

Ayrıklaştırılmış Durum Uzayı ile Öğrenme Tabanlı Görev Atama Yaklaşımı

Hatice Hilal Ezercan Kayır

DOKTORA TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Mart 2014

Learning-Based Task Allocation Approach with Discretized State Space

Hatice Hilal Ezercan Kayır

DOCTORAL DISSERTATION

Department of Electrical and Electronics Engineering

March 2014

Ayrıklaştırılmış Durum Uzayı ile Öğrenme Tabanlı Görev Atama Yaklaşımı

Hatice Hilal Ezercan Kayır

Eskişehir Osmangazi Üniversitesi

Fen Bilimleri Enstitüsü

Lisansüstü Yönetmeliği Uyarınca

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Kontrol ve Kumanda Bilim Dalında

DOKTORA TEZİ

Olarak Hazırlanmıştır

Danışman: Prof. Dr. Osman PARLAKTUNA

Mart 2014

ONAY

Elektrik-Elektronik Mühendisliği Anabilim Dalı Doktora öğrencisi Hatice Hilal Ezercan Kayır'ın DOKTORA tezi olarak hazırladığı “Ayrıklaştırılmış Durum Uzayı ile Öğrenme Tabanlı Görev Atama Yaklaşımı” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman : Prof. Dr. Osman PARLAKTUNA

İkinci Danışman : -

Doktora Tez Savunma Jürisi:

Üye : Prof. Dr. Osman PARLAKTUNA

Üye : Prof. Dr. Abdurrahman KARAMANCIOĞLU

Üye : Prof. Dr. Aydın AYBAR

Üye : Yrd. Doç. Dr. Nihat ADAR

Üye : Yrd. Doç. Dr. Metin ÖZKAN

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Bu çalışmada, çok robotlu sistemlerde görev atama problemlerinin çözümünde sistem performansının artırılması amacıyla öğrenme tabanlı görev atama yaklaşımı önerilmiştir. Bu amaçla Q-öğrenme yöntemi kullanılmıştır. Teorik olarak tek erkinli yapılar üzerinde tanımlanmış olan Q-öğrenme yönteminin çok robotlu sistemlerde uygulanması araştırılmıştır. Çok erkinli sistemlerde Q-öğrenme yönteminin uygulanmasına dair mevcut yaklaşımlar olan dağıtık ve merkezi öğrenme yaklaşımları ele alınmıştır. Bu iki temel yaklaşımın avantajlarını birleştirmek üzere Strateji Planlamalı Dağıtık Öğrenme yaklaşımı önerilmiştir. Q-öğrenme yönteminin çok robotlu sistemlerde uygulanmasında ortaya çıkan önemli bir problem ayırık ve sonlu durum ve hareket uzayları belirlenmesidir. Uygulamada kullanılan sistemde sürekli nitelikte olan durum uzayının temsili için Sabit Aralıklı Ayırık Durum Uzayı (SAADU), Dağılım Fonksiyonlu Sürekli Durum Uzayı (DFSDU) ve Değişken Aralıklı Ayırık Durum Uzayı (DAADU) yöntemleri önerilmiştir. Ayırık durum uzayı kullanımı yaklaşımı olan SAADU yöntemi sürekli durum uzayını sabit temsil hassasiyeti kullanarak ayırıklaştırırken, DAADU yönteminde durum uzayı ayırıklaştırma işlemi ardışıl kümeleme tabanlı bir yaklaşım ile adaptif olarak gerçekleştirilmektedir. DFSDU yönteminde ise durum uzayı sürekli bir yapıda temsil edilmektedir. Önerilen yaklaşımların etkinliği gerçekleştirilen uygulamalarda gösterilmiştir.

Anahtar Kelimeler: Çok robotlu sistemlerde görev atama, Q-öğrenme yöntemi, çok erkinli Q-öğrenme yöntemi, merkezi öğrenme, dağıtık öğrenme, ayırık durum uzayı, ardışıl kümeleme

SUMMARY

In this study, a learning-based task allocation approach is proposed in order to increase the overall system performance. For this purpose, Q-learning algorithm is preferred. Theoretically, Q-learning algorithm is defined on single-agent frame. The difficulties of scaling up the multi-agent Q-learning to multi-robot systems are investigated. Two major approaches of multi-agent Q-learning in literature, distributed learning and centralized learning, are examined. To combine the advantages of these approaches, Strategy-Planned Distributed Learning approach is proposed. An important problem that appears in the application of Q-learning algorithm in multi-robot domain is to define discrete and finite state and action spaces. To represent the continuous state space, three methods, Fixed-Interval Discrete State Space (FIDSS), Continuous State Space with Distribution Function (DFCSS) and Variable-Interval Discrete State Space (VIDSS), are proposed. The continuous state space is discretized by using a fixed resolution value in FIDSS, whereas the discretization process is realized by a sequential clustering-based approach in an adaptive manner in VIDSS. DFCSS method represents the continuous state space by distribution functions in continuous way. The effectiveness of proposed approaches on system performance are demonstrated by applications.

Keywords: multi-robot task allocation, Q-learning algorithm, multi-agent Q-learning algorithm, centralized learning, distributed learning, discrete state space, sequential clustering

TEŞEKKÜR

Doktora çalışmalarına başladığım günden beri yoğun programına rağmen hiçbir zaman desteğini esirgemeyen, gerek ders gerekse tez aşamasında yönlendirmeleriyle ufkumu açan, bilgi ve tecrübeleriyle her alanda bana yol gösteren danışmanım Sayın Prof. Dr. Osman PARLAKTUNA'ya teşekkürü bir borç bilirim.

Değerli fikirleriyle tez çalışmalarımı şekillendiren hocalarım Prof. Dr. Aydın AYBAR ve Yrd. Doç. Dr. Metin ÖZKAN'a teşekkür ederim.

Bu süreçte destekleriyle bana yardımcı olan tüm çalışma arkadaşlarıma teşekkür ederim. Tanıştığımız günden beri hep yanımda olan arkadaşlarım Yrd. Doç. Dr. Efnan ŞORA GÜNAL ve Doç. Dr. Serkan GÜNAL'a çok teşekkür ederim.

Çalışmalarımı ve akademik hayatımı maddi-manevi katkılarıyla destekleyen, her şartta yanımda olan eşim, annem, babam ve kardeşime ve özellikle yoğun çalışmalarım sırasında beni anlayışla karşılayan biricik oğluma sonsuz şükranlarımı sunarım.

İÇİNDEKİLER

Sayfa

ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xii
SİMGELER VE KISALTMALAR DİZİNİ	xiii
1. GİRİŞ.....	1
1.1. Problemin Tanımı ve Önerilen Yaklaşım	2
1.2. Katkılar.....	4
2. ÇOK ROBOTLU SİSTEMLER.....	5
2.1. Çok Robotlu Sistemlerin Yapısı.....	6
2.2. Çok Robotlu Sistemlerin Koordinasyonu	7
2.3. Çok Robotlu Sistemlerde Görev Atama Problemi	9
2.4. Çok Robotlu Sistemlerde Erkin Yaklaşımı	14
3. Q-ÖĞRENME YÖNTEMİ	16
3.1. Tek Erkinli Sistemlerde Q-Öğrenme Yöntemi	18
3.2. Çok Erkinli Sistemlerde Q-Öğrenme Yöntemi	20
3.3. Çok Erkinli Q-Öğrenme Yöntemi Yaklaşımları	22
3.4. Çok Robotlu Sistemlerde Q-Öğrenme Yöntemi Uygulamaları	26
3.5. Problemler	28
4. ÖNERİLEN YAKLAŞIM.....	31
4.1. Problemin Tanımlanması	34
4.1.1. Problem I: Öğrenme ihtiyacı	35
4.1.2. Problem II: Durum uzayının yapılandırılması	37

İÇİNDEKİLER (devam)

4.2. Çok Robotlu Bir Sistemin Q-Öğrenme Yöntemi Tabanlı Tanımlanması.....	38
4.2.1. Robot ortamı.....	38
4.2.2. Durum uzayı.....	39
4.2.3. Hareket uzayı	40
4.2.4. Ödül değerinin hesaplanması	41
4.2.5. Örnek bir sistem	41
4.3. Q-Öğrenme Yönteminin Çok Robotlu Sistemlerde Uygulanması.....	42
4.3.1. Merkezi öğrenme yaklaşımı	44
4.3.2. Dağıtık öğrenme yaklaşımı	46
4.3.3. Strateji planlamalı dağıtık öğrenme yaklaşımı.....	46
4.4. Durum Uzayının Belirlenmesi	49
4.4.1. Sabit aralıklı ayırık durum uzayı (SAADU) yöntemi	50
4.4.2. Dağılım fonksiyonlu sürekli durum uzayı (DFSDU) yöntemi.....	51
4.4.3. Değişken aralıklı ayırık durum uzayı (DAADU) yöntemi.....	53
4.4.3.1. Ardışıl kümeleme	54
4.4.3.2. Durum uzayının ardışıl kümeleme algoritmaları ile yapılandırılması.....	56
5. UYGULAMA	62
5.1. Uygulama Ortamı.....	62
5.2. Çok Robotlu Sistemlerin ve Görevlerin Yapısı	63
5.2.1. Görevlerin özellikleri	64
5.2.2. Çok robotlu sistemlerin yapısı	65
5.2.2.1. Çok robotlu sistem-I (ÇRS-I).....	65
5.2.2.2. Çok robotlu sistem-II (ÇRS-II)	67
5.3. Uygulama ve Hesaplamalar	68
6. UYGULAMA SONUÇLARI	70
6.1. Uygulamada Kullanılan Çok Robotlu Sistemlerin Modellenmesi.....	70
6.1.1. ÇRS-I modeli	71

İÇİNDEKİLER (devam)

6.1.2. ÇRS-II modeli	72
6.2. Q-Öğrenme Yönteminin Çok Robotlu Sistemlerde Kullanım Yaklaşımları ...	74
6.2.1. ÇRS-I için öğrenme yaklaşımlarına ilişkin sonuçlar	76
6.2.1.1. Görevlerin yapılma oranları	77
6.2.1.2. İşlemsel yük	82
6.2.2. ÇRS-II için öğrenme yaklaşımlarına ilişkin sonuçlar	84
6.2.2.1. Görevlerin yapılma oranları	84
6.2.2.2. İşlemsel yük	88
6.2.3. Öğrenme yaklaşımlarının değerlendirilmesi	89
6.3. Durum Uzayı Belirleme Yaklaşımları	90
6.3.1. Durum uzayı belirleme yaklaşımlarının incelenmesi.....	91
6.3.2. ÇRS-I için durum uzayı belirleme yaklaşımlarına ilişkin sonuçlar	93
6.3.3. ÇRS-II için durum uzayı belirleme yaklaşımlarına ilişkin sonuçlar	97
6.3.4. Durum uzayı belirleme yaklaşımlarının değerlendirilmesi	100
7. DEĞERLENDİRME	101
KAYNAKLAR DİZİNİ	104

