ÇOK AJANLI KAÇMA KOVALAMA PROBLEMLERİNE TAKVİYELİ ÖĞRENME YAKLAŞIMI

AHMET TUNÇ BİLGİN

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

NİSAN 2013

ANKARA

Fen Bilimleri Enstitü onayı

_____

Prof. Dr. Ünver KAYNAK

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

_____

Doç. Dr. Erdoğan DOĞDU

Anabilim Dalı Başkanı

Ahmet Tunç Bilgin tarafından hazırlanan ÇOK AJANLI KAÇMA KOVALAMA PROBLEMLERİNE TAKVİYELİ ÖĞRENME YAKLAŞIMI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

_____

Yrd. Doç. Dr. Esra KADIOĞLU ÜRTİŞ

Tez Danışmanı

Tez Jüri Üyeleri

Başkan: Doç. Dr. Bülent TAVLI _____

Üye: Yrd. Doç. Dr. Tansel ÖZYER _____

Üye: Yrd. Doç. Dr. Esra KADIOĞLU ÜRTİŞ _____

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Ahmet Tunç BİLGİN

Üniversitesi : **TOBB Ekonomi ve Teknoloji Üniversitesi**

Enstitüsü : **Fen Bilimleri**

Anabilim Dalı : **Bilgisayar Mühendisliği**

Tez Danışmanı : **Yrd. Doç. Dr. Esra KADIOĞLU ÜRTİŞ**

Tez Türü ve Tarihi : **Yüksek Lisans – Nisan 2013**

Ahmet Tunç BİLGİN

ÇOK AJANLI KAÇMA KOVALAMA PROBLEMLERİNE TAKVİYELİ ÖĞRENME YAKLAŞIMI

ÖZET

Güvenlik başta olmak üzere yaşamın birçok alanında uygulamalarını gördüğümüz kaçma-kovalama problemleri, her dönem için popüler bir araştırma konusu olmuştur. Özellikle son on yılda, süreç içerisine öğrenmenin de katılmasıyla ajanlar akıllı ajanlar halini almış ve bir haritaya gereksinim duymaksızın çevreleri hakkında topladıkları bilgileri kendi faydaları için kullanmaya başlamışlardır. Bu yönelim, problem çözümüne farklı disiplinlerden yeni bakış açıları kazandırmayı başarmış ve konuya olan ilginin tekrar yoğunlaşmasını sağlamıştır.

Takviyeli öğrenme, kaçma-kovalama problemlerinin çözümünde kullanılan ve ajanların çevre ile etkileşiminden faydalanan bir yöntemdir. Bu yöntemle ajanlar, karmaşık algılayıcılar ve haritalar kullanmadan çevrelerinden aldıkları geribildirimler (ödüller ve cezalar) ile davranışlarını optimize ederler. Yapılan çalışmalarda, bir kaçan ajan, bir kovalayan ajan içeren senaryolar için başarılı deneyler gerçekleştirilmişse de, birden fazla kovalayan ajan bulunan takip senaryoları için yeterli sayıda araştırma bulunmamaktadır.

Bu tezde, çok ajanlı kaçma-kovalama problemlerinde takviyeli öğrenme yaklaşımı araştırılmış ve buna yönelik olarak deneyler sunulmuştur. Problemin çözümüne ilişkin benimsenen yöntemde ajanlar Watkins'in Q(λ) öğrenmesi metodunu kullanmaktadırlar. Q-öğrenmesi, uyguladığı politikadan bağımsız, optimal olarak aksiyon-değer tablosunu güncelleyen bir Geçici Farklar Kontrolü algoritmasıdır. Bizim çalışmalarımızda kullanılan Watkins'in Q(λ) yöntemi ise Q-öğrenmesinin uygunluk izleri mekanizmasıyla genişletilmiş bir hali olup, ajanın uygulayacağı keşif

niteliğindeki ilk hamleye kadar takip eden tecrübeleri kullanmaktadır. Çalışmamızda kovalayan ajanlar takımı için eşzamanlı öğrenme yaklaşımı uygulanmıştır. Bu yaklaşımda, aynı takımdaki ajanların her biri kendi aksiyon-değer tablosuna sahiptir ve takım arkadaşlarından bağımsız olarak bilgi uzayını günceller. Çalışmamızda, bahsi geçen yöntemler kullanılarak, bir kaçma kovalama problemi simülasyonu düzenlenmiş ve yapılan deneylerde elde edilen sonuçlar paylaşılmıştır.

Anahtar Kelimeler: Kaçma-kovalama problemleri, takviyeli öğrenme, Watkins'in Q(λ) algoritması, eşzamanlı öğrenme.

University : **TOBB University of Economics & Technology**

Institute : **Institute of Natural and Applied Sciences**

Science Program : **Computer Engineering**

Supervisor : **Asst. Prof. Dr. Esra KADIOĞLU ÜRTİŞ**

Degree Awarded and Date : **Master of Science – April 2013**

Ahmet Tunç BİLGİN

AN APPROACH TO MULTI-AGENT PURSUIT EVASION GAMES USING REINFORCEMENT LEARNING

ABSTRACT

The game of pursuit-evasion, which is encountered frequently in applications of security, has always been a popular research subject in the field of robotics. Especially in the last two decades, when computer scientists gave rise to learning, the agents turned into intelligent agents and they started to use the information about their enviroment for their own purposes, without using the help of a map. This tendecy drew considerable amount of attention and opened the area to newcomers from several different disciplines.

Reinforcement learning, which takes the advantage of an agent's interaction with the environment, is a method widely used in pursuit-evasion domain. With the help of this method, agents use the feedbacks (rewards and punishments) taken from the environment to optimize their behaviour, without using complex sensors and maps. Although there are successful examples of the one-pursuer one-evader scenario, there is not enough research on multi-agent pursuit-evasion problems in literature.

In this master's thesis, a research is conducted on multi-agent pursuit-evasion problem using reinforcement learning and the experimental results are submitted. The intelligent agents use Watkins' Q(λ)-learning algorithm for the solution of the problem. Q-learning is an off-policy temporal difference control algorithm. The method we used on the other hand, Watkins' Q(λ) learning algorithm, is a unified version of Q-learning and eligibility traces. It uses backup information until the first occurence of an exploration. In our work, concurrent learning is adapted for the

learning of the pursuit team. In this approach, each member of the team has got its own action-value function and updates its information space independently.

Keywords: Pursuit-evasion problem, reinforcement learning, Watkins's Q(λ) algorithm, concurrent learning.

TEŞEKKÜR

Yüksek lisans eğitimim boyunca destek ve katkılarıyla beni yönlendiren, bu tezi hazırlamam konusunda yardımlarını esirgemeyen değerli danışmanım Yrd. Doç. Dr. Esra KADIOĞLU ÜRTİŞ'e teşekkür ederim.

Üniversitemde geçirdiğim süre zarfında bana değerli katkılarda bulunan TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliği bölümü öğretim üyelerine, çalışmalarımı keyif içerisinde sürdürebilmem adına bana büyük yardımları olan değerli ofis arkadaşlarıma ve değerli aileme teşekkürü borç bilirim.

İÇİNDEKİLER

GRAFİKLERİN LİSTESİ

ŞEKİLLERİN LİSTESİ

TABLOLARIN LİSTESİ

KISALTMALAR VE SEMBOLLERİN LİSTESİ

Kısaltma/Sembol	Açıklama
TD	Temporal Difference (Geçici Farklar)
MC	Monte Carlo
ÇAS	Çok Ajanlı Sistemler
α	Öğrenme Oranı (Learning Rate)
γ	Iskonto Oranı (Discount Rate)
λ	Uygunluk İzlerinde İz Kaybolma Oranı (Trace Decay Rate)
δ	Hata (Error)
ϵ	ϵ-açgözlü (ϵ-greedy) politikada keşif hamlesi olasılığı
π	Politika
r	Ödül / ceza
e(s,a)	Durum - aksiyon çifti için uygunluk izi
Q(s,a)	s durumunda a hamlesini yapmanın tahmini değeri
V(s)	s durumunda olmanın tahmini değeri

BÖLÜM 1

1. GİRİŞ

Kaçma-kovalama (Pursuit-evasion), temellerini oyun teorisindeki diferansiyel oyunlardan alan bir problemler ailesidir. Asıl olarak matematik ve bilgisayar bilimleri disiplinlerinin araştırma konularında yer alan bu problem, günümüzde robotik alanındaki çalışmalarda sıklıkla incelenmektedir. Üzerinde on yıllardır çalışılan bu problemin güncelliğini korumasının en önemli sebeplerinden birisi çok sayıda uygulama alanına sahip olmasıdır. Bunun sonucu olarak, problemin ortaya atıldığı ilk tarihten günümüze dek birçok versiyonu geliştirilmiş ve bu problemlere çok sayıda farklı çözüm önerileri sunulmuştur. Kaçma-kovalama problemlerinin uygulamalarına özellikle trafik kontrolü, kaçak tespiti ve takibi, arama kurtarma gibi güvenlik temelli konularda rastlanmaktadır. Bu problemle ilgili olarak yapılan önemli çalışmalardan birinde, Berkeley AeRobot (BEAR) projesi kapsamında sınırlı ancak bilinmeyen bir çevre üzerinde insansız hava ve kara araçlarının, diğer bir insansız kara aracını yakalamaya çalışması deneyi gerçekleştirilmiştir [1]. Kaçma-kovalama probleminin jenerik olması, temel oyun mantığının çok basit ve anlaşılır olması, üstelik literatürde de önemli bir araştırma derinliğini içermesi, problemin çok ajanlı sistemler alanında geliştirilen yöntemler için standart bir uygulama alanı olmasını sağlamıştır. Bu savı örneklerle desteklemek gerekirse, çok ajanlı kooperatif öğrenme üzerine yapılan araştırmalarda, geliştirilen algoritmaların test edilmesi için uygulama alanı olarak çoğunlukla kaçma kovalama problemleri kullanılmaktadır [5, 7, 10, 11, 12, 13, 15, 22]. Söz konusu problem için, takviyeli öğrenme yaklaşımıyla çevresindeki bilgileri amaçları doğrultularında kullanan ve kendilerini eğiten ajanların kullanımı son yıllarda yaygınlığını arttırmıştır.

Takviyeli öğrenme, davranış psikolojisindeki hedonizmden ilham bulan ve öncelikle kendileri için bir şeyler isteyen öğrenen sistemleri temsil eden yaklaşımdır [23]. Bu fikir yalın olarak, davranışlarını çevresiyle olan etkileşimleri doğrultusunda güncelleyen ve ondan maksimum faydayı sağlamayı öngören ajanları barındırır. Takviyeli öğrenmenin modelden bağımsız olması, önbilgi veya öğretmen

gerektirmemesi, bu yaklaşımı kaçma kovalama problemleri için uygun kılmaktadır. Akıllı bir ajan, haritasını bilmediği ve hakkında hiçbir önbilgisinin bulunmadığı bir alanda, öğretmen desteği dahi almadan sadece çevresinden aldığı ödül ve cezalarla kendi performansını iyileştirebilmektedir. Böyle bir ajan, yönlendirmeye ihtiyaç duymadığı ve harita bilgisi kullanmadığı halde ilk defa dâhil olduğu bir ortamda kendi öğrenme yöntemlerini kullanarak optimal yolları keşfedebilir. Açıkça programlamaya ve yeniden yapılandırmaya ihtiyaç duymadan, kendi özüne ait bir beceriyle öğrenebilmesi bu ajanları eşsiz kılmaktadır [30].

Son yıllarda, öğrenme bakış açısıyla kaçma kovalama problemlerinin araştırıldığı kayda değer sayıda çalışma bulunmaktadır. Bir kaçan - bir kovalayan ajanın bulunduğu oyun senaryosu üzerinde birçok deney yapılmış ve başarılı sonuçlar elde edilmiştir. Yalnız kaçağın birden fazla akıllı ajan ile yakalanması konusunda tatmin edici sonuçlara ulaşan yeterince çalışma bulunmamaktadır. Ayrıca, bu alanda yapılan araştırmalarda ajanların tek bir öğrenici (akıl) kullandıkları takım öğrenmesi (team learning) yöntemi yaygındır. Yani, her ajan edindiği bilgiyi takım adına merkezi bir akla (aksiyon-değer tablosu) işlemektedir. Diğer bir yöntem olan eşzamanlı öğrenmede (concurrent learning) ise takımdaki her üyenin kendine ait bir aklı vardır. Bu iki yöntem arasında yapılan tercih, ölçeklenebilirlik ve işbirliği arasındaki ödünleşmenin (trade-off) sonucunda alınan karardan ileri gelir. Son olarak, literatürdeki mevcut çalışmalarda sadece kovalayan takımın takviyeli öğrenme uyguladığı örnekler yoğunluktadır. Bu çalışmalarda, kaçan oyuncu için genellikle bir kontrol stratejisi kullanılmakla beraber, rastgele yürüme ve duvar takibi kullanılan araştırmalar da bulunmaktadır. Bu bilgiler ışığında, takviyeli öğrenme ile desteklenen kaçma-kovalama problemleri alanında bütün ajanların öğrendiği, çok ajanlı sistemler üzerinde araştırma yapılması ihtiyacı görülmüştür.

Bu tez çalışması kapsamında çok ajanlı kaçma kovalama problemlerine takviyeli öğrenme yaklaşımı uygulanmıştır. Kullanılan öğrenme yöntemi Watkins'in Q(λ)-öğrenmesi yöntemidir. Yapılan araştırmalar simülasyonlar ile desteklenerek elde edilen sonuçlar detaylı olarak sunulmuştur. Bu araştırma ile ilgili konuya yapılan katkılar aşağıdaki gibidir:

- Takviyeli öğrenme yapan ajanların bulunduğu kaçma-kovalama problemlerinde çok robotlu sistemler üzerinde yeterli düzeyde araştırma bulunamamıştır. Bu eksiklik, eşzamanlı öğrenme ile bağımsız öğreniciler (independent learner) kullanan çok ajanlı sistemler için daha da fazladır. Bu tezde, eşzamanlı öğrenme yaklaşımını benimseyen çok robotlu bir sistem kullanılarak literatürde derinleştirilebileceğini düşündüğümüz bir alana giriş yapmak amaçlanmıştır.
- Bu araştırma kapsamında, probleme dâhil olan bütün ajanlar takviyeli öğrenme gerçekleştirmektedirler. Oysa ilgili alanda taranılan mevcut çalışmalarda sadece kovalayan ajan takımının eğitilmesine önem verildiği gözlemlenmiştir. Birçok senaryoda, kaçak sadece kendisi için belirlenen kontrol stratejisini uygular. Dolayısıyla, birinci oyundan sonuncu oyuna kadar geçen süreçte rakibi gelişirken, kaçağın stratejisinde değişen herhangi bir şey olmamaktadır.
- Bu tez çalışması boyunca yapılan ve çeşitli senaryolara sahip deneyler için karşılaştırma ortamı sunulmaktadır. Her ne kadar, araştırmada göstermek istediğimiz yaklaşım çok ajanlı sistemlere takviyeli öğrenme yaklaşımı olsa da; simülasyonlarda elde edilen sonuçların değerlendirilebilmesi amacıyla yalın oyun sistemleriyle karşılaştırılmaları gerekir. Bu yüzden, çevresel koşullar aynı olacak şekilde, asıl deneyin sonuçları ile tek ajanlı kaçma kovalama probleminin sonuçları kıyaslanmaktadır. Ayrıca, akıllı kaçma uygulayan ajanın performansının anlaşılması amacıyla, akıllı kaçma uygulamayan ajan ile kıyaslaması da sunulmuştur.

Bu tez çalışması şu şekilde düzenlenmiştir: İlk bölümde, tez çalışmasının ana temellerini, ilgili alanda yapılan çalışmalarda gelinen son durumu ve bu çalışmaya neden ihtiyaç duyulduğunu kısaca açıklayan giriş bölümü bulunmaktadır. 2. bölümde, kaçma kovalama problemleri, çok ajanlı sistemler, ϵ-greedy aksiyon seçimi algoritması, takviyeli öğrenme, Q-öğrenmesi, uygunluk izleri gibi yapılan çalışmanın temellerini oluşturan konularla ilgili genel bilgilere yer verilmiştir. 3. bölümde, bu tez çalışması için düzenlenen oyunların senaryosu, problemin tanımı ve önerilen yöntemler açıklanmaktadır. Bölüm 4'te, yapılan deneyler ve simülasyonlardan elde

edilen sonuçlar detaylı olarak sunulmuş, bu sonuçlar birbirleriyle karşılaştırılmış ve benimsenen yöntemin diğerlerine kıyasla avantajları ve dezavantajları belirtilmiştir. Sonuç bölümünü içeren 5. bölümde ise, bu tez çalışmasından elde edilen kazanımlara dair görüşler açıklanarak, gelecek çalışmalara değinilmiş ve tez çalışması sonlandırılmıştır.

BÖLÜM 2

2. İLGİLİ ÇALIŞMALAR

Bu tez çalışması kapsamında, konumuzla ilgili kavramlar üzerinde araştırmalar yapılmış ve deneylerde çeşitli yöntemler kullanılmıştır. Bu bölümde, bahsi geçen kavramlar ve kullanılan yöntemler, tezde yapılan çalışmalara ışık tutacak şekilde açıklanarak, literatürdeki örneklerine de kısaca değinilecektir.

2.1 Kaçma-Kovalama Problemleri

Bir kavram olarak kaçma-kovalama problemi, kolayca anlaşılacağı üzere ilhamını yaşamın içinden alır. Bu problemde temel olarak kovalayan oyuncu, kaçan oyuncuyu minimum sürede yakalamaya çalışırken, kaçan oyuncu ise bu süreyi uzatmaya veya izini kaybettirmeye çalışmaktadır. Problem ilk defa 1965 yılında Rufus Isaacs tarafından "Diferansiyel Oyunlar" isimli kitapta tanımlanarak literatüre kazandırılmıştır [19]. Bu eserde, "Öldürmeye Meyilli Şoför" (Homicidal Chauffeur) oyunu üzerinden formüle edilen problemin füze rehberlik sistemi uygulamasında kullanılması amaçlanmıştır. Daha sonraki bir dönemde yapılan başka bir araştırmanın sonucunda ise saklambaç oyunu baz alınarak, oyundaki iki taraf için de optimal yöntemlerin sorgulandığı çalışma sunulmuştur [20]. Ardından, 1976 yılında T.D. Parsons yazdığı makalede daha farklı bir soru sorarak probleme yeni bir boyut kazandırmıştır: "Bir mağarada kaybolan ve rastgele hareket eden birini, kişinin nasıl hareket ettiğinden bağımsız olarak en az kaç kişilik bir kurtarma grubuyla bulabilirsiniz?" [21]. Bu yönelimle beraber, problemin çözümünde ilk defa çizgeler kullanılmış ve sürekli sistemlerden kesikli sistemlere adım atılmıştır. Ayrıca, daha önceki oyun tanımlarının aksine, bu oyundaki ajanların amaçları birbirleriyle aynı doğrultudadır. Literatürde, kaçma-kovalama problemlerinin doğuşu bu şekilde gerçekleşmiştir.

