T.C.

FIRAT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ETMENLERİN İÇ MODELİNİ KULLANARAK ÇOKLU ETMENLERDE BULANIK TAKVİYELİ ÖĞRENME

Alper KILIÇ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

ELAZIĞ

2002

T.C.

FIRAT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ETMENLERİN İÇ MODELİNİ KULLANARAK ÇOKLU ETMENLERDE BULANIK TAKVİYELİ ÖĞRENME

Alper KILIÇ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİMDALI

Bu tez, ... tarihinde aşağıda belirtilen jüri tarafından oybirliği /oyçokluğu ile başarılı / başarısız olarak değerlendirilmiştir.

Danışman: Doç. Dr. Ahmet ARSLAN

Üye: Doç. Dr. Erhan AKIN

Üye: Yrd. Doç. Dr. Fikret ATA

Üye:

Üye:

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ...../...../..... tarih ve ..... sayılı kararıyla onaylanmıştır.

TEŞEKKÜR

Bu tez çalışmasında yardımlarını esirgemeyen tez danışmanım Sayın Doç. Dr. Ahmet ARSLAN, çalışma arkadaşlarım Mehmet KAYA, Mehmet KARAKÖSE, İrfan GÜLTEKİN, K. Koray GÜNDOĞAN, M. Temel ÖZDEMİR ve Erkan DUMAN'a teşekkürlerimi sunmayı borç bilirim.

İÇİNDEKİLER

SAYFA NO

ŞEKİLLER LİSTESİ

SAYFA NO

SİMGELER

S_i	: i. Durum.
a_i	: i. Hareket.
r_i	: i. Ödül.
γ	: Azalma faktörü.
V^*	: Optimal değer.
$Q(s,a)$	: s durumunda a hareketi yapılınca oluşan Q değeri.
Δ_n	: $\hat{Q}_n$ 'de oluşan maksimum hata.
$\hat{Q}_n$	: n. güncellemeden sonra tahmin edilen Q değeri.
$\prod$	: Optimal politika.
$Pr(s'\|s,a)$	: Etmenin a hareketini seçmesi için s durumundan yeni bir s' durumuna geçme olasılığı.
d	: Etmenin görsel alanı.
a_{self}	: Etmenin kendi yaptığı hareket.
a_{other}	: Gözlemlenen etmenin yaptığı hareket.
0	: Kontrol parametresi.

ÖZET

Yüksek Lisans Tezi

ETMENLERİN İÇ MODELİNİ KULLANARAK ÇOKLU ETMENLERDE BULANIK TAKVİYELİ ÖĞRENME

Alper KILIÇ

Fırat Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

2002, Sayfa :49

Çoklu etmenli sistemlerde birlikte hareket etme ve işbirliği davranışlarının öğrenilmesi işleminin gerçekleştirilmesi Yapay Zeka sistemleri için son yıllarda gelişen yeni bir bakış açısı ortaya koymuştur. Özellikle Takviye Öğrenme yöntemleri çoklu etmenli sistemlere ve dinamik ortamlara uygulanabilirliğinden dolayı bir çok çalışmanın odak noktası haline gelmiştir. Takviye öğrenme metotları içerisinde en popüler algoritmalardan olan Q-öğrenme; çoklu etmenli sistemlere uygulandığı durumlarda optimum politikaya yönelme hızı bakımından ve sürekli- tam olarak gözlemlenemeyen ortamlarda ek bazı dezavantajları da beraberinde getirmektedir.

Bu tez çalışmasında çoklu etmenli sistemlerde konu olan dezavantajların giderilmesi amacıyla hem standart Q-öğrenme algoritmasına alternatif bir algoritma sunulmuş (FQ-Öğrenme), hem de diğer etmenlerin iç modelinin çıkarılması için bulanık küme ve bulanık mantık yaklaşımı üzerinde durularak sonuçlar performans bakımından incelenmiştir.

Anahtar Kelimeler: Çoklu etmenli sistemler, Makine öğrenmesi, Takviye öğrenme, Q-Öğrenme, Bulanık Mantık

ABSTRACT

Masters Thesis

FUZZY REINFORCEMENT LEARNING IN MULTI-AGENT SYSTEMS USING INTERNAL MODEL OF AGENTS

Alper KILIÇ

Fırat University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

2002 : Page 50

Recently, delayed reinforcement learning (RL) has been proposed as a strong method for learning in multi-agent systems (MASs). In this method, agents are concerned with the problem of discovering an optimal policy, a function mapping states to actions. The most popular RL technique, Q-learning, has been proven to produce an optimal policy under certain conditions. In this thesis, we consider a multi-agent cooperation problem, and propose a multi-agent reinforcement learning method based on the other agents' actions.

In our learning method, the agent under consideration observes other agents' action, and uses the minimax Q-learning using fuzzy state and fuzzy goal representation for updating fuzzy Q values. We also proposed a new method called FQ-learning so as to accelerate convergence of learning process.

Keywords: Multi-Agent Systems, Machine Learning, Reinforcement Learning, Q-Learning, Fuzzy Logic.

1. GİRİŞ

Günümüzde standart programlama metotları önceki yıllara göre oldukça farklılıklar göstermektedir. Bunların içerisinde göze çarpan yeniliklerden biri de sisteme akıllı hareket edebilme özelliklerini katmak ve giriş-çıkış arasındaki ilişkiler için yorum yeteneğini kazandırmaktır. Hem yorum yeteneğini kazandıran öğrenme metotları, hem de insan tecrübelerini sisteme kazandıran bulanık mantık yaklaşımı bu tez çalışmasında ele alınan yöntemlerdir. Bunun için öncelikle bu bölümde akıllı etmen kavramının ne olduğu konusu tartışılacak, takip eden diğer bölümde akıllı etmenlerin öğrenme algoritmaları –özellikle q-öğrenme- üzerinde durulacaktır. Bulanık küme ve bulanık mantık teorilerinin özetlenmesinin ardından, standart Q-öğrenme algoritmalarının var olan dezavantajlarını göz önünde bulundurarak, tezde amaçlanan çoklu etmenli sistemlerde FQ-Öğrenme, Bulanık Q-Öğrenme, Bulanık Minimax Q-Öğrenme, metodu ve son bölümde de deneysel ortam tanıtılıp sonuçları yorumlanacaktır.

Akıllı etmen kavramı daha çok 1990'ların başlarından itibaren kullanılmaya başlanmıştır. Başlangıçta farklı anlamlarda kullanılan bu kavramın zamanla hem yapı hem de karakteristik özellikleri üzerinde ortak bir anlayış oluşmuştur. Akıllı etmen için yapılan her bir tanımlamada kuşkusuz bir takım özelliklerden yola çıkılmıştır. İlk zamanlarda dar kapsamlı tanımların yanında, bir uzman sistem veya bir yapay zeka aracı da bir akıllı etmen olarak algılanmıştır.

Canlıların doğal ortamdaki yaşamları, duyularla dış dünyayı algılama ve buna karşılık en uygun davranışla tepki gösterme çevrimi olarak özetlenebilir. Dış dünyayı algılama, duyu organları ile tepki gösterme eylemleri de el ve ayaklar gibi bir çok organlarla yürütülür. Canlılardaki bu etkinlikler gibi, akıllı bir etmenin gerek sanal ve gerekse gerçek ortamdaki özerk çalışabilmesi bugün yazılım teknolojisi ile başarılabilmektedir. Bu tür özerk çalışabilen yazılımlara akıllı etmen adı verilir.

Akıllı etmenler için literatürde yapılan tanımlardan bazıları şunlardır:

"Akıllı etmenler, gerçek zaman şartlarında içinde bulunduğu dinamik ortamla etkileşimli olarak çalışan ve hareket eden yapılarıdır." [5].
"Akıllı etmenler yapacakları işler için muhakeme yapmaya izin veren ve değişen çevre şartlarına davranışlarıyla dinamik adaptasyon sağlayan yapılardır." [5].

"Bir akıllı etmen; kanaat getirme, kabilliyetli olma, seçim yapabilme ve taahhüt yapabilme gibi mantıksal birimlerden oluşan ve de mantıksal faaliyet yapan bir varlıktır" [3].

Yapılan tanımların yoğunlaştığı ana karakteristikler;

- Sahip olunan bilgiyi kullanabilme yeteneği,
- İçinde bulunulan çevreye adaptasyon,
- Yürütülen işlev için gerekirse muhakeme yapabilme ve karar verebilme,

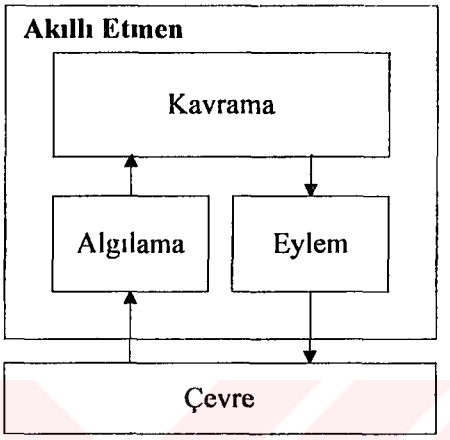
şeklinde ifade edillebilir. Yapısal olarak akıllı etmenler insanların psikolojik ve biyolojik yapılarından esinlenerek geliştirilmişlerdir. Buradan yola çıkılarak bir tanım yapılacak olursa: Akıllı etmenler; insanların psikolojik ve biyolojik yapılarından esinlenerek bir takım görev ve hedefleri başarmak üzere tasarlanan, içinde çalıştığı çevre ile dinamik bir etkileşim halinde olabilen ve yaptığı işlem için gerekli muhakeme ve karar verme yetkisi ile özerk davranabilen sistemlere denir.

1.1 Bir Akıllı Etmenin Genel Yapısı

Akıllı etmenler canlıların bir takım psikolojik ve/veya biyolojik yapılarından yola çıkılarak geliştirilen yapılar olması dolayısıyla tasarlanmalarında istenen işlevlerin yerine getirilmesi için canlıların taklit edilen özelliklerine benzerlik göstermesi hedeflenmektedir. Bu hedeflenen özelliklerin başında akıllı etmenlerin özerk olarak eylem gerçekleştirmeleri gelir. Özerk bir akıllı etmen çevresi ile herhangi bir dış yönlendirmeye gerek duymadan algılama ve eylem gösterme şeklinde etkileşebilmektedir. Etkileşim yoluyla yapılan eylemler ile çevreye rahatlıkla cevap verilebilmektedir.

Bir akıllı etmen şekil 1.1 de görülebileceği gibi üç temel yapısal alt birimden oluşur. Çevre ile etkileşimi, mesajların algılanabilmesini sağlayan algılama ve mesaja uygun cevabı verebilmeyi sağlayan eylem birimleri sağlar. Alınan mesaja göre en uygun eylemi tesbit etmede verilecek kararı da kavrama birimi belirler. Bir akıllı etmende bir eylem gerçekleştirme çevrimi şu şekilde gerçekleşir; çevresinden mesajları algılama birimi ile alır, öz bilgisine dayanarak gerekli davranış için karar verir ve bir eylem üretmek üzere içinde bulunduğu çevrede ilgili davranışını uygular. Böylece verilen bir görev veya hedefi başarmak için uygun bir adım atmış olur.

Diğer yapay zeka ürünlerinden farklı olarak akıllı etmenler, bir dış müdahele veya komut ile değil tamamen öz bilgisi ve her zaman aktif olması ile işlevlerini yürütürler. Anlamlı bir algılamayı gerçekleştirdikleri an gerekli tepkiyi gösterirler. Bu özellik bir akıllı etmene diğer yazılım sistemlerinden farklı bir kişilik sağlar.



Şekil 1.1: Genel olarak bir akıllı etmenin yapısal şeması.

1.2 Akıllı Etmenlerin Özellikleri

Akıllı etmenlerin özelliklerini yapısal ve yeteneksel olmak üzere iki kategoride incelemekte yarar vardır. Yapısal özellikler derken sistemin mimari özellikleri kasdedilmektedir. Yeteneksel özelliklerden de sistemin edinebildiği yetenekleri anlaşılmaktadır.

1.2.1 Yapısal özellikler

Bir akıllı etmenin mimari yapısı, yukarıda da belirtildiği gibi işlevlerin yürütülmesi sırasında hangi davranışın ne zaman ve nasıl gösterileceği, bilgilerin nasıl temsil edileceği, davranışların hangi sıra ve kurala göre yürütüleceği, sistemin parçalarının ne şekilde entegre olacağı sorularını yanıtlar. Bundan dolayı mimari yapılar bir takım yaklaşım tarzlarına göre yapılandırılırlar. Söz konusu yaklaşımlar temel olarak geleneksel ve davranış eğilimli yaklaşımlardır [5].

Geleneksel yaklaşımda akıllı etmenlerin tasarlanması daha çok insan bilgi işleme mekanizması ve algılama psikolojisi baz alınarak gerçekleştirilmiştir. Bu bakımdan söz konusu yaklaşımla tasarlanan sistemlerde bulunan temel birimler algılama, kavrama ve eylem

birimleridir. Bu birimler özellikle ilk dönem akıllı etmenlerde belirgin olarak mevcuttur. Nitekim ilk yapay zeka çalışmalarını başlatan Amerikalı dört ekolden birinin lideri olan Allen Newel bir psikologdur. Bu ekol daha sonraları GPÇ (Genel Problem Çözücü) ve SOAR sistemlerini geliştirmişlerdir [4]. Gerek Newel ekolünün geliştirdiği sistemler ve gerekse ilk dönem akıllı etmenlerin yapılarında asıl yoğunlaşan ana birim kavrama birimi olmuştur. Diğer iki birime sistemin performansını etkileyecek çok fazla bir rol verilmemiştir.

90'ların başından itibaren ortaya çıkan bu ikinci yaklaşım davranış psikolojisinden esinlenerek ortaya atılmış ve bu anlayışı akıllı etmen tasarımına taşımıştır. Bu yüzden davranış tabanlı yaklaşım adını da almaktadır. Mümkün olduğunca bilgi tabanlı sistemlerden faydalanmama düşüncesini taşıyan bu yaklaşımın temsilcileri bilgi tabanlı sisitemlerin bazı dezavantajlarından kurtulmayı amaçlamışlardır. Yaklaşımların farklılığı her iki yaklaşımın akıllı etmen tasarımında ön gördüğü ana birimleri de etkilemiştir. Özellikle Brooks (1986) ve Agre ve Chapman (1987) bu anlayışın ilk temsilcileridir. Bu iki çalışmadan sonra özellikle robotik etmenlerde araştırıcıların yoğun ilgisini çeken bu yaklaşım, öğrenme kaabiliyeti gibi bir çok yeteneğin akıllı etmenlere kazandırılmasını sağlamıştır. Davranış tabanlı yaklaşım ile bilgi tabanlı sistemlerin hantallığı, rijitliği ve yavaşlığı gibi dezavantajlarının önlenmesi başarılmıştır. Bu yaklaşımın en önemli gerekçesi geleneksel sistemlerle geliştirilen akıllı etmenlerin gerçek dünyadaki uygulanırlıklarının olmayışı veya çok zor oluşudur.

1.2.2 Yeteneksel özellikler

Bir akıllı etmenin mimari özelliklerinin yanında başarısını doğrudan etkileyen yeteneksel özelliklerinden de bahsetmek gerekir. Akıllı etmenlerin kişilik bulmasına bu karakteristikler sebep olmaktadır. Verilen bir görevin başarılabilmesi için gerekli olan davranışın seçilmesi ve herbir davranış için ilgili mekanizmaların harekete geçirilebilmesi gerekir. Akıllı etmenlerin karakteristiklerinin belirlenmesini elde etme yolu da mimari yapılarının özelliklerine bağlıdır. Zaten mimari yapılardaki değişikliğin sebebi de bundandır. Yani akıllı etmenlerin mimari yapılarına göre değişmeyen karakteristikleri ve değişebilen karakteristikleri vardır. Değişmeyenler akıllı etmenin tanımı gereği sahip olduğu özelliklerdir. Değişebilenler ise mimari yapılara bağlı karakteristiklerdir. Genel olarak akıllı etmenlerde bulunan temel karakteristikleri sıralamak gerekirse:

1. Özerklik (Autonomy)
2. Adaptasyon Yeteneği (Adaptivity)
3. İşlev Yürütme (Functionality)
4. Tutarlılık (Coherency)

Özerklik karakteristiği, sistemin kendi ortamında tamamen kendi başına hareket edebilmesi özelliğidir. Akıllı etmen, içinde bulunduğu çevrede, algıladığı uyarılara göre kendi başına tepki gösterir ve hiçbir yerden destek almadan hareket eder.

Adaptasyon karakteristiği bir akıllı etmenin daha önce karşılaşmadığı yeni durumlarla karşılaşması halinde anlamlı ve mantıklı davranış üretebilmesi anlamına gelir. Canlılar nasıl rahatlıkla çevrelerine uyum sağlayabiliyorlarsa bir akıllı etmenin de çevresiyle uyumlu çalışabilmesi gerekir.

Her akıllı etmenin işlevsellik karakteristiğine sahip olması gerekir. Hiçbir akıllı etmen amaçsız, görevsiz ve de işlevsiz tasarlanmaz. Akıllı etmenler görevleri doğrultusunda işlev yürütürler. Birden fazla görevi olan akıllı etmenler, aktif olan görevi ne ise o doğrultuda davranış sergiler.

Tutarlılık karakteristiği, akıllı etmenin sergilediği davranışlarında bir uyum içinde bulunması olarak tanımlanabilir. Benzer şartlarda benzer davranışları gösterebilmesi, akıllı etmenin tutarlılığını gösterir.

