-408.

59. Meng, S. and Kan, Z. (2022). Deep reinforcement learning-based effective coverage control with connectivity constraints. *IEEE Control Systems Letter*, 6, 283-288.

60. Liu, C. H., Chen, Z., Tang, J., Xu, J. and Piao, C. (2018). Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 36(9), 2059-2070.

61. Liu, C. H., Ma, X., Gao, X. and Tang, J. (2020). Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning. IEEE Transactions on Mobile Computing, 19(6), 1274-1285.

62. Nemer, I. A., Sheltami, T. R., Belhaiza, S. and Mahmoud, A. S. (2022). Energy-efficient UAV movement control for fair communication coverage: A deep reinforcement learning approach. *Sensors*, 22(5), 1919–1946.

63. Pham, H. X., La, H. M., Feil-Seifer, D. and Nefian, A. A. (2018), Cooperative and distributed reinforcement learning of drones for field coverage. *CoRR*, 1803 (07250), 1-7.

64. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P. and Mordatch, I. (2017, 04-09 Aralık). *Multiagent Actor-critic for Mixed Cooperative-competitive Environments*, NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing, Long Beach CA, ABD, 6379-6390.

65. Dong, J., Chen, S., Ha, P. Y. J., Li, Y., and Labi, S. (2020). A drl-based multiagent cooperative control framework for cav networks: a graphic convolution q network. *CoRR*, 2010 (05437), 1-20.

66. Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009, 14-18 Haziran). *Curriculum learning*. Machine Learning, New York, NY, ABD, 41-48.

67. Pateria, S., Subagdja, B., Tan, A. and Quek, C. (2021). Hierarchical reinforcement learning: a comprehensive survey, *ACM Computing Surveys*, 54(5), 1–35

68. Deng, L. and Yu, D. (2014). Deep Learning: Methods and Applications, *Foundations and Trends in Signal Processing*, 7 (3-4), 197-387.

69. Song, H.A. and Lee, S. Y. (2013 03-07 Kasım). *Hierarchical Representation Using NMF*, International Conference on Neural Information Processing (ICONIP), Daegu-Güney Kore, 466-473.

70. Sarker, I.H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, 2(6), 1-20.

71. Sarker, I.H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3), 1-21.

72. LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-480.

73. Coelho, I.M., Coelho, V.N., da Eduardo, J., Luz, S., Ochi, L.S., Guimarães, F.G. and Rios, E. (2017). A gpu deep learning metaheuristic based model for time series forecasting. *Applied Energy*, 201, 412–418.

74. Zhang, H., Nie, R., Lin, M., Wu, R., Xian, G., Gong, X., Yu, Q. and Luo, R. (2021). A deep learning based algorithm with multi-level feature extraction for automatic modulation recognition. *Wireless Networks*, 27, 4665-4676.

75. Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T. and Rellermeyer, J. S. (2020). A survey on distributed machine learning. *ACM Computing Surveys*, 53(2), 1-33.

76. Mathews, J., Marie, J. (2021 8-15 Aralık). *Critical empirical study on black-box explanations in AI*. ICIS 2021: 42nd International Conference on Information Systems, Association for Information Systems (AIS), Austin, Texas, ABD.

77. Oroojlooy, A. and Hajinezhad, D. (2022). A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence.*

78. Watkins, C.J.C.H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8, 279–292

79. Aydemir F. and Çetin A. (2023). Multi-agent Dynamic Area Coverage Based on Reinforcement Learning with Connected Agent, *Computer Systems Science and Engineering*, 45 (1), 215–230.

80. Kozen, D.C. (1992). Depth-First and Breadth-First Search. In: The Design and Analysis of Algorithms. *Texts and Monographs in Computer Science.*

81. Yuan, Y., Tasik, R., Adhatarao, S. S., Yuan, Y. and Liu, Z. (2020). RACE: Reinforced Cooperative Autonomous Vehicle Collision Avoidance. *IEEE Transactions on Vehicular Technology*, 69(9), 9279-9291.

82. Jain, R.K., Chiu, D.M.W. and Hawe, W.R. (1984). *A Quantitative Measure of Fairness and Discrimination*, Eastern Research Laboratory, Digital Equipment Corporation: Hudson, MA, ABD.

Gazili olmak ayrıcalıktır