Malum olduğu üzere, problem ilk defa ortaya atıldığı andan günümüze kadar birçok değişime uğramış ve sonucunda kaçma-kovalama problem ailesini oluşturmuştur.

Problemin çeşitli sürümleri hırsızlar ve polisler, av ve avcı, prenses ve canavar ve öldürmeye meyilli şoför gibi adlarla anılmaktadır. Bu problemlerin özelinde oyuncu sayısı, problemin gerçekleştiği harita (dünya), oyunculara bir haritanın veya çevre modelinin sunulup sunulmadığı, kovalayan veya kaçan oyuncuların işbirliği, kazanma/kaybetme koşulları gibi birçok farklı parametre tanımlanır. Kaçma kovalama problemlerinin gerçek dünya uygulamaları gözetleme, robot takip sistemleri, trafik kontrolleri gibi kritik alanlarda kullanılmaktadır.

Literatür gözden geçirildiğinde, kaçma-kovalama problemleri üzerinde çok sayıda çalışma yapıldığı ve problemin hala güncel araştırma konularının bir parçası olduğu görülür. Yalnız bu alandaki en temel problem yapılan çalışmaların simülasyon ortamından gerçek dünya ortamına taşınmasındaki güçlüklerdir. Giriş bölümünde de bahsedildiği üzere bu alanda yapılmış en önemli çalışmalardan birisi, 2002 yılında BEAR projesi kapsamında gerçekleştirilmiştir [1]. Kaçma kovalama oyunlarının olasılıksal analiz kullanılarak değerlendirildiği bu çalışmada, havadan ve karadan takip yapabilen bir robot takımı, kendilerinden kaçan bir robotu yakalamayı amaçlar. Bulundukları ortamın haritasını çıkarma ve iletişim kurma yetilerine sahip olan robot takımı, dağıtık ve hiyerarşik bir düzene sahiptir. Khepera 3 robotu kullanılarak gerçekleştirilen diğer bir araştırmada ise dağıtık sistem mimarisine sahip ve yakın mesafelerde iletişim kurabilen bir robot takımı, yeterli sayıda robot ile bırakıldıkları ortamı temizlemeyi garanti eder [11]. Ortamın haritasına ihtiyaç duymayan ve küresel konumlandırma (global localization) yapmayan robotlar, bu garantiyi yerel sınırlar (local frontiers) kullanarak sağlarlar. Kaçaklardan temizledikleri bir ortamın tekrar kontamine olmayacağını garanti ettikleri için takımda yeterli sayıda robot yoksa sınır güvenliğini tehlikeye atmayarak genişlemeyi durdururlar. Yine fiziksel donanımla gerçekleştirilen bir çalışmada, birden fazla hareketli hedefin kooperatif bir robot takımıyla izlenmesi konu alınmıştır [10]. Burada incelenen anahtar mesele, hareketli hedeflerin verimli bir şekilde tespit edilmesi için robotlar üzerindeki kısıtlı mesafe sensörlerinin nasıl yerleştirileceğidir. Kaçma-kovalama problemlerinde gerçek dünya uygulamalarına yer verilen bazı kayda değer çalışmalar bu şekilde sıralanabilir.

Yalın bir amaç olarak kaçma-kovalama problemlerinin araştırıldığı çalışmalar son yıllarda azalmasına rağmen etkisini korumaktadır. Amigoni ve Basilico tarafından ayrık durum uzayı üzerinde yapılan çalışmada "Sınırları bilinen bir ortamda, bir takipçinin kaçağı ortamdan temizlemesi için kullanılabilecek optimal strateji nedir?" sorusuna cevap aranmıştır [3]. Takipçi, tek girişi ve tek çıkışı bulunan harita üzerinde, kaçağı ortama girdiği andan itibaren elinden kaçırmadan minimum sürede yakalayabilmek için en uygun stratejiyi aramaktadır. Chung ve Burdick ise bir uzaysal arama (spatial search) problemine, olasılıksal arama stratejisiyle çözüm aramışlardır [8]. Bu çalışmada hareketli bir ajandan, sınırları bilinen bir bölge üzerinde, bölgenin içinde yer aldığı muhtemel, sabit konumdaki bir kaçağı tespit etmesi veya verilen sınırlar içinde olmadığını ilan etmesi istenmektedir. Kaçma-kovalama probleminin 3 boyutlu yapılar üzerinde araştırıldığı bir problemde ise oyunun simülasyonu, çok katlı ofis binası gibi birden fazla seviye bulunan yapılarda gerçekleştirilmiştir [16, 18].

Kaçma-kovalama oyun ailesi, çok ajanlı sistemlerin uygulanmaları için uygun alanlardan biridir; çünkü bu konu üzerinde çok geniş bir yelpazede yaklaşımlar denenmiştir ve konunun farklı senaryolarının gösterilmesi açısından birçok değişik yapılandırması olduğu bilinmektedir [6]. Oyunun işleyişi incelendiğinde, çok ajanlı robotik sistemler için uygun bir soyutlama ortamı sunduğu görülmektedir. Çok ajanlı sistemlerin kaçma kovalama oyunu üzerinde uygulandığı ilk örneklerden biri Haynes ve Sen'in gerçekleştirdikleri çalışmadır [2]. Burada av/avcı modeli üzerinden dağıtık yapay zekâ (Distributed AI) araştırması yapılmıştır. Genetik programlama kullanarak, avcıların içinde davranışsal stratejilerin evrilmesi ve çaprazlama yöntemiyle heterojen takımların içindeki uzmanların öne çıkarılması amaçlanmıştır. 2009 yılında, kapalı alan kaçma-kovalama problemleri için sunulan bir çizge arama algoritmasında ise koordineli bir arama takımı fiziksel bir ortamın grafik temsilinde düşman hedefini bulmaya çalışmaktadır [4, 12]. Bu oyun üzerinden, çok ajanlı sistemlerde kaynak dağıtım problemi (resource allocation problem) araştırılmaktadır. Kolling ve Carpin'in makalesinde ise, haritasız çok robotlu kaçma kovalama problemleri özgün bir yaklaşımla temsil edilmiştir [15]. Bu çalışmayı farklı kılan şey, robotların kısıtlı becerileriyle sadece duvarları ve yakınlarındaki takım arkadaşlarını

takip edebilmeleridir. Dağıtık bir algoritma kullanıp kısa mesafede iletişim kurabilen robot takımı, karşılıklı duvarlar arasında bir hat oluşturarak haritayı temizler.

Kaçma-kovalama problemleri, çok ajanlı sistemler için olduğu gibi, öğrenme algoritmaları için de uygun bir test ortamı sunar. Bu sebeple, son zamanlarda takviyeli öğrenme alanında gerçekleştirilen birçok çalışmada akıllı av ve avcı modelleriyle kaçma kovalama problemleri incelenmektedir [7, 9, 17, 28]. Ishizawa, Sato ve Kakazu tarafından 2003'te yayınlanan bir makalede, kaçma-kovalama problemine takviyeli öğrenme gerçekleştiren çok ajanlı heterojen bir sistemle yaklaşım sunulmuştur [7]. Bu çalışmada 4 avcıdan oluşan bir ajan takımı, avı yakalama görevini gerçekleştirebilmek için takviyeli öğrenme yardımıyla işbirliği yapmaktadır. Oyunun başında bütünüyle homojen olan ajanların becerileri, öğrenme sürecinde heterojen bir hale dönüşmektedir. Li, Pan ve Hong ise yaptıkları araştırmada çok robotlu kooperatif kaçma-kovalama problemlerine takviyeli öğrenme ve veri madenciliği kullanarak yaklaşmışlardır [5]. Bu çalışmada, sayısı birden fazla olan ve heterojen özellikler gösteren kaçaklar hakkındaki bütün faktörler gözetilerek öznitelik ilişkilerinin bir veri kümesi oluşturulur. Ardından bu veri kümesi üzerinde veri madenciliği yöntemleriyle ilginç kurallar tespit edilir ve bu kurallar doğrultusunda her kaçak için bir takip takımı oluşturulur. Takip takımları, en iyi yolları bulmak amacıyla takviyeli öğrenme metotlarından faydalanır. 2011'de Desouky ve Schwartz tarafından gerçekleştirilen araştırmada Q(λ)-öğrenmesi ve bulanık kontrolör kullanılarak öldürmeye meyilli şoför problemi incelenmiştir [17]. Bilindiği üzere, Q-öğrenmesi durum ve aksiyon uzaylarının ayrık yapıda olduğu durumlarda kullanılmaktadır. Ancak bu çalışmada kaçma kovalama problemlerini daha gerçekçi bir ortamda sunmak adına bulanık sistemlerden yararlanılmış ve Q(λ) bulanık sonuç çıkarma (zero-order Takagi-Sugeno) sistemi ortaya atılmıştır. Sonuç olarak, bu çalışmayla takviyeli öğrenme kullanılan kaçma-kovalama problemlerinin ayrık yapıdan sürekli yapıya çıkarsanabileceği gösterilmiştir.

Bu bölümde kaçma-kovalama problemleri kısaca tanımlanmış ve oyunun farklı türlerine dair bilgilere yer verilmiştir. Ayrıca, yapılan literatür araştırması kapsamında problemin ortaya atılışından günümüze kadar gerçekleştirilen çalışmalarla ilgili açıklamalar sunulmuştur.

2.2 Çok Ajanlı Sistemler

Günlük yaşantıda karşımıza çıkan veya otonom bilgisayarlı sistemlerle formüle etmeye çalıştığımız kimi problemleri tek birey/ajan ile tanımlamak veya çözmek mümkün değildir. Bazen çözümü uygulanabilir veya verimli (feasible) hale getirmek için, bazen de zorunluluktan birden fazla ajana ihtiyaç duyarız. Bunu daha net ifade etmek için birden fazla ajana ihtiyaç duyulan üç durum somut örneklerle aşağıdaki gibi gösterilebilir:

- İşbirliğine gereksinim duyulması. (*Örn. Takım sporları.)*
- Rekabet ve/veya taraf içeren durumlar. (*Örn. Kaçma-kovalama problemleri.)*
- Tek ajanla gerçekleştirilmesi uzun sürebilecek veya imkânsıza yakın olan görevleri, birden fazla ajanla parçalara bölmek ve kaynak dağıtımı yapmak. (*Örn. Bir arazinin mayınlardan temizlenmesi.)*

Bunun yanında çok ajanlı sistemlerin kullanımının zorunlu olmadığı hallerde dahi bize sunduğu bazı avantajlar vardır [30]. Bunlar aşağıdaki tabloda (Tablo 2.1) gösterilmektedir.

Kavram	**Sağladığı Avantaj**
Ölçeklenebilirlik:	Monolitik bir sisteme yeni beceriler kazandırmaktansa, çok ajanlı bir sisteme bu ihtiyacı karşılayan bir ajan eklemek daha kolaydır.
Paralel iş gerçekleştirimi:	Yapılan iş parçalara bölünebiliyorsa, bu bağımsız görevler farklı ajanlar tarafından eşzamanlı gerçekleştirilir.
Dayanıklılık:	Ajanlardan bir tanesi arızalandığında, diğer bir ajanın bunu tolere edebilmesi dayanıklı bir sisteme işaret eder.
Coğrafi dağılım	Bir coğrafyaya yayılmış olan problemi, birden fazla ajan alanda dağılarak eşzamanlı gerçekleştirir.
Programlanabilirlik	Bütün görevlerin merkezi bir sisteme programlanması yerine, alt görevler belirlenerek bunlar farklı ajanlara atanır.
Maliyet	İhtiyaç duyulan bütün özellikleri barındıran maliyetli bir ajan yerine, farklı görevler için alınmış basit ve ucuz ajanlar kullanılır.

Tablo 2.1 Çok ajanlı sistemlerin getirileri

Öte yandan, her karmaşık sistem için çok ajanlı sistemlerin kullanılması gerektiğini iddia edemeyiz; çünkü bunun getireceği avantajların yanında bazı zorlukları da mevcuttur [35]. Örneğin; tek ajanlı bir sistemde, eğer dünyaya ajan haricinde hükmeden bir şey yoksa bu dünya ajan için durağandır. Yani ajanın iki hamlesi arasında yaşadığı dünya sabit kalmaktadır. Çok ajanlı bir sistemde ise dünyaya birden fazla ajan müdahil olduğu için, ajanlardan biri beklerken dünya değişebilmektedir. Çok ajanlı sistemler ele alındığında karşılaşılan en önemli zorluklardan biri budur. Öğrenme özelinde ise, eşzamanlı öğrenme yönteminde birden fazla öğrenicinin (learner) bulunması dünyayı hareketli kılar. Bu durum geleneksel makine öğrenmesi teknikleri için bir ihlal oluşturduğundan, eşzamanlı öğrenmede modern veya modifiye edilmiş öğrenme tekniklerine ihtiyaç duyulmuştur [29]. Çok ajanlı bir sistemin getireceği diğer bir zorluk ise kontroldür. Burada, tek ajanlı bir sistemin aksine birbirleriyle etkileşime girecek ajanlar bulunur. Dolayısıyla ajanların kendi aralarındaki iletişimi, görev dağılımı, birbirlerine nasıl tepki verecekleri üzerinde durulması gereken konulardır.

Çok ajanlı sistemlere şiddetle ihtiyaç duyulan bir problem "Robotik Mayın Temizleme (Robotic Demining)"dir [34]. Ne kadar büyük olursa olsun bir araziyi tek bir robotla mayınlardan temizlemek tabii ki mümkündür; ancak mantıklı değildir. Coğrafi dağılımın avantajı kullanılarak, bu görev birden fazla homojen robotla parçalara ayrılabilir. Arazi, robot sayısı kadar eşit alana bölünüp her alana bir robot atandığında süre anlamında kazanç sağlanacağı aşikârdır. Bu örnekteki robotlar görevlerini birbirlerine paralel ancak bağımsız olarak yaptıkları için kendi aralarında bir iletişime de ihtiyaç duymazlar. Ayrıca, robotlardan biri arızalanıp görevini yapamayacak hale gelirse, diğer bir ajan onun görevini üstlenebilecektir. Yine mayın temizleme görevinin farklı bir gerçekleştiriminde ise, mayınları tespit edebilen ve etkisiz hale getirebilen iki tür robot olduğunu varsayalım. Mayınları tespit eden robot onları etkisiz hale getiren robota haber verecek ve ardından mayınlar araziden temizlenecektir. Yalnız bu örnekte heterojen robotlar, kaynak paylaşımı ve işbirliği söz konusu olacaktır. Bu durumda neyin, hangi amaçla ve nasıl kullanılacağını imkânlar ve ihtiyaçlar belirler.

2.2.1 Öğrenme Bakış Açısıyla Çok Ajanlı Sistemler (ÇAS)

Öğrenme bakış açısıyla kooperatif çok ajanlı sistemleri incelediğimizde karşımıza iki ana başlık çıkmaktadır [31, 33]. Bunlardan ilki olan takım öğrenmesinde (team learning), tüm ajan takımının davranışlarına bütünüyle tek bir merkez, bir *öğrenici* karar verir. İkinci kategori olan eşzamanlı öğrenmede (concurrent learning) ise her takım üyesi kendi davranışını kendi öğrenme işlemiyle belirler.

2.2.1.1 Takım Öğrenmesi

Bu öğrenme yönteminde takımdaki bütün oyuncuların adına karar veren merkezi bir öğrenici vardır. Öğrenici her oyuncunun içinde bulunduğu durumu gözeterek onlar adına davranışlarını belirler. Takımın genel başarısını dikkate alırken, ajanların bireysel performansına dair bir endişe taşımadığı için eşzamanlı öğrenmeye göre daha basit olduğu söylenebilir. Yalnız bu yöntemle ilgili en büyük sorun, kimi durumlarda çok boyutluluğun laneti (curse of dimensionality) fenomenine ulaşabilecek geniş durum uzayıdır. N adet durumda yer alabilecek m adet ajanın durum uzayı büyüklüğü n^m olacaktır. Bu yaklaşımla ilgili ikinci bir problem ise bütün bilgilerin aynı anda tek bir merkezde işlenmiş olmasının gerekmesidir [29].

Takım öğrenmesi kendi içinde de homojen takım öğrenmesi ve heterojen takım öğrenmesi olarak iki ana başlığa bölünür. Homojen ajanlardan oluşan takımda bütün oyuncular aynı görev bilincine sahiptirler ve kendi aralarında ayrışmazlar. Heterojen takımlarda ise, ya oyuncuların arasında başlangıçtan itibaren görev dağılımları bulunur ya da zaman içerisinde karşılaştıkları koşullardan dolayı kendi aralarında ayrışırlar.

2.2.1.2 Eşzamanlı Öğrenme

Eşzamanlı öğrenme yönteminde, takımın her üyesi kendi bireysel performansını iyileştirmekten sorumludur. Bunun için her biri kendi adına öğrenir ve davranışına sadece kendisi karar verir. Burada desteklenen mantık, bir takımı oluşturan parçaların başarısı artarsa, takımın genel başarısının da artacağıdır. Eşzamanlı öğrenme söz konusu olduğunda karşılaşılan en büyük güçlük, ajanların kendilerini, üzerlerinde kontrol haklarının bulunmadığı diğer ajanları düşünerek adapte

etmeleridir [29]. Yani bir ajan davranışını belirlerken sadece kendisini düşünür; fakat diğerlerini de kendi fiillerine maruz bırakır. Bu açıdan, takım öğrenmesine kıyasla daha küçük bir durum uzayı; fakat iç içe geçmiş daha karmaşık bir problem vardır [32].

Bu tezdeki yaklaşım dikkate alındığında, sunduğumuz kaçma-kovalama probleminde eşzamanlı öğrenme yöntemini benimseyen bir avcı takımı mevcuttur. Ayrıca bu yaklaşımda çok ajanlı sistemlerin sağladığı avantajlardan bölgesel dağılım ve dayanıklılık kavramlarının kullanıldığı söylenebilir.