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 2.1 Müzayede Süreci Akış Şeması	11
Şekil 3.1 Pekiştirmeli Öğrenme Modeli	17
Şekil 4.1 Temel Çok Robotlu Sistem Kontrol Mimarisi	33
Şekil 4.2 Örnek Bir Görev Akış Şeması	36
Şekil 4.3 Öğrenme Tabanlı Çok Robotlu Sistem Kontrol Mimarisi.....	43
Şekil 4.4 Sabit Aralıklı Ayırıklaştırılmış Durum Uzayı	51
Şekil 4.5 $Q_f(t)$ Fonksiyonu.....	52
Şekil 4.6 Değişken Aralıklı Ayırıklaştırılmış Durum Uzayı	53
Şekil 4.7 Dört Elemanlı Bir C_i Kümesi için Küme Parametreleri	58
Şekil 5.1 Deneysel Uygulama Ortamı	62
Şekil 6.1 ÇRS-I'de Tüm Öğrenme Yaklaşımları için Düşük ve Yüksek Öncelikli Görevlerin Yapılma Oranları	78
Şekil 6.2 ÇRS-II'de Tüm Öğrenme Yaklaşımları için Düşük ve Yüksek Öncelikli Görevlerin Yapılma Oranları	86
Şekil 6.3 ÇRS-I Ortamında Bulunan R_1 Robotuna ait Durum Uzayının SAADU Yaklaşımı için Elde Edilen Q Değerleri Grafiği	92
Şekil 6.4 ÇRS-I Ortamında Bulunan R_1 Robotuna ait Durum Uzayının DFSDU Yaklaşımı için Elde Edilen Q Değerleri Grafiği	92
Şekil 6.5 ÇRS-I Ortamında Bulunan R_1 Robotuna ait Durum uzayının DAADU Yaklaşımı ile Elde Edilen Q Değerleri Grafiği	93
Şekil 6.6 ÇRS-I'de Durum Uzay Belirleme Yaklaşımları için Düşük Öncelikli ve Yüksek Öncelikli Görevlerin Yapılma Oranları	95
Şekil 6.7 ÇRS-II'de Durum Uzay Belirleme Yaklaşımları için Düşük Öncelikli ve Yüksek Öncelikli Görevlerin Yapılma Oranları	98

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
Çizelge 4.1 R_1 ve R_2 Robotları için Robot Ortamları.....	42
Çizelge 5.1 Uygulama Ortamının Fiziksel Ölçüleri	63
Çizelge 5.2 Odalar Arası Mesafeler.....	63
Çizelge 5.3 Görevler ve Nitelikleri.....	64
Çizelge 5.4 Görevler ve Tanımlama Parametreleri	64
Çizelge 5.5 ÇRS-I için Robotlar ve Yapabildikleri Görevler	66
Çizelge 5.6 ÇRS-I için Robot Parametreleri.....	66
Çizelge 5.7 ÇRS-II için Robotlar ve Yapabildikleri Görevler.....	67
Çizelge 5.8 ÇRS-II için Robot Parametreleri	68
Çizelge 6.1 ÇRS-I’de Bulunan Robotlara İlişkin Oj^R Bilgileri.....	71
Çizelge 6.2 ÇRS-II’de Bulunan Robotlara İlişkin Oj^R Bilgileri	72
Çizelge 6.3 ÇRS-I’de Tüm Öğrenme Yaklaşımları için Görevlerin Yapılma Oranları	80
Çizelge 6.4 ÇRS-I için Öğrenme Uzayı Boyutları	83
Çizelge 6.5 ÇRS-II’de Tüm Öğrenme Yaklaşımları için Görevlerin Yapılma Oranları	87
Çizelge 6.6 ÇRS-II için Öğrenme Uzayı Boyutları	89
Çizelge 6.7 Öğrenme Yaklaşımlarının Değerlendirilmesi.....	90
Çizelge 6.8 Durum Uzayı Belirleme Yaklaşımları için Görevlerin Yapılma Oranları .	96
Çizelge 6.9 Durum Uzayı Belirleme Yaklaşımları için Görevlerin Yapılma Oranları .	99
Çizelge 6.10 Durum Uzayı Belirleme Yaklaşımlarının Değerlendirilmesi	100

SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
T_i	i 'nci görev
T	Görevler kümesi
R_j	j 'nci robot
R	Robotlar kümesi
A	Sonlu ve ayrık hareket uzayı
S	Sonlu ve ayrık durum uzayı
P	Durum geçiş olasılık fonksiyonu
ρ	Ödül fonksiyonu
γ	Azaltma faktörü
α	Öğrenme hızı
h	Hareket politikası
Q	Öğrenilen Q değeri
$ \cdot $	Uzay/ küme büyüklüğü
Δt^G	Görevlerin anons zamanları arasında geçen gerçek süre
I_j	R_j robotu için görev kümesi
S_j	R_j robotu için durum uzayı
A_j	R_j robotu için hareket uzayı
O_j^R	R_j robotu için robot ortamı
O^R	Sistem için robot ortamı
$Price(\cdot)$	Anons edilen görevin ücreti
$Rev(s, a)$	Durum-hareket çifti için ödül değeri
$I_{ışb}$	Robotlar arası işbirliğine tabi görev kümesi
ÇRS	Çok Robotlu Sistem
MKS	Markov Karar Süreci

BÖLÜM 1

GİRİŞ

Günümüzde teknolojinin gelişimiyle birlikte robotların uygulama alanları da genişlemiştir. Robotların kullanım alanları, günlük hayattaki basit işlemlerden ileri teknoloji gerektiren uygulamalara kadar geniş bir yelpazede dağılmaktadır. Ortam temizliği, ofis hizmetleri, turist rehberliği, ortam gözlemlenmesi, nesne taşıma işleri gibi gündelik süreçler, zehirli atıkların veya mayınların toplanması, kimyasal tehlike taşıyan tesislerdeki işler gibi insan sağlığı açısından risk taşıyan uygulamalar ile arama-kurtarma alanları, sualtı ve uzay araştırmaları gibi insanların bizzat ulaşamadığı yerlerdeki çalışmalar robot sistemlerinin kullanıldığı çeşitli alanlardır. Bu ve benzeri birçok uygulamada kullanılan otonom gezgin robotlar, çalışma ortamlarının ihtiyaçlarına göre dışarıdan bir yöneticiye ihtiyaç duymadan kendi kararlarını alma ve bağımsız hareket etme yetilerine sahip olmalıdır. Bu amaçla robot sistemlerinin mimarisi ve kontrolü ile ilgili araştırmalar süregelmektedir.

Karmaşık görevleri yerine getirmek üzere aynı veya farklı nitelikteki robotların belirli stratejiler doğrultusunda ortak bir çalışma alanında birlikte kullanılmaları ile çok robotlu sistemler elde edilmektedir. Çok robotlu sistemler, gerek gürbüz yapıları, işlemleri daha kısa sürede tamamlama yeteneğine sahip olmaları ve daha iyi çözümler sunmaları gibi avantajları nedeniyle, gerekse bazı uygulamaların tek bir robot tarafından yapılamayacak olması gibi mecburiyetler sonucunda çok çeşitli uygulamalarda sıklıkla kullanılmaktadır. Bununla birlikte, çok robotlu sistemlerden beklenen verimin elde edilebilmesi ancak bu sistemlerin koordinasyonlarının gerekli doğruluk ve hassasiyette yapılması ile sağlanabilmektedir. Bu sistemlerin tek robotlu sistemlere göre sundukları avantajlardan etkin şekilde yararlanabilmek için sistem kaynakları en uygun şekilde kullanılmalı, sistem koordinasyonu ve kontrolü robotların gerekli zamanda gerekli yerde olmalarını sağlayacak şekilde yapılmalıdır. Birçok gerçek ortam uygulamalarında sistem kaynaklarının sınırlı olması nedeniyle gerekli sistem performansına ulaşabilmek için robotlar arası görev paylaşımını sağlayacak koordinasyonun, sistemin toplam

kazancını en büyükmeyi veya maliyeti en küçükmeyi amaçlayarak yapılması gerekmektedir. Bu noktada, “hangi robotun hangi şartlar altında hangi görevleri yapması gerektiği” kararı büyük önem kazanmaktadır ve araştırmacıların uzun zamandır cevap aradığı bir sorudur. Bu soruya verilecek en doğru cevap için sistem kaynakları, sistemin yapısı ve uygulamanın ihtiyaçları göz önüne alınmalıdır.

Çok robotlu sistemlerin çalışma ortamları çoğu kez durağan olmayan, birçok parametreye bağlı ve ancak kısmen gözlemlenebilen bir yapıya sahiptir. Çok robotlu sistemlerin tasarımı aşamasında çalışma ortamında karşılaşılabilecek tüm olası durumların ve robotların bunlara karşı tutumlarının tam anlamıyla kestirilebilmesi mümkün değildir. Ortaya çıkabilecek problemlerin sistem performansını en az etkileyecek şekilde çözülebilmesi için robotların, bu zaman içerisinde değişebilen ortam şartlarına uyum sağlayabilmeleri ve davranışlarını bu değişen şartlara göre düzenleyebilmeleri gerekmektedir. Sistem içi koordinasyonun gerektiği şekilde gerçekleştirilebilmesi ve sistemin tam anlamda esnek ve otonom olabilmesi için robotların öğrenme yeteneğine sahip olmaları oldukça önemlidir. Öğrenme yeteneğine sahip robotların oluşturduğu sistemlerin, karşılaşılan belirsizliklere ve problemlere karşı daha gürbüz, daha güvenilir ve istenilen çalışma performansını sağlayabilecek yapıda olacakları açıktır.

1.1 Problemin Tanımı ve Önerilen Yaklaşım

Çok robotlu sistemlerde robotlar arası görev paylaşımı problemi uzun zamandır araştırmacıların üzerinde çalıştıkları bir konudur. Çok robotlu sistemlerin kullanıldığı birçok uygulamada yapılması gereken görevlerin gelme süreleri ve sıraları önceden bilinmemektedir. Dolayısıyla tüm sistem kazancını en iyileyecek şekilde bir koordinasyonun önceden yapılması mümkün olamamaktadır. Görevlerin çalışma esnasında öngörülemez zamanlarda ortaya çıkması ve görev paylaşımlarının o esnada bir göreve atanmamış robotların arasında yapılabilecek olması robot grubunun performansını olumsuz yönde etkilemektedir. Özellikle görevler arası bir hiyerarşi mevcut olduğunda öncelikle ve mutlaka yapılması istenen görevler, bu görevi

yapabilecek olan robotların daha düşük öncelikli görevlerle meşgul olmaları durumunda yapılamamaktadır. Bu noktada, akıllı görev atama yöntemi olarak nitelendirilebilecek yaklaşımlara ihtiyaç olduğu düşünülmektedir.

Bu çalışmada, çok robotlu sistemlerde robotlar arası görev paylaşım problemlerinin çözümünde sistemin verimini artırmak amacıyla öğrenme yöntemleri tabanlı yaklaşımların kullanımı, uygulanması ve uygulamada karşılaşılan problemlerin çözüm yöntemleri araştırılmıştır. Gerçekleştirilen çalışmanın temel amacı, çok robotlu bir sistemde öğrenme yaklaşımları kullanarak robotların geçmiş görev paylaşım deneyimlerini daha sonraki uygulamalarda kullanmalarını sağlamaktır. Genellikle çok robotlu sistemlerde bir görev anons edildiğinde robotlar o esnada başka bir görev yapmamaları halinde o görevi yapmaya istekli olmaktadır. Ancak, robotların görevlere doğrudan istekli olmak yerine görevleri yapma konusunda devam edecek süreçte anons edilecek yeni görevlere ilişkin beklentilerini de hesaba katarak karar verme durumunda olmaları ve geçmişte edindikleri tecrübeleri bilgiye dönüştürerek karar aşamasında bu bilgiden yararlanmaları mümkün olduğu takdirde sistemin toplam kazancında artış sağlanabileceği planlanmaktadır.