2. Bulanık Mantık ve Bulanık Sistemler

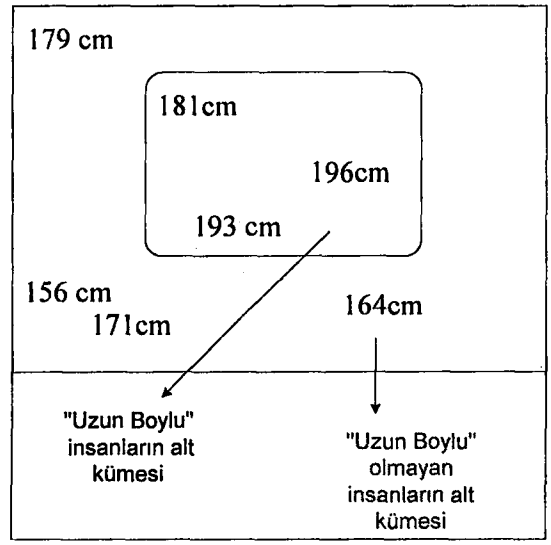
Bulanık mantık model temelli sistemler insan tecrübelerine dayanan bir teknolojidir. Bulanık mantık, sistem davranışını tanımlamak için dilsel değişkenler kullanır ve bu sayede tam bir matematiksel modellemeye olan ihtiyaç ortadan kalkar. Bulanık mantık kullanımı, çok karmaşık sistemler tasarlamanın, optimize etmenin ve bakımını yapmanın etkili bir yoludur. Hata toleranslı ve duyarlığı az sistemlere yeni bir yaklaşım tarzı olması ve akıllı sistemleri hayata geçiren bir teknoloji olma özelliği bulanık mantık kullanımını oldukça çekici bir hale getirmektedir.

Verilen bir sistem için tam bir matematiksel modelin kolayca bulunamadığı; nonlineerliklerin, zaman kısıtlamalarının ve çok parametrenin olması durumlarında, problem hakkında tasarım bilgisi varsa veya tasarım sırasında elde edilebildiğinde bulanık mantık kullanılması kolaylıklar sağlar. Ayrıca, sistem davranışının belirli kümeler çerçevesinde ortak çıkışlar ürettiği durumlarda bulanık sistemin oldukça başarılı sonuçlar verdiği gözlenmiştir.

Yapısı ve ortaya çıkma sebebi insan tecrübelerinin sisteme aktarılması şeklinde olan bulanık mantık ve bulanık küme yaklaşımı, tecrübe ve bilgi paylaşımı gerektiren yapay zeka platformunda da kullanıldığı hakkında literatürde bir çok çalışmaya rastlanmaktadır.

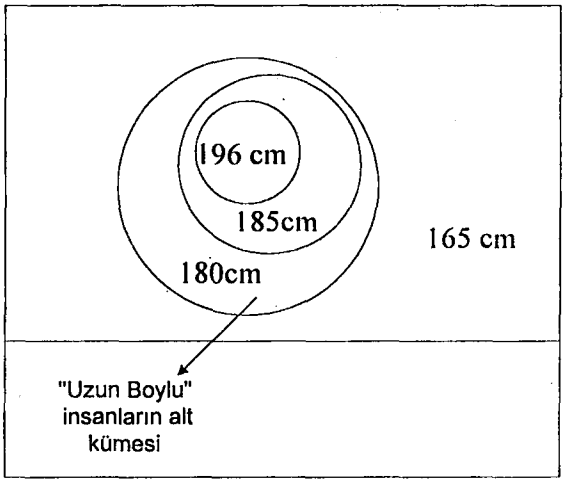
Mühendislik tecrübe ve deneyimlerini, konvansiyonel modelleme anlayışı ile birleştirmenin sonucunda sistem performansının artması, varolan tasarımlara yeni fonksiyonlar kazandırması, geliştirme sürecini ve pazara sunma süresini, ileri geliştirme programları kullanarak kısaltması dikkate alınabilecek bir avantajdır.

Burada klasik kümeler ile bulanık kümeler arasındaki farklı bir örnek sayesinde açıklamak, konunun prensibinin anlaşılmasında kolaylık sağlayacaktır.



Şekil 2.1 Bir klasik keskin küme

İnsanlar için "uzun boylu" olma ölçüsü olarak boy>180 cm olduğu düşünülürse, Şekil 2.1'de görüldüğü gibi uzun boylu olma sınırı çok kesin bir ifade ile belirtilmiştir. Halbuki aynı olay bulanık küme ile gösterilebilir ;



Şekil 2.2 Bir Bulanık Küme

Şekildeki bulanık kümenin matematiksel tanımı şu şekilde yapılabilir :

⇒ Bulanık küme "Uzun Boylu" UB

⇒ X, tüm boy değerlerinin kümesi

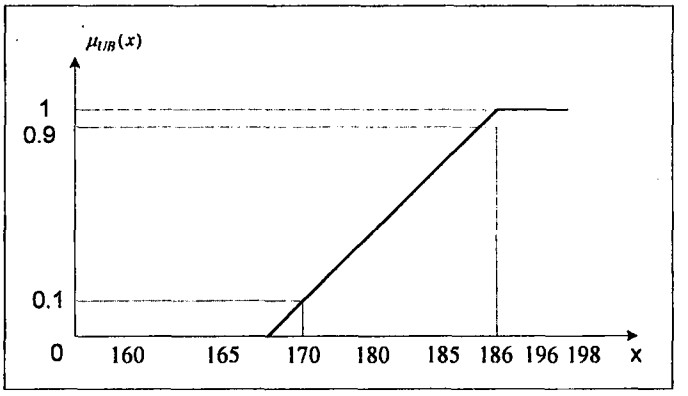
⇒ Her x∈X ve de üyelik fonksiyonları μUB(x)

⇒ μ, üyelik derecesi, μ ∈ [0,1]

Örnek olarak

$\mu_{YA}(165)=0$, $\mu_{YA}(170)=0.1$, $\mu_{YA}(185)=0.9$, $\mu_{YA}(160^0C)=0$,

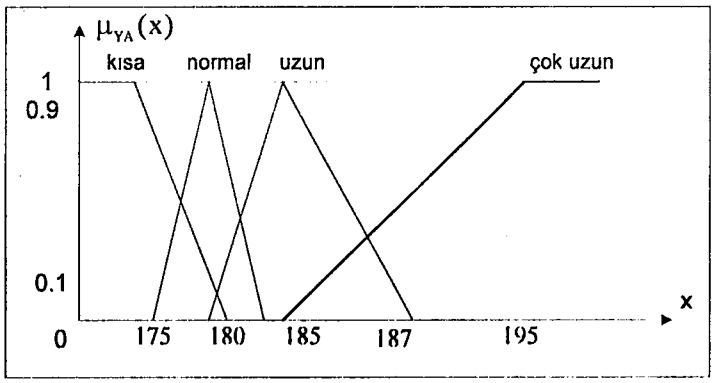
$\mu_{YA}(178)=0.35$, $\mu_{YA}(196)=1$, $\mu_{YA}(170)=0$, $\mu_{YA}(181)=0.65$,



Şekil 2.3 Bulanık Üyelik dereceleri

Şekil 2.3'teki değerlerden yararlanarak bulanık kümeyi sürekli bir fonksiyon şeklinde tanımlamak mümkün olmuştur.

Bu durumda, dilsel değişkenlerin bulanık kümelerle ifadesi Şekil 2. 4'deki gibi yapılabilir.



Şekil 2.4 Çok değişkenli bir bulanık küme

2.1 Dilsel Değişkenlerin Tanımlanması

Dilsel terimlerin sayısı ve isimlerin belirlenmesi yapılır. Genelde 3, 5 veya 7 terim kullanılır.

a- Kontrol stratejisinde kullanılacak kurallardan bazılarının tesbiti,

b- Değişken aralığının dilsel olarak bulunması,

c- Deneyim ; birçok bulanık mantık sisteminin tasarımından sonra seçim kolaylaşır.

Üyelik fonksiyonunun tanımlanması,

a- Terim ismine en uygun temel değişkenin belirlenmesi,

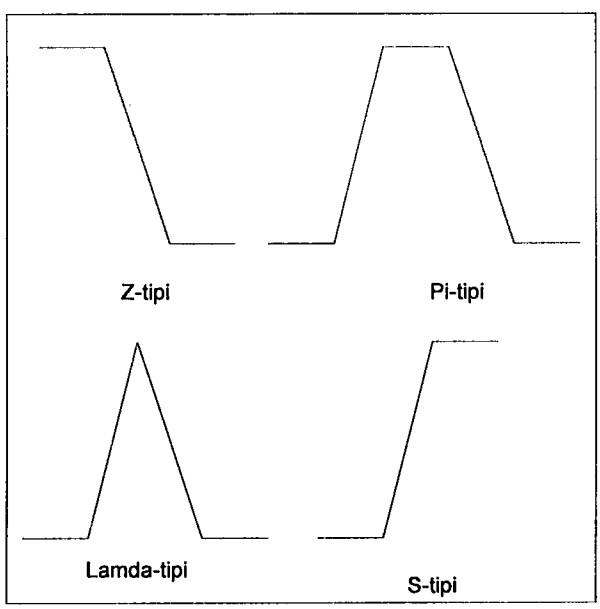
b- Bu değer için $\mu=1$ koyulur,

c- Komşu terimler için $\mu=0$ olacak yerden başlanır,

d- Bu noktalar doğrular ile birleştirilir,

e- Her terim için aynı işlemler tekrarlanır.

2.2 Standart Üyelik Fonksiyonları



Şekil 2.5 Standart Üyelik Fonksiyonları

Standart üyelik fonksiyonları (Şekil 2.5), insanların gerçek değerleri yorumlama şekillerini sadece yaklaşık olarak ifade edebilirler. Gerçekte, psikodilsel çalışmalar üyelik fonksiyonlarının şu özelliklere sahip olmaları gerektiğini ortaya çıkarmıştır.

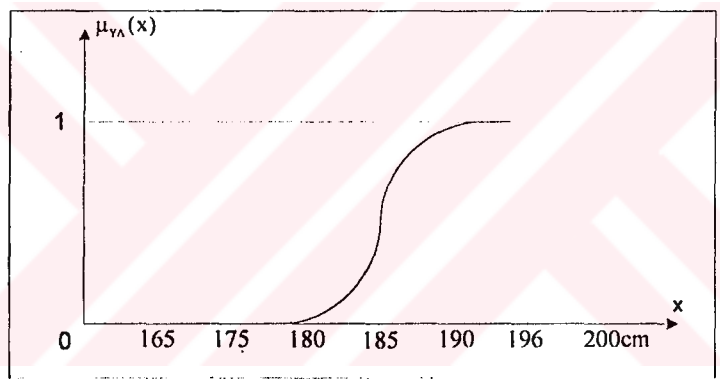
1- $\mu(x)$ X'de süreklidir, yani temel değişkenlerdeki küçük bir değişim, çıkışın bir adım fonksiyonu şeklinde değerlendirilmesine neden olmamalıdır.
2- $d\mu(x)/dx$ X'de süreklidir, aynı kural bu sefer değerlendirilme oranı için mümkündür.
3- $d^2\mu(x)/dx^2$ X'de süreklidir.
4- $\mu(x)$: min $\mu(\max_x(d^2\mu(x)/dx^2))$ eğim değişimi minimum olmalıdır.

μ : üyelik derecesi

$\mu(x)$: üyelik fonksiyonu

x : temel değişkenin tanımlı olduğu söylem evreni

matematiksel olarak, bu dört özelliği aynı anda sağlayabilen fonksiyon "kübik spline" dir.



Şekil 2.6 Kübik Spline üyelik fonksiyonu

Uygulamalar Sistem-tipindeki (kübik spline) üyelik fonksiyonlarının karar verme ve veri analiz sistemlerinde iyi sonuçlar verdiklerini göstermiştir. Ancak kontrol uygulamalarında standart üyelik fonksiyonları yeterli olmaktadır.

2.3 Giriş Değişkenlerinin Bulanıklaştırılması

Eğer bir insanın boyu 184 cm ise ;
çok uzun 0.1
uzun 0.9 olacaktır.

2.4 Kural Kümesinin Tanımı

Bulanık mantık sistemlerinde, kontrol stratejisi "EĞER... O HALDE" kurallarıyla tanımlanır.

EĞER <ön şart> O HALDE <sonuç>

<ön şart> belirli bir durumu tanımlar, <sonuç> ise sistemin bu durum için vermesi gereken tepkiyi kapsar. <ön şart > birden fazla şartı içinde tutabilir. Bu şartlar ve, veya deyimleri ile birleştirilir.

2.5 Bulanık Çıkarım

Bulanık çıkarım iki adımda yapılır :

1- Toplama : Toplama adımında giriş değişkenlerinin bulanıklaştırılmış değerleri ön şart kuralı altında işlenir. Bu sonuç her kuralın o andaki durum için uygulanabilme derecesini belirtir. Bu işlemin "Boole" cebrindeki karşılığı "VE" işlemidir.

2- Derleme : Bu adımda o anki duruma göre yapılacak işlev belirlenir. İşlev, kuralların <sonuç> 'ları olarak tanımlıdır ve kuralı o anki duruma uygulanabilirliği ne kadar fazla ise sonuç da o kadar fazla etkilenecektir. Bu işlemin "Boole" cebrindeki karşılığı "⇒" dır.

2.6 Durulama

Bulanık çıkarımın sonucu kural sonuçlarının doğruluk dereceleridir. Bulanık mantık sisteminin çıkışının bir gerçek değer olması gerektiğinden durulama (defuzzification) gereklidir. Böyle bir dönüşüm dilsel terimlerin temel değişken ile aralarındaki ilişkiyi göz önüne almalıdır.

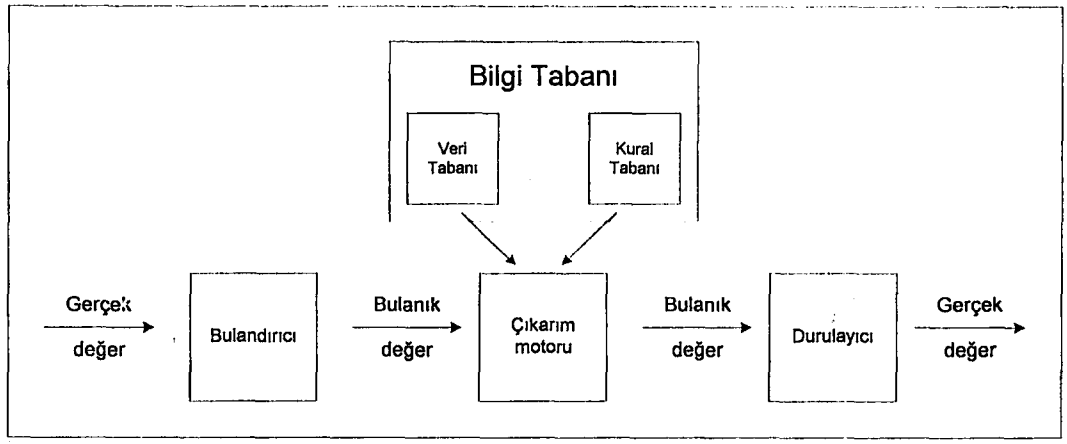
Tüm kuralların sonuçları, karşılıklı terimlerin üyelik fonksiyonlarıyla birebir eşlenirler. Sonuçta elde edilen üyelik fonksiyonları tek bir üyelik fonksiyonlarıyla birebir eşlenirler. Elde edilen tek üyelik fonksiyonu, istenen çıkışı tanımlayan bulanık bir kümedir. Ancak, çıkışta tek bir değer istendiğinden, bu üyelik fonksiyonu tek bir değere sıkıştırılmalıdır.

Durulama işlemi, ağırlık merkezi, maksimumların ortalaması, soldakilerin maksimumları ve sağdakilerin maksimumları kullanılarak yapılabilir.

Sonuç olarak ;

- Bulanık mantık, lineer kontrol modelinin bir üst kümesi gibidir. Bu yüzden, bulanık mantık konvansiyonel kontrolün bir genelleştirilmiş halidir.
- Bulanık mantığın gücü basit şeyleri basit olarak tutmaktır; zaten birçok uygulama karmaşık model temelli çözüm gerektirmemektedir. Bulanık mantık çözümü, daha az bir çaba ile daha iyi iş yapabilir.
- Bulanık mantık, mühendislik deneyimini etkin bir çözüme aktarmanın en kısa yoludur.

2.7 Bulanık Denetleyicinin Yapısı



Şekil 2.7 : Bulanık Denetleyicinin Yapısı

a- **Bulandırıcı :** Bu bölüm giriş değişkenlerini (gerçek sayılan) ölçer, onlar üzerinde bir ölçek değişikliği yapar ve bulanık kümelere dönüştürür. Yani onlara birer etiket vererek dilsel nicelik kazandırır.

b- **Çıkarım Motoru :** Burada kurallar üzerinde bulanık mantık yürütülür yani insan beyninin düşünüş şeklinin bir benzetimi yapılmaya çalışılır.

c- **Veri Tabanı :** Çıkarım motoru, kural tabanında kullanılan bulanık kümelerini bu bölümden alır.

d- **Kural Tabanı :** Kontrol amaçlarına uygun dilsel denetim kuralları burada bulunur ve çıkarım motoruna buradan verilir.

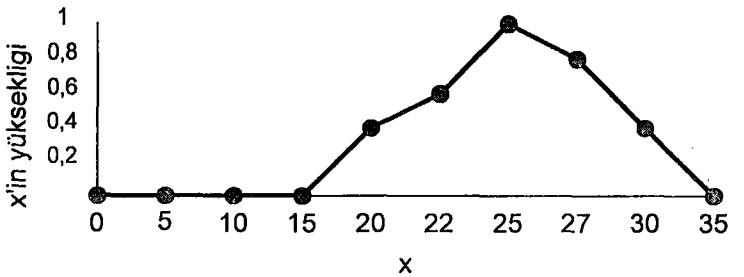
e- **Durulayıcı :** Çıkarım motorunun bulanık küme çıkışları üzerinde gerekli ölçek değişikliklerini yapar ve bunları gerçek sayılara dönüştürür.