2.3 Takviyeli Öğrenme

Eğer bir ajan, içinde yaşadığı dünyayla ilgili gözlem yapıyor ve bu gözlemleri daha sonraki görevlerinde performansını arttırmak için kullanıyorsa bu ajanın öğrendiğini söyleriz [36]. Bu tezde kullanıldığı şekilde daha spesifik bir tanım yapmak gerekirse; Bir ajana öğreniyor diyebilmek için yaptığı bir hamleden dolayı çevresinden aldığı ödül veya cezanın ona daha sonra alacağı kararlarda yol göstermesini bekleriz. Kaçma-kovalama problemi özelinde düşünürsek, bir ajan neden öğrenmeye ihtiyaç duyar? Öğrenmesini beklediğimiz bilgileri ona problemin en başında veremez miyiz? Bunun temel sebebi, bir kaçma-kovalama problemine başlarken rekabetin iki tarafındaki ajanların da ellerinde herhangi bir veri olmamasıdır. Dolayısıyla rakiplerini alt etmek için lehlerine kullanabilecekleri hiçbir kozları yoktur ve bunu ancak oyunun içindeki geribeslemelerden öğrenebilirler.

Bu tezde takviyeli öğrenme yönteminden yararlanılmıştır. Takviyeli öğrenmenin arkasındaki esas fikir, yapacağı hamlelerin karşılığında çevresinden ödül isteyen "hedonistik" bir öğrenme sistemi oluşturmaktır [14]. Oyuncular, yapacakları hamleleri ileride onlara sayısal anlamda maksimum toplam faydayı getirmesi beklentisiyle seçerler. Bunun için oyundaki her akıllı ajanın bir öğrenicisi (learner) vardır. Oyunun başlangıcında içinde hiçbir veri bulunmayan bu öğrenici, yaptığı hamlelerin karşılığında aldığı geribeslemelerle zihnini doldurmaya başlar.

Takviyeli öğrenme söz konusu olduğunda verilmesi gereken en önemli kararlardan birisi keşif (exploration) ve uygulama (exploitation) arasındaki ödünleşmedir (trade-

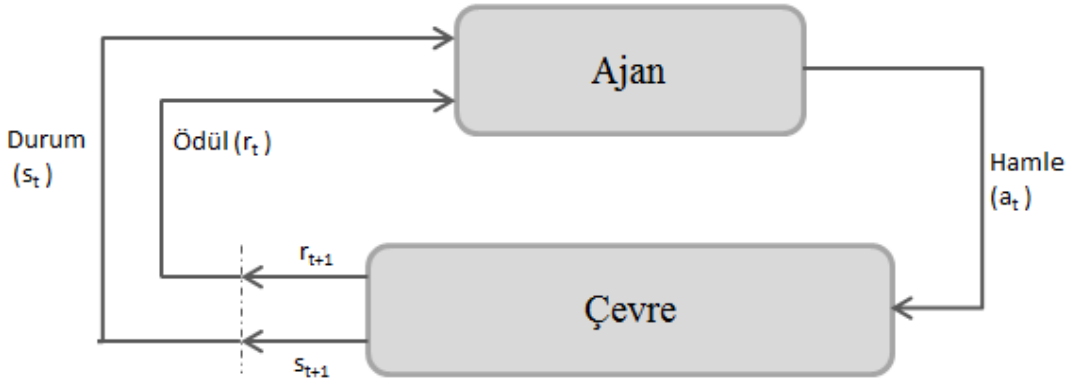
off) [23]. Akıllı ajan, bir hamle yapacakken aldığı ödül ve cezalarla doldurduğu zihnini kullanır. Normal şartlar altında ajanın, ona en yüksek ödülü getirecek olan açgözlü hamleyi uygulaması beklenir. Yalnız önceliği her zaman maksimum fayda sağlayan aksiyona vermek oyuncu için uzun vadede başarılı bir yöntem olmayabilir; çünkü bu durumda haritada keşfedilmemiş yollar kalacaktır. Bir öğrenen ajan yeni şeyler keşfetmeye çalışmadığında, ancak ve ancak zihnindeki kısıtlı bilgilerle o ana kadar kendisine en çok fayda getiren hamleyi seçer. Sonuç olarak, keşfetmeye inanmayan açgözlü bir ajan kendi kısır dünyasında yaptığı tercihlerin faydalı olduğunu düşünebilir; fakat saklı kalan dünyada daha faydalı yolların bulunması olasıdır. Bu anlamda, keşif ve uygulama arasındaki denge bir akıllı ajan için karar verilmesi gereken kritik bir etmendir. Oyuncu ödülleri kazanmak için sahip olduğu bilgileri uygulamak zorundadır; öte yandan gelecekte daha iyi hamleler yapabilmek için de dünyasını keşfetmesi gerekir.

Oyuncu ve dünya haricinde takviyeli öğrenme sistemlerinin dört temel öğesi bulunmaktadır [23]. Bunlar;

- Belirli bir zamanda, oyuncunun içinde bulunduğu koşullara göre davranışını belirleyen aksiyon seçimi, bir *politika (policy)*,
- Oyuncunun bir durumdan diğer bir duruma geçmesinin getireceği anlık ödülü belirleyen *ödül fonksiyonu (reward function)*,
- Oyuncunun bir durumdan diğer bir duruma geçmesinin uzun vadede getireceği faydayı öngören *değer fonksiyonu (value function)*,
- Oyuncunun içinde yaşadığı dünyanın davranışlarını modelleyen *çevre modeli (environment model)*'dir.

Takviyeli öğrenmenin temelinde ajan ve çevre arasındaki ilişki yatar. Ajan, öğrenici ve karar vericidir. Ajanın etkileşim kurduğu, onu verdiği ödül ve cezalarla eğiten şey ise çevredir. Oyun devam ettiği sürece bu etkileşim sürmek zorundadır. Problemin her yeni bölümünde ajan ve çevre arasındaki etkileşim ayrık zaman dilimleriyle gösterilir, $t = 0, 1, 2, \ldots$ S, çevre üzerindeki bütün olası konumların bir kümesiyse, bir t ayrık anında ajanın bulunduğu konum $s_t \in S$ ile gösterilir. Benzer şekilde, $A(s_t)$, ajanın t anında yapabileceği hamlelerin kümesiyse, uyguladığı aksiyon $a_t \in A(s_t)$'dir.

Ajan, verdiği kararın neticesinde çevreden bir ödül, r_{t+1} alır. Bahsedilen bu etkileşime dayalı süreç şekil 2.1'de gösterilmektedir [23], [24].



Şekil 2.1 Takviyeli öğrenmede ajan ve çevre etkileşim süreci

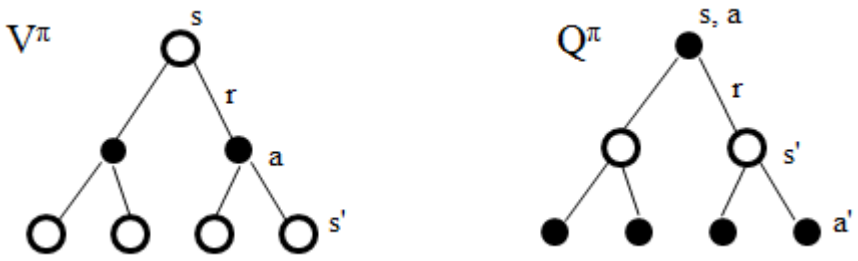
Oyundaki her ayrık *t* anında, ajanın bulunduğu konumdan yapabileceği her hamlenin bir ihtimali vardır. s_t = s ve a_t = a olmak üzere, ajanın s konumunda bulunduğu *t* anında *a* hamlesini gerçekleştirme olasılığı π_t (s,a) ile gösterilir. Bu olasılığın nasıl hesaplanacağı aksiyon seçiminde kullanılan yönteme bağlıdır.

Takviyeli öğrenme gerçekleştiren bir ajan, oyunun sonuna geldiğinde toplayabileceği en yüksek ödülü toplamış olmayı arzular. Oyun süresince alınan ödüllerin r_{t+1},r_{t+2},$r_{t+3,\ldots}$ şeklinde sıralandığını kabul edelim. Bunların toplamı, yani *beklenen kazanç*, şu şekilde gösterilir:

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \ldots + r_{T,} \qquad (2.1)$$

Bu gösterimde T, ajanın hedefe ulaştığı, yani problemin sonlandığı anı temsil eder. Epizodik (bölümlerden oluşan) bir oyunun bir bölümü sonlandığında, ajanlar başlangıç konumlarına döner ve oyun baştan başlar. Burada, elde edilen kazancın gerçek değerini hesaplamak için iskontoya ihtiyaç duyulur. İskonto, ajanın o an içinde bulunduğu bir durumda alabileceği ödülün büyüsüyle, gelecekteki bir ödülü kaçırmamasını sağlar. Diğer bir deyişle, gelecekteki ödülün şimdiki değerini belirler. Bunun için 0 ve 1 aralığında bir iskonto oranı, γ, oluşturulur.

Takviyeli öğrenme yöntemlerinde, bir ajanın o an içinde bulunduğu konumun değerini belirleyen bir değer fonksiyonu bulunur. Bir *s* durumunun değerini, o konumdan yola çıkıldığında ajanın toplaması beklenen ödül miktarı belirler. Buradan yola çıkarak, π politikası altında bir *s* konumunda bulunmanın matematiksel değeri $V^{\pi}(s)$ ile gösterilir. Benzer şekilde, π politikasının uygulandığı bir dünyada, *s* konumundayken *a* hamlesini yapmanın değeri $Q^{\pi}(s,a)$'dır. Burada Q'ya aksiyon-değer fonksiyonu denir. V^{π} ve Q^{π} değer fonksiyonları edinilen tecrübeler neticesinde tahmin edilebilirler. Bu iki fonksiyonun uygulama diyagramları şekil 2.2'de gösterilmektedir [23]. $V^{\pi}(s)$ fonksiyonu için ajanın bir *s* konumuyla karşılaşma sayısı ve $Q^{\pi}(s)$ fonksiyonu için ajanın *s* konumunda *a* hamlesini uygulama sayısı sonsuza yaklaştığında ortaya çıkan değerlerin yakınsamaları beklenir.



Şekil 2.2 V^{π} ve Q^{π} için uygulama diyagramları

Bu bölümün başında bahsedildiği üzere bir takviyeli öğrenme probleminin en kritik kararı keşif ve uygulama birbirinden ayrıştırılırken verilir. Bu esnada izlenecek politikayı belirlemek için de bir aksiyon seçim yöntemine karar verilmelidir. Bu tez çalışmasında, ajanlar hamlelerini ϵ-greedy aksiyon seçim algoritmasına göre seçmektedirler.

2.3.1 ϵ-Greedy Aksiyon Seçimi

Takviyeli öğrenme metodunu uygulayan bir ajan, etkin bir keşif yapma içgüdüsüne sahiptir. Böylelikle yaşadığı her deneyimi bilgi dağarcığına katar. Oyunun bir *t* anında, ajan, yaşadığı dünya üzerinde bulunduğu *s* konumundayken yapabileceği hamlelerin matematiksel değerlerini karşılaştırır. Açgözlü yaklaşım ona değeri en yüksek olan aksiyonu seçmesini söyler. Öte yandan, içindeki keşfetme içgüdüsü de

diğer tercihlerin onu ileride daha büyük ödüllere ulaştırabileceğini anımsatır. İşte bu noktada kararı vermek üzere aksiyon seçim algoritması devreye girer. Bu tezde uygulanan kaçma-kovalama probleminde ϵ-greedy aksiyon seçim yöntemi kullanılmıştır.

Bilindiği gibi, her zaman için bir karar alınırken nitel veya nicel olarak öne çıkan en az bir tercih bulunur. İşte bu tercihe açgözlü tercih denir. Bir ajan, açgözlü tercihi uyguluyorsa onun mevcut değer bilgilerinden faydalandığını anlarız (exploitation). Aksi durumda, aynı ajan açgözlü hamle dışında bir tercihte bulunuyorsa da, bu sefer onun keşfettiğini söyleriz (exploration). Bu noktada, her zaman açgözlü hamleleri tercih etmenin basit bir alternatifi çoğunlukla açgözlü seçimler yapıp, ϵ olasılıkla da diğer hamlelerden birini rastgele seçmektir. Açgözlü-yaklaşık bir aksiyon seçim kuralı sunan bu yönteme ϵ-greedy yöntemi denir. Oynanan oyun sayısı sonsuza giderken her aksiyon sonsuz defa örnekleneceği için, bir *a* aksiyonunun değerinin optimal değere yakınsayacağı söylenebilir. Yalnız burada ϵ olasılık değerinin asimptotik anlamda kademeli olarak azalacağı kabul edilmektedir.

2.3.2 Q-öğrenmesi

Takviyeli öğrenme söz konusu olduğunda kullanılan en önemli yöntemlerden bir tanesi geçici farklar (TD, temporal difference) öğrenmesidir. TD öğrenmesi, Monte Carlo (MC) yöntemleri ve dinamik programlamanın bir bileşimi olarak tanımlanabilir. Yani TD yöntemleri, dinamik programlama örneğinde olduğu gibi problemin sonuçlanmasına gerek duymadan tahminlerini güncelleyebilirler. Bunun yanı sıra, TD öğrenmesinin Monte Carlo yöntemlerinden miras aldığı özellik ise, çevre modelinden bağımsız olarak doğrudan yalın tecrübelerden öğrenebilmeleridir.

Tecrübe ve tahmin kavramlarının tanımı, TD öğrenmesi ve MC yöntemleri için birbirine benzerdir. Yalnız bu iki yöntemin birbirinden ayrıştığı nokta temelde çevreden alınan geribeslemenin değerlendirildiği zamandır. Bir MC metodunda, *t* anında ajanın içinde bulunduğu *s* konumunun değerini hesaplayabilmesi için R_t beklenen kazancının değerini de bilmesi gerekir. Dolayısıyla da ajanın bitiş konumuna ulaşması ve oyunun içinde bulunulan bölümünün sonlanması gerekir. MC'deki değer fonksiyonunun formal olarak güncellemesi;

$$V(s_t) \leftarrow V(s_t) + \alpha [R_t - V(s_t)] \quad (2.2)$$

şeklinde gösterilir. Buradaki α simgesi adım-genişliği (step-size) katsayısıdır. Öte yandan, TD yöntemlerinin $V(s_t)$ değerine karar verebilmesi için sadece bir sonraki adımı beklemeleri yeterlidir. En basit TD yöntemi olan TD(0) yönteminin formal gösterimi;

$$V(s_t) \leftarrow V(s_t) + \alpha [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (2.3)$$

şeklindedir. TD( ) gösteriminde parantezin içindeki sayı uygunluk izlerinde bahsedileceği üzere ağırlık faktörünü belirlerken kullanılır.

Bilindiği üzere kaçma-kovalama problemlerinin de aralarına dâhil olduğu kimi problemlerde oyun çok uzun sürebilmektedir. MC yöntemleri uyguladıkları politika gereği değer fonksiyonunu güncellemek için oyunun sonlanmasını bekleyeceğinden, böyle bir problemde MC yöntemlerinin kullanılması anlamsızdır. TD yöntemleri ise çevrimiçi olarak her adımda değer fonksiyonlarını güncelledikleri için uzun bölümlerden yılmazlar. Dolayısıyla bu tür problemlerde TD öğrenmesi daha avantajlıdır denilebilir.

Q-öğrenmesi, uygulanan politikadan bağımsız olarak (off-policy) öğrenme yetisine sahip bir geçici farklar kontrolüdür [25, 26]. Yani ajanın hakkında öğrendiği politika, aksiyon seçiminde uyguladığıyla aynı olmak zorunda değildir. Watkins, bu değerli yöntemi 1989 tarihinde yayınladığı doktora tezinde sunmuştur [25]. Q-öğrenmesinin matematiksel gösterimi şu şekildedir:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha [r_{t+1} + \gamma Q\max_{\hat{a}}(s_{t+1},\hat{a}) - Q(s_t,a_t)]. \quad (2.4)$$

Buradaki $Q\max(s_{t+1},a)$ fonksiyonu, ajanın bir sonraki adım olan t+1 anında seçebileceği açgözlü hamlenin değeridir. Dolayısıyla, ajan *t* anındaki aksiyon-değer denklemini hesaplarken bir sonraki adımda yapacağı hamlenin Q değerini değil de, yapmış olabileceği açgözlü hamlenin Q değerini dikkate alır. Bu durumda, öğrenilen aksiyon-değer fonksiyonu, Q, doğrudan optimal aksiyon-değer fonksiyonu, Q^*'a yaklaşır. Uygulanan politikanın öğrenme fonksiyonu üzerinde doğrudan bir etkisi

yoksa da, hala yapılacak hamleyi o belirler. Q-öğrenmesi algoritması şekil 2.3'te gösterilmektedir.

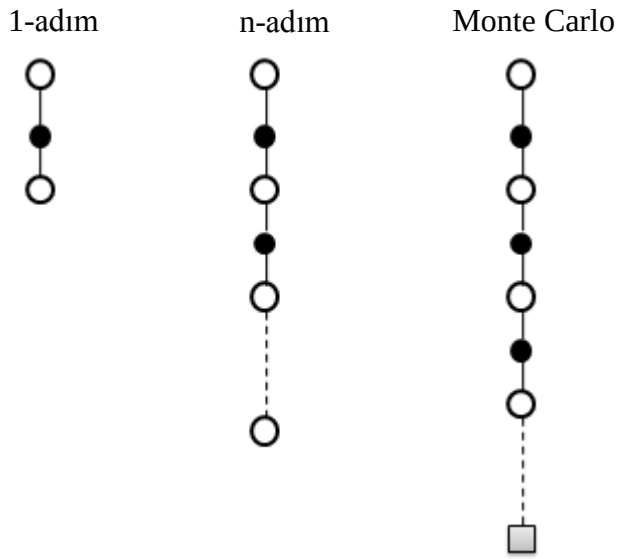
Q(s,a) başlangıç değerlerini belirle
Tekrarla (Her yeni bölüm için):
    s ← başlangıç durumu
    Tekrarla (bölümdeki her adım için):
        π politikasını kullanarak *s* durumdan *a* hamlesini seç
        Hamle a'yı uygula, ödülü ve s''yi gözlemle
        $Q(s,a) \leftarrow Q(s,a) + \alpha\,[r_{t+1} + \gamma.Qmax_{\hat{a}}(s',\hat{a}) - Q(s,a)]$
        s ← s'
    s bitiş konumuna erişene kadar devam et

Şekil 2.3 Q-Öğrenmesi algoritması

2.3.3 Uygunluk İzleri (Eligibility Traces)

Geçici farklar yöntemi, ajan için bir *s* durumunda olmanın matematiksel değerini tahmin eder. Durum-değer fonksiyonu ise, TD(0) yöntemi özelinde, bu değeri hesaplarken bir sonraki adıma dair gözlemlerini kullanır. Peki ya durum değeri hesaplanırken, sadece bir sonraki gözlemden faydalanmak yeterli olmuyorsa ne yapılabilir? Bu noktada ilk akla gelen Monte Carlo yöntemlerini kullanmak olur. Yalnız MC yöntemleri kullanıldığında da, sadece tek bir durum değeri oluşturmak için oyunun sonlanmasını beklemek gerekecektir. Bu problemi ortadan kaldırmak amacıyla uygunluk izleri (eligibility traces) mekanizması kullanıma sunulmuştur. TD metotları uygunluk izleri vasıtasıyla genişletildiğinde, yönteme daha genel ve daha tepeden bir yaklaşım sağlanır. Diğer bir ifadeyle, TD ve Monte Carlo yöntemleri birleştirilerek ikisi arasında bir orta yol oluşturulur. Bu sonuçla ortaya çıkan TD yöntemi, kaç adım sonrasına kadar destek (backup) alacağını kendisi belirler. TD(0), TD(λ) ve Monte Carlo'nun destek diyagramlarının karşılaştırması şekil 2.4'te gösterilmektedir [23].