Yapılan çalışmalarda, yukarıda anlatılan temel amaca ulaşmak üzere Q-öğrenme yöntemi kullanılmaktadır. Q-öğrenme yöntemi, durum ve hareket uzaylarının sonlu ve ayrık olarak belirlenmiş olduğu ortamlar üzerinden tanımlanmıştır. Q-öğrenme yönteminin çok robotlu sistemlere uygulamasında karşılaşılan en büyük problemler; ayrık durum ve hareket uzayı kullanım zorunluluğunun getirdiği ayrıklaştırma sıkıntıları ile genelleştirilmiş durum ve hareket uzaylarının kullanımlarından doğan büyük öğrenme uzayının getirdiği işlemsel yüküdür. Bu çalışmada, bu problemleri çözmek için durum uzayının ardışıl kümeleme yaklaşımıyla ayrık olarak ve sistem gereksinimlerine göre adaptif şekilde belirlendiği bir yaklaşım önerilmiştir.

Çok robotlu sistemlerde Q-öğrenme yönteminin kullanılmasına yönelik iki temel yaklaşım bulunmaktadır. Bu yaklaşımlar merkezi ve dağıtık öğrenme yaklaşımlarıdır. Merkezi öğrenme yaklaşımı en iyi çözüme yakınsama imkanı verirken büyük öğrenme uzayı ve haberleşme yükü gibi dezavantajlara sahiptir. Dağıtık öğrenme ise uygulama

kolaylığına sahiptir fakat düşük sistem performansına neden olabilmektedir. Yapılan çalışmalarda bu iki temel yaklaşım incelenmiştir. İlaveten, dağıtık öğrenme yapısına sahip olmakla birlikte merkezi planlama yapan alternatif bir yaklaşım önerilmiş ve etkin sonuçlar sunduğu gösterilmiştir.

1.2 Katkılar

Ele alınan çözüm yaklaşımlarının çok robotlu sistemlerin görev paylaşım problemlerine sağladığı katkıları üç madde halinde sıralamak mümkündür:

- 1- Robotların görev paylaşımında öğrenme yeteneğine sahip olmaları sağlanmıştır. Böylece, robotlar geçmişte edindikleri tecrübeleri bilgiye dönüştürerek devam eden süreçte bu bilgilerden yararlanmışlardır. Böylece sistem performansının artırılması sağlanmıştır.
- 2- Q-öğrenme yönteminin çok robotlu sistemlere uygulanması için bilinen iki temel öğrenme yaklaşımı incelenerek avantaj ve dezavantajları uygulamalı olarak ortaya konulmuş ve her iki yaklaşımın avantajlarını birleştiren yeni bir yöntem önerilmiştir.
- 3- Q-öğrenme yönteminin çok robotlu sistemlere uygulanması sırasında karşılaşılan problemler ele alınmıştır. En temel problem olan ayrıklaştırma sürecine dair, sürekli durum uzayını ardışıl kümeleme yaklaşımları kullanarak adaptif olarak ayrıklaştıran bir yöntem ortaya konulmuştur. Böylece, durum uzayını sistemin yapısına göre belirleyen bir yapı sunulmuştur.

Tezin ilerleyen bölümleri şu şekildedir. Bölüm 2’de çok robotlu sistemler ve çok robotlu sistemlerde görev atama problemi anlatılmaktadır. Bölüm 3’te Q-öğrenme yöntemi teorik olarak ele alınmakta ve çok robotlu sistemlerde kullanımı incelenmektedir. Bölüm 4 karşılaşılan problemleri ve önerilen yaklaşımları içermektedir. Bölüm 5’te önerilen yöntemlerin etkinliğini göstermek için gerçekleştirilen uygulamalara ilişkin bilgiler yer almaktadır. Bölüm 6’da yapılan uygulamaların sonuçları ve Bölüm 7’de genel değerlendirme verilmektedir.

BÖLÜM 2

ÇOK ROBOTLU SİSTEMLER

Robotların kullanıldığı birçok gerçek ortam uygulamalarında, ihtiyaç duyulan sistem performansının sağlanabilmesi için tek bir robot yerine robot gruplarının kullanımı tercih edilmektedir. Çok robotlu sistemler (ÇRS) üzerindeki çalışmalar 1980'lerin sonlarında yoğunluk kazanmaya başlamıştır. Halen, gerek robot gruplarının mimarileri gerekse robotlar arası koordinasyon ve kontrol konularında birçok çalışma yapılmaktadır.

Robotik araştırmacıları için “tek robotlu bir yapı mı, yoksa çok robotlu bir sistem mi kullanılmalı?” sorusu kilit öneme sahiptir. Bu soruya verilecek tek bir cevap olamayacağı, karar verilecek noktada çalışma ortamının, sistem ihtiyaçlarının ve imkanlarının göz önünde bulundurulması gerektiği açıktır. Yapılan çalışmalarda, çok robotlu sistemlerin tek robotlu yapılara göre tercih edilmelerini sağlayacak çok sayıda avantaja sahip oldukları, bununla birlikte başta sistem maliyeti olmak üzere birçok olumsuz hususla da karşılaşıldığı belirtilmiştir (Arkin, 1998). Çok robotlu sistemlerin avantajlarını şöyle sıralamak mümkündür:

- Çok robotlu sistemler, verilen görevleri tek bir robota göre çok daha hızlı tamamlama imkanına sahiptir. Özellikle karmaşık görevlerin daha basit alt görevlere ayrılmasının mümkün olduğu durumlarda her bir alt görevin farklı robotlar tarafından eş zamanlı olarak yapılması görev tamamlama sürecini hızlandırmaktadır.
- Çok robotlu sistemlerde robotlar, aynı anda ortamdaki daha fazla alana ulaşabilir ve ortam hakkında daha fazla bilgi sahibi olabilirler. Özellikle farklı nitelikteki robotlardan oluşan bir grupta her robot, yeteneği ölçüsünde edindiği farklı bilgileri diğer robotlarla paylaşarak grubun performansını artırabilmektedir.

- Çok robotlu sistemler bir robotun tek başına yapamayacağı görevleri yapma imkanı sunmaktadır. Bir robotun, imkanlarının yetersiz kalması halinde diğer robotlarla işbirliği yaparak görevi tamamlaması olasıdır. Ayrıca eşzamanlı olarak yapılması gereken birçok görevin tek bir robot tarafından yapılması mümkün değilken, gruptaki robotlar arasında paylaşılarak tamamlanması söz konusu olabilmektedir.
- Çok robotlu sistemler meydana gelebilecek hatalara karşı daha gürbüzdür. Çalışma esnasında robotların bir veya bir kaçının fonksiyonlarını kısmen veya tamamen yitirmeleri durumunda, bu robotların yapacakları işlerin diğer robotlar tarafından tamamlanması mümkün olabilmektedir. Bu durum performans kaybı ve gecikmeye neden olsa da görevler yerine getirilebilmektedir.

Tüm bu avantajların yanı sıra çok robotlu sistemlerin kullanılmasının getirdiği bir takım dezavantajlar da mevcuttur. İlk olarak, daha büyük sistemlerin daha fazla maliyet gerektirdiğini söylemek mümkündür. Buna ilaveten, robotlar arası bilgi paylaşımını sağlamak üzere ihtiyaç duyulan haberleşme hem ek maliyet hem de ek planlama yükü anlamına gelmektedir. Birden fazla robotun bir arada kullanılması robotların hareket alanlarında kısıtlamalara ve dahası robotlar arası davranış çakışmalarına neden olabilmektedir. Ayrıca, diğer robotların davranış modelleri hakkında yeteri kadar bilgi sahibi olunamaması çok robotlu sistemlerin performansını önemli ölçüde düşürmekte ve gerekli verimin alınamamasına neden olmaktadır. Bu sayılan teknik problemlerin üstesinden gelinerek gerekli sistem performansının elde edilebilmesi, çok robotlu sistemlerin koordinasyonlarının gerekli doğruluk ve hassasiyette yapılmasına bağlıdır. Bu noktada, oluşturulacak sistem mimarisi ve sistemin kontrolünü sağlayacak yapıların seçilmesi büyük önem taşımaktadır.

2.1 Çok Robotlu Sistemlerin Yapısı

Çok robotlu sistemlerin kontrol ve koordinasyon planlamalarında grubu oluşturan robotların yapıları önemli rol oynamaktadır. Stone ve Veloso yaptıkları bir çalışmada,

robot gruplarının homojen ve heterojen olmaları durumunu ele alarak bir sınıflandırma oluşturmuşlardır (Stone and Veloso, 2000). Homojen gruplarda tüm robotlar yazılım ve donanım bakımından aynı niteliklere sahip iken heterojen gruplarda robotlar farklı yapısal özelliklere ve görev yapma yeteneklerine sahiptirler.

Birbirine eş vasıflara sahip robotların oluşturduğu homojen gruplarda robotlar genellikle aynı işi birlikte yapmak üzere organize edilmişlerdir. Robotik araştırmaları kapsamında sürü robotlar olarak adlandırılan bu yapılar kontrol ve koordinasyon kolaylıkları nedeniyle kullanımları tercih edilmektedir (Mataric, 1995).

Heterojen gruplar ise, robotların birbirlerinden farklı yetenekte olmaları sayesinde aynı ortamda bulunan farklı nitelikteki görevleri yapabilme avantajına sahiptir. Her robot, yetenekleri bağlamında bir uzmanlık alanına sahiptir. Robotların farklı alanlardaki farklı becerileri çok çeşitli görevlerin yapılmasına imkan sağlayarak sistemlerin performansına önemli katkılar sağlamaktadır. Bu sebeple yaygın olarak kullanılmaktadırlar. Bununla birlikte, heterojen robot gruplarında robotlar arası koordinasyonun hassasiyetle yapılması belirtilen avantajlardan yararlanmak için oldukça önemlidir (Rossi et al., 2009).

2.2 Çok Robotlu Sistemlerin Koordinasyonu

Çok robotlu sistemlerde kontrol ve koordinasyon yaklaşımlarını merkezi ve dağıtık olmak üzere iki farklı yapıda ele almak mümkündür. Tamamen merkezi koordinasyondan tamamen dağıtık koordinasyona kadar çok çeşitlilik gösteren kontrol yapısının seçilmesi, çalışma ortamlarının gereksinimlerine ve sistem imkanlarına bağlı olarak belirlenmektedir.

Merkezi koordinasyonda tek bir merkezi birim tüm grubun kontrolünü üstlenmekte ve robotlar arası işbirliği ve görev paylaşımını sağlamaktadır (Chaimowicz, et al., 2001). Teorik olarak ele alındığında merkezi koordinasyon yaklaşımlarında en iyi çözümlere ulaşılması beklenmektedir. Ancak, sistem performansının tek bir birime

bağlı olması nedeniyle sistemin hata toleransının azalması, özellikle büyük robot gruplarında sistemin meydana gelen değişikliklere yeterince hızlı cevap verememesi ve yoğun haberleşme ihtiyacı gibi dezavantajlar, bu yaklaşımların uygulama güçlüğüne neden olmaktadır. Yine de, sağladığı avantajlar nedeniyle çok büyük olmayan robot gruplarının özellikle durağan olarak nitelendirilebilecek ortamlarda kullanımlarında etkin bir yöntem olarak tercih edilmektedir (Dias et al., 2006).