Bir bulanık denetleyicinin gerçekleşmesinde, denetlenecek sistemin bir matematiksel modelinden daha çok o sistemi çalıştıran operatörün sistem davranışı konusunda sahip olduğu bilgiler daha önemlidir. Tasarım sırasında genellikle bu tür bilgilerden yararlanılır. Böyle bir yaklaşım, uzun yıllar boyunca kazanılan deneyimlerin denetleyici içerisine, yorumlanmış halde kolaylıkla yerleştirilmesine olanak sağlar. Bu yararın yanında getirdiği sakınca kontrolör tasarımında belirli bir otomasyon elde edilememesidir. Buna rağmen, bulanık kontrolörün en önemli kısmını oluşturan kural tabanının oluşturulması için kullanılabilecek çeşitli yaklaşımlar şunlardır:

- Bir uzmanın bilgi ve/veya deneyimlerinden yararlanmak
- Sürecin bir bulanık modelinin kullanılması
- Operatörün süreç üzerinde yaptığı işlemleri kullanmak
- Öğrenen algoritmaları uygulamak

2.8 Bulanık Küme Teorisinin Bazı Özellikleri

U evrensel kümesinin A bulanık alt kümesi $\mu_A : U \rightarrow [0, 1]$ üyelik fonksiyonuyla tanımlanır. Bu fonksiyon her bir $u \in U$ elemanına [0, 1] aralığında bir değer atar. Bu değer de u elemanının A kümesine ait olma derecesini gösterir (Şekil 2.8). Bu noktadan sonra U evrensel uzayının her noktada tanımlı ve sonlu olduğu kabul edilecektir.



Şekil 2.8. Üyelik fonksiyonu

İki bulanık küme, A ve B için

$\forall u \in U, \mu_A(u) = \mu_B(u)$

sağlandığında bu kümelerin eşitliği söz konusudur ve A=B yazılabilir.

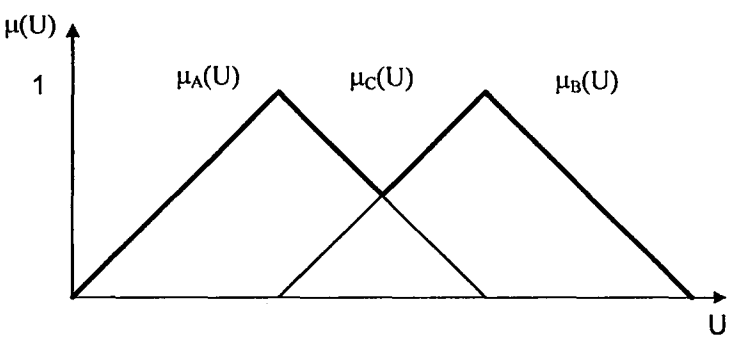
$\forall u \in U, \mu_A(u) \leq \mu_B(u)$

şartı sağlandığında ise A bulanık kümesi (kısaca küme) B kümesinin alt kümesidir ve $A \subseteq B$ şeklinde ifade edilir.

Bulanık kümelerin birleşimi, her bir kümenin tüm elemanlarının *maksimum* işlemine tabi tutulmasıyla elde edilir (Şekil 2.9).

$C = A \cup B$ ise,

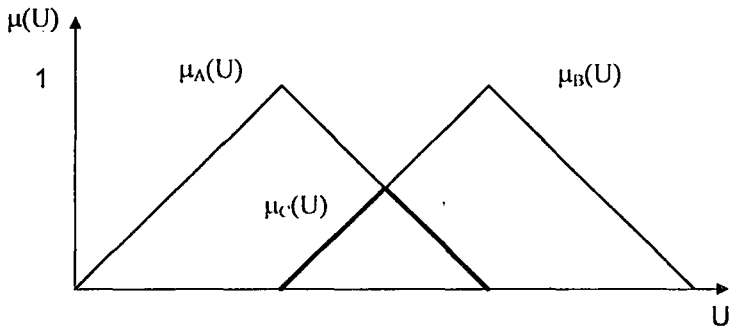
$\forall u \in U, \mu_C(u) = \max(\mu_A(u), \mu_B(u))$



Şekil 2.9. İki bulanık kümenin birleşimi

Bulanık kümelerin kesişimi, her bir kümenin tüm elemanlarının *minimum* işlemine tabi tutulmasıyla elde edilir (Şekil 2.10).

$C = A \cap B$ ise,

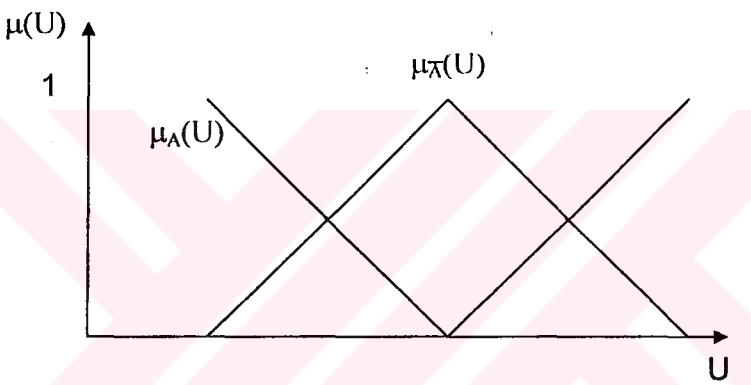


Şekil 2.10. İki bulanık kümenin kesişimi

$\forall u \in U, \mu_C(u) = \min(\mu_A(u), \mu_B(u))$

Bir bulanık A kümesinin tümleyeni $\bar{A}$ aşağıdaki gibi tanımlanır (Şekil 2.11).

$\forall u \in U, \mu_{\bar{A}}(u) = 1 - \mu_A(u)$



Şekil 2.11. Bulanık kümenin tümleyeni

3. AKILLI ETMENLER İÇİN ÖĞRENME

3.1 Çıkartım Sistemleri İçin Öğrenme

Bir sistemin akıllı özelliğini kazanabilmesi için öğrenme işlemini gerçekleştirebilmesi gerekmekte olduğu akıllı etmen tanımının en tabi sonucudur. Bu varsayımdan yola çıkılacak olursa öğrenme, tecrübenin bir sonucu olarak davranışta gözlenen değişiklikdir. Uzun vadeli gerçekler etmenin sembolik yapay zeka formatında ne bildiğini ve kurallar ise bu gerçekler arasındaki ilişkileri tanımlar. Eğer etmen, dünyası değiştiği zaman doğrudan bir insan müdahalesine gerek duyacağına bu bilgiyi kendi kendine geliştirebilir veya en azından genişletebilirse etmene öğrenme yeteneği kazandırılmış olur. Firefly ve OpenSesame dünyalarından öğrenen iki etmen örneğidir[3]. Firefly müzik çeşitlerini grublandırmak için istatistiksel teknikler, OpenSesame ise davranış modellerini tanımlamak için yapay sinir ağlarını kullanır. Arama makinelerinden Yahoo ve AltaVista web'i gezerken yeni linklerle karşılaştığı zaman yeni anahtar kelimeler öğrenir. Bu anahtar kelimeler arama makinelerinin sözcüklerine ve daha sonra da arama paneline eklenir. Bu şekilde kullanıcıların gerektiğinde bu kelimelere göre de arama yapılabilmesine imkan verir. Böyle etmenlere öğrenen veya adaptif etmenler denir. Kural tabanlı etmenlerde öğrenme, etmenin bir veya birkaç yoldan kural tabanı ve uzun vadeli gerçekleri kendiliğinden değiştirebilme yeteneğidir. Bu yollar:

1. Yeni Kurallar Ekleme veya Eski Olanları Değiştirme:

Eğer etmen arzu edilen yeni bir davranışı tanımlayabilirse, yeni bir kural önerebilir veya en azından arzu edilen yeni davranış hakkındaki inancını kullanıcı veya geliştiricinin dikkatine çekebilir. Daha önce ifade edildiği gibi bu işi kendiliğinden yapmak zordur. Buradaki zorluk kuralların yazılmasından değil, mantıksal olarak uyumlu kuralları bulmaktan gelmektedir. Mevcut kuralları değiştirme, her zaman için sağlanılan veya asla sağlanılmayan yüklemleri silme veya eklemeyi ihtiva edebilir. Bu kural optimizasyonunun bir formudur. Buradaki ilginç problem kuralların nasıl değiştirileceği değil, arzu edilen yeni davranışın nasıl tanımlanacağıdır.

2. Yeni Gerçekler Ekleme veya Eski Olanları Değiştirme:

Yeni uzun vadeli gerçekleri tanımlama yeteneği, etmenin kavram aralığını genişlettiğinden yeni bir kuralı oluşturmaktan daha zordur. Örneğin, bir etmen kullanıcısının yeni bir projeye başlaması nedeniyle yeni bir çalışma ilişkisi biçimlendirdiği farz edilsin. Etmen, elektronik mektup için önemli yeni bir gönderici ve alıcının olduğunu tanımlayabilir, fakat bu kişinin proje maliye müdürü görevi üstlendiğini tanımlayamaz (Eğer maliye müdürlerini önceden tanımıyorsa). Kendiliğinden yeni kavramlar tanımak yapay zekada çok ileri bir problemdir.

Mevcut gerçekleri değiştirmek mümkündür. Örneğin, eğer kullanıcı yeni bir göndericiden maliye konuları üzerine birçok mektup almaya başlarsa, etmen bu göndericiyi yeni bir maliye müdürü olarak tanıyabilir (Maliye müdürü kavramına sahip olduğu kabul edilerek). Yine burada da gerçek problem değişikliğin nasıl yapılacağı değil, yeni gerçeklerin kendiliğinden nasıl tanımlanacağıdır.

3. Güvenirlik Katsayısının Değiştirilmesi:

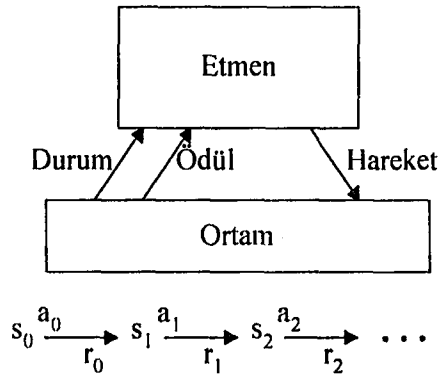
Bir olasılık muhakeme sisteminde, etmen için, verilen bir inançta güvenirlik derecesini belirleyen katsayıyı değiştirmek çok kolaydır ve bu işlem yeni kuralları tanıtmaktan daha güvenli olmalıdır.

3.2 Takviyeli Öğrenme

Takviyeli öğrenme, kendi ortamında algılama ve hareket yapan bir özerk etmenin amacını gerçekleştirmek için en uygun hareketleri yapmayı nasıl öğrenebileceği sorusuna cevap verir. Bu çok genel problem, bir hareketli robotun, kontrol etmeyi öğrenme, fabrikalardaki işlemleri optimize etmeyi öğrenme ve oyun oynamayı öğrenme gibi işleri kapsar. Etmen ortamında bir hareket yaptığı zaman, bir eğitici ortaya çıkan durumun istenilebilirliğini göstermek için bir ödül veya ceza sağlar. Mesela bir oyunu oynama etmeni eğitildiği zaman, eğitici oyun kazanıldığında pozitif bir ödül, kaybedildiğinde negatif bir ödül ve diğer durumlarda sıfır ödül sağlayabilir. Etmenin amacı, en büyük toplam ödülü üreten hareketlerin sırasını öğrenmektir.

Yukarıdaki amaca yönelik bir robotun inşa edilmesi istenirse, robot veya etmen, ortamının durumunu gözlemek için sensörlere ve bu durumu değiştirmek için de yapabileceği hareketlerin bir kümesine sahip olmalıdır. Örneğin hareketli robot, bir kamera veya sonarlar gibi sensörlere ve ileri git ve dön gibi hareketlere sahip olabilir. Onun görevi, amacını gerçekleştirecek hareketlerin seçimi için bir kontrol stratejisi veya politikasını öğrenmektir. Mesela, robotun batarya seviyesi ne zaman düşerse, kendi şarj edicisine gitme gibi bir amacı olabilir. Etmen amacı, etmenin her bir farklı durumdan çıkarabileceği her bir farklı harekete sayısal bir değer atayan (anlık ödül) bir ödül fonksiyonuyla tanımlanabilir. Örneğin, bir şarj ediciye yaklaşma amacı, şarj ediciye derhal bağlanma sonucunu doğuran durum-hareket geçişine pozitif bir ödül atanarak (100 gibi) sağlanabilir. Bu durumda diğer bütün durum-hareket geçişlerine sıfır ödülü atanır. Bu ödül fonksiyonu robot içinde inşa edilebilir veya sadece robot tarafından yapılan her bir harekete ödül veren harici bir öğretmen tarafından da bilinebilir. Robotun işi, sıralı hareketleri yapmak, onların sonuçlarını gözlemek ve kontrol politikasını öğrenmektir. İstenilen kontrol politikası, herhangi bir başlangıç durumundan etmen

tarafından zamanla toplanılan ödülü maksimize eden hareketleri seçmektir. Robot öğrenmesi için bu genel ortam şekil 3.1'de gösterilmiştir.



Şekil 3.1: Ortamıyla etkileşen etmen.

Şekil 3.1'de görüldüğü gibi, toplam ödülü maksimize etmek için bir kontrol politikasını öğrenme problemi çok geneldir ve robot öğrenme işleri ötesinde bir çok problemi kapsar. Problem genelde, süreçlerin sırasını bulmak ile uğraşır. Örneğin, üretim optimizasyon problemlerinde, öyle üretim hareketleri seçilmelidir ki maksimize edilecek ödül, üretilen malın satış değerinden mal olma maliyetinin çıkarılma değeridir. Başka bir örnek olarak da planlama problemleri gösterilebilir. Büyük bir şehirdeki yolculara hangi taksilerin gönderileceğini seçmede maksimize edilecek ödül, yolcuların bekleme zamanına ve hareket halinde iken taksilerin harcayacakları toplam benzin miktarına bağlıdır. Amaç, verilen bir hareketin sırasının kalitesini tanımlamaktır. İstenilen duruma erişmek için arama yöntemini kullanan bir sistem her bir adımda alternatif hareketler arasından bir seçim yapar. Takviyeli öğrenmede ise hareketlerin belirsiz sonuçlar çıkarabileceği ve öğrenicinin kendi hareketlerinin sonuçlarını tanımlayan bir alan (domain) bilgisine sahip olmadığı göz önüne alınır. Takviyeli öğrenme ile yapılan uygulamaların belki de en tanınmışı Tesauro'nun (1995) TD-GAMMON oyun programıdır. Bu program birinci sınıf tavla oyuncusu olmak için takviyeli öğrenmeyi kullanmıştır.

Uygun hareketleri seçmek için bir kontrol politikasını öğrenme problemi, fonksiyon yaklaşım problemlerine benzerlik gösterir. Öğrenilmesi gereken hedef fonksiyon Π: S $\rightarrow$A kontrol politikasıdır. Bir başka deyişle S kümesinden şu andaki s durumu verildikten sonra A kümesinden uygun bir hareket a'yı çıkarma politikasıdır. Bununla birlikte takviyeli öğrenme birkaç yönden diğer fonksiyon yaklaşım yöntemlerinden ayrılır (Mitchell, 1997):

1. Gecikmiş Ödül: Etmenin amacı, şu andaki s durumundan, optimal hareket a= Π(s)'i planlayan bir hedef fonksiyon öğrenmektir. Diğer bir çok öğrenme yöntemlerinde Π gibi bir

hedef fonksiyon öğrenildiği zaman her bir eğitme örneği (s, Π(s)) şeklindedir. Takviyeli öğrenmede ise eğitme bilgisi bu formda değildir. Bunun yerine eğitici; etmen hareketlerini yürütürken sadece anlık ödül değerlerin sırasını sağlar.

2. Kısmen Gözlenebilir Durumlar: Her ne kadar herhangi bir zaman adımında etmen sensörlerinin ortamın bütün durumunu algılayabildiğini kabul etmek uygun olsa da, bir çok pratiksel durumda sensörler yeterince bilgi sağlayamaz. Örneğin, sadece önünü görmeye yarayan kameraya sahip bir robot arkasında ne olduğunu idrak edemez. Böyle durumlarda hareketler seçildiği zaman şu andaki gözlemlerle birlikte öncekileri de dikkate almak etmen için gerekli olabilir.

3. Hayat Boyu Öğrenme: Birbirinden izole edilmiş fonksiyon yaklaşımından farklı olarak, robot öğrenme aynı sensörler kullanarak aynı ortamda birden fazla ilgili işi öğrenmeyi ihtiva edebilir. Örneğin, hareketli bir robotun aynı anda, şarj edicisine yaklaşma, dar koridorlar boyunca gezinme ve lazer yazıcıdan çıktıları toplama gibi öğrenme amaçları olabilir.

3.2.1 Öğrenme İşlemi

Sıralı kontrol yöntemlerini daha kesin bir ifadeyle formülize etmek için bir takım kabullerin yapılması gereklidir. Örneğin etmen hareketlerinin belirli veya belirsiz oldukları düşünülebilir. Aynı şekilde, etmen bir hareketden sonra meydana gelecek durumu tahmin edebilir. Diğer bir taraftan etmen, ona optimum hareketlerin sırasını gösteren bir uzman tarafından eğitilebilir veya hareketlerin seçimi tamamen etmene bırakılabilir.

Bu bölümde öncelikle Markov Karar Süreç (MKS) tabanlı problemlerin öğrenilmesi üzerinde durulmuştur. MKS'de bir etmen, ortamındaki farklı durumlar kümesi S'i algılayabilir ve yapabileceği A hareketlerinden birini uygulayabilir. Her bir ayrık zaman adımı t'de etmen şu andaki durum s_t 'yi algılar ve ona uygun a_t hareketini yürütür. Ortam etmene anlık ödül $r_t=r(s_t, a_t)$'ı vermekten ve bir sonraki durum $s_{t+1} = \delta(s_t, a_t)$'yi üretmekten sorumludur. Burada δ ve r fonksiyonları ortamın parçasıdır ve etmenin bilmesine gerek yoktur. Bir MDP'de $\delta(s_t, a_t)$ ve $r(s_t, a_t)$ fonksiyonları sadece şu andaki durum ve harekete bağlıdır, daha öncekilere bağlı değildir. Burada S ve A'nın sonlu bir kümeye sahip olduğu göz önüne alınmıştır.