Uygunluk izine dair tanımlamayı kaçma-kovalama oyunu üzerinden yapmak gerekirse; bir uygunluk izi, oyundaki akıllı ajanın yaptığı hamle veya ziyaret ettiği durum gibi, bir *olayın* gerçekleşmesinin geçici kaydıdır. Bu *iz (trace),* olayla ilişkili hafıza katsayılarını, öğrenme değişikliklerini yapmak için *uygundur (eligible)* şeklinde işaretler [23]. Böylelikle bir TD hatası gerçekleştiğinde, sadece seçilmiş durum ve aksiyonlar bundan sorumlu tutulur. Hata fonksiyonu, bir zaman adımında tahmin edilen ödül ile gerçekte hak edilen ödül arasındaki farklı hesaplamaktadır.



Şekil 2.4 TD(0), n-adımlı TD ve Monte Carlo yöntemlerinin destek diyagramları

Uygunluk izlerinin etkisi, herhangi bir oyunun herhangi bir bölümündeki V^{π} durum değerinin tahmin edilmesi üzerinden tartışılabilir. Monte Carlo yönteminde, her bir *s* durumu için, içinde bulunulan o durumdan oyunun sonuna kadar alınan bütün ödüller gözetilerek değer tahmini yapılır. Öbür taraftan, TD(0) yöntemindeki tahminde sadece bir sonraki adımda alınan ödül doğrudan dikkate alınıp gelecekteki potansiyel ödülleri temsilen bir sonraki adımın durum değeri kullanılır. Dolayısıyla, bu iki yöntemin ortasında bulunan üçüncü bir yöntemin hesaba katacağı ödül sayısı, bir ile maksimum adım sayısı arasında olacaktır. Örneğin; bir n-adımlık destekte *t*. adımdaki beklenen ödül hesaplanırken ilk n adet ödülün karşılığı doğrudan, sonradan alınması beklenen ödüllerin değeri ise t+n. adımın durum değeri üzerinden denkleme katılır. Bunun matematiksel gösterimi;

$$R_t^{(n)} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots\ldots + \gamma^{n-1} r_{t+n} + \gamma^n V_t(s_{t+n}) \quad (2.5)$$

şeklindedir.

N-adımlı yöntemlerde, önceki bir tahminin değeri daha sonraki bir tahminle arasında çıkan fark gözetilerek değiştirilebildiği için TD yönteminin özelliğinin korunduğu söylenebilir. Son olarak, n-adımlı TD tahminindeki yapılan artırımla TD(0) tahmininde yapılan artırım çevrimiçi güncellemenin iki farklı örneğini gösterir. Çevrimiçi güncellemede, bölüm henüz tamamlanmamışken fonksiyon gelecekteki hesaplamalardan geri dönüş beklemeyi kestiği anda artırımlar işleme konulabilir. Bu an, n-adımlı bir TD tahmini için $R_t^{(n)}$'in hesaplandığı andır.

TD tahmini yapılırken, sadece bir n-adımlı tahmin kullanılır gibi bir kısıtlama yoktur. Bu fikirden yola çıkılarak ortaya konan TD(λ) tahmini yönteminde, *ileri doğru* birden fazla n-adımlı tahmin hesaplanıp bunların ortalaması alınmaktadır. Bu noktada zayıflama-oranı (decay-rate) katsayısı, λ, farklı n-adımlı tahminlerin ağırlıklarını belirmekte kullanılır ve [0,1] kapalı aralığındadır. N-adımlı destekler λ^{1-n}'le orantılı olarak ağırlıklandırılır. Her birine çarpan olarak eklenen 1-λ normalizasyon faktörü ağırlıkların toplamda 1'e eşitlenmesini sağlar. Sonuç olarak bu tahminlerin toplamı (λ-toplamı) şu şekilde gösterilir:

$$R_t^{(\lambda)} = (1-\lambda) \cdot \sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)} \quad (2.6)$$

Yukarıda bahsi geçen ileriye yönelik bakış, sadece uygunluk izleri fikriyle neyin amaçlandığını ve bunu kullanan yöntemlerin neyi hesapladığı açıklamaktadır. Aslında burada anlatılan bakış açısında, çok daha sonraki adımlara dair bilgiler gerektiği için yöntem uygulanabilir değildir.

Pratik olarak uygulanabilir olan geriye doğru bakış yöntemi ise, ileri doğru bakış yöntemiyle eşdenik olup aynı sonuca ters yoldan ulaşılabilmektedir. Bu metotta problem içerisindeki her duruma karşılık gelen bir hafıza katsayısı, yani bir *uygunluk izi* mevcuttur ve bir *t* zamanı için $e_t(s)$ ile gösterilir. Problemin her adımında ziyaret edilen durumun uygunluk izi 1 arttırılırken, diğer bütün durumlar γλ ile zayıflatılmaktadır: (γ: ıskonto faktörü, λ: iz-kaybolma katsayısı)

$$e_s(s) = \begin{cases} \gamma\lambda e_{t-1}(s), & eğer\ s \neq s_t; \\ \gamma\lambda e_{t-1}(s) + 1, & eğer\ s = s_t; \end{cases} \quad (2.7)$$

Bahsedilen uygunluk izi denklemi sayesinde, problem üzerindeki bir durum ziyaret edilmediğinde değeri kademeli olarak düşmektedir. Buradaki uygunluk izi, bir takviyenin gerçekleşmesi için her bir durumun öğrenme değişikliğine olan uygunluk derecesini belirler. Bu noktada takviye ile kastedilen şey, geri doğru sinyal gönderildiği hesaba katılarak, bir adımlık TD sapmasıdır. Bir önceki adımda gerçekleşmiş durum-değer tahmininin hatası:

$$\delta_t = r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t) \quad (2.8)$$

denklemiyle gösterilir. Artırımlar her adımda uygulandığı takdirde yapılan düzenleme çevrimiçi gerçekleştirilmiş olur.

Son kısımda bahsedilen geriye yönelik bakış yöntemi zamanda da geriye doğru etki yapar. Ajan geçmişte yaptığı bir tahminin TD hatasını mevcut bilgilerini kullanarak hesaplar ve bu bilgiyi zamanda geriye taşıyarak hatanın bağlı olduğu durumu uyarır. Daha önceden, TD(0)'ın sadece 1-adımlık gözlemde bulunduğunu belirtmiştik. Pratik olarak incelersek, TD(λ) gösteriminde λ'nın yerine 0 konduğunda o an ziyaret edilen durum dışındaki bütün durumların izi 0'la çarpılacaktır. Yani ajanın hafızasında geçmişe dair bütün izler yok olacaktır. Öte yandan büyük λ değerleri kullanıldığında da, ajan uzak geçmişinden, o anlarda içinde bulunduğu durumlara fazla hata sorumluluğu yükleyemeyecek dahi olsa, bir türlü kopamayacaktır.

Bu bölümde, ilgili çalışmalar ana başlığı altında tezde kullanılan kavramlar ve yöntemlerle ilgili bilgilere yer verilmiştir. Sonraki bölümlerde, yapılan deneyler anlatılırken bu yöntemler uygulanış şekliyle beraber açıklanacaktır.

BÖLÜM 3

3. YÖNTEM

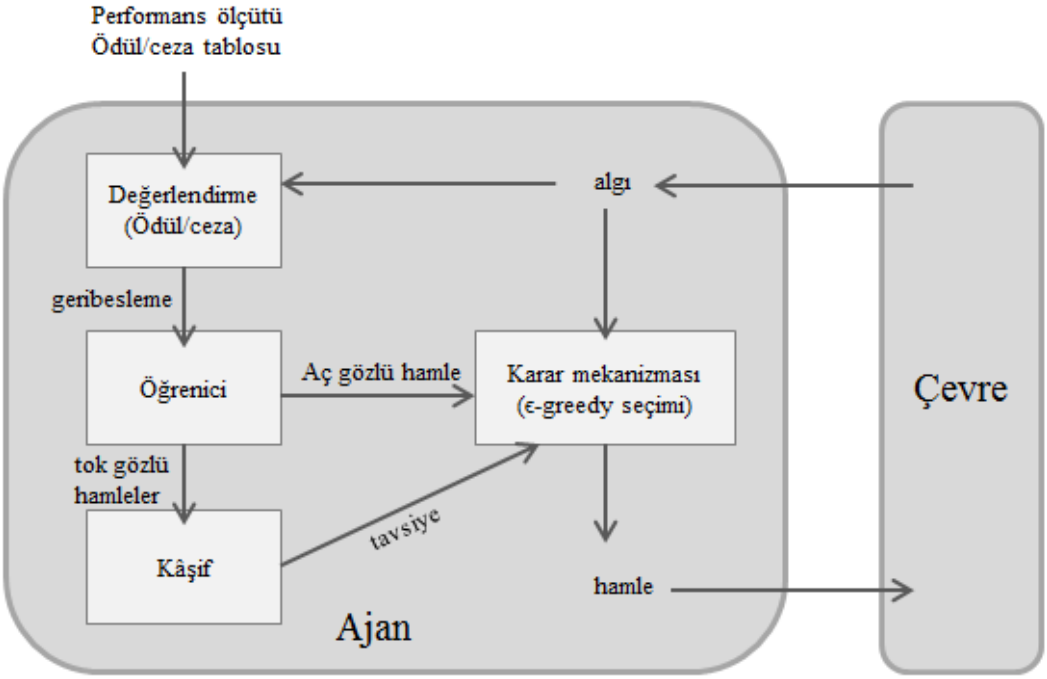
Bu tez çalışmasında, robotların Q(λ)-öğrenmesi yöntemini kullanarak takviyeli öğrenme yaptığı çok ajanlı bir kaçma-kovalama problemi sunulmaktadır. Problemin aktörleri birbirlerine zıt amaçlar taşıyan akıllı ajanlardır. Oyunun başlangıcında homojen özelliklere sahip olan bu ajanlara kaçan robot (*av*) ve kovalayan robot (*avcı*) adları verilmiştir. Oyuncular problemin başında, hakkında bilgi sahibi olmadıkları bir haritanın iki farklı ucuna bırakılırlar. Kovalayan robotun oyundaki amacı mümkün olan en az sayıda hamle ile kaçan robotu yakalamaktır. Kaçan robotun hedefi ise hiç yakalanmamaktır. Bir oyuncunun kazancının diğer oyuncuların kaybıyla dengede olduğu bu tür oyunlara sıfır-toplamlı oyunlar (zero-sum game) denir. Kaçma-kovalama problemleri doğal olarak bu oyun türünün özelliklerini göstermektedir. Uygulanan yaklaşıma göre farklılık gösterebilen bitiş koşulu ise bizim çalışmamızda "*iki ajanın aynı birim kare içerisinde bulunması*", bir başka deyişle avcının avı yakalamasıdır. Her yakalanma veya kaçışın ardından oyunun yeni bir bölümü tekrar başlar. Kaçma-kovalama problemleri bu yönü itibariyle de tekrarlanan oyunlar (repeated game) sınıfına dâhildirler. Bu çalışmada, oyunun tekrarlayan bir bölümü içinde ajanın kendi performansını arttırabilmek için ne yapabileceği sorgulanmaktadır. Bunun için kullanılan yöntem takviyeli öğrenmedir. Ajanlar, bir bölüm içerisinde yaptıkları hamlelere göre ödül veya cezalar alıp bu bilgileri durum-aksiyon matrisi kanalıyla belleklerine işlerler.

Bu çalışmada üzerinde durulan diğer bir önemli nokta ise kaçma-kovalama probleminde kullanılan oyuncu sayısıdır. Basit bir kaçma-kovalama oyununda bir avcı bir de av bulunur. Yalnız, konuyla ilgili gerçek dünya senaryoları dikkate alındığında sıklıkla birden fazla robota ihtiyaç duyulabilmektedir. Çok ajanlı sistemlerin anlatıldığı bölümde bahsi geçtiği üzere bir problemde birden fazla ajan kullanmak için farklı motivasyonlar olabilir. Tezdeki yaklaşım söz konusu olduğunda bu motivasyonlardan bölgesel dağılım ve dayanıklılık (robustness) etkili olmuştur. Baz alınan problemde, bir avcının büyük bir haritayı yalnız başına keşfedip

av araması güçtür. Bu noktada, kapsanan alanın ölçeğine göre birden fazla ajan kullanılarak bölgesel dağılımın getirilerinden faydalanılabilir. Ayrıca robotik sistemlerde, bir robotun arızalanması veya bataryasının tükenmesi gibi sebeplere dayalı olarak bir görev başarısız olabilmektedir. Kaçma-kovalama problemleri paralelinde bir arama kurtarma senaryosunu dikkate alırsak; bir bölgede mahsur kalan birinin tek bir robot ile bulunmaya çalışılması durumunda robotun herhangi bir sebeple görevini tamamlayamaması kabul edilebilir bir hata değildir. Güvenlik ve arama kurtarma gibi kritik öneme sahip konularda bir görev şansa bırakılamaz. Dolayısıyla böyle bir senaryoda birden fazla ajanın kullanılması sistemin dayanıklılığını sağlar. Öte yandan, çok robotlu sistemlerin kontrol güçlükleri, durum uzayının büyümesi ve maliyet gibi bilinen zaaf ve zorlukları da bulunabilir. Sonuçta birden fazla robot kullanmanın kazandıracakları ve kaybettirecekleri kıyaslanarak akla yatkın bir karar verilmesi gerekir.

Yaptığımız çalışmadaki kaçma-kovalama problemi 1 avcı - 1 av ve 2 avcı - 1 av olmak üzere iki farklı biçimde oynanmıştır. Kullanılan bütün ajanlar donanım seviyesinde homojendir. 2 avcı – 1 av probleminde avcı takımında birden fazla ajan bulunması sebebiyle bu ajanların arasında doğan etkileşim de bir inceleme konusudur ve yönetilmesi gerekir. Ajanlar arasındaki etkileşimin yönetilmesi için çok robotlu kooperatif sistemler düzeyinde iki farklı yaklaşım kabul görmektedir. Bunlar takım öğrenmesi ve eşzamanlı öğrenmedir. Takım öğrenmesinde ajanlar bir ekip olarak tek bir merkezden yönetilirler ve öğrenmede kullanılan akıl bu merkezde bulunur. Bu fikre göre ajanların yönetilmesi kolay olsa da takımdaki ajanların tek başlarına bir söz hakkı yoktur. Sadece çevreden aldıkları geri beslemeleri merkezlerine bildirip hamle yapmak için yine merkezden komut beklerler. Bizim bu tez çalışması için edindiğimiz yaklaşımda eşzamanlı öğrenme kullanılmaktadır. Eşzamanlı öğrenmede her ajan tek başına bir bireydir ve kendi hamlelerinden sorumludur. Öğrenmeyi gerçekleştirebilmek için her ajan diğerlerinden yalıtılmış olarak kendi aklını kullanır. Çalışmamızda eşzamanlı öğrenme yöntemini tercih etmemizin sebebi çok robotlu sistemleri kullanmaktaki motivasyonlarımızla örtüşmesidir. Takviyeli öğrenme konusunun üzerinde durmak istememiz sebebiyle ajanların öğrenme alışkanlıklarının yanına bir takım arkadaşı katıldığında nasıl

değişeceğini görmek önem arz eder. Takım öğrenmesi kullanıldığında bunun gözlemlenmesi mümkün olmayacaktır. Ayrıca bu durum robotlardaki öğrenme üzerine yapılan bir araştırmada vurgulanmak istenen noktayı karşılamayacaktır. Kaçma-kovalama problemine yaklaşmak istediğimiz tarafta bir akıllı ajan takımındaki ajanın öğrendiklerini bir merkeze gönderip oradan komut beklemesi göstermeyi amaçladığımız sonucu ortaya çıkarmaz. Bu anlamda öğrenme yetisine sahip, bağımsız ajanlara ihtiyaç duyulduğu için eşzamanlı öğrenme yöntemi dikkate alınmıştır.



Şekil 3.1 Oluşturulan kaçma-kovalama problemindeki akıllı ajan

Çalışmamızdaki kaçma-kovalama problemine yaklaşım yönteminden bahsedilirken üzerinde durulması gereken en önemli konu kullanılan öğrenme algoritmasıdır. Tanımlar kısmında bahsedildiği üzere (bölüm 2.3) takviyeli öğrenme; bir ajanın -hedeflerine giden yolda- çevresiyle gerçekleştirdiği etkileşim kanalından aldığı ödül ve cezaları daha sonraki adımlarında yol göstermesi amacıyla kullanmasıdır. Q-öğrenmesi en yaygın kullanıma sahip takviyeli öğrenme algoritmalarından biridir. Bir geçici farklar (TD) öğrenmesi yöntemi olan Q-öğrenmesi, kullanılan politikadan bağımsızdır [27]. Yani ajan aksiyon seçimini ϵ-greedy veya softmax gibi yöntemler

kullanarak yapsa dahi arka planda kendisine maksimum faydayı getireceğine inandığı açgözlü yöntemi kullanarak öğrenmektedir. Ajan bu metodu kullanarak hem keşfe yönelik yapacağı hamlelere şans tanır, hem de keşfe yönelik olan hamleden, aslında ona açgözlü aksiyonu kullanmasını öneren durumun sorumlu tutulmasını engeller. Bu tez çalışmasında kullanılan Q-öğrenmesi yöntemiyle öğrenen ajanlara dair hazırlanan aksiyon diyagramı şekil 3.1'de gösterilmektedir.