Dağıtık koordinasyon yaklaşımlarında, her bir robot kendi yerel bilgilerini kullanarak kendi kararlarını alma ve bağımsız hareket etme yetisine sahiptir. Bu yapısı dolayısıyla merkezi koordinasyon yaklaşımlarının dezavantajlarının üstesinden gelmektedir (Dias, 2004). Dağıtık koordinasyon yaklaşımlarında, sistem meydana gelebilecek değişimlere karşı çözüm üretme kapasitesine sahiptir. Planlamanın merkezi olmaması robot grubunda meydana gelen değişikliklere karşı sistemin dayanıklılığını artırmakta ve modülerlik özelliği sağlamaktadır. Bu avantajlar ve kolay uygulanabilme vasıfları nedeniyle çok robotlu sistemlerde yaygın olarak kullanılmaktadır. Ancak, robotların kendi yerel bilgilerini kullanarak kendi davranışlarını belirlemesi sonucu ortaya çıkan davranış çakışmaları ve yerel bilgilere bağlı kalınması, bulunan çözümlerin sistemin performansını artırma açısından yetersiz kalmasına neden olabilmektedir (Dias et al., 2006).

Market-tabanlı yaklaşımlar, merkezi ve dağıtık koordinasyon yaklaşımlarının avantajlarını birleştiren ve çok robotlu sistemlerde koordinasyon problemini çözmek üzere yaygın olarak kullanılan yöntemlerdir (Dias et al., 2006). Bu yaklaşımlarda, bir robot yine kendi karar mekanizmasına sahiptir ancak karar alma sürecine diğer robotların da dahil olması söz konusudur. Robotlar arası haberleşme trafiği merkezi yaklaşımlarda olduğu kadar yoğun ve kritik değildir. Karar alma sırasında robotlar arası müzakerelerin gerçekleştirilmesi amacıyla kullanılmaktadır. Bu yaklaşımlar için önemli olan husus, müzakere süreci ve kurallarının hassasiyetle belirlenmesi ve karar alma mekanizmalarının sistem ihtiyacına göre tasarlanmasıdır.

2.3 Çok Robotlu Sistemlerde Görev Atama Problemi

Robotların yer aldığı birçok uygulamada, gerekli sistem performansının sağlanabilmesi için tek bir robot yerine robot gruplarının kullanılması gerekmektedir. Verilen görevlerin etkin, başarılı ve güvenilir bir biçimde yerine getirilebilmesi ancak sistem kaynaklarının robotlar arasında uygun bir şekilde dağıtılmasını sağlayacak grup koordinasyonu ile mümkün olmaktadır. Çok robotlu sistemlerde görev atama problemi, yapılması gereken görevlerin görevi yapacak robotlar arasında sistem kaynaklarının en iyi şekilde kullanılmasını sağlayacak şekilde dağıtılması süreci olarak tanımlanmaktadır (Gerkey and Mataric, 2004). Çok robotlu sistemlerde robotlar ve robotların yetenekleri sistem kaynakları olarak nitelendirilmektedir. Gerçekte, birçok durumda sistem kaynaklarının yetersizliğinden dolayı tüm işlerin yapılması mümkün olamamaktadır (Jones et al., 2007). Bu durum, robotlar arası görev paylaşımının etkin bir şekilde yapılması gerektiğinin önemini ortaya koymaktadır. Görevlerin etkin bir şekilde dağıtılması, sistem kaynaklarının en iyi şekilde kullanılmasını sağlamak üzere ya sistem kazancının en büyüklenmesi ya da maliyetinin en küçüklenmesi amaçlanarak yapılmaktadır. Sistem kazancının veya maliyetinin değerlendirilmesi görev yapma niteliği, yapılan görevden sistemin sağladığı fayda veya bu görevlerin sisteme maliyeti gibi çeşitli faktörler üzerinden gerçekleştirilmektedir (Dias et al., 2006).

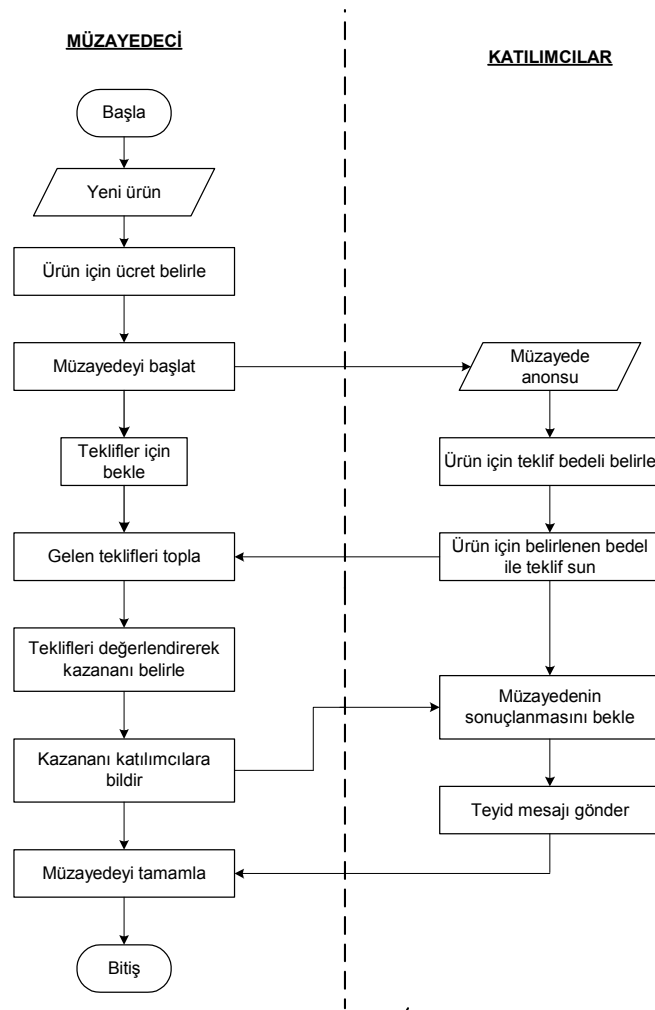
Gerkey ve Mataric, çok robotlu sistemlerde görev atama problemlerinin çözümüne yönelik bir sınıflandırma geliştirmişlerdir (Gerkey and Mataric, 2004). Görev atama problemleri robotların yapısına, görevlerin yapısına ve görev sürecinin yapısına göre üç ayrı sınıfta incelenmiştir. İlk sınıflandırma bir anda tek bir görev yapan robotlar ve aynı anda birden fazla görev yapan robotlar şeklinde robotların yapısına göre ele alınmıştır. Benzer şekilde görevlerin yapısına göre yapılan sınıflandırmada görevler tek bir robot tarafından yapılan görevler ve çok robot tarafından yapılması gereken görevler olarak ikiye ayrılmıştır. Sürecin yapısına göre yapılan sınıflandırmada ise görevler hakkındaki bilgi temel alınmıştır. İlk durumda yapılması istenen görevler ve görevlerin sırası önceden bilinmekte ve görevlerin robotlara nasıl dağıtılacağı önceden planlanabilmektedir. Bu da robotların görev atamalarını sistem performansını en iyileyecek şekilde yapmayı mümkün kılmaktadır.

İkinci durumda ise görevler aniden ortaya çıkmakta ve görev dağıtımı görevler ortaya çıktıktan sonra o esnada yapılmaktadır. Bu yapıda, en iyi çözüme ulaşılabilmesi için kullanılacak görev atama yaklaşımının ortamdaki değişiklikleri algılama ve esnek karar verme yeteneklerine sahip olması gerekmektedir .

Bir başka sınıflandırma ise çok robotlu sistemlerin çalışma ortamlarının yapısı temel alınarak yapılmıştır (Parker, 2008). Durağan ortamlar için önerilen görev atama yaklaşımları ortam, robotlar ve görevler hakkında tam bir bilgi sahibi olduğunu kabul etmektedir. Uygulama kolaylığı sağladığı halde, ortamdaki belirsizlikleri ve meydana gelen değişiklikleri temsil etmekte yetersiz kalmaktadır. Dinamik ortamlarda görev atama yaklaşımları ise ortamdaki değişikliklere gerçek zamanlı olarak uyum sağlamakta ve uygun kararları verme imkanı sunmaktadır.

Market tabanlı yaklaşımlar, merkezi ve dağıtık koordinasyon yaklaşımlarının avantajlarını birleştiren ve çok robotlu sistemlerde görev paylaşımı problemini çözmek üzere yaygın olarak kullanılan yöntemlerdir (Goldberg et al., 2003; Zlot and Stentz, 2006). Müzayede (auction) yöntemi çok robotlu sistemlerde market tabanlı koordinasyon yaklaşımı olarak yaygın şekilde kullanılmaktadır (Gerkey and Mataric, 2002). Müzayede protokollerinin temel avantajı, sistemde meydana gelebilecek arızalara karşı gürbüz bir yapı ortaya koymalarıdır (Spaan et al., 2010). Müzayede süreci, müzayedeci olarak adlandırılan merkezi bir birim (auctioneer) tarafından ortaya konan bir nesne (item) için katılımcıların (bidder) tekliflerini sunması ve müzayedeci birimin bu teklifleri değerlendirerek kazanan erkini belirlemesi şeklinde gerçekleştirilmektedir. Müzayedeci birim, belirli bir nesne için müzayedeyi başlattıktan sonra katılımcıların tekliflerini beklemeye başlar. Müzayede anonsunu alan katılımcılar, müzayedeye sunulan ürünün kendileri için kazancını veya maliyetini hesaplayarak tekliflerini hazırlar ve bu teklifleri müzayedeci birime gönderir. Tekliflerin gelmesi için belirlenen süre dolduğunda müzayedeci kazananı belirleyerek katılımcılara bildirir. Kazanan katılımcı bunu teyid edecek mesajı müzayedeciye gönderdiğinde müzayede süreci tamamlanmaktadır (Gerkey and Mataric, 2002).

Müzayedeleri, nesnelerin müzayedeye sunulma şekline göre iki grupta ele almak mümkündür. Tek ürünlü müzayede yapısı en basit müzayede yapısı olup, her seferinde sadece bir nesne için müzayede yapılması imkanı sunmaktadır. En yüksek teklifi veren katılımcı müzayedeyi kazanmaktadır. Tümleşik müzayede ise daha karmaşık bir yapıya sahiptir. Aynı anda birden fazla sayıda nesne için müzayede yapılmaktadır. Katılımcılar kendileri için uygun olan ürün veya ürün grupları için teklif sunmaktadır (Berhault et al., 2003). Tümleşik müzayede yapısının özel bir durumu olan çok ürünlü müzayede de ise müzayedeye çok sayıda ürün sunulduğu halde bir katılımcı sadece bir ürün için kazanan katılımcı olabilmektedir (Dias et al., 2006). Müzayede sürecine ilişkin akış şeması Şekil 2.1’de verilmektedir.



Şekil 2.1 Müzayede Süreci Akış Şeması

Müzayede yönteminin çok robotlu sistemlerde kullanımında, müzayedeci olarak görevi anons eden birim, katılımcı olarak görevi yapabilecek robotlar ve ürün olarak da görevler kullanılmaktadır. Süreci aşağıda verilen üç temel adım şeklinde ele almak mümkündür:

- Adım 1: Yapılması gereken görevlerin robotlara anons edilmesi
- Adım 2: Robotların görevler için uygun bedelleri belirleyerek müzayedeye katılmaları
- Adım 3: Görevleri anons eden birim tarafından görevlerin robotlara atanması

Müzayede sürecinin ilk adımı olarak görevlerin robotlara atamasını yapacak olan birim, görevleri tüm robotlara anons ederek gelecek teklifleri beklemeye başlamaktadır. Teklif bekleme süreci ya belirlenen süre dolunca ya da belirli sayıda katılımcı tekliflerini bildirince tamamlanmaktadır.