Etmenin amacı, şu anda gözlenen durum s_t 'ye göre bir sonraki a_t hareketini seçmek için $\Pi: S \rightarrow A$ politikasını, yani $\Pi(s_t)=a_t$ 'ı öğrenmektir. Π politikası etmenin zamanla en büyük toplam ödülü elde etmesiyle öğrenilebilir. Bunun için rasgele bir başlangıç durumu s_t'den rasgele bir politika takip edilerek elde edilen toplam ödül $V^{\Pi}(s_t)$ ile tanımlanabilir. Bu durum aşağıda verilmiştir.

$$V^{\Pi}(s_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$$
$$\equiv \sum_{i=0}^{\infty} \gamma^i r_{t+i} \qquad (1)$$

Burada ödüllerin sırası r_{t+i} , s_t durumundan başlanarak ve yukarıda tanımlandığı gibi, yani $a_t=\Pi(s_t)$, $a_{t+1}=\Pi(s_{t+1})$ ile hareketleri seçmede kullanılan Π politikası sayesinde üretilir. γ ise $0 \leq \gamma < 1$ arasında bir değer olup anlık ödüllere karşılık geciktirilmiş relatif değeri belirler. Özellikle gelecek i zaman adımında alınan ödüller γ^i faktörüyle üstel bir şekilde azaltılır. Eğer $\gamma=0$ olursa, sadece anlık ödül göz önüne alınır. Eğer γ bire yaklaşırsa geleceğin ödüllerine anlık ödüle göre daha büyük önem verilir.

$V^{\Pi}(s)$, başlangıç durumu s'den bir Π politikasıyla öğrenilen "azaltılmış toplam ödül" olarak adlandırılır. Gelecek ödülleri anlık ödüllere göre azaltmak mantıklıdır. Çünkü bir durumda daha sonrakinden ziyade daha yakın olan ödül tercih edilir. Bununla birlikte toplam ödülün başka tanımları da vardır. Örneğin, sonlu çevre ödülü;

$\sum_{i=0}^{h} r_{t+i}$,h adımı üzerinde azaltılmamış toplam ödülleri gösterir. Ortalama ödül ise;

$\lim_{h\to\infty} \frac{1}{h} \sum_{i=0}^{h} r_{t+i}$ etmenin bütün yaşamı üzerinde zaman adımı başına aldığı ödüldür.

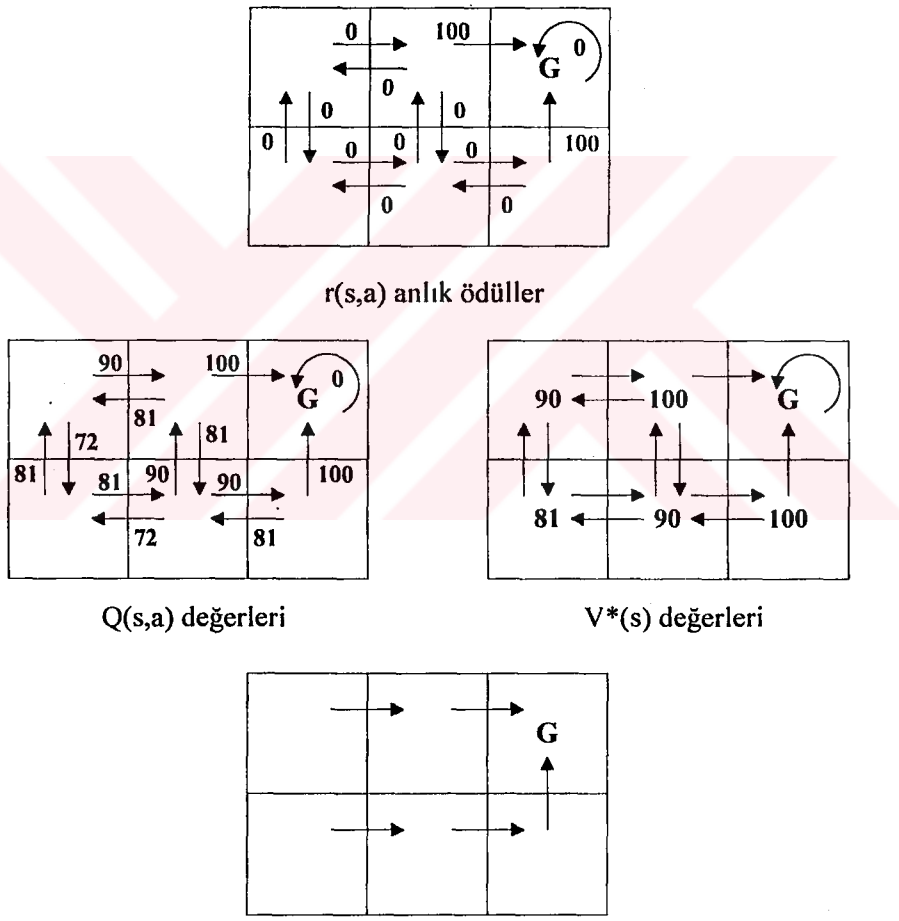
Artık bundan sonra etmenin amacı bütün s durumları için toplam ödülü maksimize eden bir $V^{\Pi}(s)$ politikası öğrenmektir. Bu politikaya "optimal politika" denir ve Π^* ile gösterilir.

$$\Pi^* \equiv \arg\max_{\Pi} V^{\Pi}(s) \qquad (2)$$

Notasyonun basitleştirilmesi için böyle bir optimal politikanın değer fonsiyonu $V^{\Pi^*}(s)$ yerine, $V^*(s)$ kullanılır. $V^*(s)$ etmenin s durumundan başlayarak elde edebileceği maksimum azaltılmış toplam ödülü, yani s durumundan başlayıp optimal politika takip edilerek elde edilen azaltılmış toplam ödülü gösterir.

Bu kavramların daha iyi açıklanması için aşağıda bir örnek verilmiştir (grid-world ortamı). Şekil 3.2'nin en üstteki diyagramında 6 bölme, etmen için 6 mümkün durumu veya yeri temsil eder. Diyagramdaki her bir ok etmenin bir durumdan diğerine geçebileceği mümkün hareketi temsil eder. Her bir okla ilişkilendirilmiş sayı, etmenin ilgili durum-hareket geçişini yürütmesi durumunda alacağı anlık ödül r(s, a)'yı gösterir. Bu özel ortamda anlık ödül, G durumuna geçme dışındaki bütün durum-hareket geçişleri için sıfırdır. G amaç durumudur (goal state). Etmen G durumuna girdikten sonra yapabileceği tek hareket bu durumda kalmaktır. Bu nedenle G yutucu durum (absorbing state) olarak adlandırılabilir.

Durumlar, hareketler, anlık ödüller tanımlandıktan ve azalma faktörü γ seçildikten sonra, optimal politika Π^* ve onun değer fonksiyonu $V^*(s)$ belirlenebilir. Örnek için $\gamma=0.9$ olarak alınmıştır. Şekil 3.2'ün en alttaki diyagramı bu ortam için bir optimal politikayı gösterir, fakat bundan başka optimal politikalar da vardır. Tüm politikalar gibi optimal politika da etmenin herhangi bir durumda seçeceği her hareketi belirleyebilir. Aynı zamanda optimal politika, etmeni G duruma en kısa yoldan ulaştırır. Şeklin ortadaki diyagram her bir durum için V^* değerlerini gösterir. Örneğin, bu diyagramda en alttaki sağ durum göz önüne alındığında bu durum için V^* değeri 100'dür. Çünkü bu durumda optimal politika, 100 anlık değeri olan yukarıya git hareketini seçer. Bundan sonra etmen yutucu durumda kalacak ve herhangi bir ödül



Şekil 3.2: Ayrık durumlardan meydana gelen belirli bir etmen ortamı.

almayacaktır. Benzer şekilde en altdaki orta durum için V^* değeri 90'dır. Çünkü optimal politika etmeni bu durumdan önce sağa (sıfır değerinde bir anlık ödül üreterek) sonra da yukarı doğru (100 anlık değeri üreterek) hareket ettirecektir. Böylece en alttaki orta durum için azaltılmış gelecek ödül;

$$0 + \gamma 100 + \gamma^2 0 + \gamma^3 0 + ... = 90 \text{ dır.}$$

Bu özel ortamda etmen yutucu durum G'ye ulaştıktan sonra onun sonsuz geleceği artık bu durumda kalmaktan ve sıfır ödüller almaktan ibarettir.

3.2.2 Q Öğrenme

Q Öğrenmenin amacı, rasgele bir ortam içinde etmene optimal politikayı nasıl öğrenebileceğini göstermektir. Fonksiyon $\Pi^*: S \rightarrow A$ 'yı doğrudan öğrenmek zordur. Çünkü kullanılabilir eğitme verileri (s, a) formundaki eğitme örneklerini sağlamaz. Bunun yerine öğrenici için kullanılabilir eğitme bilgisi sadece anlık ödüller $r(s_i, a_i)$'in sırasıdır. Bu çeşit eğitme bilgisi verildikten sonra, durumlar ve hareketlerin tanımladığı bir sayısal değerlendirme fonksiyonunu öğrenmek ve sonra bu değerlendirme fonksiyonuna göre optimal politikayı gerçekleştirmek daha kolaydır.

Birden fazla değerlendirme fonksiyonları içinde etmen V^*'ı öğrenmeye çalışmalıdır. Örneğin, $V^*(s_1) > V^*(s_2)$ ise etmen s_2'nin yerine s_1'i öğrenmeyi tercih etmelidir. Etmen politikası durumlar yerine hareketler arasında seçim yapmalıdır. s durumunda optimal hareket; anlık ödül r(s, a) ile γ değeriyle azaltılmış V^* 'nın toplamını maksimize eden a hareketidir. Yani;

$$\Pi^*(s) = \arg\max_a [r(s,a) + \gamma V^*(\delta(s,a))] \qquad (3)$$

δ(s, a), s durumuna a hareketi uygulandıktan sonra meydana gelen durumu gösterir. Böylece etmen, eğer mükemmel bir anlık ödül fonksiyonu r ve durum geçiş fonksiyonu δ bilgisine sahip ise V^* 'yı öğrenerek optimal politikayı kazanabilir. Yani etmen, hareketine cevap olarak ortam tarafından kullanılan r ve δ fonksiyonlarını bilirse, herhangi bir s durumu için optimal hareketi (3) nolu denklemden hesaplayabilir. Fakat etmenin her durum-hareket geçişi için anlık sonuçları (anlık ödül ve sonraki durum) tahmin edebilmesi mümkün değildir. Bir başka deyişle; robot kontrolü gibi bir çok pratiksel problemlerde rasgele bir duruma rasgele hareket uygulayarak tam sonucu önceden tahmin etmek, etmen veya onun programcısı için

mümkün değildir. δ ve r'nin bilinmediği durumlarda etmen (3) nolu denklemi değerlendiremediğinden V^* öğrenilemez.

3.2.2.1 Q fonksiyonu

Bir değerlendirme fonksiyonu Q(s, a) tanımlansın. Bu şekilde onun değeri s durumundan başlayarak ve ilk hareket olarak a'yı uygulayarak başarılabilen maksimum azaltılmış toplam ödüldür. Diğer bir deyişle, Q'nun değeri; s durumunda a hareketi yürütülerek anlık olarak alınan ödül ile bundan sonraki durumun optimal politikayı takip etme değerinin γ ile azaltılmış toplamıdır. Yani;

$$Q(s,a) \equiv r(s,a) + \gamma V^*(\delta(s,a)) \qquad (4)$$

Q(s, a), s durumunda optimal hareket a'yı seçmek için (3)'deki denklemin maksimize edilen terimidir. Bundan dolayı (3) nolu denklem Q(s, a)'ya göre yeniden yazılabilir.

$$\Pi^*(s) = \arg\max_a Q(s,a) \qquad (5)$$

Bu denklem gösterir ki eğer etmen, V^* fonksiyonu yerine Q fonksiyonunu öğrenirse r ve δ fonksiyon bilgilerine sahip olmasa bile optimal hareketleri seçebilir. Bu denklem ile s durumunda sadece Q(s, a)'yı maksimize edecek a hareketleri göz önüne alınır. Q öğrenme algoritmasının olumlu bir yönü; şu andaki durum ve hareket için Q değeri, eğer s durumunda a hareketi seçilirse gelecekte azaltılmış toplam ödülü belirlemek için gerekli bütün bilgiyi tek bir sayıda toplayabilir. Bu durum şekil 3.2 deki örnekten de görülmektedir. Dikkat edilirse her bir durum hareket geçişleri için Q değerleri; r değeri ile V^*'nın γ ile azaltılmış değer toplamıdır. Ayrıca şekilde gösterilen optimal politika, maksimum Q değerli hareketleri seçmede de uygunluk gösterir.

3.2.2.2 Q Öğrenme için bir algoritma

Q fonksiyonunu öğrenme, optimal politikayı öğrenmeye benzer. Burada anahtar problem, sadece anlık ödüller r'nin zamanla sırası verildikten sonra Q'nun eğitme değerlerini belirleyen yolu bulmaktır. Bu iteratif bir yaklaşımla başarılabilir. Bunun için aşağıda Q ve V^* arasında bir ilişki verilmiştir.

$V^*(s) = \max_{a'} Q(s,a')$ Bu durumda (4) nolu denklem yeniden yazılırsa;

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(\delta(s,a),a') \qquad (6)$$

Q'nun bu özyinelemeli tanımı iteratif olarak Q'ya yaklaşan algoritmaların temelini oluşturur (Watkins, 1989). Öğrenici, gerçek Q fonksiyonunu tahmin edebilmesi için $\hat{Q}$ sembolünü kullanır. Bu sembol bir anlamda öğrenicinin Q hakkındaki hipotezidir. $\hat{Q}$, her bir durum hareket çifti için ayrı bir girişi olan tablo ile gösterilirse, her (s, a) çifti için tablo da $\hat{Q}(s,a)$ değeri vardır. $\hat{Q}(s,a)$ değeri öğrenicinin gerçek Q(s, a) değeri için şu andaki tahminini gösterir. Tablo başlangıç olarak rasgele değerlerle doldurulabilir. Fakat başlangıç değerleri olarak sıfır alınırsa algoritmayı anlamak daha kolay hale gelir.

Etmen şu andaki durum s'i sürekli olarak gözler, bir a hareketini seçer ve bu hareketi yürütür. Bu yapılırken ortaya çıkan ödül r=r(s, a) ve yeni durum $s' = \delta(s, a)$'i bir sonraki hareket için gözler. Daha sonra her bir geçiş için $\hat{Q}(s,a)$'yı aşağıdaki eğitme kuralına göre güncelleştirir.

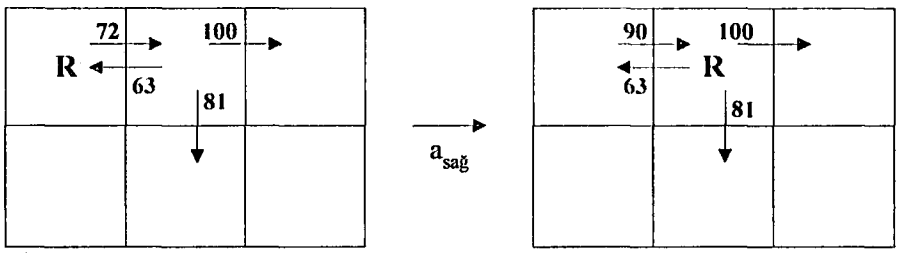
$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a') \qquad (7)$$

Bu eğitme kuralı her ne kadar $\hat{Q}$ değerini kullansa da (6) nolu denlemden çıkarılmıştır. (6) nolu denklemde Q, δ(s, a) ve r(s, a)'ye göre tanımlansa da (7) deki denklemde eğitme kuralının bu fonksiyonları bilmesine gerek yoktur. Bunun yerine etmen ortamındaki hareketi yürüttükten sonra meydana gelen yeni durum s' ve r ödülünü gözler ve daha sonra buna göre hareket eder. Aşağıdaki tabloda belirli bir Markov karar süreci (MKS) için Q öğrenme algoritması verilmiştir. Bu algoritma kullanılarak etmenin tahmini $\hat{Q}$'su sonuçta gerçek Q fonksiyonuna yaklaşır.

- Her bir s, a için $\hat{Q}(s,a)$ 'yı sıfıra ata.
- Şu andaki s durumunu gözle
- Öğrenme gerçekleşinceye kadar tekrarla:
- Bir a hareketini seç ve onu yürüt,
- Anlık ödül r'yi al,
- Yeni durum s' durumunu gözle,
- $\hat{Q}(s,a)$ değerini yandaki kurala göre güncelleştir: $\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')$
- $s \leftarrow s'$

Q öğrenme algoritmasının çalışmasını anlamak için aşağıda bir örnek verilmiştir. Bu örnekte etmen ızgara dünyasında bir hücre sağa doğru hareket eder ve bu geçiş için sıfır anlık ödülü alır. Sonra, henüz yürütülen durum-hareket geçişi için tahmini $\hat{Q}$, (7) nolu denklemdeki eğitme kuralına göre hesaplanır. Eğitme kuralına göre bu geçiş için yeni $\hat{Q}$ tahmini, alınan anlık ödül (sıfır) ile sonraki durumun en yüksek $\hat{Q}$'sunun γ (0.9) ile azaltılmış değerinin toplamıdır. Şekil 3.2 de gösterilen bu ortamın anlık ödül değerleri ise şekil 3.3'deki gibi kabul edilsin. Bu durumda anlık ödül, amaç duruma girilmesi haricinde her yerde sıfırdır. Bu ortam yutucu bir durum içerdiğinden dolayı eğitme bölümlerden ibaret olarak kabul edilebilir. Her bir bölüm boyunca etmen rasgele seçilen bir durumdan başlar ve yutucu duruma ulaşıncaya kadar hareketlerin yürütülmesine müsaade edilir. Amaç duruma ulaşıldığı zaman bölüm sonlanır ve etmen bir sonraki bölüm için rasgele seçilen yeni bir başlangıç duruma aktarılır.