Uygunluk izleri (eligibility traces), TD yöntemlerinin kapsamını genişleterek Monte Carlo yöntemleriyle birleştirilmiş bir yaklaşım sunar. Bu sayede problemlere daha genel ve tepeden bir bakış şansı yakalanmıştır. Bilindiği gibi standart TD yöntemlerinde ajanın yaptığı her hamle bir durum sonrası gözetilerek değerlendirilir. Bu yöntem kullanıldığında ise ajan yaptığı hamleler neticesinde geçtiği yollara ileriki zamandan bildirimde bulunarak geçmişte yapılmış olan tahminlerin TD hatası değerlerini döndürür. Uygunluk izleri mekanizması kullanılırken problemin durum uzayındaki her durum için hafızada bir uygunluk izi parametresi tutulması gerekir. Geçmişe yapılan bildirimlerde zincir büyüdükçe daha eski hamlelerin ödül veya cezalardan sorumlu tutulmaları da mantıksız olacaktır; çünkü o ödül veya ceza üzerindeki izleri kaybolmaya başlamıştır. Bu noktada, geçmişe doğru giderken hamlelerin geri bildirimlerden alacağı sorumluluğu belirlemek üzere iz-kaybolma parametresi, λ, devreye girer. $0 \leq \lambda \leq 1$ olmak üzere her n-adımlık desteğin ağırlığı λ^{n-1} ile orantılıdır.

Q-öğrenmesi yöntemini uygunluk izleriyle birleştiren iki farklı algoritma mevcuttur. Bunlar Watkins'in ve Peng'in Q(λ)-öğrenme algoritmalarıdır. Bu tez çalışmasında Watkins'in Q(λ)-öğrenmesi yaklaşımı kullanılmıştır. Standart Q-öğrenmesi metodunda, ajan tarafından optimal olmayan keşif hamleleri yapılsa dahi açgözlü politika öğrenilir. Uygunluk izleri yaklaşımında zamanda geriye doğru hata bildirimi yapıldığı göz önünde bulundurularak, Q-öğrenmesinde uygulanması muhtemel bir keşif hamlesinin yapılan değer tahminini etkilemediği için TD hatasından sorumlu tutulamayacağı ortadadır. Bu nedenle, Watkins'in Q(λ)-öğrenmesi metodunda TD hatası bildirimi sadece başlangıçtan itibaren açgözlü hamleler takip edildiği sürece yapılabilir. İlk defa keşfe yönelik bir hamle denendiğinde o ana kadar tutulan izlere göre hata bildirimi yapılır. Bu noktada durum uzayındaki durumların mevcut

uygunluk izleri silinir ve süreç yeniden başlar. Bu durum haricinde Watkins'in Q(λ) yöntemi TD(λ) ve Sarsa(λ) yöntemlerine benzer. Yalnız, Q-öğrenmesinde bir aksiyon-değer tahmini yapılırken hamleden sonra gidilen durumun optimal hamlesinin değeri de dikkate alındığından, yani henüz ziyaret edilmeyen bir durumun bilgisine başvurulduğundan, Watkins'in Q(λ) yönteminin de ilk keşif hamlesinden bir sonraki duruma kadar baktığı söylenebilir. Bu bağlamda, eğer yapılan ilk keşif hamlesi a_{t+n} ise, en uzun destek de şöyle olacaktır:

$$r_{t+1} + \gamma r_{t+2} + \ldots\ldots + \gamma^{n-1} r_{t+n} + \gamma^{n} Q_t \max_{\hat{a}}(s_{t+n}, \hat{a}). \quad (3.1)$$

Ayrıca durum-aksiyon fonksiyonunun güncellenmesi de aşağıdaki gibidir:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha \delta_t e_t(s, a). \quad (3.2)$$

Burada kullanılan δ_t simgesi, t anında karşılaşılan durum-aksiyon hatasını verir ve gösterimi

$$\delta_t = r_{t+1} + \gamma Q_t \max_{\hat{a}}(s_{t+1}, \hat{a}) - Q_t(s_t, a_t) \quad (3.3)$$

şeklindedir. Watkins'in Q(λ)-öğrenmesi yönteminin sözde kodu şekil 3.2'de gösterilmektedir. Bu kodun içinde bulunan $\text{argmax}_b Q(s', b)$ ifadesi, s''den yapılabilen hamleler içinde en yüksek değere sahip olanı vermektedir.

Son olarak, çalışmamızda karşılaştırmalı bir yaklaşım sunmak istememiz sebebiyle kaçan ve kovalayan ajanlar için farklı davranış biçimleri kullanılmıştır. Kovalayan ajan tarafında kullanılan algoritmaların sabit bir hedefi veya rastgele yürüyüş yapan bir avı yakalamaktaki başarısı, diğer algoritmaların yönteme ne kattığını göstermek açısından önemlidir. Bu sayede oyunun hangi zaman aralığında yakınsadığını tespit etmek ve gelişmiş yöntemleri bununla kıyaslamak imkânı doğar. Bunun yanında her iki takımdaki ajanların da Q öğrenmesi veya Q(λ) öğrenmesi kullandığı çalışmalar eşit zekâ koşullarında kaçan ve kovalayan takımların nasıl bir performans ortaya koyduğunu gösterir. Monte Carlo yöntemleri ise tahmin değerlerini belirlerken oyunun her bölümünde son aşamaya kadar ilerledikleri için uygulanan yaklaşımın karşılaştırmasında önemli bir referans noktası sunar.

Q(s,a) başlangıç değerlerini belirle ve bütün s, a çiftleri için e(s,a) = 0

Tekrarla (Her yeni bölüm için):

s ve a değerlerini belirle

Tekrarla (bölümdeki her adım için):

a hamlesini yap, r ve s''yi gözle

π politikasını kullanarak s' durumdan *a*' hamlesini seç

$a^* \leftarrow \text{argmax}_b Q(s',b)$

$\delta = r + \gamma Q(s',a^*) - Q(s,a)$

$e(s,a) \leftarrow e(s,a) + 1$

Bütün s, a çiftleri için:

$Q(s,a) \leftarrow Q(s,a) + \alpha\delta e(s,a)$

eğer $a' = a^*$ ise, $e(s,a) \leftarrow \gamma\lambda e(s,a)$

değilse, $e(s,a) \leftarrow 0$

$s \leftarrow s'$;

$a \leftarrow a'$;

s bitiş konumuna erişene kadar devam et

Şekil 3.2 Watkins'in Q(λ) algoritmasının sözde kodu

BÖLÜM 4

4. DENEYLER

Bu tez çalışması kapsamında, araştırmalarımızı üzerinde yoğunlaştırdığımız kaçma-kovalama problemleri alanında çeşitli deneyler hazırlanmış ve bunlar Ubuntu işletim sistemi üzerinde çalışan Player/Stage simülasyon ortamında gerçekleştirilmiştir. (Player, birçok farklı robot donanımını destekleyen bir arayüzdür. Stage ise üzerinde birden çok robotun çalıştırılabileceği, gerçeğe yakın bir robot simülatörüdür.) Oluşturulan kaçma-kovalama problemi üzerinde çok robotlu Q(λ) öğrenmesi araştırmaları yapılmıştır. Bu kapsamda farklı ajanlar için farklı davranış biçimleri dikkate alınarak öğrenme yöntemlerinin karşılaştırmaları sunulmuştur. Problem oyuncu sayısı, kovalayan ajanın davranış biçimi ve kaçan ajanın davranış biçimi olarak üç ana başlıkta incelenebilir. Araştırmada kullanılan bütün deney tipleri tablo 4.1'de gösterilmektedir.

Deneylere oyuncu sayısı açısından bakıldığında iki farklı yaklaşım görülmektedir. Bunlar; 1 avcı – 1 av ve 2 avcı – 1 av olarak belirtilebilir. Anlaşılacağı üzere araştırmalarımızda üzerinde durmak istediğimiz konulardan biri çok robotlu sistemlerdir. Deney sonuçlarının, bu sistemlerin getirilerinin ve götürülerinin ortaya çıkarılmasında yardımcı olması beklenir. Ayrıca çok robotlu sistemlerde coğrafi dağılımın başarıya olan etkisi incelenmektedir.

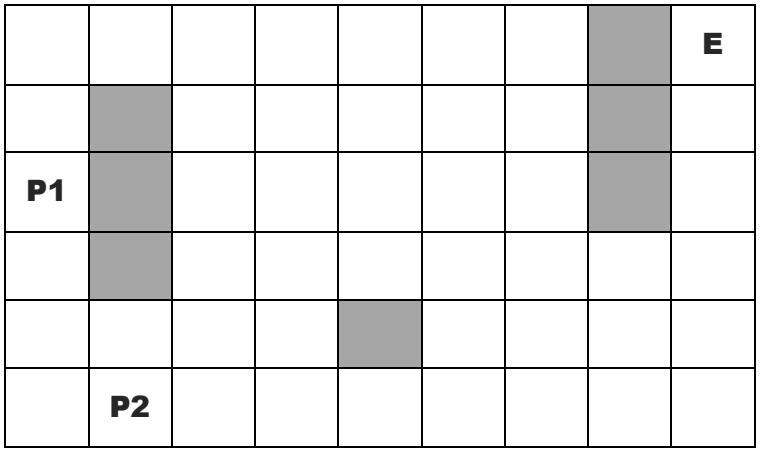
Kovalayan ajan davranışı dikkate alındığında bunlar iki kümede toplanmaktadır. Bu çalışmaların temelini oluşturan Q ve Q(λ) yöntemleri bu noktada üzerinde durulmak istenen asıl konuyu göstermektedirler. Kullanılan Q-öğrenmesi algoritması şekil 2.3'te, Watkins'in Q(λ)-öğrenmesi algoritması ise şekil 3.2'de gösterilmektedir. Kaçan ajan davranışı söz konusu olduğunda bu yöntemlerin yanında sabit kalma ve rastgele yürüme (random walk) kullanılmıştır. Kaçan ajanın sabit kalmasının, avcı için onu kolay bir av haline getirdiği ortadadır. Bu yöntemin kullanılmasındaki amaç basit koşullar altında avcının kaç bölüm içerisinde optimal değere yakınsayacağını görmektir. Rastgele yürüme metodu ise beklentinin aksine avcının işlerini hayli zorlaştırmaktadır. Avın yaptığı bu hareket, bilinçli olarak avcıdan herhangi bir kaçma

eylemi sunmaz; fakat kovalayan ajan açısından bilinmezliği arttırarak işlerini tahmin edilenden daha güç bir seviyeye getirebilir.

No	Oyuncu Sayısı	Kovalayan Davranışı	Kaçan Davranışı	Öğrenme Oranı (α)	İskonto Faktörü (γ)	İz-Zayıflama Oranı (λ)
1	1 P – 1 E	Q	Sabit	0.1	0.95	-
2	1 P – 1 E	Q(λ)	Sabit	0.9	0.95 – 0.99	0.9 – 1
3	1 P – 1 E	Q(λ)	RandomWalk	0.05	0.9	0.1 – 1
4	1 P – 1 E	Q(λ)	Q(λ)	0.05, 0.9	0.9	0.1 – 1
5	2 P – 1 E	Q(λ)	Sabit	0.9	0.99	0.9
6	2 P – 1 E	Q(λ)	RandomWalk	0.05	0.9	0.1 – 1
7	2 P – 1 E	Q(λ)	Q(λ)	0.05, 0.9	0.9	0.1 – 1

Tablo 4.1 Kaçma-kovalama problemine uygulanan deneyler. (P=avcı, E=av)

Kaçma-kovalama problemi için kullanılan deney düzeneği de oyunun işleyişi anlamında değerli bilgiler vermektedir. Bilindiği gibi haritadaki durum uzayı sürekli değil ayrık, ızgara (grid) yapısındadır. Tez çalışması boyunca, çok sayıda farklı harita ve düzenek kullanılmıştır. Yalnız, deneylerin anlatıldığı bu kısımda yöntemlerin karşılaştırmaların mantıksal bir zemine oturması için sabit bir deney düzeneğinden bahsedilmektedir. Deneyde kullanılan harita 6x9 birim boyutlarında olup üzerinde engeller barındırmaktadır. Dolayısıyla oyunun durum uzayında 54 farklı konum bulunmaktadır. Haritanın çevresinde ajanlar için giriş ve çıkış yolları bulunmamaktadır. Bunun yerine ajanlar oyuna doğrudan harita üzerinde başlarlar. Oyunun başlangıcında karşılıklı iki takımın ajanları birbirlerinden uzak noktalarda olacak şekilde konumlandırılmışlardır. Oyunun örnek bir başlangıç hali şekil 4.1'de gösterildiği gibidir. Burada renkli kareler harita üzerinde engelleri, P1, P2 ve E değişkenleri ise sırasıyla birinci avcı, ikinci avcı ve avı göstermektedir.



Şekil 4.1 Örnek bir oyun başlangıç haritası (P1=avcı 1, P2=avcı 2, E=av)

Durum uzayı gibi aksiyon uzayı da ayrık yapıdadır. Oyuncular engellerle veya duvarlarla sınırlanmadıkları durumlarda dört ana yöne doğru hamle yapabilirler. Hamleler eşzamanlı değildir ve bir sıra ile gerçekleştirilmektedirler.

Her Q(λ) deneyinin başlangıcında takviyeli öğrenme yöntemi ile alakalı olarak belirlenmesi gereken 3 farklı parametre bulunmaktadır. Bunlar:

- α : öğrenme oranı (learning rate)
- γ : iskonto faktörü (discount factor)
- λ : zayıflama oranı (decay-rate)

olarak tanımlanmaktadır. Bunlardan α ve γ standart Q-öğrenmesi yönteminde de mevcutken λ'ya sadece uygunluk izleri mekanizması kullanıldığı durumlarda ihtiyaç duyulur. Kullanılan simgelerden α, yani öğrenme oranı, yeni elde edilen bir bilginin eski bilgiyi ne denli değiştireceğini belirler. Bu oran sıfıra yaklaştıkça oyuncunun yeni bilgileri değersizleşmeye başlar. Bir başka deyişle ağır ama daha emin adımlarla ilerlemeyi tercih etmiştir. Bu durumda oyunun hemen yakınsaması beklenmez, bunun yerine uzun vadede tutarlı bir sonuç verir. Buna karşın öğrenme oranı bire yaklaştıkça yeni edinilen bilgi çok daha önemli hale gelir. Hatta bu değer bire eşitlendiği takdirde her edinilen bilgi bir öncekinin izini tamamen silerek yerine geçer. γ simgesi ile gösterilen iskonto oranı, gelecekteki ödülün oyuncunun kararı üzerinde ne kadar etkili olduğunu belirler. Bir başka deyişle gelecekteki ödülün ajan

için mevcut durumdaki karşılığını simgeler. Bu değer sıfıra yaklaştıkça oyuncu daha "fırsatçı" bir davranış biçimini benimser ve yakında alabileceği yüksek değerli bir ödülü kovalar. Aksi durumda ise oyuncu ileriye yönelik bir yatırımı önemser ve sonraki süreçte alacağı değerli ödüllere odaklanır. Bahsi geçen son parametre ise λ ile gösterilen iz zayıflama oranıdır. Bilindiği gibi Watkins'in Q(λ) yönteminde ajan adım adım ilerlerken açgözlü yaklaşımı takip ettiği sürece geçmiş hamlelerin, çizdiği yol üzerindeki sorumluluklarını belirler. λ ise, oyunda ileri gidildikçe ajanın seyahatnamesindeki izlere yüklenen bu sorumluluğun hangi katsayıyla kaybolacağını belirler. Bu değer 0 olduğunda, yani Q(0) durumunda, standart Q-öğrenmesi yaklaşımı izlenmiş olur; çünkü ajan geçmişteki hamlelere hiçbir sorumluluk yüklemez. Öte yandan bu değer 1 olduğunda da Monte Carlo yönteminden esinlenmiş bir Q(λ)-öğrenmesi ortaya çıkar. Yalnız Watkins'in Q(λ) öğrenmesi yönteminde yapılan ilk keşfe yönelik hamlede bu izler silindiği için 1'e çok yakın λ değerlerinin pratikte çok fazla işlevi olmayacaktır.

Çalışmamız süresince akıllı ajanların izlediği politikada aksiyon seçimi yöntemi olarak ϵ-greedy kullanılmıştır. ϵ değeri olarak da 0.1 uygun görülmüştür. Bu politikayı uygulayan bir ajanın %90 ihtimalle açgözlü hamleyi seçmesi beklenmektedir. Öte yandan geriye kalan %10 ihtimal dâhilinde de açgözlü olmayan hamlelerden birini, her birini eşit olasılıkla olmak üzere, seçecektir.

Yapılan deneylerde sürecin izlenmesi ve yakınsamaların tespit edilmesi anlamında bölüm sayısı ciddi bir öneme sahiptir. Kaçan ajanın sabit olduğu deneylerde genelde oyun 25. bölümden sonra yakınsamaya başlar. Bu deneylerde bölüm sayısı olarak genelde 100 seçilmiştir. Öte yandan ilk 100 hamlede herhangi bir tespit yapılamadığı durumlarda bölüm sayısı da arttırılmıştır. İki takımdaki ajanların da öğrendiği senaryolarda 1000 hamleye kadar çıkılmıştır.

Son olarak, kullanılan farklı yöntemlerin eşit koşullarda karşılaştırılabilmeleri amacıyla deneylerdeki rastgeleliğin oyuna etkisinin düşük tutulması istenmiştir. Bu nedenle bütün deneylerin başında sözde rastgele sayı üretecine aynı tohum (seed) verilmiştir. Ayrıca deneylerdeki öğrenme oranları [0.05-0.9], iskonto faktörleri de [0.7-0.95] aralıklarında seçilmiştir. Dolayısıyla ajanlar öğrenme yaklaşımı olarak

hem yavaş hem de hızlı öğrenmeyi denerler. Iskonto faktörü özelinde bakıldığında da ajanlar gelecekteki ödüllere yüksek değer veren ajanlar olarak adlandırılabilir.