Müzayede sürecinin ikinci adımı robotların yapacakları görevler için müzayedeye sunacakları teklif bedellerinin hesaplanmasıdır. Bu teklif bedelleri temel olarak robotların ilişkili görevleri yapmaları durumuna uygunluklarını temsil etmektedir. Gezgin robot uygulamalarında teklif bedeli genellikle görevi yapma sürecinde alınan yol, harcanan süre veya her ikisinin birlikte kullanımı şeklinde hesaplanmaktadır (Mosteo and Montano, 2007). İlaveten, bir diğer çalışmada, müzayede teklif bedeli görevlerin yapılması sırasında harcanan gerçek enerji modelleri kullanılarak hesaplanmış ve etkin çözümler sunduğu gösterilmiştir (Kaleci et al., 2010).

Üçüncü adımda ise müzayedeye katılım süreci tamamlandıktan sonra, müzayedeci gelen teklifleri değerlendirerek görevlerin robotlara atamasını gerçekleştirmektedir. Bu görev atama problemi yöneylem araştırması çalışmalarında yaygın olarak kullanılan optimal atama problemi (optimal assignment problem) olarak tanımlanmaktadır (Gerkey and Mataric, 2004). Bu problem, robot gruplarında görev atama problemine uygulandığında, n adet görevin m adet robota atanması problemi olarak aşağıdaki (2.4)-(2.6) denklemleriyle tanımlanmaktadır (Gerkey and Mataric, 2004).

Görevler kümesi:

$$T = \{T_i | i = 1, 2, \dots, n\} \quad (2.1)$$

Robotlar kümesi:

$$R = \{R_j | j = 1, 2, \dots, m\} \quad (2.2)$$

T_i görevinin R_j robotuna atanması durumunda elde edilecek kazanç değeri c_{ij} ve karar değişkeni:

$$x_{ij} = \begin{cases} 1, & T_i \text{ görevi } R_j \text{ robotuna atanmış ise,} \\ 0, & \text{değilse} \end{cases} \quad (2.3)$$

olmak üzere, robot grubunun toplam kazancını en büyükmek üzere amaç fonksiyonu:

$$f = \left(\sum_{i=1}^n \sum_{j=1}^m x_{ij} c_{ij} \right) \quad (2.4)$$

ve kısıtlar:

$$\sum_{i=1}^n x_{ij} = 1, \forall j \quad (2.5)$$

$$\sum_{j=1}^m x_{ij} = 1, \forall i \quad (2.6)$$

şeklinde tanımlanmaktadır. Denklem (2.5) ile verilen kısıt görevin tek bir robota atanabileceğini ve Denklem (2.6) ile verilen kısıt ise her robota sadece bir görev atanabileceğini belirtmektedir. Bu problemi en iyileyen çözüm, Denklem (2.4) ile tanımlanan toplam kazanç fonksiyonunu en büyükleyecek x_{ij} tamsayılarının belirlenmesiyle mümkün olmaktadır.

Hatime ve diğerleri, yukarıda tanımlanan optimal atama problemini çözmek üzere üç farklı yöntem kullanmışlar ve elde edilen sonuçları farklı performans kriterleri

açısından değerlendirmişlerdir (Hatime et al., 2013). Kullanılan üç yöntem; yapay sinir ağları tabanlı bir yaklaşım, tamsayılı doğrusal programlama yaklaşımı ve Macar Algoritması (Hungarian Algorithm) yaklaşımlarıdır. Macar Algoritması yaklaşımının özellikle robotlar arası yük dağılımı açısından en iyi çözüme ulaştığı ve gerek maliyet gerekse süre kriterleri açısından ise kabul edilebilir sonuçlar verdiği ortaya konulmuştur. Bu optimal atama problemini bir lineer programlama çözüm yöntemi olan Macar Algoritması ile çözülmesi birçok çalışmada önerilmiştir (Gerkey and Mataric, 2004). Macar Algoritması Denklem (2.7) ile verilen kazanç matrisini kullanarak

$$\mathcal{C} = [c_{ij}] \quad (2.7)$$

karar değişkenleri x_{ij} tamsayılarını belirlemektedir. x_{ij} değeri '1' ise T_i görevi R_j robotuna atandığı anlamına gelmektedir.

2.4 Çok Robotlu Sistemlerde Erkin Yaklaşımı

Erkin, dışarıdan bir başka etken tarafından bir yönlendirme yapılmadan bulunduğu ortamda algılarını kullanarak amaca yönelik olarak karar verme ve hareket etme yeteneğine sahip yazılım veya donanım birimi olarak tanımlanmaktadır (Wooldridge, 2002). Otonom olmaları yani kendi başına karar verme ve harekete geçme yeteneklerinin olması, sadece algılarına göre değil amaca yönelik hareket etmeleri ve diğer erkinlerle iletişim kurabilme yeteneklerine sahip olmaları erkinlerin en önemli özellikleridir.

Çok erkinli sistem (ÇES), ortak bir çalışma alanını paylaşan ve etkileşim halinde olan otonom erkinlerin oluşturduğu grup olarak tanımlanmaktadır. Çok erkinli sistemler tek erkinli sistemlerin tek başına çözemediği karmaşık problem ortamlarında kullanıldıkları için erkinler arası koordinasyon, işbirliği, görev paylaşımı ve kontrollerin hassasiyetle yapılması gerekmektedir. Çok erkinli sistemlerin temel özelliklerini aşağıdaki gibi özetlemek mümkündür (Stone and Veloso, 2000):

- Esnek problem çözme yeteneği: Farklı erkinlerin farklı özelliklere sahip olmaları sayesinde karmaşık ve zor görevlerin yerine getirilmesi mümkün olmaktadır.
- Paralel işlem yapma kapasitesi: Karmaşık görevlerin yerine getirilmesi sırasında farklı erkinleri farklı görevlerle yetkilendirerek eş zamanlı çalışma ortamı oluşturmak mümkündür. Paralel çalışma yeteneği sistemin karmaşık işler karşısında başarısını artırmaktadır.
- Gerçek zamanlı çalışma yeteneği: Paralel çalışma yeteneği sayesinde erkinlerin aynı anda farklı görevler icra edebilmeleri sistemin çalışmasını hızlandırmaktadır. Böylece, zaman kısıtına sahip uygulamalarda gerçek zamanlı çalışma imkanı sağlanmaktadır.
- Gürbüzlük: Erkinlerin iyi koordine edildiği sistemlerde bir veya daha fazla erkinin hatası veya çalışma dışı kalması diğer erkinler tarafından telafi edilebilmektedir.
- Modülerlik: Sistemlerin esnek yapısı yeni erkinlerin ve dolayısıyla yeni yeteneklerin sisteme dahil edilmesine imkan sağlamaktadır.

Yukarıda yer alan özellikler çok erkinli sistemlerin farklı alanlarda, farklı görevleri yerine getiren sistemlerin modellenmesinde kullanılmalarına olanak sağlamaktadır. Çok erkinli sistemlerin kullanıldığı önemli alanlardan biri de çok robotlu sistemlerin modellenmesidir. Özellikle gerçek zamanlı çalışma, gürbüzlük ve modülerlik özellikleri sayesinde çok robotlu sistemlerin modellenmesinde ve kontrolünde oldukça etkin olan yaklaşımlar sağlamaktadır. Böylece çok erkinli sistemler için önerilen yaklaşımların çok robotlu sistemlerde de uygulanması mümkün olmaktadır. Her bir robot bir erkin olarak modellenenebildiği gibi ilaveten algılayıcı, karar verme birimi, koordinasyon merkezi gibi çeşitli yazılım erkinlerinin tanımlanması da mümkündür. Robot sistemlerinin erkin tabanlı olarak tanımlandığı, erkinler üzerinde teorik olarak başarımları ispatlanmış çeşitli yöntemlerin robot sistemlerine uygulandığı ve etkin çözümlerin elde edildiği çok sayıda çalışma bulunmaktadır (Hwang,et al., 2009a).

BÖLÜM 3

Q-ÖĞRENME YÖNTEMİ

Erkinler, çoğunlukla ancak kısmi olarak gözlemlenebilen ve dinamik yani zaman içerisinde değişen ortamlarda kullanılmaktadır. Her zaman için ortam hakkında net bilgilere sahip olunamaması, ortamın tüm durumlarının gözlemlenememesi ve özellikle çok erkinli yapılarda diğer erkinlerin davranışlarının sistem üzerindeki etkilerinin kestirilememesi nedeniyle karşılaşılan tüm kontrol ve koordinasyon problemlerinin klasik yöntemlerle çözülmesi mümkün olamamaktadır. Bu durumda oluşabilecek problemlerin üstesinden gelmek üzere erkinlerin değişen durumları algılayıp buna uygun çözümler üretme yeteneğine sahip olmaları gerekmektedir. Bu da, ancak sistemlerin öğrenme özelliğine sahip olması ile sağlanabilmektedir. Öğrenme yeteneği, sisteme yeni bilgi katması, farklı verileri bir araya getirerek genelleştirilmiş sonuçlar elde etmesi ve geçmiş tecrübeleri gelecek çalışmalarda kullanabilir hale getirmesi gibi nitelikleri sebebiyle sistemin performansını geliştirmektedir.

Makine öğrenmesi yaklaşımlarının amacı bir erkinin değişen ortam durumlarına rağmen verilen bir girdi kümesi için uygun çıktı değerlerini elde edebilme yeteneğini artırmaktır (Mitchell, 1997). Makine öğrenmesi yöntemlerini üç temel kategoride ele almak mümkündür (Russel and Norvig, 2003):

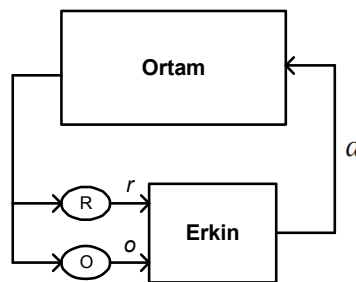
- Öğreticili Öğrenme (supervised learning)
- Öğreticisiz Öğrenme (unsupervised learning)
- Pekiştirmeli Öğrenme (reinforcement learning)

Öğreticili öğrenme yöntemlerinde sistem verilen girdi-çıkı veri kümeleri üzerinden aradaki ilişkiyi tanımlayacak bir model üretmektedir. Kesin tanımlı girdi-çıkı verilerine ihtiyaç duymaktadır. Erkinlerin, tamamen gözlemlenebilir ortamlarda yaptıkları hareketlerin sonuçlarını gözlemleyerek hareket-çıkı ilişkilerini öğrenmeleri mümkündür. Fakat kısmi gözlemlenebilir ortamlarda, hareketlerin sonuçlarının etkileri

tam olarak kestirilememesi problemi karmaşık hale getirmektedir (Russel and Norvig, 2003).

Öğreticisiz öğrenme yöntemlerinde sistem sadece girdi bilgilerine sahiptir. Öğrenme, girdi değerleri arasındaki ilişki kullanılarak gerçekleştirilmektedir. Genellikle sınıflandırma çalışmaları için kullanılmaktadır (Russel and Norvig, 2003).