Q öğrenme algoritması sıfır ile yüklenmiş bütün $\hat{Q}$ değerlerinde, etmen amaç duruma erişinceye kadar veya sıfır olmayan bir ödül alana kadar $\hat{Q}$ tablo girişi için herhangi bir değişiklik yapmayacaktır. Fakat amaç durumdan bir önceki durum için $\hat{Q}$ değeri artık sıfırdan farklı olacaktır. Bir sonraki bölümde ise eğer etmen amaç duruma komşu olan duruma geçerse aynı şekilde bu durum için sıfırdan farklı bir $\hat{Q}$ değeri elde edilmiş olur. Algoritma bu şekilde



Başlangıç Durumu: s_1 Sonraki Durum: s_2

Şekil 3.3: Bir hareket yürütüldükten sonra $\hat{Q}$ değerinin güncelleştirilmesi

bölümlerin istenilen değere kadar ilerledikten sonra, etmen bütün durum hareket geçişleri için sıfırdan farklı $\hat{Q}$ değerleri üretecektir. Örnek olarak şekil 3.3'de s_1 'den s_2 duruma $a_{sağ}$ hareketi uygulanırsa aşağıdaki $\hat{Q}$ değeri elde edilir.

$$\hat{Q}(s_1, a_{sağ}) \leftarrow r + \gamma \max_{a'} \hat{Q}(s_2, a')$$
$$\leftarrow 0 + 0.9 \max\{63, 81, 100\}$$
$$\leftarrow 90$$

Sonuç olarak da optimum politikaya yönelmiş Q değerlerini içeren $\hat{Q}$ tablosu elde edilir. Sonraki bölümde de görüleceği gibi belirli kabuller altında algoritmada ifade edilen Q öğrenme algoritması Q fonksiyonuna yakınsayacaktır. Eğer Q öğrenme algoritması ödülleri negatifden farklı belirli bir Markov Karar Süreç işlemi için uygulanacak olursa ve başlangıçta bütün $\hat{Q}$ değerlerine sıfır değeri atanacak olursa iki önemli özellik ortaya çıkar. Bunlardan biri;

$\hat{Q}$ değerleri eğitme boyunca asla azalmaz. Yani; $\hat{Q}_n(s,a)$, eğitme prosedürünün n.'inci iterasyonundan sonraki öğrenilen değeri olarak kabul edilirse (yani etmen tarafından yapılan n.'inci durum hareket geçişinden sonra) o zaman;

$(\forall s, a, n) \quad \hat{Q}_{n+1}(s,a) \geq Q(s,a)$ dir.

İkinci bir özellik ise bütün yer değiştirme sürecine her $\hat{Q}$ değeri, sıfır ile Q değeri arasındadır. Yani;

$(\forall s,a,n) \quad 0 \leq \hat{Q}_n(s,a) \leq Q(s,a)$ dir.

3.2.2.3 Yakınsama

Q değerlerinin bulunması için sunulan algoritma ile gerçek Q fonksiyonuna eşit $\hat{Q}$'ya doğru bir yakınsamanın sağlanabilmesi için belirli şartların kabul edilmesi gerekir. Bu kabullerden biri sistemin belirli MKS'ye sahip olduğu, ikincisi bütün durum ve hareketler için anlık ödüllerin var olduğu ($|r(s, a)| < c$ olmak üzere) ve üçüncüsü ise etmenin her mümkün durum hareket çiftini sonsuz bir şekilde sık ziyaret ettiğidir. Bu üçüncü kabul, eğer a hareketi s durumu için geçerli bir hareket ise, o zaman etmenin a hareketini s durumundan tekrarlı bir şekilde yürütebileceğini ifade eder. Bu şartlar bazı durumlarda genel bazılarında ise oldukça sınırlı olabilir. Öyle ki daha önce verilen örnekten daha genel ortamlar tanımlanabilir. Bu genel ortamlarda ödüllerin değeri pozitif ve negatif olabileceği gibi sıfırdan farklı ödül üreten birden fazla durum hareket geçişi de olabilir. Etmenin sonsuz şekilde sık durum hareket geçişine maruz kalma şartları sınırlıdır. Bu ancak geniş alanlarda kabul edilen bir yaklaşımdır. Yakınsamanın temelini teşkil eden anahtar fikir, en büyük hatalı $\hat{Q}(s,a)$ ne zaman güncelleştirilirse, bir önceki değerinden γ kadar azaltılmış bir hataya sahip olur.

Teorem 1: Belirli bir MKS için Q öğrenmenin yakınsaması.

$(\forall s,a)|\, r(s,a)| \leq c$ ödülleriyle sınırlandırılmış belirli bir MKS'de Q öğrenme algoritmasının kullanıldığı farz edilsin. Q öğrenme algoritması (7) nolu eğitme kuralını kullanır. $\hat{Q}(s,a)$ tablosu ise başlangıç olarak keyfi sonlu değerler ile yüklensin. $\hat{Q}_n(s,a)$, $\hat{Q}(s,a)$'in n.'inci güncelleştirilmiş durumunu gösterirken; eğer her bir durum hareket çifti sonsuz şekilde sık ziyaret edilirse, o zaman $\hat{Q}_n(s,a)$, $n \to \infty$ iken bütün s ve a için $\hat{Q}(s,a)$'ya yaklaşır.

İspatı: Her biri durum hareket geçişi sonsuz şekilde sık meydana geldiğinden, bu geçişin en az bir defa meydana geldiği ardışık taramalar göz önüne alınsın. $\hat{Q}$ tablosundaki bütün girişlerin

maksimum hatası her aralık boyunca en az γ faktörüyle azaltılır ($0 \le \gamma < 1$). $\hat{Q}_n$, n.'inci güncelleştirmeden sonra tahmin edilen Q değeridir. Δ_n, $\hat{Q}_n$'de maksimum hata olsun. Yani;

$$\Delta_n \equiv \max_{s,a} |\hat{Q}_n(s,a) - Q(s,a)|$$

$\delta(s, a)$ yerine s' kullanarak, n+1 iterasyon üzerinde güncelleştirilecek $\hat{Q}_n(s,a)$ tablo girişi için düzeltilen $\hat{Q}_{n+1}(s,a)$ deki hata bulunmak istenirse;

$$|\hat{Q}_{n+1}(s,a) - Q(s,a)| = |(r + \gamma \max_{a'} \hat{Q}_n(s',a')) - (r + \gamma \max_{a'} Q(s',a'))|$$

$$= \gamma |\max_{a'} \hat{Q}_n(s',a') - \max_{a'} Q(s',a')|$$

$$\le \gamma \max_{a'} |\hat{Q}_n(s',a') - Q(s',a')|$$

$$\le \gamma \max_{s''a'} |\hat{Q}_n(s'',a') - Q(s'',a')|$$

$$|\hat{Q}_{n+1}(s,a) - Q(s,a)| \le \gamma \Delta_n$$

Yukarıdaki ikinci ve üçüncü satır arasındaki ilişki aşağıdaki gibidir.

$$|\max_a f_1(a) - \max_a f_2(a)| \le \max_a |f_1(a) - f_2(a)|$$

Üçüncü satırdan dördüncü satıra geçildiğinde ise yeni bir s'' değişkeni tanımlanmıştır. Bu değişken sayesinde Δ_n'in tanımına uygun bir ifade elde edilebilir. Böylece herhangi bir (s, a) için güncelleştirilen $\hat{Q}_{n+1}(s,a)$, $\hat{Q}_n$ tablosunda maksimum hata Δ_n'in en fazla γ katıdır. Başlangıç tablosundaki en geniş hata Δ_0 ile gösterilirse, bütün (s, a)'ların ziyaret edildiği birinci taramadan sonra tablodaki en geniş hata $\gamma \Delta_0$ olacaktır. Bu şekilde k'ıncı taramadan sonra hata

en fazla $\gamma^k \Delta_0$ olur. Her bir durum sonsuz bir şekilde sık ziyaret edildiğinden, $n \to \infty$ iken tarama sayısı sonsuz ve $\Delta_n \to 0$'dır.

3.2.2.4 Deneysel stratejiler

Dikkat edilirse sunulan algoritmada hareketlerin, etmen tarafından nasıl seçildiğini belirlemez. Bunun için gerekli olan strateji, etmen için güncel yaklaşım $\hat{Q}$'yu kullanmak suretiyle $\hat{Q}(s,a)$'yı maksimize eden s durumunda a hareketini seçmektir. Bununla birlikte bu strateji ile etmen, daha yüksek değerli diğer hareketleri keşfetmede başarısız iken yüksek $\hat{Q}$ değerlerine sahip olmak için ilk eğitme sırasında bulunan hareketleri sürekli olarak işleme riskini alır. Gerçekde yukarıdaki yakınsama teoremi, her bir durum hareket geçişinin sonsuz şekilde sık meydana gelmesine bağlıdır. Eğer etmen daima şu andaki $\hat{Q}(s,a)$'yı maksimize eden hareketleri seçerse, açıkca bu durum meydana gelmez. Bu nedenle hareketleri seçmede bir olasılık yaklaşımı kullanmak Q öğrenmede esasdır. Daha yüksek $\hat{Q}$ değerlikli hareketler daha yüksek olasılıklara atanır. Fakat her hareketin de sıfırdan farklı bir olasılığı vardır. Her bir hareketin olasılığı aşağıdaki formülle ifade edilebilir.

$$P(a_i \mid s) = \frac{k^{\hat{Q}(s,a_i)}}{\sum_j k^{\hat{Q}(s,a_j)}}$$

Burada $P(a_i \mid s)$, etmen s durumunda iken a_i hareketini seçme olasılığıdır. k $(k > 0)$ yüksek $\hat{Q}$ değerli hareketi onaylayan seçimin ne kadar güçlü olduğunu belirleyen bir sabittir. k'nın daha büyük değeri $\hat{Q}$'nun ortalaması üzerindeki hareketlere daha yüksek olasılık atar. Etmenin öğrendiğini kullanma ve inandığı hareketleri arama ödülünü maksimize eder. Bunun aksine küçük k değerleri diğer hareketlere daha yüksek olasılık verir. Etmenin şu anda yüksek $\hat{Q}$ değerlerine sahip olmayan hareketleri keşfetmesine imkan sağlar. Bazı durumlarda k iterasyon

sayısıyla değiştirilir. Böylece etmen öğrenmenin ilk safhaları boyunca keşfetme amaçlıdır, daha sonra derece derece kullanma stratejisine doğru kayar.

3.2.3 Belirsiz Ödüller ve Hareketler

Şimdiye kadar Q öğrenme hep belirli ortamlar için göz önüne alındı. Burada ise belirsiz durumlar ele alınacaktır. Ödül fonksiyonu r(s, a) ve hareket geçiş fonksiyonu δ(s, a) olasılık sonuçlarına sahip olabilir. Örneğin, Tesauro'nun (1995) tavla oynama programında, hareket sonuçları doğal şekilde olasılıklıdır. Çünkü herbir hareket, bir çift zarın yuvarlanmasını içerir. Benzer biçimde gürültü sensörlü ve effektörlü robot problemlerinde hareket ve ödülleri belirsiz olarak modellemek çoğu zaman uygundur. Böyle durumlarda δ(s, a) ve r(s, a) fonksiyonlarına önce s ve a'ya dayandırılmış sonuçlar üzerinde bir olasılık dağılımı üretmek ve sonra bu dağılımdan bir sonuç çıkartmak gerekir. Bu olasılık dağılımları sadece s ve a'ya bağımlı (yani önceki durum ve harekete bağlı değillerdir) olurlarsa, böyle sistemler belirsiz Markov karar süreçleri (MKS) olarak adlandırılırlar.

Belirsiz durumlarda, hareketlerin sonuçları artık belirli değildir gerçeği göz önüne alınarak öğrenicinin amacı yeniden ifade edilmelidir. Π politikası takip edilerek elde edilen azaltılmış toplam ödülün beklenti değeri (belirsiz sonuçlar için) aşağıdaki gibi ifade edilebilir.

$$V^{\Pi}(s_t) \equiv E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right]$$

Burada r_{t+i} ödüllerinin sırası s durumundan başlayıp Π politikası takip edilerek üretilir. Dikkat edilirse bu (1) nolu denklemin genelleştirilmiş halidir. Daha önceki gibi optimal politika Π^*, bütün s durumları için $V^{\Pi}(s)$'i maksimize eden Π politikası olarak tanımlanır. (4) nolu denklemin beklenti değeri alınarak, Q tekrar ifade edilmek istenirse;

$$\begin{aligned} Q(s,a) &\equiv E[r(s,a) + \gamma V^*(\delta(s,a))] \\ &= E[r(s,a)] + \gamma E[V^*(\delta(s,a))] \\ &= E[r(s,a)] + \gamma \sum_{s'} P(s' \mid s,a) V^*(s') \end{aligned} \qquad (8)$$

Burada P(s' |s, a), s durumunda a hareketi uygulanarak s' durumunu üretme olasılığıdır. Buradan Q aşağıdaki gibi recursive olarak tekrar yazılabilir.

$$Q(s,a) = E[r(s,a)] + \gamma \sum_{s'} P(s' \mid s,a) \max_{a'} Q(s',a') \qquad (9)$$

Bu denklem (6) nolu denklemin genelleştirilmiş halidir. Daha önce belirli durum için çıkartılan eğitme kuralı, belirsiz ortamın yakınsamasında başarısız kalır. Örneğin, (s, a) geçişinin tekrarlandığı her bir zaman farklı ödüller üreten bir belirsiz ödül fonksiyonu r(s, a) göz önüne alındığında, eğitme kuralı; $\hat{Q}$ tablo değerlerini uygun Q fonksiyonuyla yüklese bile $\hat{Q}(s,a)$ değerlerini sürekli olarak değiştirecektir. Kısaca bu eğitme kuralı yakınsama sağlamayacaktır. Bu zorluğun üstesinden eğitme kuralı değiştirilerek gelinebilir. Düzeltilmiş eğitme kuralı, $\hat{Q}$'nun Q'ya yakınsamasını temin etmede yeterlidir.

$$\hat{Q}_n(s,a) \leftarrow (1-\alpha_n)\hat{Q}_{n-1}(s,a) + \alpha_n[r + \max_{a'}\hat{Q}_{n-1}(s,a)] \quad (10)$$

Burada; $\alpha_n = \frac{1}{1+\text{visits}_n(s,a)}$ dir. (11)

s ve a n'inci iterasyon boyunca güncelleştirilen durum ve hareketlerdir. $\text{visits}_n(s, a)$ ise n'inci iterasyonu da içeren durum hareket çiftinin ziyaret edilme sayısıdır. Düzeltilmiş kuralda anahtar fikir, $\hat{Q}$ revizyonlarının belirli durumdan daha derecesel olarak yapıldığıdır.

Eğer α_n 1'e set edilirse, belirli durum için eğitme kuralı elde edilir. α_n değeri, n artar iken düşer. Böylece güncellemeler, eğitme ilerlerken daha küçük olur. Eğitme boyunca uygun bir oranda α_n azaltılarak Q fonksiyonuna yakınsama sağlanabilir. Yukarıda verilen α_n'in seçimi Watkins ve Dayan'ın (1992) aşağıdaki teoremine göre yakınsama şartlarını sağlayan yöntemlerinden biridir.

Teorem 2: Belirsiz bir MKS için Q öğrenmenin yakınsaması.

$(\forall s,a)|\, r(s,a)\,| \leq c$ ödülleriyle sınırlandırılmış belirsiz bir MKS'de Q öğrenme algoritmasının kullanıldığı farz edilsin. Q öğrenme algoritması (10) nolu eğitme kuralını kullanır. $\hat{Q}(s,a)$ tablosu ise başlangıç olarak keyfi sonlu değerler ile yüklensin ve azalma faktörü de $0 \leq \gamma < 1$ olsun. n(i, s, a), a hareketinin s durumuna i defa uygulanma sayısını da gösteren bir iterasyon bilgisi olsun.

Eğer her bir durum hareket çifti sonsuz şekilde sık ziyaret edilirse, $0 \leq \alpha_n < 1$ ve;

$$\sum_{i=0}^{\infty}\alpha_n(i,s,a) = \infty, \quad \sum_{i=0}^{\infty}[\alpha_n(i,s,a)]^2 < \infty \text{ ise,}$$

o zaman $n \to \infty$ iken 1 olasılıkla $\hat{Q}_n(s,a) \to Q(s,a)$ dir[20].

4. ÇOKLU ETMENLİ ORTAMLARDA GELİŞTİRDİĞİMİZ ÖĞRENME METOTLARI

Çoklu etmenli sistemler ayrık yapay zeka sistemlerinin önemli bir yapısını oluşturmaktadır. Belirli bir amacı gerçekleştirmek için programlanan etmenler ortamda tek başına bulunmayacakları için etmenlerin birlikte ve koordineli bir şekilde hareket etmeleri tek başına hareket etmelerinden daha verimli olacağı açıktır. Bu nedenle günümüzde yapay zeka sistemleri için önerilen tüm öğrenme metotlarının çoklu etmenli sistemlere uygulanabilmesi amaçlanmaktadır. Çoklu etmenli sistemlerde en etkili öğrenme metodu olan takviyeli öğrenme modeli, iskelet olarak önceki kısımlarda da belirtildiği gibi Markov modellerini ve Markov Karar Süreç işlemini baz alır.

Markov Karar Süreç işleminde karar veren etmen, içinde bulunduğu olasılıksal geçiş fonksiyonu olarak tanımlanmış ortamla sürekli etkileşim halindedir. Bu açıdan birincil etmenler kendi davranışlarını ikincil etmene göre yeniden düzenlemek durumundadırlar. Markov modellerinin iskelet yapısı, bu bakımdan, çoklu etmenlerin kendilerine verilen amacı gerçekleştirebilmesi için geniş bir bakış açısı ortaya çıkarır.