4.1 Sabit Kaçağı Bulmak (Deneyler 1, 2 ve 5)

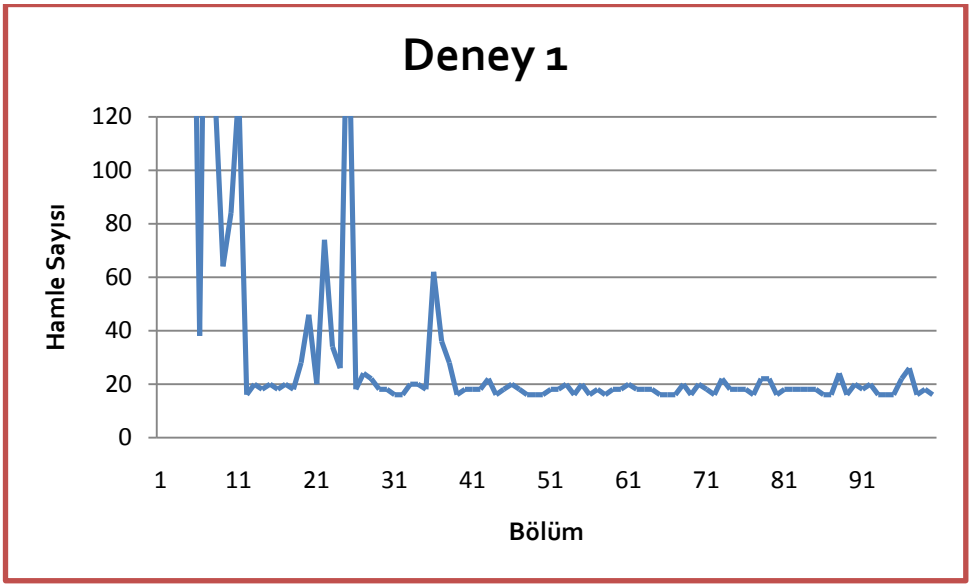
Deney çalışmaları kapsamındaki kaçma-kovalama problemini çözmek adına gerçekleştirilen simülasyonların özet bilgileri tablo 4.1'de gösterilmektedir. Burada görüleceği üzere 1., 2. ve 5. deneyler sabit konumdaki bir kaçağı yakalamak üzerine kuruludur. Bu üç deneyin özellikleri,

No	Oyuncu S.	P Davranış	E Davranış	α	γ	Λ
1	1 P – 1 E	Q	Sabit	0.1	0.95	-
2	1 P – 1 E	Q(λ)	Sabit	0.9	0.95 – 0.99	0.9 – 1
5	2 P – 1 E	Q(λ)	Sabit	0.9	0.99	0.9

Tablo 4.2 Deney 1, 2 ve 5'in özellikleri (P=avcı, E=av)

şeklindedir. İlk deneyde avcı Q-öğrenmesi uygularken, deneyler 2 ve 5'te Watkins'in Q(λ) yöntemi kullanılmıştır. Av ise her üç durumda da sabit konumdadır. Bu deneylere ilişkin sonuçlar grafikler 4.1, 4.2, 4.3 ve 4.4'te gösterilmektedir.

Deney 1'in sonuçlarından anlaşılacağı üzere avcının sabit bir avı yakalamaya çalıştığı durumda henüz oyunun başlarında adım sayısı optimal-yaklaşık bir değere yakınsamaktadır. Avcıyla avın arasında minimum 14 adımlık mesafe bulunan bu örnekte değerler 25. bölümün ardından optimal sonuca yaklaşık olan 16 çevresine yerleşmiştir. Ajan 0.1'lik öğrenme oranıyla her ne kadar yavaş bir öğrenme gösterse de, süreç içerisinde eski bilgilerini yalanlayacak bir durumla karşılaşmadığı için kısa bir sürede aradığı sonucu bulabilmiştir. Diğer bir deyişle, avcının avı sabit olarak kaldığı için harita üzerinde ödül hiçbir zaman yer değiştirmeyecektir. Dolayısıyla öğrendiği bir bilginin ileride onu hayal kırıklığına uğratması söz konusu olamaz.



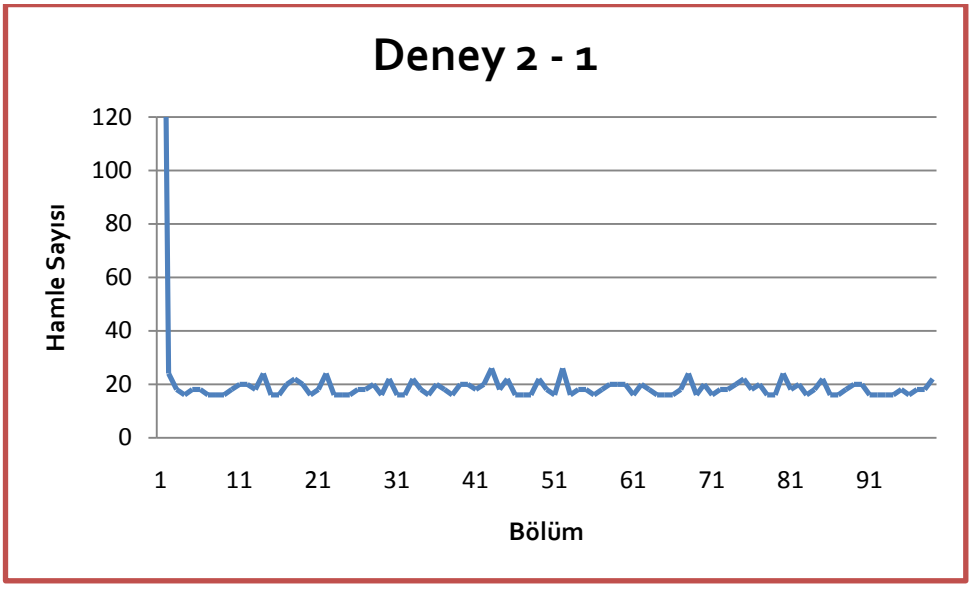
Grafik 4.1 Deney 1 Sonuçları (1 avcı - 1 av, avcı→Q, av→sabit)

Deney 2'yi incelediğimizde bir önceki örnekte Q-öğrenmesi gerçekleştiren ajanın aynı fiziki koşullar altında Watkins'in Q(λ) yöntemini uyguladığını görmekteyiz. Bu deneyin simülasyonunda λ, γ ve α değerleri için çok sayıda deneme yapılıp en başarılı sonucun hangi parametrelerle elde edildiği gözlemlenmiştir. Buna ilişkin tablo (tablo 4.3) aşağıda gösterilmektedir. Tablodaki yakınsama sütunu ajanın kaçıncı aşamada oyunu bitiren hamle sayısını optimal değere yakınsadığını gösterir. Bununla ilgili karar verilirken, ajanın o aşamadan oyunun sonuna kadar hiç optimal değerin 5 adım üstüne çıkmaması koşulu dikkate alınmıştır.

α	γ	λ	Yak. Adım	α	γ	λ	Yak. Adım
0.1	0.95	0.1	186	0.1	0.95	0.95	84
0.6	0.95	0.1	58	0.3	0.95	0.5	64
0.9	0.95	0.1	28	0.3	0.95	0.95	19
0.9	0.95	0.5	17	0.6	0.95	0.5	23
0.9	0.95	0.9	6	0.6	0.95	0.95	7
0.9	0.7	0.9	12	0.9	0.99	0.9	3

Tablo 4.3 Deney 2 yakınsama adımları

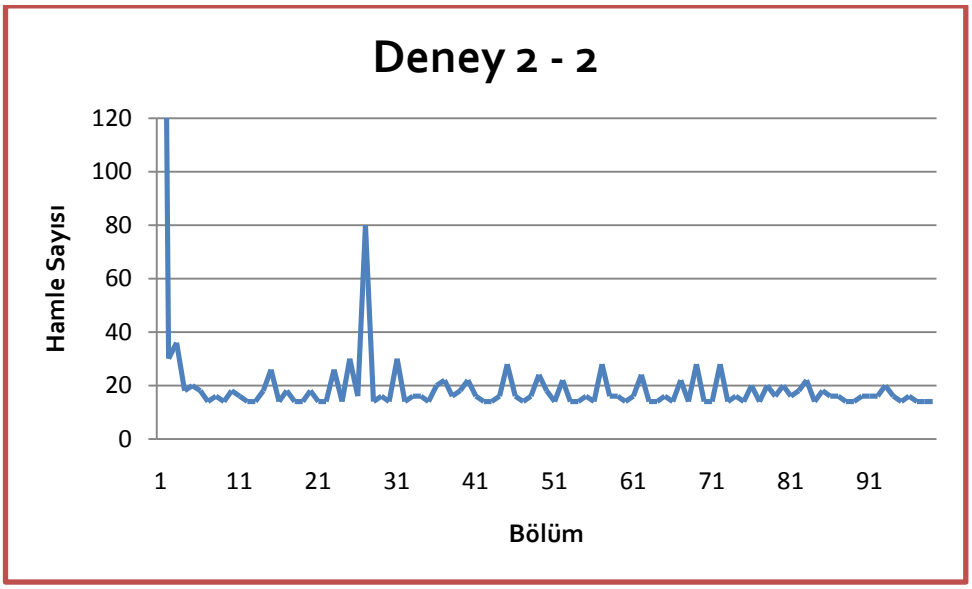
Deney 2'nin sonuçları incelendiğinde parametreler doğru atandığı takdirde Q(λ) yönteminin verilen problem için çok başarılı olduğu söylenebilir. Bu sonuçları yalın Q-öğrenmesinin sonuçlarıyla karşılaştırdığımızda uygunluk izleri mekanizmasının Q-öğrenmesi ile birleştirildiğinde yöntemin gücünü arttırdığı görülmektedir. α=0.9, γ=0.99 ve λ=0.9 değerleri kullanılarak yapılan simülasyonda oyun 3. adımda yakınsama noktasına varmıştır. Bu deneye ilişkin sonuçlar grafik 4.2'de verilmiştir. Yalnız, bu sonuçlara ilişkin tek problem ajanın hiçbir zaman optimal yolu keşfedememiş olmasıdır. Öte yandan α=0.9, γ=0.95 ve λ=1 atamalarıyla yapılan deneyde yakınsama 6. adımdan itibaren görülse de ajan, optimal yol olan 14 adım uzunluğundaki yolu keşfedebilmiştir. Bu durum grafik 4.3'deki sonuçlardan görülebilir.



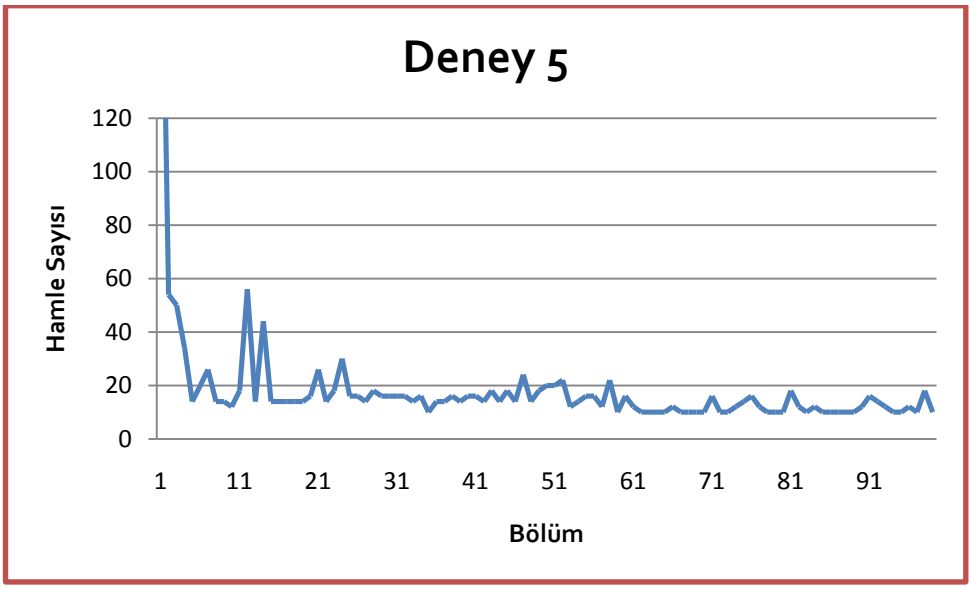
Grafik 4.2 Deney 2-1 Sonuçları (1 avcı - 1 av, avcı→Q(λ), av→sabit)

5. deney ile beraber avcı takımına ikinci bir üye katılmış ve çok robotlu sistemlere geçilmiştir. Takıma dâhil olan yeni ajan ava 10 adımda ulaşabilecek şekilde, yani diğer avcıdan daha yakın olarak konumlandırılmıştır. Bu deneyde iki avcı kullanılması beklenildiği gibi coğrafi dağılımın avantajının gözlenmesini sağlamıştır. Grafik 4.4'te görüldüğü üzere haritanın iki farklı ajan tarafından araştırılması başlangıçtan itibaren yakalama süreçlerini kısaltmaktadır. Ayrıca, iki avcı da ayrı ayrı kendi optimal adım sayısı değerlerine yakınsamaktadır. Takımdaki ajanlar

öğrenme ve politikalarını uygulama işlevlerini yalnız başlarına gerçekleştirdikleri için birbirlerine karşı bir bağımlıkları yoktur. Bu noktada, eğer robotlardan biri devre dışı kalırsa diğer robotun yalnız başına görevi tamamlayabileceği görülmektedir.



Grafik 4.3 Deney 2-2 Sonuçları (1 avcı - 1 av, avcı→Q(λ), av→sabit)



Grafik 4.4 Deney 5 Sonuçları (2 avcı - 1 av, avcılar→Q(λ), av→sabit)

Sabit bir kaçağı hedef alarak yapılan ilk üç deney göstermektedir ki, her ne kadar Q-öğrenmesi başarılı bir yöntem olsa da, uygunluk izleriyle genişletildikleri zaman çok daha verimli bir öğrenme ortaya çıkmaktadır. Uygunluk izleri mekanizması kullanıldığında, ajan oyunun bir bölümünü sonlandırdığı zaman kendi başarısında katkısı bulunan aksiyonları da ödüllendirir. Bir başka deyişle, hedefe ulaştığı yolda geçtiği duraklara λ faktörüne bağlı olarak teşekkürlerini sunar. Yalnız, Watkins'in Q(λ) algoritmasında keşif değeri taşıyan ilk hamlede sorumluluk zinciri bozulur ve o ana kadarki ödül dağıtıldıktan sonra süreç tekrar başlar. Bunun sebebi, politikadan bağımsız bir yöntem olan Q-öğrenmesinde bir keşif hamlesi yapılsa dahi aç gözlü hamlenin öğrenilmesidir. Bu noktada açgözlü hamle, yapılan keşiften dolayı sorumluluğu almaktan kaçınır ve uygunluk izleri silinir. Bu aksamaya rağmen sonuçlar incelendiğinde uygunluk izleri mekanizmasının Q-öğrenmesine olumlu katkı yaptığı söylenebilir.

4.2 Akıllı Avcılar – Rastgele Kaçan Av (Deney 3 ve 6)

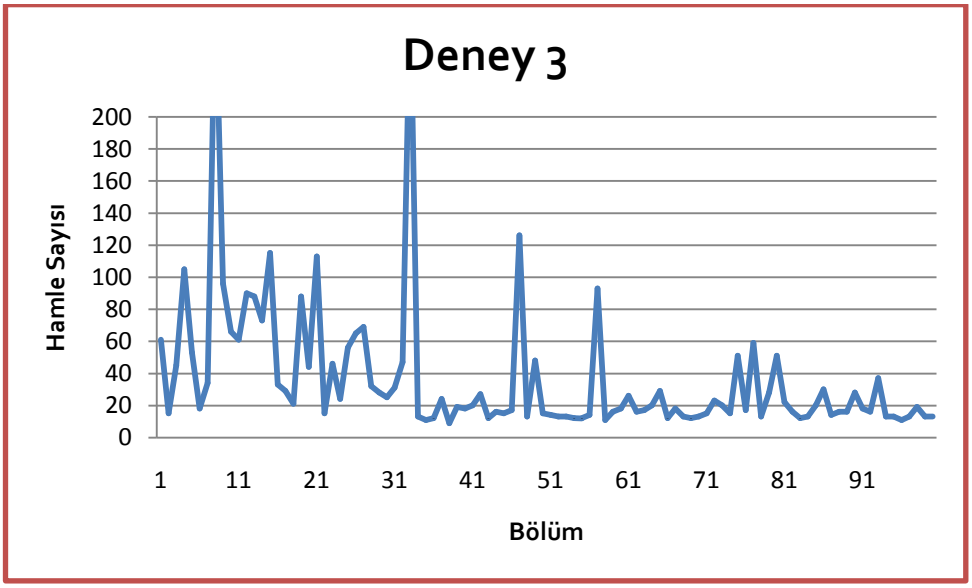
Yapılan 3. ve 6. deneylerde bilinçsiz bir şekilde kaçan av ve onu yakalamaya çalışan akıllı avcılar bulunmaktadır. Bu senaryoda oyuncu sayısının, avcı takımı Watkins'in Q(λ) yöntemini uyguladığında nasıl şekilleneceğini görebilmek adına iki deneyin sonuçları karşılaştırılmaktadır. Deney 3 ve deney 6'nin özellikleri,

No	Oyuncu S.	P Davranış	E Davranış	α	γ	Λ
3	1 P – 1 E	Q(λ)	RandomWalk	0.05	0.9	0.1 – 1
6	2 P – 1 E	Q(λ)	RandomWalk	0.05	0.9	0.1 – 1

Tablo 4.4 Deney 3 ve 6'nın özellikleri (P=avcı, E=av)

şeklindedir. Bu deneylerde avcılar Watkins'in Q(λ) yöntemini kullanırken av ise rastgele hareket etmektedir. Kullanılan zayıflama oranı ise söz konusu örnek için 1'dir. Yani oyunun herhangi bir bölümü için baştan sona açgözlü seçimler uygulandığı takdirde, ajan ilk hamleden itibaren yaptığı bütün hamlelere hata

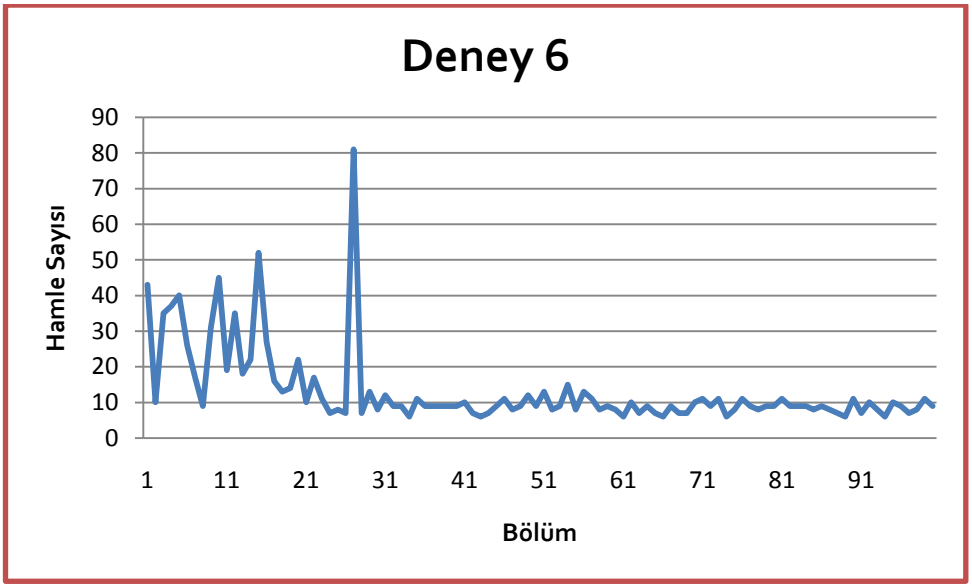
bildirimi yapar ve tahminlerini değiştirir. 1 avcı - 1 av ile gerçekleştirilen deney 3'e ilişkin sonuçlar grafik 4.5'te gösterilmektedir.