Pekiştirmeli öğrenme yöntemleri ise doğrudan girdi-çıkı veri kümelerine ve bir öğreticiye ihtiyaç duymayan makine öğrenmesi yaklaşımlarıdır. Pekiştirmeli öğrenme yöntemlerinde erkin ortama algılama ve hareket birimleri üzerinden doğrudan bağlıdır. Öğrenme süreci, yapılan bir hareketin karşılığında ortam tarafından erkine gönderilen geri besleme bilgisi ile gerçekleşmektedir. Temel bir pekiştirmeli öğrenme yapısı Şekil 3.1'de gösterilmektedir. Başlangıçta "s" durumunda bulunan erkin, algılayıcıları aracılığıyla sistemden "o" gözlem bilgilerini almakta ve buna bağlı olarak bir "a" hareketi üretmektedir. Gerçekleştirilen hareketin sonucunda sistem durum değiştirerek "s'" durumuna geçmekte ve sistemin durumuna ilişkin bir değerlendirme sunan "r" ödül değeri erkine gönderilmektedir. Hareketin sonucunda sistem durumlarının amaçlanan duruma yakınsaması halinde bu geri besleme hediye, beklenenden uzaklaşması halinde ise ceza şeklini almaktadır. Erkin davranışını, bu ödül değerinin uzun vadeli toplamını en büyükleyecek şekilde belirlemektedir (Kaelbling et al., 1996). Bir öğreticiye ihtiyaç duymaması, kısmi gözlemlenebilen ve belirsizlik içeren ortamlarda kullanılma imkânı sağlaması, yapısal olarak basit olmaları ve gerçek-zamanlı uygulamalardaki yüksek performansı sebebiyle özellikle çok erkinli sistem uygulamalarında tercih edilmektedir.



Şekil 3.1 Pekiştirmeli Öğrenme Modeli

3.1 Tek Erkinli Sistemlerde Q-Öğrenme Yöntemi

Standart pekiştirmeli öğrenme yaklaşımlarında ortam Markov Karar Süreci (Markov Decision Process) olarak tanımlanmaktadır. Ortamın ayrık (discrete) ve sonlu (finite) durumları kümesi S , erkinin ayrık ve sonlu hareket kümesi A , her bir durum-hareket çifti için durum geçiş olasılık fonksiyonu $P: S \times A \times S \rightarrow \Pi(S): [0,1]$ ve her bir durum-hareket çifti için erkinin aldığı ödül fonksiyonu $\rho: S \times A \times S \rightarrow \mathbb{R}$ olmak üzere bir Markov Karar Süreci (MKS) $\langle S, A, P, \rho \rangle$ kümesi ile tanımlanmaktadır (Buşoniu et al., 2008).

Herhangi bir ayrık k adımı için $s(k) \in S$ durumunda bulunan ortam erkinin yaptığı $a(k) \in A$ hareketi ile durum geçiş olasılık fonksiyonuna bağlı olarak $s(k+1)$ durumuna geçmektedir. Yaptığı $a(k)$ hareketinin sonunda erkin $r(k) = \rho(s(k), a(k), s(k+1))$ ödül değerini almaktadır. Bu ödül değeri, ortamı $s(k)$ durumundan $s(k+1)$ durumuna geçiren $a(k)$ hareketinin ortam üzerindeki anlık etkisini temsil etmektedir (Buşoniu et al., 2008). Bir erkinin mevcut durum için hareketini nasıl seçtiğini erkinin hareket politikası h belirlemektedir. Bir hareket politikası olasılıksal ($h: S \times A \rightarrow [0,1]$) veya deterministik ($h: S \rightarrow A$) olabilmektedir. Zaman içerisinde değişmeyen hareket politikalarına durağan hareket politikası denir. Her MKS deterministik, durağan ve optimal bir hareket politikasına sahiptir (Yang and Gu, 2004). MKS olarak tanımlanan bir ortamda bulunan bir erkinin amacı, her adımda beklenen uzun vadeli toplam ödül değerini en büyükmektir. $\gamma \in [0,1)$ azaltma faktörü olmak üzere herhangi bir ayrık k adımı için uzun vadeli toplam kazancın beklenen değeri aşağıdaki ifade ile verilmektedir (Kaelbling et al., 1996).

$$E\{\sum_{i=0}^{\infty} \gamma^i r(k+i)\} \quad (3.1)$$

Erkin, ortamdan sadece anlık hareketine ilişkin ödül değeri aldığı halde uzun vadeli toplam kazancını en büyükmeyi hedeflemektedir. Bunun için kullanılan bir yöntem hareket-değer fonksiyonunu hesaplamaktır (Buşoniu et al., 2008). Hareket-

değer fonksiyonu $Q^h: S \times A \rightarrow \mathbb{R}$; mevcut "h" hareket politikasına bağlı olarak durum-hareket çiftinin beklenen toplam kazancını ifade etmektedir.

$$Q^h(s, a) = E\{\sum_{i=0}^{\infty} \gamma^i r(k+i) | s(k) = s, a(k) = a, h\}, \forall k \quad (3.2)$$

Optimal durum-hareket fonksiyonu olarak tanımlanan Q-fonksiyonu Denklem (3.3) ile verilmektedir.

$$Q^*(s, a) = \max_h Q^h(s, a), \forall s \in S \text{ ve } \forall a \in A \quad (3.3)$$

Bu ifade, Denklem (3.4) ile verilen Bellman optimallik eşitliğini sağlamaktadır (Tuyls and Nowè, 2005). Bellman optimallik prensibi; bir optimal hareket politikasının başlangıçtaki durum ve başlangıçtaki hareketten bağımsız olarak mevcut andan sonraki kararların mevcut andaki hareketin sonucunda elde edilen duruma bağlı olarak bir optimal politika oluşturma özelliğine sahip olması gerektiğini söylemektedir.

$$Q^*(s, a) = \sum_{s' \in S} P(s, a, s') [\rho(s, a, s') + \gamma \max_{a'} Q^*(s', a')] \quad (3.4)$$

Denklem (3.4) ile verilen eşitlik, s durumunda a hareketini yapmanın optimal değerinin anlık ödülün beklenen değeri ile sonraki durumlardan elde edilebilecek optimal değerlerin toplamına eşit olacağını ifade etmektedir. Durum uzayı sonlu boyutlu olduğunda toplamsal olarak yazılabilmektedir (Sutton and Barto, 1998).

Erkin öğrenme amacına ulaşmak için önce optimal Q^* değerini hesaplamalı, sonra da Q^* değerini sağlayacak hareket politikasını kullanarak yapacağı hareketi seçmelidir. Bu öğrenme sürecini gerçekleştirmek üzere birçok pekiştirmeli öğrenme yöntemi bulunmaktadır. Bu yöntemleri; dinamik programlama tabanlı model destekli yaklaşımlar ve değer fonksiyonunun tahmin edilmesi tabanlı model desteksiz yaklaşımlar şeklinde sınıflandırmak mümkündür (Yang and Gu, 2004). Q-öğrenme yöntemi, erkin tabanlı sistemler için yaygın olarak kullanılan değer fonksiyonu tabanlı model desteksiz bir pekiştirmeli öğrenme yöntemidir ve her bir durum-hareket çifti için

elde edilen optimal Q değerlerini öğrenmektedir (Sutton and Barto, 1998). İlk olarak Watkins tarafından ortaya konulmuştur (Watkins, 1989). MKS olarak tanımlanan ortam için dinamik programlama yöntemleriyle uygulanabileceği açıklanarak çeşitli algoritmalar önerilmiştir.

Q -öğrenme algoritması optimal Q değerinin yinelemeli olarak hesaplandığı aşağıdaki denklem ile ifade edilmektedir:

$$V = \max_{a' \in A} Q_k(s(k+1), a')$$

$$Q_{k+1}(s(k), a(k)) = (1 - \alpha(k))Q_k(s(k), a(k)) + \alpha(k)[r(k) + \gamma V] \quad (3.5)$$

Bu denklem ortam modeline, durum geçiş olasılık fonksiyonlarına ve ödül fonksiyonuna ihtiyaç duymamaktadır. Burada $\alpha(k)$ öğrenme hızını, γ azaltma faktörünü ifade etmektedir. $\max_{a' \in A} Q_k(s(k+1), a')$ ifadesi $a(k)$ hareketi ile geçilen $s(k+1)$ durumunda yapılabilecek en iyi harekete ilişkin Q değerini göstermektedir. Öğrenme hızı $\alpha(k)$ aşağıda verilen şartları sağlamalıdır. Bu şartlar $\alpha(k) \in (0,1]$ olması gerektiğini söylemektedir.

$$\sum_{k=1}^{\infty} \alpha(k) = \infty \text{ ve } \sum_{k=1}^{\infty} (\alpha(k))^2 < \infty \quad (3.6)$$

Eğer her hareket, her bir durum için sonsuz defa gerçekleşiyor ve $\alpha(k)$ öğrenme hızı her bir adımda uygun şekilde azaltılıyorsa yukarıdaki denklem ile hesaplanan Q değerleri Q^* 'a 1 olasılık ile yakınsamaktadır (Watkins and Dayan, 1992).

3.2 Çok Erkinli Sistemlerde Q -Öğrenme Yöntemi

Olasılıksal oyun (Stochastic Game) Markov Karar Sürecinin çok erkinli sistemler için genelleştirilmiş halidir. m erkin sayısı, S ayrık ve sonlu ortam durumları kümesi, $A = A_1 \times A_2 \times \dots \times A_m$ erkinlerin genelleştirilmiş hareket kümesi, $P: S \times A \times S \rightarrow \Pi(S): [0,1]$ durum geçiş olasılık fonksiyonu, $\rho_j: S \times A \times S \rightarrow \mathbb{R}, j = 1 \dots m$ her bir

durum-hareket çifti için her bir erkinin aldığı ödül fonksiyonu olmak üzere olasılıksal oyun (OO) $\langle S, A, P, \rho_j \rangle$ kümesi ile tanımlanmaktadır (Buşoniu et al., 2008). Burada, durum geçişleri tüm erkinlerin tümleşik hareketleri sonucunda gerçekleşmektedir.

Olasılıksal oyunlar için önerilen çözüm yöntemlerinden biri Nash dengesinin sağlanmasıdır (Yang and Gu, 2004). Bir j erkinin diğer erkinlerin hareket politikalarına karşı ortaya koyduğu hareket politikası, h_j^* , verilen tüm hareket politikaları için beklenen ödül değerini en büyüklüyorsa aşağıda verilen ifade sağlanmaktadır.

$$E\{\rho_j | h_1, \dots, h_j^*, \dots, h_m\} \geq E\{\rho_j | h_1, \dots, h_j, \dots, h_m\}, \forall h_j, j = 1, \dots, m \quad (3.7)$$

Tüm erkinler için bu şartı sağlayan tümleşik hareket politikaları $\langle h_1^*, \dots, h_j^*, \dots, h_m^* \rangle$ kümesi Nash dengesi olarak tanımlanmaktadır (Buşoniu et al., 2008). Buna göre, diğer erkinlerin hareket politikaları aynı kaldığı sürece bir erkinin hareket politikasını değiştirerek toplam kazancı artıramayacağı açıktır.