Genel olarak n-etmenli bir Markov Modeli $<S,A_1,...,A_n,T,R_1,...,R_n>$ olarak gösterilir. Burada,

- S, ortamın durumunu barındıran sonlu bir kümeyi ifade etmektedir.
- $A_1,...,A_n$ her bir etmen için mümkün olan hareketlerin oluşturduğu sonlu kümeleri göstermektedir (A_i ,i etmeninin sonlu hareket kümesini gösterir).
- $T:SxA_1x...xA_n \rightarrow \prod(S)$ geçiş fonksiyonu olarak tanımlanır ve ortamın o anki durumuna göre her bir etmenin hareketlerinin kartezyen çarpımı ile ortamın bir sonraki durumu hakkında olasılık dağılım fonksiyonunu tanımlar. Burada ortamın, içinde bulunduğu s durumundan bir sonraki s' durumuna geçme olasılığını her bir etmenin yaptığı hareket belirler.
- $R:SxA_1x...xA_n \rightarrow \Re$ her bir etmenin alacağı ödül fonksiyonu olarak tanımlanmıştır. Burada etmenin almayı beklediği anlık ödülü, ortam ve her bir etmenin grup olarak yaptıkları hareketler belirlemektedir.
- β değeri azaltma faktörü olarak belirlenir. Bu β azaltma faktörü bazı uygulamalarda reel sayı olabileceği gibi bazı uygulamalarda ise bir fonksiyon olarak tanımlanabilir.

Önceki kısımlarda da belirtildiği gibi, etmenin amacı alacağı azaltılmış toplam ödülleri maksimize etmektir. Özetle bu ödül $E\left\{\sum_{j=0}^{\infty}\beta^j r_{t+j}^i\right.$ şeklinde formüle edilebilir. Burada r_{t+j}^i, j adım içerisinde *i.* etmen tarafından alınan ödüldür. Eğer etmenler birlikte hareket edip, kendi aralarında koordinasyonu sağlamak için optimum politikayı izleyeceklerse, çoklu etmenli ortamlarda bir Q fonksiyonu tanımlanacaktır. Bu Q fonksiyonunu yukarıdaki Markov modeline göre şu şekilde yazabiliriz:

$$Q_i^{\pi}(s,a_1,...,a_n) = R_i(s,a_1,...,a_n) + \beta\sum_{s'\in S} T(s,a_1,...,a_n,s')$$
$$\cdot \sum_{a'_1\in A_1,...a'_n\in A_n} \pi(s',a'_1).\pi(s',a'_n).Q_i^{\pi}(s',a'_1,...,a_n)$$

Burada Q fonksiyonu her bir etmen için grup politikasının kesin çözümünü verecek şekilde tanımlanmıştır. Her bir etmen kendi ödül fonksiyonuyla tanımlanmış anlık ödülü alır ve bu ödül grup politikası dahilinde yapacağı tekil hareketi belirler. Bu şekilde grup politikasını belirlemek için oluşturulan Q fonksiyonu en iyi sistem cevabını verebilmesi için aşağıdaki gibi *max* operatörüyle basite indirgenebilir.

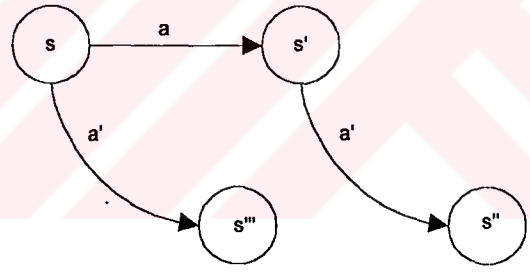
$$Q_{\Delta i}^{\pi}(s,a_1,...,a_n) = R_i(s,a_1,...,a_n) + \beta\sum_{s'\in S} T(s,a_1,...,a_n,s')$$
$$\cdot \max_{a'_i\in A_i} \sum_{a'_1,...,a'_{i-1}} \sum_{a'_{i+1},...,a'_n} \pi_1(s',a'_1)....\pi_{i-1}(s',a'_{i-1}).\pi_{i+1}(s',a'_{i+1})\pi_n(s',a'_n).Q_{\Delta i}^{\pi}(s',a'_1,...,a'_n)$$

Littman [12], yukarıdaki gibi tanımlanan grup politikasını belirlemek için özellikle çoklu etmenli mimarilerde takım q-öğrenmesi kavramını ortaya koymuştur. Bu yapıya göre çoklu etmenli bir sistemde etmenin öğrenmesi kolaylıkla lineer programlama teknikleri ile programlanabilmektedir. Ortamdaki tüm etmenler ortak bir Q tablosunu eş zamanlı olarak güncellemektedir. Dolayısıyla teorik olarak etmen, ortamdaki her bir durumu sonsuz sayıda ziyaret ederek optimum politikayı izleyeceği ispatlanmıştır. Son zamanlarda yapılan çalışmaların bir çoğu etmenin öğrenmesini hızlandırmak yönündedir. Kaya [6] ve Kılıç[8], etmenlerin ortamdaki bir durumda almış oldukları ödülü ve yapmış olduğu hareketi göz önünde bulundurarak henüz ziyaret edilmeyen durumları da güncellemeyi amaçlamışlardır. Kuter [3], etmenlerin gözlemleyebildiği durumları belirli bir Manhattan uzaklığı ile tarif ederek etmenin içinde bulunduğu durumda yapacağı hareketleri gruplandırma yoluna giderek S-learning kavramını ortaya koymuştur.

4.2 Hızlı Q-Öğrenme Metodu: FQ-Öğrenme

Standart Q-Öğrenme metodunda Q(s, a), ortam *s* durumundayken *a* hareketini seçtiği zaman almış olduğu azaltılmış maksimum ödülleri tanımlamaktadır. Bu durumda etmenin davranışlarını belirlemek ve öğrenme işlemini gerçekleştirmek için her bir Q değeri durum-hareket çifti olarak bir tabloya yerleştirilir. Etmen ortamın o anki durumunda bir hareket gerçekleştirdiği zaman bu Q tablosu güncellenir[1-2].

Çoklu etmenli sistemlerde ise ortamdaki tüm etmenler aynı anda ortak bir Q tablosunu güncellerler. Bu şekilde bir etmen kendi *s* durumunda edinmiş olduğu tecrübeyi diğer etmenlerle paylaşmış olur. Bununla birlikte Markov Modellerinin yapısının doğası gereği sistemde tek optimum politika olmasının yanı sıra birden fazla alt-optimum politikanın var olduğu önceki çalışmalarımızda gösterilmiştir [6]. Bu varsayıma göre Şekil 4.1 de gösterilen ve alt-optimum politikalar için $T(s, a', s''')$ geçiş fonksiyonu tanımlanabilmektedir.



Şekil 4.1: $T(s, a', s''')$ şekilsel olarak gösterimi

Şekil 4.1 de görüldüğü gibi ortam *s* durumunda iken etmenin $a' \in A(s)$ hareketi mevcut ise $T(s, a', s''')$ geçiş fonksiyonunun var olduğu söylenebilir. Sonuç olarak ortamdaki etmenler $T(s, a', s''')$ yolunu da güncelledikleri takdirde diğer yolları ve dolayısıyla tablodaki diğer Q değerlerini de güncellemiş olacaktır. Sonuç olarak etmen $\sum_a \pi(s,a)Q^{\pi}(s,a)$ optimum politikasına göre bir harekette bulunur ve alt-optimum politika olarak adlandırılan $Q^*(s, a')$ değerini günceller. Bu değer de 2. bölümde gösterilen Q fonksiyonuna limit durumda yakınsar.

FQ-öğrenme adı verilen bu takviyeli öğrenme yöntemi aşağıdaki gibi formüle edilebilir:

Eğer $a' \in A(s)$ ise :

$$Q(s,a') = k[(1-\beta)Q(s,a') + \beta(r + \gamma \max_{a' \in A} Q(s' ,a'))],$$

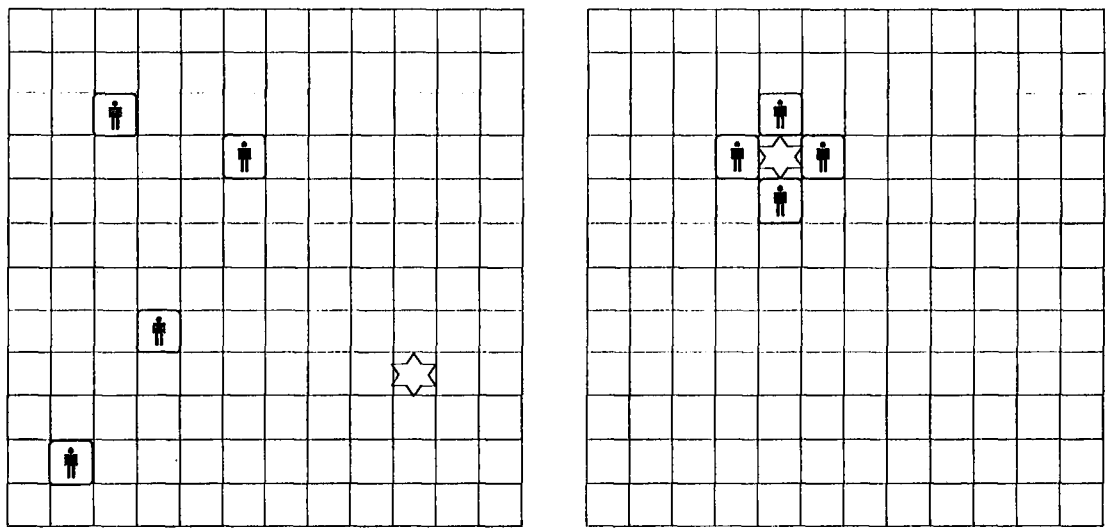
$$k = \begin{cases} 1 & a' = a \\ (0,1] & diger \end{cases}$$

fonksiyonu yazılabilir. Burada s, ortamın o an içinde bulunduğu durum; a, etmenin s durumunda yaptığı hareket ve k ($0<k\leq1$) sabit değeri ise aşağıdaki gibi tanımlanan yönelme parametresidir.

4.2.1 Deneysel Ortam: Av- Avcı Problemi

Problemin amacı, avcı etmenlerin av etmenini birlikte koordineli şekilde ve işbirliği yaparak en iyi yakalama politikasını keşfetmesi esasına dayanır. [11]

- Ortam 12x12 boyutlarında bir ızgara ortamıdır ve kenarlar bağlantılıdır.
- Başlangıç durumunda her bir etmen farklı bir yerde ve rasgele bir noktada konumlandırılmıştır.
- Her bir zaman adımında, etmenler senkronize olarak yukarı, aşağı, sola veya sağa doğru olan hareketlerden birisini yürütür yada yerini muhafaza eder.
- Aynı anda ve aynı ızgara hücresinde birden fazla sayıda avcı etmeni bir arada bulunabilir. Bununla birlikte bir veya daha fazla sayıda avcı etmeni ile bir veya daha fazla sayıda av etmeni aynı zaman diliminde aynı ızgara hücresinde yer alamazlar.
- Av etmeni herhangi bir anda hareketini seçerken, görsel alanında bulunan avcı etmenlerinin kendisine olan uzaklıklarının toplamını maksimize etmeye çalışır. (Manhattan Uzaklığı)
- Bir etmen hareket ederken kendi görsel alanında bulunan nesneleri göz önünde bulundurarak hareketlerini seçer.
- Av etmeninin yakalanması şartı, bütün avcı etmenlerinin, av etmenlerini çevrelemesi sonucunda av etmenlerinin hareket edecek bir alan bulamaması esasına dayanır. Bu durumun gerçekleşmesi durumuna *amaç durum* denir. Şekil 4.2'de herhangi bir başlangıç durumu ve amaç durum gösterilmiştir. Amaç durum gerçekleştikten sonra bütün av ve avcı etmenleri rasgele bir konumda yeni bir oyuna başlamak üzere konumlandırılırlar.

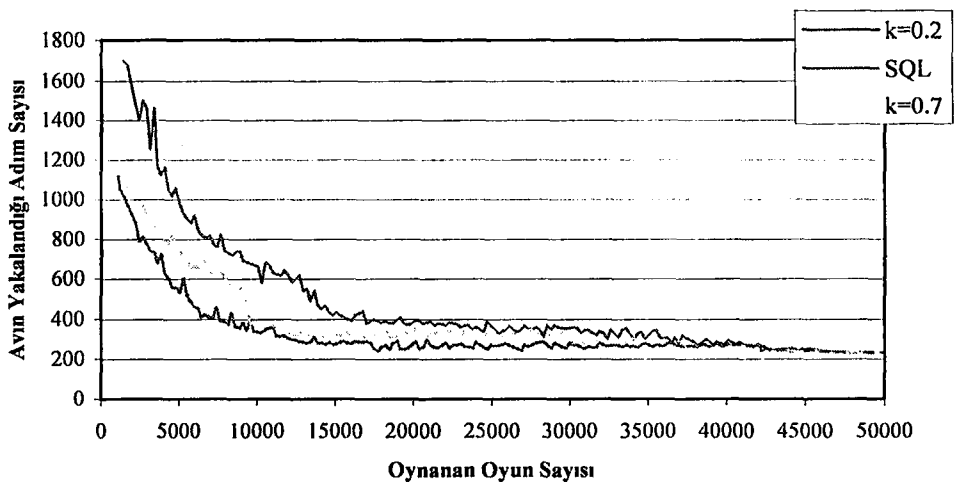


a) b)

Şekil 4.2 a) Problem ortamı için herhangi bir başlangıç durumu. b) Hedef durum.

Önerilen yöntemleri değerlendirmek için yapılan deneylerde dört adet avcı etmen ve bir adet av etmeni oyun ortamına gelişigüzel olarak yerleştirilirler. Yukarıda anlatıldığı gibi avcı etmenler av etmeninin etrafını sarana kadar oyun devam eder, 2000 adım sonunda hala avcı etmenler avı yakalayamadıkları durumda oyun tekrar başa alınır. Av etmeni yakalandığı zaman ortam, avcı etmenlere +1.0 puan ödül verir, av etmeninin yakalanmadığı her harekette ise ortam avcı etmenlere -0.1 puan ceza verir.

Şekil 4.3 te FQ-öğrenme yönteminin standart yönteme göre k katsayısının farklı değerlerine göre grafiği gösterilmiştir. Burada FQ-öğrenmenin parametreleri: öğrenme oranı β=0.8, azaltma



Şekil 4.3 Standart Q-Öğrenme ile FQ-Öğrenmenin farklı k değerleri için karşılaştırılması

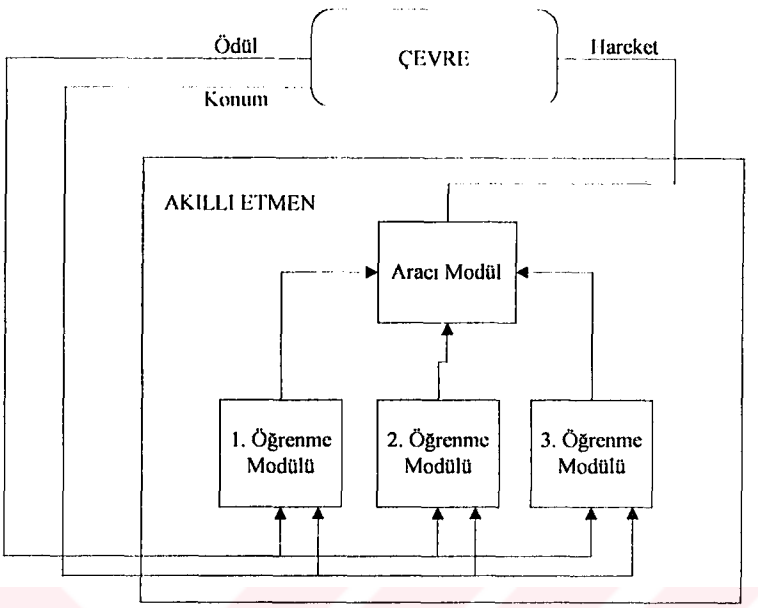
faktörü γ=0.9 ve Q tablosunun başlangıç değerleri 0.1 olarak seçilmiştir.

Çoklu etmenli ortamlarda belirsiz ödüller olduğu zaman yada ortam tam olarak modellenemediği durumlarda en popüler Takviyeli Öğrenme metotlarından olan Q-Öğrenme optimum politikayı bulacaktır. Bununla birlikte her zaman adımında sadece bir hareketi güncellediği için durum uzayının oldukça büyük olduğu durumlarda standart Q-öğrenme yöntemi etkisizdir. Şekil 4.3'te de görüldüğü gibi FQ-öğrenme yöntemi hem optimum politikaya yakınsamakta hem de standart Q-öğrenme yönteminden daha hızlı öğrenmektedir. Sonuç olarak, Markov Karar Süreç modeline göre oluşturulmuş deneysel ortamda alt-optimum politikaların da araştırılması nedeniyle daha hızlı bir öğrenme gerçekleştirilmiştir.

4.3 Modüler Q-Öğrenme

Önceki bölümlerde geleneksel takviyeli öğrenme yöntemlerinin dezavantajlarından bahsedilmişti. Bu dezavantajlardan en önemlileri arasında, eğitim sürecinin aldığı zamanın uzun olması ve etmenlerin her birisinin görsel alanlarında oluşan durum uzayı boyutunun büyüklüğü yer almaktadır. Bu dezavantajları ortadan kaldırabilmek için önerilen yöntemlerden birisi de modüler yaklaşım yöntemidir[9, 10].