Grafik 4.5 Deney 3 Sonuçları (1 avcı – 1 av, avcı→Q(λ), av→rastgele yürüme)

Beklendiği ve görüldüğü üzere, 100 bölüm üzerinden gerçekleştirilen bu deneyde bir değer etrafında yakınsama görülememiştir. Avın rastgele hareket yapması ve durum uzayı üzerinde ödülün sürekli yer değiştirmesi tam bir yakınsama gözlenmemesine sebep olmuştur. Ancak yine de avcı, her bölüme aynı konumdan başlayan avın bulunduğu bölgeyi kestirmekte başarılı olmuştur. Bu doğrultuda, simülasyonun başında zaman zaman bir bölüm için çok fazla hamleye ihtiyaç duyan ajan, kademeli olarak bu problemi aşmıştır. Ayrıca, ava başlangıçta 14 hamle mesafede bulunan avcı, oyunun ortasından itibaren birçok kez 14 hamle dahi gerçekleştirmeden avı yakalayabilmiştir. Bu bölümlerde, henüz oyunun başında av daha bulunduğu konumdan uzaklaşamadan avcı ona doğru yönelerek oyunu kazanmaktadır. Son olarak, Q(λ) yöntemi ve sabit avcı ile gerçekleştirilen oyunlarda, iz kaybolma faktörü 1'e yakınken öğrenme oranının da yüksek bir değer seçilmesinin uygun olduğu görülmüştü. Yalnız bu deneyde av hareketli olduğu ve ödül yer değiştirdiği için bu stratejinin başarısız olduğu tespit edilmiştir. Dolayısıyla öğrenme faktörü 0.05 olarak alınmıştır. Üçüncü deneyde ajanın oyunu sonlandıran hamle sayılarının ortalaması alındığında 36 değerine ulaşılmıştır.

Rastgele yürüyüş yapan bir avın iki akıllı avcı ile yakalanmaya çalışıldığı deney 6'nın sonuçları da deney 3 ile paraleldir. Beklendiği gibi harita üzerindeki coğrafi dağılımın katkısı açıkça görülmektedir. Bu simülasyondan elde edilen sonuçlar grafik 4.6'da sunulmaktadır.



Grafik 4.6 Deney 6 Sonuçları (2 avcı – 1 av, avcılar→Q(λ), av→rastgele yürüme)

Grafik 4.6'daki sonuçlara göre 30. bölümden itibaren ajan takımının rastgele kaçan ajanı nasıl yakalayabileceğini anladığı görülmektedir. Bu noktada, takıma yeni katılan ve ava daha yakın bir konumda olan avcının hemen oyunun başında ava doğru yönelerek oyunu sonlandırdığı tespit edilmiştir. Deney 6 için kritik nokta 27. bölümdür. Söz konusu bölüme kadar ortalama 25 hamlede sonlanan oyun, bu bölümden sonra ortalamasını 9'a düşürmüştür. Sonuç olarak, av rastgele hamleler yaparak avcıya kendi davranışları hakkında bilgi edinmesi için hiçbir koz vermemesine rağmen, avcı takımı oyunu kolayca sonlandıran güzergâhı keşfedebilmiştir.

Rastgele yürüyen bir avın yakalanmaya çalışıldığı bu söz konusu deneylerde, yavaş gerçekleştirilen öğrenme ve uygunluk izleri mekanizmasının kullanımının getirileri görülmektedir. Yalnız bu noktada dikkat edilmesi gereken kritik bir nokta farklı oyun senaryoları için farklı konfigürasyonların uygun olabileceğidir. Bu bölümün

sonucunda ajanın hareketli bir hedefi aradığı durumlarda hızlı öğrenmesinin aklında yanlış bilgilerin yer etmesine yol açacağı anlaşılmıştır. Ödülün sürekli yer değiştirdiği ama yine de belli bir bölgede yoğunlaştığı durumlarda temkinli öğrenme yaklaşımının daha uygun olabileceği görülmüştür.

4.4 Akıllı Avcılar – Akıllı Av (Deney 4 ve 7)

Akıllı bir avcının kendisiyle rekabet edebilecek düzeyde öğrenme gücüne sahip bir av ile karşılaştırılması bu çalışmanın asıl ilgilendiği konudur. Bu doğrultuda bütün ajanların Q(λ)-öğrenmesi yöntemini uyguladıkları 1 avcı – 1 av ve 2 avcı – 1 av simülasyonları gerçekleştirilmiştir. Bu deneylere ilişkin bilgiler,

No	Oyuncu S.	P Davranış	E Davranış	α	γ	Λ
4	1 P – 1 E	Q(λ)	Q(λ)	0.05 , 0.9	0.9	0.1 – 1
7	2 P – 1 E	Q(λ)	Q(λ)	0.0.5, 0.9	0.9	0.1 – 1

Tablo 4.5 Deney 4 ve 7'nin özellikleri (P=avcı, E=av)

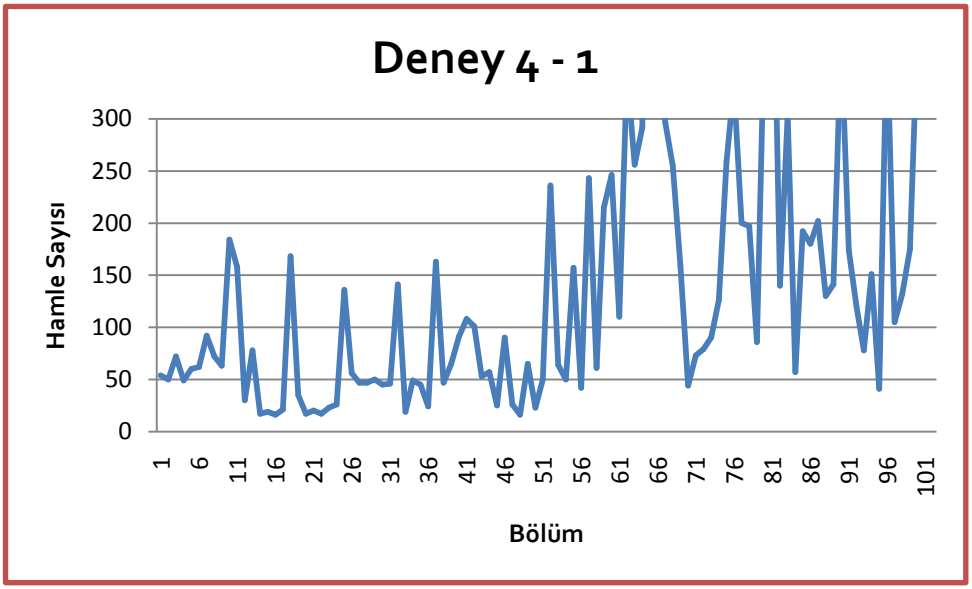
şeklindedir. Bu simülasyonlarda hem avcı takımı hem de av Watkins'in Q(λ) yöntemini kullanmaktadırlar. Deneylerin bu aşamasında ilk kez av akıllı ajan halini alarak öğrenme yetisine sahip olmuştur. Yalnız avın öğrenme stratejisi avcı kadar kolay anlaşılır değildir. Bizim çalışmamızda uygulanan stratejiye göre eğer av, avcı takımının herhangi bir üyesi tarafından yakalanırsa büyük bir ceza almakta ve sakladığı uygunluk izleri sayesinde onu yakalanma durumuna getiren hamlelere hata bilgisi göndermektedir. Bir başka deyişle av, avcıda olduğu gibi bir hedefi kovalama motivasyonuna sahip değildir. Aksine yakalanması durumunda alması muhtemel bir cezadan kaçmaya çalışmaktadır. Avcı ise bu noktada ilk defa öğrenebilen bir ava karşı mücadele etmektedir.

Hem av, hem de avcı Watkins'in Q(λ) öğrenmesi yöntemini kullanmalarına rağmen ödül ve ceza anlayışları birbirlerinden farklı olduğu için, öğrenme faktörüne de farklı tepki gösterirler. Daha önceki deneylerden yola çıkarak hareketli bir avı yakalamaya

çalışan avcının yavaş öğrenmesinin kendisi adına daha verimli olduğunu bilmekteydik. Akıllı av için ise bu detaya dair bir tecrübe bulunmamakta, bu deneylerde bu bilginin ortaya çıkması beklenmektedir. Akıllı avcılar – akıllı av kullanarak yapılan kaçma-kovalama deneylerine ilişkin sonuçlar grafikler 4.7'den 4.12'ye kadar gösterilmektedir. Bu sıraya göre ilk üç deney 1 avcı – 1 av senaryosunu, sonraki üç deney ise 2 avcı – 1 av senaryosunu temsil etmektedir. Ayrıca, söz konusu simülasyonların hangi konfigürasyonlarla gerçekleştirildiğine dair bilgiler ve bunun sonucunda oyunu sonlandıran ortalama hamle sayıları tablo 4.6'da bulunabilir.

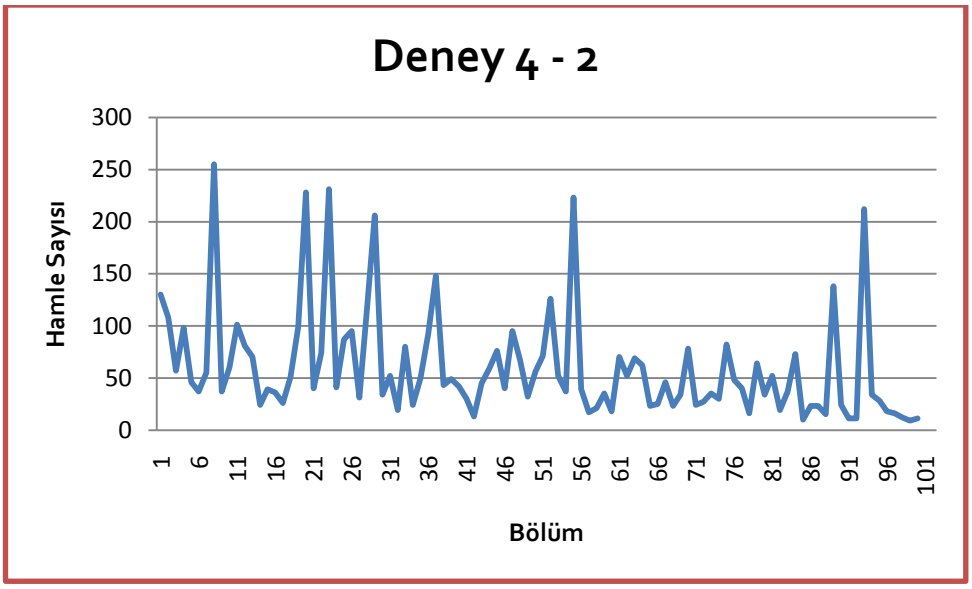
Deney	**α - Avcı 1**	**α - Avcı 2**	**α - Av**	**Ortalama hamle**
4-1	0.9	-	0.9	137
4-2	0.05	-	0.05	60
4-3	0.05	-	0.9	102
7-1	0.9	0.9	0.9	50
7-2	0.05	0.05	0.05	33
7-3	0.05	0.05	0.9	42

Tablo 4.6 Akıllı avcı(lar) – akıllı av deney düzenlemeleri



Grafik 4.7 Deney 4-1 Sonuçları (1 avcı – 1 av, $\alpha_{avcı,\ av}$=0.9)

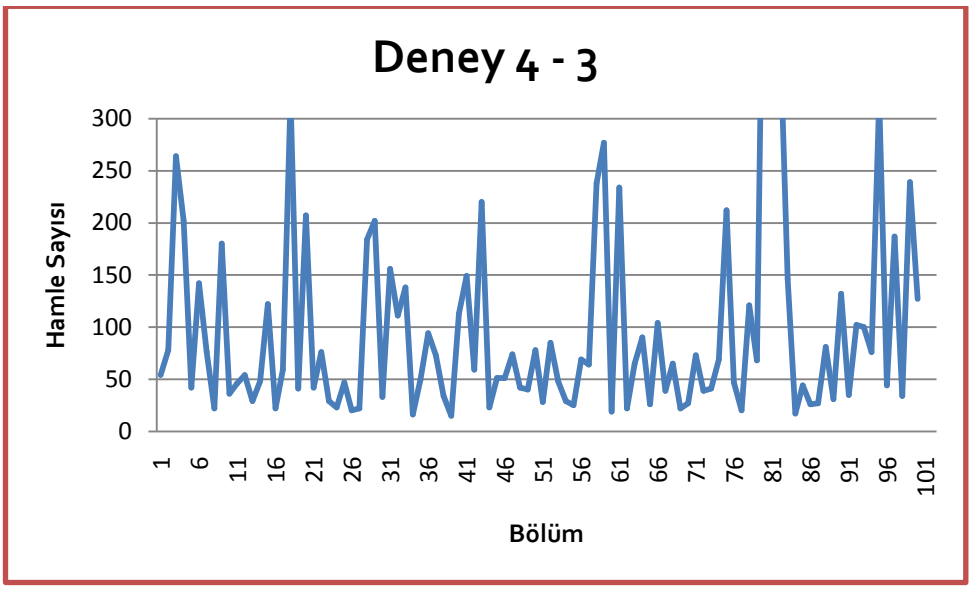
Tek akıllı avcı ve tek akıllı av ile gerçekleştirilen deneyler (4-1, 4-2 ve 4-3) incelendiğinde simülasyon sonuçlarının öğrenme faktörüne göre kritik düzeyde şekillendiği görülmektedir. Deney 4-1'e bakıldığında iki ajan için de öğrenme faktörü 0.9'dur. Önceki deneyler göz önüne alındığında yüksek öğrenme faktörünün hareketli ava karşı mücadele eden avcı söz konusu olduğunda verimsiz olduğu gözlemlenmişti. Bu sonuçlara göre de bölümler ilerledikçe işlerin avcı açısından yolunda gitmediği rahatlıkla söylenebilir. Öyle ki, avcı performansını iyileştirmek yerine son bölümlere doğru daha önce hiç maruz kalmadığı düzeyde başarısızlığa maruz kalmıştır. Avcı için, oyunu sonlandıran hamle sayısının giderek arttığı grafikte kolaylıkla görülebilmektedir. Tablo 4.6'daki ortalama hamle değerine bakıldığında da bir bölümün ortalama 137 hamle sürdüğü anlaşılır. Öğrenme faktörünün 0.9 seçilmesi avcı için ne kadar başarısız bir performansa yol açtıysa; avcının öğrendiği hamlelere hızlıca yanıt verebilen av için de o kadar başarılı olduğu söylenebilir.



Grafik 4.8 Deney 4-2 Sonuçları (1 avcı – 1 av, $\alpha_{avcı,\ av}$=0.05)

Deney 4-2 incelendiğinde, bu simülasyonda öğrenme faktörünün 0.05 seçildiği görülmektedir. Önceki tecrübeler ışığında bunun avcı için olumlu sonuçlar doğurması beklenecektir. Grafikteki deney sonuçları incelendiğinde bu yorumun doğru olduğu anlaşılmaktadır. Bir önceki deney ile karşılaştırıldığında oyunun bir bölümünün bu sefer ortalama 60 hamlede tamamlandığı ve avcının kendini süreç

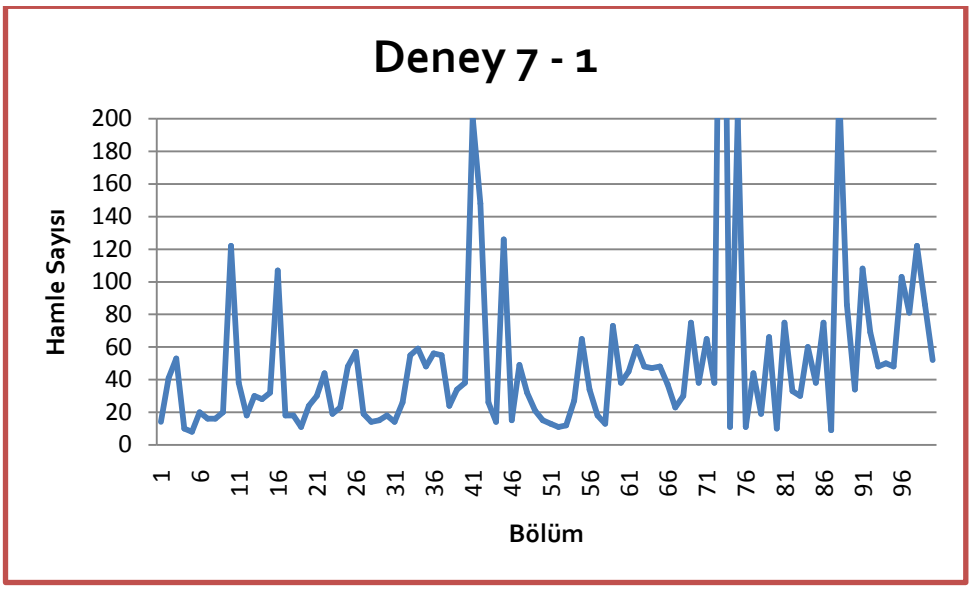
içerisinde geliştirdiği görülmektedir. Av açısından bakılırsa da, bu deney içerisinde çok yavaş öğrenmesi onun avcının hamlelerine karşı adapte olamayıp kendini savunamaması anlamına gelmiştir. 4-1 ve 4-2 deneylerinin sonucunda avcı ve av için farklı öğrenme değerlerinin verimli olduğu görülerek bu doğrultuda bir düzenleme yapılmıştır. (Deney 4-3).