Hu ve Wellman, öğrenme sürecinin Nash dengesinin hesaplanarak gerçekleştirildiği Nash-Q-öğrenme yöntemini geliştirmişlerdir (Hu and Wellman, 1998). Nash-Q-öğrenme yönteminin belirli şartlar altında optimal sonuca yakınsadığı gösterilmiştir (Boeling, 2000; Hu and Wellman, 2003). Bu yöntemde her bir erkin j için Q fonksiyonları

$$Nash_j(s, Q^1, \dots, Q^j, \dots, Q^m) = Q^j(s, h_1^*, \dots, h_j^*, \dots, h_m^*) \quad (3.8)$$

olmak koşuluyla Denklem (3.9) ile gösterildiği şekilde güncellenmektedir.

$$V_N = Nash_j(s, Q^1, \dots, Q^j, \dots, Q^m) \\ Q^j(s, a_1, \dots, a_m) = (1 - \alpha)Q^j(s, a_1, \dots, a_m) + \alpha[\rho_j + \gamma V_N] \quad (3.9)$$

Eğer bir olasılıksal oyun (OO) için $\rho_1 = \dots = \rho_m$ ise tüm erkinler aynı amaca yönelik olarak ortak bir kazancı en büyüklemeye çalışırlar. Bu durumda OO tamamen

işbirlikçi olarak adlandırılır. Eğer $\rho_1 + \dots + \rho_m = 0$ ise OO tamamen rekabetçi olarak nitelendirilir ve bazı erkinler kendi kazançlarını en büyükmeye, diğerlerinin kazançlarını da en küçükmeye çalışırlar. Bu iki durumun arasında kalan oyunlar da mevcuttur. Tamamen işbirlikçi bir oyun için Nash dengesi şöyle ifade edilir:

$$Nash_j(s, Q^1, \dots, Q^j, \dots, Q^m) = \max_{a_1 \in A_1, \dots, a_m \in A_m} Q^j(s, a_1, \dots, a_m) \quad (3.10)$$

Q-değerleri ise Denklem (3.11) ile güncellenir.

$$V = Nash_j(s, Q^1, \dots, Q^j, \dots, Q^m) \\ Q^j(s, a_1, \dots, a_m) = (1 - \alpha)Q^j(s, a_1, \dots, a_m) + \alpha[\rho + \gamma V] \quad (3.11)$$

Tamamen işbirlikçi bir oyunda tüm erkinlerin tümleşik hareketleri için ortak Q-değerlerinin öğrenildiği durumda bu oyunun çok erkinli MKS olarak tanımlanabileceği belirtilmiştir (Boutlier, 1996). Ancak, bir OO'da genellikle birden çok Nash denge noktası bulunmaktadır. Birbirlerinden bağımsız karar verme ve hareket etme yeteneğine sahip erkinlerin aynı Nash dengesini sağlayacak tümleşik hareketleri seçmelerini sağlamak büyük bir problem teşkil etmektedir. Yukarıdaki tanımlamanın geçerli olabilmesi için tüm erkinlerin aynı Nash dengesine ulaşacak hareket politikalarına sahip olmaları gerekmektedir. Bunun için ya merkezi bir koordinasyon biriminin bulunmasına ya da erkinler arası güvenilir ve sürekli haberleşme ağının sağlanmasına ihtiyaç duymaktadır.

3.3 Çok Erkinli Q-Öğrenme Yöntemi Yaklaşımları

Q-öğrenme yönteminin tek erkinli sistemlere ve çok erkinli sistemlere uygulanması arasındaki temel fark ortamın durağan ve tamamen gözlemlenebilir olmasıyla yani MKS olarak modellenebilmesiyle ilgilidir. Çok erkinli sistemlerde, erkinlerin bağımsız olarak hareket edebilme yeteneğine sahip olmaları ve hareketlerinin sonucunda ortam durumlarında meydana gelen öngörülemeyen değişiklikler sebebiyle

ortamı durağan olarak tanımlamak mümkün değildir. Bu durum, pekiştirmeli öğrenme yöntemlerinin temel varsayımı olan ortamın MKS olarak tanımlanması şartı ile çelişmektedir. Tek erkinli sistemler için geleneksel Q-öğrenme algoritması başarıyla uygulanırken, bu algoritmanın çok erkinli sistemlerdeki doğrudan uygulamalarında optimal sonuca yakınsama garanti edilememektedir (Matignon et al., 2007).

Lauer ve Riedmiller gerçekleştirdikleri çalışmalarında, erkinlerin ortamdan aldıkları bilgileri dikkate alan iki temel yapıyı ele almışlardır (Lauer and Riedmiller, 2000). İlk yapıda erkinler hem kendi hareketleri hem de diğer erkinlerin hareketleri hakkında bilgi edinmektedir. İkinci yapıda ise erkinler sadece kendi hareketlerini bilmektedir. Erkinler ilk yapıda tümleşik hareket kullanıcılar olarak, ikinci yapıda ise bağımsız hareket ediciler olarak adlandırılmışlardır (Claus and Boutilier, 1998).

Q-öğrenme yönteminin çok erkinli sistemlerde uygulamasına ilişkin yukarıda yer alan yapıya dayanan iki temel yaklaşım bulunmaktadır: dağıtık öğrenme ve merkezi öğrenme (Wang and de Silva, 2006a; Matignon et al., 2007).

İlk yaklaşım olan dağıtık öğrenme yaklaşımı tek erkinli Q-öğrenme yönteminin doğrudan çok erkinli sisteme uyarlanması ile elde edilmektedir. Bu yaklaşımda, robotlar kendi özel durum ve hareket uzaylarına ilişkin Q-değerlerini kendi yerel bilgileri dahilinde diğer robotların hareketlerinden bağımsız olarak öğrenmektedir.

Bu yaklaşımın en büyük avantajı öğrenme uzayının boyutunun çok büyük olmaması dolayısıyla işlemsel yükün az olmasıdır. İlaveten, erkinler arası haberleşmeye ihtiyaç duymamaktadır. Ancak, erkinlerin sadece kendi yerel bilgileri ile yetinerek diğer erkinlerin ortam üzerindeki etkilerini dikkate almamaları pekiştirmeli öğrenme yöntemlerinin temel kabulü olan ortamın MKS olarak tanımlanması şartı ile çelişmektedir. Bu da optimal sonuca yakınsamanın önündeki en büyük engeldir. Ayrıca, dağıtık öğrenme algoritmalarının kullanılması, özellikle erkinler hareketlerini eş zamanlı olarak seçtikleri ve diğer erkinlerin hareket seçimlerini dikkate almadıkları durumlarda davranış çakışmalarına neden olmaktadır (Wang and de Silva, 2006b). Bu da işbirliğinin zayıf kalmasına ve öğrenme hızının azalmasına neden olarak gerçek

zamanlı kullanıma imkân tanımaması ile sonuçlanmaktadır. Yine de özellikle uygulama kolaylığı ve hızlı çalışması yüzünden birçok sistemde kullanılmış ve etkinliği deneysel olarak gösterilmiştir (Martinson and Arkin, 2003; Wang and de Silva, 2005; Mataric, 1997).

Bir diğer yaklaşım olan merkezi öğrenme yaklaşımında, robotlar işbirliği içerisinde oldukları diğer robotlarla birlikte tümleşik hareketlerinin birlikte değerlendirildiği ortak Q-değerlerini birlikte öğrenmektedir. Bu yaklaşımda Q değerleri Denklem (3.12) ile verilen yinelemeli denklem ile elde edilmektedir.

$$V = \max_{a'_1 \in A_1, \dots, a'_m \in A_m} Q(s, a'_1, \dots, a'_m)$$

$$Q(s, a_1, \dots, a_m) = (1 - \alpha)Q(s, a_1, \dots, a_m) + \alpha[\rho + \gamma V] \quad (3.12)$$

Bu durumda ya tüm erkinler aynı anda ortamdaki tüm değişiklikleri gözlemleyerek aynı bilgileri aynı anda öğrenmekte ya da merkezi bir koordinasyon birimi öğrenme sürecini tüm erkinlerle haberleşerek gerçekleştirmektedir. Merkezi öğrenme yaklaşımında öğrenme işbirlikçi tüm robotların tümleşik hareket uzayı üzerinden gerçekleştirildiği için ortamın MKS olarak tanımlanmış olması şartının sağlandığı düşünülmekte ve optimal çözümlere yakınsanacağı beklenmektedir (Matignon et al., 2007). Ancak, tümleşik hareket uzayı kullanımı nedeniyle öğrenme uzayının boyutunun çok büyük olması sisteme büyük işlemsel yük getirmektedir. Tek erkinli bir Q-öğrenme algoritmasında; $|S|$ durum uzayı boyutunu ve $|A|$ hareket uzayının boyutunu göstermek üzere öğrenme uzayının boyutu $|S||A|$ olmaktadır. Çok erkinli bir sistemde merkezi bir öğrenme yapısının kullanıldığı durumda ise m erkin sayısını göstermek üzere öğrenme uzayının boyutu $|S||A_1| \dots |A_m|$ olmaktadır. Tamamen işbirlikçi bir oyun için $|A_1| = \dots = |A_m| = |A|$ alınabileceğinden dolayı öğrenme uzayının boyutu $|S||A|^m$ olmaktadır. Öğrenme uzayının boyutunun erkin sayısına bağlı olarak üstel olarak artması işlemsel yükü oldukça artırmaktadır (Hu and Wellman, 2003).

Ayrıca merkezi öğrenme yaklaşımları ya harici bir öğrenme veya koordinasyon merkezine ya da işbirlikçi erkinler arası güvenilir haberleşme ağına ihtiyaç duymaktadır. Tüm bu dezavantajlar merkezi bir öğrenme yaklaşımının kullanılmasının ne kadar problemlere açık olduğunu göstermektedir (Wang and de Silva, 2006a). Birçok gerçek ortam uygulamasında bu iki unsurun sağlanması her zaman mümkün olamamaktadır. Erkinler arası yoğun haberleşme akışı işlemsel karmaşıklığı mevcut erkin sayısına bağlı olarak çok fazla artırıyor olması sebebiyle tercih edilmemektedir. Özellikle çok robotlu sistemlerin kullanıldığı birçok uygulamada ortam şartları gereği güvenilir haberleşme garanti edilememektedir.

Merkezi ve dağıtık öğrenme yaklaşımlarının avantajlarını birleştirmek üzere dağıtık yapıda fakat merkezi karar verme mekanizmasına sahip yaklaşımlar da bulunmaktadır. Dağıtık öğrenme algoritmalarının kullanıldığı fakat merkezi bir kontrol mekanizmasının da bulunduğu modüler Q-öğrenme algoritması önerilmiş ve robot futbol oyunlarına uygulanmıştır (Park et al., 2001). Modüler Q-öğrenme algoritmasında her bir robot diğerlerinden bağımsız olarak Q-öğrenme algoritmasını kullanmaktadır. Robotlar arası koordinasyonun sağlanması ve davranış çakışmalarının engellenmesi için harici bir kontrol birimi yer almaktadır. Bu birim her bir robotun öğrendiği Q değerleri üzerinden merkezi bir karar verme birimi gibi çalışmaktadır.