Modüler yaklaşım yönteminde getirilen en büyük yenilik etmenlerin mimarilerinin ve karar verme modüllerinin değiştirilmesi ve geliştirilmesi olmuştur. Geleneksel takviyeli öğrenme yöntemlerini kullanan bir etmenin yapısı ve dış dünya ile etkileşim mimarisinde bir takım yenilikler yapılmıştır. Bu yeni mimari yapısı ile inşa edilen etmen ile dış dünyası arasındaki iletişimi ve etmenin mimari yapısı şekil 4.4.'de verilmiştir



Şekil 4.4 Modüler mimariye sahip olan bir akıllı etmenin dış dünya ile iletişimi.

Bu mimaride açıkça görülebileceği gibi akıllı etmenin mimari yapısına iki yeni modül eklenmiştir. Bunlardan birisi öğrenme modülü diğeri ise aracı modüldür. Bu yaklaşımı açıklarken tez çalışmamızda deneysel sonuçları gözlemlemek için uygulama yaptığımız ızgara ortamını kullanalım. Daha önceki bölümlerde de değinildiği gibi ızgara ortamı, 12x12 boyutundadır. Bu ortam içinde dört adet avcı etmeni ve bir adet av etmeni yer almaktadır. Geleneksel takviyeli öğrenme yöntemleri ile böyle bir ortamda etmenlerin eğitilmesi sürecinde her bir etmenin görsel alanında oluşan durum uzayı boyutu 1,021,700'dir. Modüler yapı yaklaşımı kullanılarak durum uzayı sayısı büyük oranda azaltılabilmektedir.

Etmenin içerisinde yer alan öğrenme modüllerinin sayısı, ortamda bulunan toplam etmen sayısı n olmak üzere, n-1'dir. Herhangi bir etmen yapacağı harekete karar verirken, kendi görsel alanındaki avcı ve av etmenlerinin konumlarını göz önünde bulundurur. Ancak geleneksel takviyeli öğrenme yöntemlerinde bu etmenlerin hepsinin konumlarını tek bir karar verme modülünde değerlendirir. Bundan dolayı durum uzayı boyutu oldukça büyük olur. Bu mimaride ise, 1.öğrenme modülü sadece 1.avcı etmeninin ve av etmeninin konumlarını dikkate alarak yapılması gereken hareketi tespit eder ve bu hareketin yapılması durumunda alınacak ödülü saklar. Aynı şekilde 2.öğrenme modülü de sadece 2. avcı etmeninin ve av etmeninin konumlarını dikkate alır ve yapılacak en iyi hareketi ve bu hareketin yapılması durumunda kazanılacak ödülü saklar. Örneğin, eğer etmenin görsel alanında sadece 1. ve 2. avcı etmeni ve av etmeni yer almıyorsa, 1 öğrenme

modülü 1. avcı etmeni ile av etmeninin, 2. öğrenme modülü 2.avcı etmeni ve av etmeninin, 3 öğrenme modülü de sadece av etmeninin konumlarını dikkate alarak yapılabilecek en iyi hareketleri ve bu hareketlerin yapılması durumunda alınacak ödülleri saklarlar. Böylece bir anlamda yapılaması gereken görev parçalara ayrılarak her bir modüle paylaştırılır. Her bir öğrenme modülü kendi öğrenme politikaları doğrultusunda yapılması gereken hareketi ve bu harekete ait ödülü aracı modüle gönderir. Aracı modül aslında son kararı veren ve etmenin yapacağı hareketi belirleyen modüldür. Bu modül yapılacak olan harekete aşağıdaki formülde verilmiş olan seçme politikasına göre karar verir:

$$\arg\max_{a \in A} \sum_{i=1}^{i=3} Q_i(s_i, a)$$

Burada $Q_i(s_i, a)$ ifadesi, i. öğrenme modülündeki s_i durumunda yapılacak olan a hareketi sonunda elde edilecek olan Q ödülünün değeridir. Aracı modüle ulaştırılan bu ödüllerden yola çıkılarak yapılacak olan harekete karar verilir.

Bu yaklaşımda ise durum uzayı gözlemleyen etmenin görsel alanına bağlıdır. Geliştirilen yeni mimari gereğince akıllı etmenin içerisinde oluşturulan öğrenme modülleri sayesinde durum uzayının boyutu ortamdaki etmen sayısından bağımsızdır. Oluşan durum uzayı sadece bir modülün durum uzayı ile toplam modül sayısının çarpımına eşittir. Herhangi bir öğrenme modülü ortamda gözlemleyen etmenin dışında sadece iki etmen varmış gibi davrandığı için ortamdaki etmen sayısından bağımsız olarak durum uzayı oluşur. Bu mimaride oluşan durum uzayı sayısı, etmenin görüş alanı *d* ise, $m=(2.d + 1)$ olmak üzere, $m^4 + 1$' dir.

4.4 Bulanık Q-Öğrenme Metodu

Geleneksel takviyeli öğrenme yöntemlerine önceki bölümlerde değinilmişti. Klasikleşen bu yöntemlerin bazı dezavantajlarını ortadan kaldırmak üzere takviyeli öğrenme yöntemlerine bulanık mantık bakış açısından yaklaşılmıştır. Bu dezavantajlardan en büyüğü, klasik takviyeli öğrenme yöntemlerinin bazı problemlerde kullanılması sonucunda oluşan durum uzayı sayısının çok büyük olmasıdır. Bu durumun doğurduğu olumsuz sonuçlar da karar verme zamanının fazla olması ve dolayısı ile etmenlerin eğitim süreçlerinin durum uzayının boyutu ile orantılı olarak uzamasıdır.

Bu dezavantajın ortadan kaldırılmasına bir çözüm olarak geliştirilmiş olan bulanık takviyeli öğrenme yöntemi durum uzayının boyutunu çok büyük oranda azaltmıştır. Bu durumun

daha iyi anlaşılabilmesi için bu tez çalışmasında tanımlanmış av-avcı deneysel ortam üzerinde açıklama yapılması uygun olacaktır.

12x12 boyutundaki, deneysel ortam bölümünde detaylandırılan kurallar çerçevesinde geleneksel takviyeli öğrenme yöntemi kullanılarak etmenlerin eğitilmesinin sağlanması durumunda her bir etmenin görsel alanı ile orantılı olarak durum uzayının boyutu artacaktır. Bununla birlikte aynı ortamda birden fazla etmen bulunuyorsa, etmen sayısı ile üstel olarak durum uzayı boyutu artacaktır. Bu durumun matematiksel ifadesi aşağıdaki gibidir:

Eğer bir etmenin görüş alanı d ise etmenin görüş alanındaki durum sayısı $(2d+1)^2 + 1$'dir. Ancak yukarıda da bahsettiğimiz gibi ortamda birden fazla etmen yer alıyorsa oluşacak durum sayısı etmen sayısına bağlı olarak artacaktır. Bizim problemimizdeki gibi bir sistemde 4 avcı etmeni ve bir av etmeninden oluşan bir sistemde etmenlerin her birinin görsel alanı $m=2d+1$ olmak üzere oluşacak durum sayısı;

$$4m^2 + 6(m^2 - 1)^2 + 2(m^2 - 1)(m^2 - 2)^2 + (m^2 - 1)(m^2 - 2)(m^2 - 3)^2/6$$'dir.

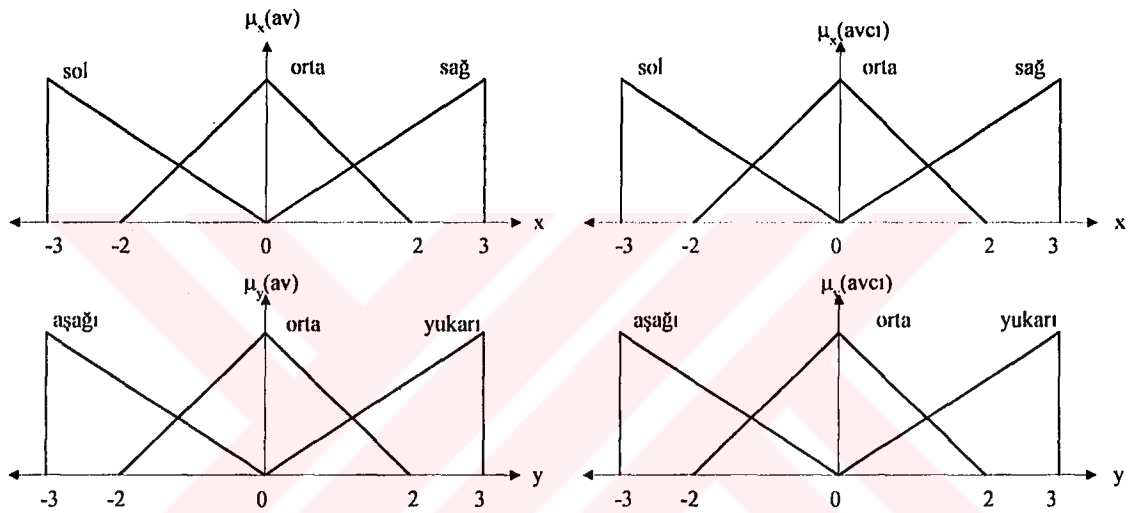
Bu durumda etmenlerin her birisinin görüş alanı 3 ızgara hücresi ise, her bir etmenin durum uzayı sayısı 1,021,700'dür. Bunun anlamı; her bir etmen herhangi bir anda yapacağı harekete karar verirken bu durumlardan hangisinde olduğuna karar verecek ve hangi hareketi yaptığında en yüksek puanı alacağını bulmaya çalışacaktır. Bu da işlemin gerçek zamanda gerçekleştirilmesini imkânsız hale getirecek ve öğrenme sürecinin oldukça büyümesine yol açacaktır. Ayrıca gerçek bir ortamda robotların kullanılması be bu robotların eğitilmesi söz konusu olursa robotun ortamı bir ızgara ortamı olarak görmesi ve bu şekilde değerlendirme yapması da oldukça büyük bir problem teşkil etmektedir. Dolayısıyla bir robotun görsel alanında bulunan herhangi bir nesnenin koordinatlarını tam olarak tespit etmesi de oldukça zordur. Bunun gerçekleştirilmesi durumunda robot ortamı aşağıdaki gibi algılaması gerekir.

Bu ortamda H etmeninin P etmeninin bulunduğu yeri (2,2) olarak belirlemesi oldukça güçtür. Ayrıca eğer H etmeni yapacağı hareketi P etmeninin bulunduğu konuma göre yapıyorsa, P etmeninin (2,2) konumunda bulunması ile (2,3) veya (3,2) konumlarının herhangi birisinde bulunması arasında çok fazla bir fark yoktur. P etmeninin bu konumlardan herhangi birisinde olması durumunda H etmeninin yapacağı hareket aynı olacaktır.

Bu problemin bulanık – takviyeli öğrenme yöntemi ile gerçekleştirilmesi durumunda ise ortamın ızgara ortamı olup olmaması veya gözlemlenen etmenin bulunduğu yerin tam koordinatlarının bir önemi yoktur. Şekil 4.1.'de olduğu gibi bir etmenin (2,2), (2,3), (3,2) veya (3,3)

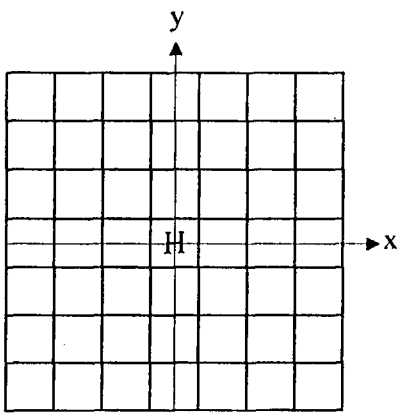
konumlarında olması arasında çok fazla bir fark yoktur. Bu konumlardan her hangi birisinde bulunan gözlemlenen etmen, gözleyen etmene göre sağ – üst konumdadır. Etmen bütün bu durumların hepsini tek bir durum gibi algılar ve yapacağı harekete karar verir. Böylece güncellenecek olan Q tablosundaki durum sayısı azaltılmış olur.

Etmenin dış dünyadan bilgi almasını sağlayan algılayıcılardan gelen bilgiler bulanık üyelik giriş fonksiyonlarını oluşturur. Tanımlanan bulanık üyelik giriş fonksiyonları şekil 4.5'de gösterilmiştir.

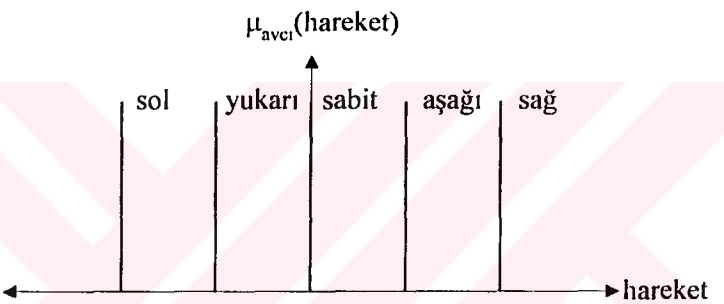


Şekil 4.5. Akıllı etmen için tanımlanmış olan bulanık üyelik giriş fonksiyonları.

Gözlemlenen bir etmenin, gözlemleyen etmene göre bulunduğu yer x ve y eksenlerine göre 9 farklı durumda sınıflandırılabilir. Örneğin bir ızgara ortamında bulunan bir akıllı etmenin görsel alanında H_{11} etmeni H_1 etmenine göre sol – üst tarafında , P etmeni de H_1 etmenine göre sağ alt tarafta bulunsun. Etmen yapacağı harekete karar verirken görsel alanında bulunan diğer etmenlerin tam koordinatları yerine bu bilgiler ışığında bir seçim yapar. Bulanık sistemin çıkışı ise şekil 4.7.'deki gibidir.



Şekil 4.6. Akıllı etmenin herhangi bir andaki görsel alanı.



Şekil 4.7. Bulanık sistemin çıkış fonksiyonu.

Önerilen yöntemde Bulanık sistemin kural tabanı Q-öğrenme tarafından oluşturulur. Bu bilgiler ışığında etmenin görsel alanı ne olursa olsun, etmenin görsel alanında oluşacak olan durum uzayı sadece bulanık sistemin giriş fonksiyonunun derecesi ile ve ortamdaki etmen sayısı ile orantılı olarak artacaktır[21,22,23]. Bizim uygulama ortamımızdaki gibi ortamda 4 adet avcı ve bir adet av etmeninden oluşan bir sistemde bulanık üyelik giriş fonksiyonları 3. mertebeden tanımlanırsa bir etmenin görsel alanında oluşabilecek durum sayısı yani durum uzayı sayısı; ortamdaki toplam etmen sayısı n olmak üzere 10^{n-1} duruma iner. Bu da bizim kullandığımız ortamda durum uzayının 10,000 duruma indirgenmesine eşdeğerdir. Yani geleneksel takviyeli öğrenme yöntemlerinde aynı ortam için oluşan durum uzayı boyutu 1,021,700 iken önerdiğimiz yöntemle durum uzayı boyutu 10,000 duruma indirgenir.

Önerilen bu yöntemde durum uzayının boyutu tanımlanan bulanık üyelik giriş fonksiyonunun derecesine ve ortamda bulunan toplam etmen sayısına bağımlıdır. Tanımlanan

bulanık üyelik giriş fonksiyonunun derecesi m ve ortamdaki toplam etmen sayısı n olmak üzere, ortamda oluşabilecek olan toplam durum uzayı sayısı;

$$\sum_{i=0}^{n-1}\binom{n-1}{i}(m^2)^{n-1-i}$$

olur.

Bu noktaya kadar bulanık küme yaklaşımı sadece öğrenme anında durum uzayının azaltılmasını amacıyla kullanıldı. 2. bölümde bulanık mantık ve bulanık kontrol yapısının özelliklerini kullanarak son yıllarda Q-öğrenme yöntemini geliştiren çalışmalar vardır[24, 27]. Glorennec [21,25], a hareketini yapan etmenin hareket uzayını bulanık değişkenler ile ifade edip kural kalitesi q kavramını göstermiştir. Buna göre her etmenin performansı şu şekilde ifade edilebilmektedir:

$$Q(s_k, a_k) = \frac{1}{2^n} \sum_{i \in R(s_k)} q(i, a_k)$$

Burada $R(s_k)$, s_k durumunda meydana gelen bulanık kurallar kümesini ifade etmekte ve n, giriş değişken sayısını göstermektedir. Formülde ifade edilen öğrenme katsayısı:

$$\Delta q(i, a_k) = 2^n \, acl(i) \Delta Q$$

şeklinde ifade edilmektedir.

Glorennec [23], aynı zamanda etmenlerin geçmişte yaptıkları hareketleri gözönünde bulundurarak Dinamik Bulanık Q-Öğrenme yöntemini önermiş, etmenin hedef durumu ile şimdiki durumu arasındaki hata katsayısını bulanık girişler şeklinde ifade etmiştir.

Berenji [26], bulanık mantık yaklaşımı kullanarak önerdiği Bulanık Q-Öğrenme yönteminde etmen bulanık kuralları kullanmadan öğrenme işlemini gerçekleştirmektedir. Bu yöntemde bulanık kısıtlamalar kullanılmış, etmenin bir sonraki hareketinin hedef duruma göre yakınlığının tahmini amaçlanmıştır. Buna göre:

$$Val(s, a) = Max_a FQ(s, a)$$

$$\Delta FQ(s_k, a_k) = \beta[(r + \gamma Val(y)) \wedge \mu_C(s_k, a_k) - FQ(s_k, a_k)]$$

4.5 İç Model Kullanarak Q-Öğrenme

Çoklu etmenli bir sistemde etmenler işbirliği yaparak ve birlikte (koordineli) hareket ederek amaçlarına ulaşmaya çalışırlar. Dolayısıyla etmenler yapacakları hareketleri seçerken diğer etmenlerin konumlarını ve yapmaları olası olan hareketleri de göz önünde bulundurmalıdırlar. Öğrenen bir akıllı etmen, kendi yapacağı harekete karar verirken, görsel alanında bulunan diğer etmenlerin yapacakları hareketleri de tahmin edebilmelidir[28,29]. Bunu gerçekleştirebilmek için diğer etmenin iç modelini ve yapacağı hareketi tahmin edebilen bir algoritmanın kullanılması gereklidir.