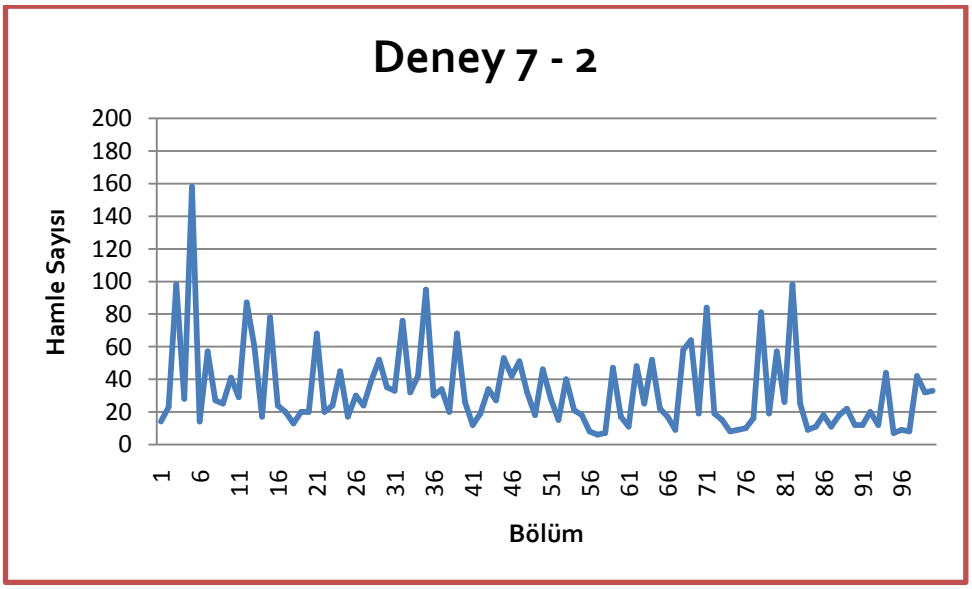
Grafik 4.9 Deney 4-3 Sonuçları (1 avcı – 1 av, $\alpha_{avcı}$=0.05, α_{av}=0.9)

Bahsedildiği üzere deney 4-3'te akıllı avcı ve akıllı avın en verimli oldukları öğrenme değerleri seçilmiştir. İki tarafın da en iyi performansını göstereceği bir çarpışmada kimin üstün taraf olacağı merak edilmektedir. Bu deneyin sonuçları incelendiğinde oyunu sonlandıran hamle sayısının 102, yani yaklaşık olarak önceki iki deneyin ortalaması olduğu görülmüştür. Grafik incelendiğinde simetriğe yakın bir görüntü ortaya çıkmakta ve oyunun başında sonuna kadar iki taraf adına da bir üstünlük göze çarpmamaktadır. Yalnız, grafik üzerinde dikkat edilmesi gereken bir nokta; süreç içerisinde zaman zaman avcının performansını iyileştirerek avı az sayıda hamleyle yakalayabildiği, buna karşın avın derhal adapte olarak cezalara maruz kalmamak için tehlikeyi bertaraf ettiğidir. Elde edilen bu sonuçlar iki tarafın da verimli bir şekilde öğrenme gerçekleştirebildiğini gösterir.

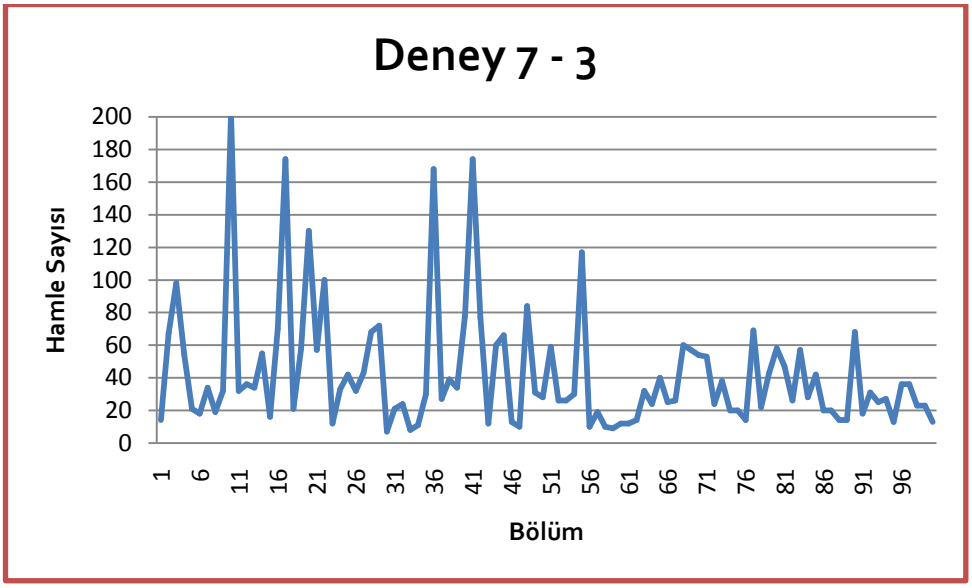
Deneyler 7-1, 7-2 ve 7-3'te avcı takımı iki ajana çıkartılmıştır. Daha önceki çok ajanlı deneylerde olduğu gibi bunun harita üzerinde düzgün dağılımı sağlaması ve dayanıklılığa sebep olması beklenmektedir. Kullanılan öğrenme değerleri 4. deneyde olduğu gibi düzenlenmiştir. Bu deneylerin sonuçları grafikler 4.10, 4.11 ve 4.12'de sunulmuştur.



Grafik 4.10 Deney 7-1 Sonuçları (2 avcı – 1 av, $\alpha_{avcı,\ av}$=0.9)



Grafik 4.11 Deney 7-2 Sonuçları (2 avcı – 1 av, $\alpha_{avcı,\ av}$=0.05)



Grafik 4.12 Deney 7-3 Sonuçları (2 avcı – 1 av, $\alpha_{avcı}$=0.05, α_{av}=0.9)

Deney 7'nin sonuçları bir önceki deneyle paralel olacak şekilde incelenirse, elde edilen verilerin aynı öğrenme faktörü düzenlemeleri için aynı örüntüyü takip ettiği görülmektedir. Diğer bir deyişle avcı takımına yeni bir üyenin katılmasıyla oyunu sonlandıran hamle sayısı beklenildiği gibi azalmış ve aynı öğrenme değerleri için grafikler benzer şekillenmiştir.

Bu bölümde sunulan simülasyonların tamamı için iz kaybolma oranı 1'dir. 0.1'den itibaren farklı değerler kullanılarak yapılan deneyler sonucunda optimal sonuca λ=1 atamasıyla ulaşıldığı görülmüştür. Bu doğrultuda, ajan yaptığı her hamlenin ardından kendisini bu yola teşvik eden hamlelere sorumluklarıyla orantılı olarak hata değerlerini döndürür ve aksiyon-değer tablosu güncellenir. Sorumluluk bilindiği gibi iz kaybolma oranıyla belirlenir. En son yapılan hamlenin sonuç üzerindeki sorumluluğu yüksekken, geçmişe doğru gidildikçe bu sorumlulukların değeri düşer. Deney 7'nin sonuçları çok ajanlı sistemlerin kullanımı açısından incelendiğinde, daha önceden de olduğu gibi coğrafi dağılımdan yararlanılarak iyi bir sonuç ortaya konduğu görülmektedir. Bu noktada iyi düzeyde öğrenme gerçekleştirebilen bir akıllı ajana kıyasla birden fazla ajan kullanmanın daha verimli bir yöntem olduğu tartışmaya açıktır; çünkü ajanlar arasında henüz bir işbirliği dahi yokken, takımın sayıca fazla olmasının getirdiği avantajlar bir alana dağılmayı kolaylaştırmaktadır.

BÖLÜM 5

5. SONUÇ

Takviyeli öğrenme, günümüzde çok çeşitli problemlere uygulanabilen ve ajanların bir öğretmene veya önbilgiye ihtiyaç duymadan sadece çevre ile etkileşimlerini kullanarak öğrendikleri bir yöntemdir. Uygunluk izleri mekanizması ise, ajanların hafızasına bir sorumluluk katsayısı ekleyerek, gerçekleştirilen bir hamleden elde edilen hata bilgisinin, sorumlulukları çerçevesinde geçmiş aksiyonlara bildirilmesini sağlar. Watkins'in Q(λ)–öğrenmesi yöntemi, standart Q-öğrenmesinin uygunluk izleri kullanılarak genelleştirilmiş bir versiyonudur.

Kaçma-kovalama problemleri ise, oyun teorisinde yer alan ve uygulamalarına özellikle güvenlik alanında rastlanan bir araştırma konusudur. Bilindiği gibi kaçma-kovalama oyunu ailesi, çok ajanlı sistemler ve öğrenme yöntemlerinin uygulanması için uygun bir alandır. Dolayısıyla çok ajanlı sistemlerde eşzamanlı takviyeli öğrenme yaklaşımı üzerine yapılan bu araştırmanın söz konusu oyun üzerinde gerçekleştirilmesi uygun olmuştur.

Bu tez çalışmasında, çok ajanlı kaçma-kovalama problemlerine takviyeli öğrenme yaklaşımı anlatılmıştır. Yapılan araştırmanın sunduğu bir yenilik olarak, oyunda kovalayan ajanların yanı sıra kaçan ajan da takviyeli öğrenme gerçekleştirmektedir. Ayrıca, kovalayan ajan takımı her ajanın bağımsız olarak öğrendiği eşzamanlı öğrenme yöntemini kullanmaktadır. Çalışmalar süresince uygulanan yöntemler ve simülasyon ortamında gerçekleştirilen deneylere ilişkin sonuçlar detaylı olarak anlatılmıştır. Yapılan deneyler için karşılaştırma ortamı sunulması adına kovalayan ajanın standart Q-öğrenmesi yöntemini uyguladığı, kaçan ajanın da sabit olduğu ve rastgele yürüme yaptığı senaryolara dair sonuçlar açıklanmıştır. Elde edilen sonuçlar neticesinde kaçan ajanın başarılı bir şekilde *akıllı* kaçış stratejisi gerçekleştirebildiği gözlemlenmiştir. Gelecek süreçte, deneyler simülasyon ortamından gerçek dünya ortamına taşınarak çalışmalar genişletilebilir.

KAYNAKLAR

[1] R. Vidal, O. Shakernia, H.J. Kim, D.H. Shim, S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation and experimental evaluation",*IEEE Transactions on Robotics and Automation*, 18-5, sayfa 662-669, 2002.

[2] T. Haynes, S. Sen, "Evolving behavioral strategies in predators and prey", *Adaptation and Learning in Multiagent Systems*, Springer Verlag: Berlin, sayfa 113-126, 1996

[3] F. Amigoni, N. Basilico, "A game theoretical approach to finding optimal strategies for pursuit evasion in grid environments", *IEEE International Conference on Robotics and Automation*, RiverCentre, Saint Paul, Minnesota, ABD,14-18 Mayıs 2012.

[4] A. Kehagias, G. Hollinger, S. Singh, "A graph search algorithm for indoor pursuit/evasion", *Mathematical and Computer Modelling*, 50, sayfa 1305-1317, 2009.

[5] J. Li, Q. Pan, B. Hong, "A new approach of multi-robot cooperative pursuit based on association rule data mining", *International Journal of Advanced Robotics Systems*, 6-4, 329-336, 2009

[6] J. Liu, S. Liu, H. Wu, Y. Zhang, "A pursuit-evasion algorithm based on hierarchical reinforcement learning",*International Conference on Measuring Technology and Mechatronics Automation*, Hunan, Çin, 11-12 Nisan 2009

[7] Y. Ishiwaka, T. Sato, Y. Kakazu, "An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning", *Robotics and Autonomous Systems*, 43, sayfa 245-256, 2003.

[8] T. Chung, J.W. Burdick, "Analysis of search decision making using probabilistic search strategies", *IEEE Transactions on Robotics*, 28-1, 2009.

[9] S.F. Desouky, H.M. Schwartz, "Q(λ)-learning adaptive fuzzy logic controllers for pursuit-evasion differential games", *International Journal of Adaptive Control and Signal Processing*, 2011

[10] L.E. Parker, "Distributed algorithms for multi-robot observation of multiple moving targets", *Autonomous Robots*, 12, sayfa 231-255, 2002

[11] J.W. Durham, A. Franchi, F. Bullo, "Distributed pursuit-evasion without mapping or global localization via local frontiers",*Auton Robot*, 32, sayfa 81-95, 2012

[12] G. Hollinger, S. Singh, A. Kehagias, "Improving the efficiency of clearing with multi-agent teams", *The International Journal of Robotics Research*, 29, 2010.

[13] S. Desouky, H. Schwartz, "Learning in n-pursuer n-evader differential games" *IEEE International Conference on Systems, Man and Cybernetics*, İstanbul, Türkiye, Ekim 2010.

[14] B. Bouzy ve M. Metivier, "Multi-agent model-based reinforcement learning experiments in the pursuit evasion game", 2007.

[15] A. Kolling, S. Carpin, "Multir-robot pursuit-evasion without maps", *IEEE Conference on Robotics and Automation*, Anchorage, Alaska, ABD, 3-8 Mayıs 2010.

[16] A. Kolling, A. Kleiner, M. Lewis, K. Sycara, "Pursuit-evasion in 2.5d based on team-visibility", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Tayvan, 18-22 Ekim 2010.

[17] B.M. Faiya, H.M. Schwartz, "Q(λ)-learning fuzzy controller for the homicidal chauffeur differential game", *20th Mediterranean Conference on Control & Automation (MED)*, Barselona, İspanya, 3-6 Temmuz 2012

[18] S. Rodriguez, J. Denny, A. Mahadevan, J. Vu, J. Burgos, T. Zourntos, N.M. Amato, "Roadmap-based pursuit-evasion in 3d structures", 2010.

[19] R. Isaacs, "Differential Games: A Theory with Applications to Warfare and Pursuit, Control and Optimization", New York: John Wiley & Sons, 1965.

[20] M.M. Flood, "The hide and seek game of Von Neumann", Management Science, 18-5, Ocak 1972.

[21] T.D. Parsons, "Pursuit-evasion in a graph", *Theory and Applications of Graphs*. Springer Verlag: Berlin, sayfa 426-441, 1978.

[22] J. Durham, A. Franchi, F. Bullo, "Distributed pursuit-evasion with limited-visibility sensors via frontier-based exploration", *IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, ABD, 3-8 Mayıs 2010

[23] R.S. Sutton, A.G. Barto, "Reinforcement Learning: An Introduction", The MIT Press: Cambridge, Massachusetts, 1998.

[24] R.S. Sutton, "Learning to predict by the method of temporal differences", *Machine Learning*, 3, sayfa 9-44, 1988.

[25] C.J. Watkins, "Learning from delayed rewards", Doktora Tezi, Cambridge University, 1989.

[26] C.J. Watkins, P. Dayan, "Q-Learning" *Machine Learning*, 8, 279-292, 1992.

[27] A.G. Barto, R.S. Sutton, C.J. Watkins, "Learning and sequential decision making", *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, The MIT Press: Cambridge, Massachusetts, 1990.

[28] F. Fernandez, D. Borrajo, L.E. Parker, "A reinforcement learning algorithm in cooperative multi-robot domains", *Journal of Intelligent and Robotic Systems*, 43, sayfa 161-174, 2005

[29] L. Panait, S. Luke, "Cooperative multi-agent learning: The state of the art", *Autonomous Agents and Multi-Agent Systems*, 11, sayfa 387-434, 2005.

[30] P. Stone, M. Veloso, "Multiagent systems: a survey from a machine learning perspective", *Autonomous Robots*, 8, sayfa 345-383, 2000.

[31] M. Tan, "Multi-agent reinforcement learning: independent vs. cooperative agents", 10^{th} International Conference on Machine Learning, Amherst, Massachusetts, ABD, 1993.

[32] N. Ono, K. Fukumoto, "Multi-agent reinforcementt learning: a modular approach", *2^{nd} International Conference on Multiagent Systems*, Kyoto, Japonya, 9-13 Aralık 1996.

[33] L.E. Parker, C. Touzet, F. Fernandez, "Techniques for learning in multi-robot teams", *Robot Teams: From Diversity to Polymorphism*, AK Peters, 2001

[34] E.U. Acar, H.Choset, Y. Zhang, M. Schervish, "Path planning for robotic demining: robust sensor-based coverage of unstructured environments and probabilistic methods", *The International Journal of Robotics Research*, 22, sayfa 441-466, 2003.

[35] A.H. Bond, L. Gasser, "An analysis of problems and research in DAI", *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers: San Mateo, California, sayfa 3-35, 1988

[36] S. Russell, P. Norvig, "Artificial Intelligence: A Modern Approach", 1995.

[37] T. Aral, E. Pagello, L.E. Parker, "Guest editorial advances in multirobot Systems", *IEEE Transactions on Robotics and Automation*, 18-5, 2002.

[38] W. Burgard, M. Moors, D. Fox, R. Simmons, S. Thrun, "Collaborative multi-robot exploration", *IEEE Conference on Robotics and Automation (ICRA)*, San Fransisco, California, ABD, 24-28 Nisan 2000.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı:	BİLGİN, Ahmet Tunç
Uyruğu:	T.C.
Doğum tarihi ve yeri:	26.07.1989 Gölbaşı/Ankara
Medeni hali:	Bekâr
Telefon:	0 (533) 437 75 99
E-posta:	abilgin@etu.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	TOBB ETÜ Bilgisayar Mühendisliği	2013 (beklenen)
Lisans	TOBB ETÜ Bilgisayar Mühendisliği	2010

İş Deneyimi

Yıl	Yer	Görev
2013 – Halen	Bankacılık Düzenleme ve Denetleme Kurumu	Bankacılık Uzman Yrd.
2011 – 2013	TOBB ETÜ Fen Bilimleri Enstitüsü	Öğretim Asistanı
2010 – 2010	TDB Dienstleistungen GmbH	Yazılım Programlama
2009 – 2009	ILG Bilişim Teknolojileri	Yazılım Programlama
2008 – 2008	BİMEL Elektronik Ltd. Şti.	Yazılım Programlama

Yabancı Dil: İngilizce