Bu tahmini yapabilmek için geleneksel takviyeli öğrenmeden farklı olarak, $Q(s, a_{self}, a_{other})$ şeklinde bir Q fonksiyonu tanımlanır. Burada $s \equiv (s_{self}, s_{other})(\in S)$'dir. Gözlemleyen etmenin içinde bulunduğu ortamdaki konumu s_{self}, diğer etmenin konumu ise s_{other} şeklinde sembolize edilmiştir. Gözlemleyen etmenin herhangi bir anda gerçekleştirdiği hareket a_{self} $(\in A_s)$, diğer etmenin gerçekleştirdiği hareket ise a_{other} $(\in A_o)$ şeklinde gösterilecektir.

Bu yöntemde gözlemleyen etmen kendi hareketini seçerken, diğer etmenin hareketini gizli bir değişken olarak kullanır. Yani etmen kendi yapacağı hareketi seçerken, diğer etmenin hareketini de göz önünde bulundurmak zorundadır. Ancak etmenler eşzamanlı hareket ettikleri için bu durumun gerçekleştirilmesi mümkün değildir. Dolayısıyla bir etmen diğer etmenin yapacağı hareketi tahmin etmek zorundadır. Bu öğrenme yönteminde, diğer etmenin yapacağı hareket tahmin edilirken, diğer etmenin geçmişte yaptığı hareketler gözlemlenerek elde edilen deneyimler sonucunda yapılacak olan hareket tahmin edilmeye çalışılır. Bunu gerçekleştirebilmek için $I(s, a_{other})$ şeklinde bir fonksiyon tanımlanır[28].

Bir gözlemleyen etmen, diğer etmenin s konumunda gerçekleştirdiği bir a^*_{other} $(\in A_o)$ hareketini gözlemlediği zaman, gözlemleyen etmen s durumunda yürütülebilecek her a_{other} $(\in A_o)$ hareketi için dahili modelini aşağıdaki gibi günceller.

$$I(s, a_{other}) \leftarrow (1-\theta) I(s, a_{other}) + \begin{cases} \theta & (a_{other} = a^*_{other} \\ 0 & (diger) \end{cases}$$

burada θ $(0 \leq \theta \leq 1)$ aralığında seçilebilecek olan bir kontrol parametresidir.

Bir çoklu etmenli sistemde, diğer etmenin dahili modelini tahmin etmeye dayalı olan Q öğrenme yönteminin süreci aşağıdaki gibidir:

1. Etmen içinde bulunduğu ortamdaki konumunu göz önünde bulundurarak, diğer etmenin yapacağı hareketi $I(s, a_{other})$ fonksiyonunu kullanarak tahmin ederek, kendi yapacağı a_{self} hareketini aşağıdaki seçme politikasına göre seçer.

$$\pi[a_i \mid s] = \frac{e^{\overline{Q}(s,a_i)/\tau}}{\sum_{a_k \in A} e^{\overline{Q}(s,a_k)/\tau}}$$

burada $Q(s, a_{self})$ ifadesi, $I(s, a_{other})$ fonksiyonu göz önünde bulundurularak elde edilen Q fonksiyonudur.

$$\overline{Q}(s, a_{self}) \equiv \langle Q(s, a_{self}, a_{other}) \rangle_{I(s,a_{other})}$$
$$= \sum_{a_{other} \in A_o} I(s, a_{other}) Q(s, a_{self}, a_{other})$$

2. Etmen yapacağı a_{self} hareketini seçip gerçekleştirdikten sonra 1. adıma döner. Aynı zamanda diğer etmende kendi a^*_{other} hareketini gerçekleştirir. Bu durumda etmenlerin ortam içinde bulunduğu konum s iken, hareketler gerçekleştirildikten sonra s' konumuna geçer ve yapılan hareket sonunda içinde bulunduğu ortamdan r ödülünü alır. Bundan sonra, etmen $I(s, a_{other})$ fonksiyonu gereğince tahmin tablosunu güncelledikten sonra $Q(s, a_{self}, a^*_{other})$ fonksiyonunu aşağıdaki gibi günceller:

$$Q(s, a_{self}, a^*_{other}) \leftarrow (1-\beta) Q(s, a_{self}, a^*_{other}) + \beta(r + \gamma \max_{a_{self} \in A_s} Q(s', a'_{self}, a'_{other}))$$

burada $a'_{other} = \arg\max_{a'_{other} \in A_o} I(s', a'_{other})$ 'dir.

3. Eğer yeni s' durumu amaç durum ise öğrenme bölümü sona ererek yeni bir öğrenme bölümü başlatılır. Aksi takdirde $s' \rightarrow s$ durumu olmak üzere 1. adıma geri dönülür.

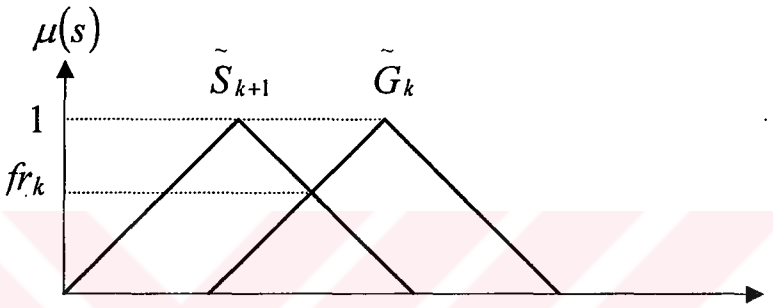
4.6 Bulanık Minimax Q-Öğrenme Metodu

Çoklu etmenli sistemlerde ortamdaki bir etmen kendi hareketini belirlerken, diğer etmenlerin de hareketlerini göz önünde bulundurması sistemin verimliliğini arttırmaktadır. Bununla birlikte bazı sistemlerde etmen kendi ödülünü maksimum yapmak isterken ortamdaki diğer etmenlerin alacağı ödülleri minimum yapmak isteyebilir. Ayrıca ortamdaki diğer etmenin alacağı ödülün amaç duruma olan uzaklığının etkili olduğu önceki bölümlerde anlatılmıştı. Bazı sistemlerde de etmenin amaç durumu 2. bölümde anlatıldığı gibi dilsel değişkenler ile ifade edilebilir. Örneğin

bir robot etmeninin amacı "düşmanını yeterli uzaklıkta takip et, uygun bir köşede dur" gibi kesin küme kavramıyla ifade edilemeyebilir[8, 18]. Bu durumda ortamın durum değişkenleri bu tezde amaçlandığı gibi bulanık küme yaklaşımı kullanarak ifade edilmelidir. Buna göre bulanık $FR(\tilde{S}, a)$ değeri aşağıdaki formüle ve şekil 4.8'e göre hesaplanır:

$$fr_k = \max(\mu_{\tilde{S}_k} \wedge \mu_{\tilde{S}_{k+1}}) = \max(\min\{\mu_{\tilde{S}_k}, \mu_{\tilde{S}_{k+1}}\})$$

$$FR(\tilde{S}, a) = r \cdot \min_{fr_k}\{fr_1^m, fr_2^m, ..., fr_n^m\}$$



Şekil 4.8 Bulanık durum ve bulanık amaç durumu

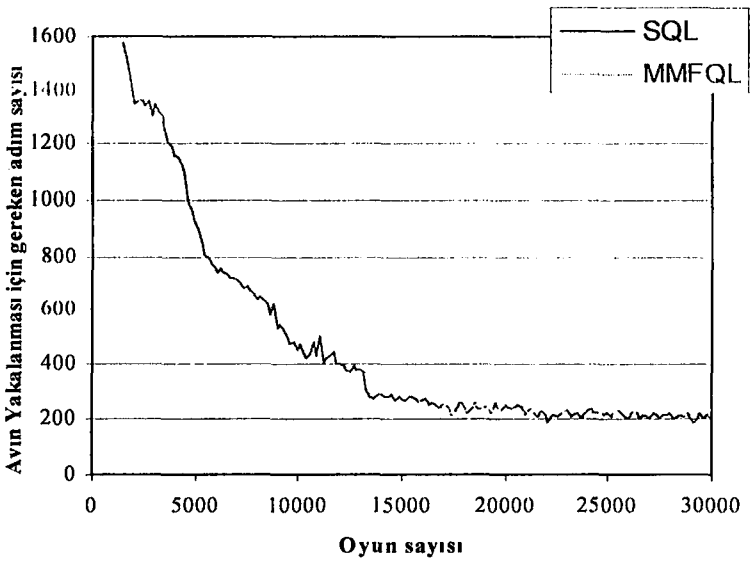
Yukarıdaki formüle göre $(0 \le \min_{fr_k}\{fr_1^m, fr_2^m, ..., fr_n^m\} \le 1)$

$$FValQ(\tilde{S}, Q_1) = \max_{\pi_1(s,\cdot)\in\Pi(A_1)} \max_{a_{other}\in A_2} \sum_{a_{self}\in A_1} \pi(\tilde{S}, a_{self}) \cdot Q_1\left[\tilde{S}, a_{self}, a_{other}\right]$$

$$FuzzyQ(\tilde{S}, a_{self}, a_{other}) =$$

$$(1-\beta) \cdot FuzzyQ(\tilde{S}, a_{self}, a_{other}) + \beta(FR(\tilde{S}, a_{self}) + \lambda \cdot FValQ(\tilde{S}, Q_1))$$

Bu tez çalışmasında göz önünde bulundurulan deneysel ortam olan av-avcı problemi üzerinde bulanık minimax algoritması uygulandığı zaman elde edilen sonuçlar Şekil 4.6'daki gibidir. Bu deneylerde avcı etmenler av etmenini yakaladıkları zaman +1.0 ödül puanı alır, yakalanmayan her harekette ise etmen 0 puan ödül-ceza alır. Elde edilen grafik, standart Q-öğrenme yöntemiyle tezde önerilen yöntemlerden biri olan minimax Q-öğrenme yönteminin karşılaştırılmasını göstermektedir.



Şekil 4.9 Standart Q-Öğrenme ve bulanık Q-Öğrenmenin karşılaştırılması

Şekil 4.6 da görüldüğü gibi etmenin şu an içinde bulunduğu durum ile amaç durum arasındaki bulanık ilişkinin kullanılması ile optimum sonuca erişme hızında büyük bir artış gözlenmiştir.

5. SONUÇ

Günümüzde yapay zeka sistemlerinde en popüler öğrenme algoritması Takviyeli öğrenme metotlarından biri olan Q-öğrenme, birçok çalışmanın odak noktası haline gelmiştir. Öğrenme işleminin optimuma yakınsaması limit durumunda ispatlanmış olan bu öğrenme yöntemi kolaylıkla program aşamasına geçirilebilmekle beraber bazı ek dezavantajlara sahiptir.

Bu tez çalışmasında standart Watkins q-öğrenme algitmasının var olan dezavantajlar incelenip düzeltilmeye çalışılarak, yerine değişik bir varyasyonu olarak bulanık mantık yaklaşımı önerilmiştir. Böylece etmenin içinde bulunduğu durum bulanık giriş değişkenleri şeklinde ifade edilerek, durum uzayının oldukça azaltılması amaçlanmıştır. Durum uzayının azaltılması öğrenme işleminin hızlanmasına, edinilen tecrübelerin ortak kullanımı ile işbirliğinin daha optimal hale gelmesine yol açmaktadır.

Bulanık kümelerinin kullanılması ile diğer avcı ya da av etmenlerinin konumu belirgin ve katı kurallar içerisinde sınırlandırılmayıp esneklik kazandırılmış ve d görsel alan uzayının derinliğinden bağımsız hale getirilmiştir.

Birinci bölümde akıllı etmen kavramı incelenerek bir etmenin akıllı özelliğini kazanabilmesi için üzerinde barındırması gereken yetenekler tartışılmış, ikinci bölümde ise q-öğrenme için kullanılacak bulanık sistemin temelleri örneklerle anlatılmış ve üyelik fonksiyonlarının sistemin yakınsama durumlarındaki etkisi incelenmiştir. Üçüncü bölümde standart takviyeli öğrenme metotları incelenmiş ve bu tez çalışmasında kullanılan standart q-öğrenme algoritması tanımlanmış ve yakınsaması incelenmiştir. Dördüncü bölümde ise bu tezde amaçlanan FQ-öğrenme, Bulanık Q-Öğrenme, Minimax Q-öğrenme algoritmaları tanımlanmıştır. Ardından önerilen yaklaşımlar için kullanılan deneysel ortam av-avcı problemi sunularak ve bu deneysel ortam için kullanılan bulanık-takviyeli öğrenme ve FQ-öğrenmenin sonuçları karşılaştırılmıştır.

KAYNAKLAR

1. Watkins, C. J. C. H., "*Learning from delayed rewards*", PhD Thesis, University of Cambridge, England, 1989.

2. Sutton, R. S., "*Learning to predict by the methods of temporal differences*", Machine Learning, 3, 9-44

3. Kuter, U., "*S-Learning: A Multi-Agent Reinforcement Learning Method*", MS Thesis, Middle East Technical University, 2000.

4. Sheppard, J. W., "*Multi agent reinforcement learning in Markov Games*", PhD thesis, John Hopkins University, 1997.

5. Andre, D., "*Learning Hierarchical Behaviors*", Draft, Submitted to AHRL Workshop, 1998.

6. Kilic, A., Kaya, M., Arslan, A.,"*Finding Sub-optimal Policies Faster in Multi-Agent Systems: FQ-Learning*", the 7th International Conference on Intelligent Autonomous Systems, California, USA, March 25-27, 2002

7. Kilic, A., Kaya, M., Arslan, A.,"*Fuzzy Reinforcement Learning in Cooperative Multiagent Systems*", International Symposium on Computer and Information Sciences (ISCIS 2001), Turkey, November 5-7, 2001

8. Kilic, A., Arslan, A.,"*Minimax Fuzzy Q-learning in in Cooperative Multiagent Systems*", Accepted for publication in Lecture Notes in Computer Science

9. Gültekin., İ, Arslan, A.,"*Moduler Fuzzy Cooperation Algorithms for Multi Agent Systems*", Accepted for publication in Lecture Notes in Computer Science

10. Ono, N., Fukomoto, K., "*Multi-agent reinforcement learning: A modular approach*". In Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS96), pp: 252-258, 1996.

11. Tan, M., "*Multi-agent reinforcement learning: independent vs. cooperative agents*" In Proceedings of the Tenth International Conference on Machine Learning (ICML-93), pp: 330-337, 1993.

12. Littman., M. L., "*Markov games as a framework for multi agent reinforcement learning*", Proceedings of the Eleventh International Conference on Machine Learning, pp. 157-163. San Francisco, CA 1994.

13. Sandholm, T.W.; Crites, R. H., "*Multi agent reinforcement learning in the Iterated Prisoner's Dilemma*", Biosystems, 37:147-166, 1995.

14. Hu, J.; Wellman, M. P., "*Multi-agent reinforcement learning: theoretical framework and an algorithm*" In Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98), pp: 242-250, 1998.

15. Sen, S.; Sekeran, M.; Hale, J., "*Learning to coordinate without sharing information*". In proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), pp: 426-431, 1994.

16. Lin, L. J. ,"*Self-improving reactive agents based on reinforcement learning, planning and teaching*", Machine Learning, Vol: 8, pp: 293-321, 1992.

17. Touzet, P., "*Neural reinforcement learning for behavior synthesis*", In Proceedings of CESA'96 IMACS Multi-conference, Lille, July 1996.

18. Berenji, H.; Vengerov, D., "*Cooperation and coordination between fuzzy reinforcement learning agents in continuous state partially observable markov decision processes*", In Proceedings of the 8th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'99) 1999.

19. Berenji, H.; Vengerov, D., "*Advantage of cooperation between reinforcement learning agents in difficult stochastic problems*", In Proceedings of the 9th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'00) 2000.

20. Littman, M. L., "*Value-function reinforcement learning in Markov games*", Journal of Cognitive Systems Research, vol: 2, pp: 55-66, 2001.

21. Glorennec, P. Y.; Jouffe, L., "*Fuzzy Q-learning*", Second European Workshop on Reinforcement Learning, Milano, Italy, September, 18-19, 1995.

22. Berenji, H.; Khedkar, P., "*Learning and tuning fuzzy logic controllers through reinforcement*", IEEE Trans. on Neural Networks, 3(5), Sept. 1992.

23. Glorennec, P. Y.; Jouffe, L., "*Reinforcement learning for autonomous robots*", Proc. of EUFIT, Aachen, Germany, Sept., 1996.

24. Castro, J. L., "*Fuzzy logic controllers are universal approximators*", IEEE Transaction on SMC, vol: 25/4, April, 1995.

25. Glorennec, P. Y., "*Fuzzy Q-learning and evolutionary Strategy for adaptive fuzzy control*", Proc. of EUFIT, ELITE Foundation, pp: 35-40, Aachen, Germany, Sept., 1994.

26. Berenji, H. R., "*Fuzzy Q-learning: a new approach for fuzzy dynamic programming*", Proc. Third IEEE Int. Conf. on Fuzzy Systems. IEEE Computer Press, Piscataway, NJ, pp: 486-491, 1994.

27. Bonarini, A., "Delayed *reinforcement, Fuzzy Q-learning and Fuzzy Logic Controllers*", Genetic Algorithms and Soft Computing, Physica Verlag (Springer Verlag), Heidelberg, Germany, pp: 447-466, 1996b.

28. Nagayuki, Y.; Ishii, S.; Kenji, D., "*Multi-agent Reinforcement Learning: An Approach Based On The Other Agent's Internal Model*", Fourth International Conference on Multi-agent Systems (ICMAS), 215-221, Los Alamitos, IEEE Computer Society, 2000.

29. Seo, H. S.; Youn, S. J.; Oh, K. W.," *A Fuzzy Reinforcement Function for the Intelligent Agent to process Vague Goals*", The 19th International Meeting of the North American Fuzzy Information Processing, NAFIPS, 2000.