Wang ve de Silva bir çalışmalarında dağıtık bir yaklaşım olan ardışıl Q-öğrenme algoritmasını önermişlerdir. Buna göre, erkinler birbirlerinden bağımsız olarak önceden tanımlanmış bir sıralamaya göre öğrenmektedirler. Bir erkin hareket seçerken kendinden önceki erkinlerin seçimlerini gözlemleyerek kendi hareketine karar vermekte, böylece erkinler arasında oluşabilecek davranış çakışmaları önlenmektedir. Ortamın anlık da olsa MKS olarak kabul edilebileceği, bu yapının başarısı nesne taşıma problemi üzerinde deneysel olarak gösterilmiştir (Wang and de Silva, 2008). Ancak yine de dağıtık öğrenme yaklaşımlarının en iyi çözüme yakınsayacağına dair teorik bir ispat bulunmamaktadır (Martinson and Arkin, 2003).

3.4 Çok Robotlu Sistemlerde Q-Öğrenme Yöntemi Uygulamaları

Çok robotlu sistemler doğaları gereği genellikle dinamik ve ancak kısmi olarak gözlemlenebilen ortamlarda kullanılmaktadır. Böyle bir durumda robotların yetenekleri ve iş yapma kapasiteleri ortamda meydana gelen değişikliklere ve diğer robotların davranışlarına göre değişiklik göstermektedir. Çok robotlu sistemler için kontrol mimarilerini tasarlarırken karşılaşılabilecek tüm olası durumları önceden kestirmek ve robotların davranışlarını önceden planlamak mümkün olamamaktadır. Robotların davranışlarını ortama ve gelişen durumlara göre adapte etme yeteneğine sahip olmaları gerekmektedir. Bu probleme karşı önerilebilecek en iyi çözüm çok robotlu sistemleri öğrenme yetenekleriyle donatmaktır. Öğrenme kapasitesine sahip bir robotun ortamdan ve diğer robotlardan kaynaklanan belirsizliklerle mücadele etmesi mümkün olacaktır. Böylece tüm sistem daha gürbüz, adaptif ve etkin hale gelmektedir.

Yapısal ve uygulanma kolaylığı sebebiyle çok robotlu sistemler üzerinde yapılan birçok çalışmada pekiştirmeli öğrenme yaklaşımları ve özellikle Q-öğrenme yöntemi tercih edilmektedir (Buşoniu et al., 2008).

Yang ve Gu literatürdeki mevcut çalışmaların kısa bir taraması niteliğinde olan çalışmalarında (Yang and Gu, 2004), çok erkinli pekiştirmeli öğrenme algoritmalarının çok robotlu sistemler için kullanımını ele almışlardır. Öncelikle erkin tabanlı öğrenme algoritmalarının teorik altyapısı anlatılmış, tek erkinli sistemlerde etkin olarak kullanılabilen Q-öğrenme algoritmasının çok erkinli sistemlere uyarlanması karşılaşılan sıkıntılara değinilmiştir. Takım-Q (Littman, 2001a), minimax-Q (Littman, 1994), Nash-Q (Hu and Wellman, 2003), dost-veya-düşman Q (Littman, 2001b) çok erkinli sistemlerin çeşitli kısıtlarına karşı önerilmiş Q-öğrenme algoritması tabanlı öğrenme yaklaşımlarıdır. Çok robotlu sistemlerin dinamik ve kısmi gözlemlenebilir çalışma ortamları ile durum ve hareket uzaylarının sürekli ve sınırsız olması problemin MKS olarak tanımlanmasını engellemektedir. Bu sebeple çok erkinli sistemler için önerilen algoritmaların doğrudan çok robotlu sistemler için kullanımında optimal çözümün yakınsanmaması gibi çeşitli problemler bulunmaktadır. Bu problemlere çeşitli çözüm önerileri sunan çalışmalar listelenmekte, bununla birlikte durum ve hareket

uzaylarının tanımlanması ve kullanılması, mevcut çok sayıdaki yöntemin doğrudan kullanımının sağlanması, çok robotlu sistemlerin çoklu denge noktalarının tespiti ve tüm bu konulara ilişkin teorik çalışmaların eksikliği gibi açık noktaların varlığı vurgulanmaktadır. Ayrıca, çok erkinli sistemler için önerilen yöntemlerin çok robotlu sistemlerde kullanımına dair yeterli teorik çalışmanın bulunmadığı belirtilmektedir (Yang and Gu, 2004).

Benzer bir diğer çalışma da çok erkinli sistemlerde kullanılan pekiştirmeli öğrenme yöntemlerini erkin teorisi ve oyun teorisi temelinde ele almaktadır (Buşoniu, et al., 2008). Çok erkinli pekiştirmeli öğrenme algoritmalarının işbirlikçi, rekabetçi, karma olmasına, dinamik veya durağan ortamlarda kullanılabilmelerine göre sınıflandırması yapılmıştır. Çok erkinli sistemler için önerilen algoritmaların çok robotlu sistemlerde kullanımları araştırılmıştır.

Mataric (Mataric, 1997) ve Martinson (Martinson et al., 2002) gürültülü ve dinamik ortamlar için çok robotlu sistemlerde öğrenme algoritmalarının kullanımını araştırmış, öğrenme uzayını küçültmek için ayırık hareketleri değil davranışları kullanmışlardır.

Robot futbol oyunları, aynı takımda bulunan robotlar kendi aralarında işbirlikçi bir strateji izlerken karşı takımdaki robotlara karşı rekabetçi bir strateji izlemeleri yani hem işbirlikçi hem de rekabetçi stratejilerin birlikte ele alınması açısından oldukça farklı uygulama ortamları sunmaktadır. Ayrıca, ortamın dinamik ve karmaşık olması tüm durumların tanımlanmasını ve futbol oyununun modellenmesini imkânsız kılmaktadır. Robot futbol oyunu için robotların görevlerinin atanması ve optimal hareketlerin seçilmesi ve tüm bu işlemlerin gerçek-zamanlı olarak gerçekleştirilmesi oldukça önemlidir. Hwang bir çalışmasında futbol oynayan robotların hem temel hem de yüksek seviyeli davranışları öğrenmesi için Q-öğrenme yöntemi kullanılmıştır (Hwang et al., 2007). Öğrenme algoritmalarının çoklu strateji seçimleri üzerinde oldukça etkili olduğu yapılan deneylerle gösterilmiştir. Robot futbol oyunları üzerinde gerçekleştirilen bir başka çalışmada, robotların golcü, savunma oyuncusu, yardımcı oyuncu veya kaleci rollerine atanması, bu rolleri gerçeklemek üzere yapacakları

hareketlerin belirlenmesi ve en temel davranış olan engelden sakınma davranışları Q-öğrenme yöntemi kullanılarak gerçekleştirilmektedir (Hwang et al., 2004). Önerilen algoritma ile Q-öğrenme algoritmalarının performansını etkileyen sistem parametreleri, öğrenme hızı (α) ve azaltma oranı (γ) uyarlamalı olarak ayarlanmaktadır. Böylece öğrenme algoritmasının yakınsama hızının artırılması sağlanmış ve yerel en iyilere takılması engellenmiştir.

Çok robotlu sistemlerde görev paylaşımı probleminin çözümüne yönelik olarak öğrenme yaklaşımlarının kullanıldığı çalışmalar da bulunmaktadır. Bunlardan biri Jones ve arkadaşları tarafından gerçekleştirilen, görevlerin dağıtılması ve planlanması sırasında öğrenme yeteneğini kullanarak ileride gelişebilecek durumlar hakkında öngörüde bulundukları çalışmadır (Jones et al., 2007). Elde ettikleri sonuçlar, öğrenme yeteneğinin sistemin performans artışına katkıda bulunduğunu göstermektedir.

Bir başka çalışmada, robotlar geçmiş müzayedede tecrübelerini öğrenerek müzayedeye katılım durumlarını değerlendirdikleri bir yaklaşımdır (Busquet and Simmons, 2006). Schneider ve arkadaşları, görev yaparken harcanan sürenin müzayedeye katılım bedeli olarak kullanıldığı bir sistemde, yüksek belirsizlikler içeren ortamlar için ortama bağlı görev süresi bilgisinin öğrenildiği bir çalışma gerçekleştirmişlerdir (Schneider et al., 2005).

3.5 Problemler

Q-öğrenme yönteminin çok robotlu sistemlerde uygulanması esnasında, robotların çalışma ortamlarının doğası gereği karşılaşılan problemleri üç ana başlık altında ele almak mümkündür:

- 1- Çok robotlu sistemlerin çalışma ortamlarının MKS olarak tanımlanamaması
- 2- Birçok uygulamada durum ve hareket uzaylarının ayrık olmaması ve ayrık olarak tanımlamada karşılaşılan güçlükler
- 3- Tümlleşik durum ve hareket uzaylarının çok büyük boyutlu olması

Geleneksel Q-öğrenme yönteminin kullanımında karşılaşılan bir problem ortamın MKS olarak tanımlanması yani durağan ve belirli olması gerekliliğidir. Oysa, çok robotlu sistemlerin gerek çalışma ortamları hakkında her zaman tam anlamında bilgi sahibi olunamaması gerekse robotların birbirinden bağımsız olarak hareket etmeleri nedeniyle ortamın durağan olarak nitelendirilmesi imkânsızdır. Q-öğrenme yönteminin en iyi çözüme yakınsaması ortamın MKS olması şartına doğrudan bağlıdır. Bu da sistemin performans kaybına neden olmaktadır. Bu probleme karşı önerilen yöntemlerden biri, diğer robotların hareketlerinin sistem üzerindeki etkilerinin tahmin edilmesine dayanmaktadır. Bu durumda sistem yarı-Markovyan ve sözde-durağan olarak tanımlanmaktadır (Kawakami et al., 1999). Bir başka yaklaşımda ise erkinlerin birbirinden bağımsız olarak öğrenmesi ama öğrenme işlemi sonrasında davranışların çizelgelendirilmesi önerilmektedir (Uchibe et al., 1998). Fakat bu durum erkinler arası harici haberleşme zorunluluğu doğurmaktadır.

Çok robotlu sistemlerde Q-öğrenme yönteminin uygulanması esnasında karşılaşılan bir diğer problem ise; robotların çalışma ortamları nedeniyle sürekli ve sınırsız olan durum ve hareket uzaylarından kaynaklanmaktadır. Oysa Q-öğrenme yöntemleri ayırık ve sonlu durum ve hareket uzayları üzerinde tanımlanmış ve optimal çözümlere yakınsadığı bu şartlar altında gösterilmiştir. Çok erkinli pekiştirmeli öğrenme yöntemlerindeki birçok teorik çalışma sadece küçük ve sınırlı durum ve hareket uzayları üzerinde gerçekleştirilmiştir (Yang and Gu, 2004). Bazı uygulamalarda ise durum ve özellikle hareket uzaylarının ayırık olarak ifade edilmesi tercih edilmemektedir. Önerilen çözümlerden birinde durum ve hareket uzayları ayırık kümeler halinde gösterilmek yerine fonksiyon yaklaşımları kullanılarak ifade edilmiştir (Smart and Kaelbling, 2002). Ancak, bu fonksiyon yaklaşımlarını elde etmek için yapay sinir ağları gibi harici yapılara ihtiyaç duyulmaktadır (Riedmiller et al., 2009). Ayrıca ayırık hareketler arasında sık geçişler yaşanması robotun istenmeyen hareketler yapmasına neden olmaktadır. Liu ve arkadaşları gerçekleştirdikleri bir çalışmada, sistemin yapması gereken hareketleri değil, hareketler bütününe ilişkin ağırlıkları öğrendiği bir yöntem önermektedir (Liu et al., 2005). Sonuçta, robotun davranışı ayırık hareketlerin ağırlıklı toplamı alınarak belirlenmektedir. Böylece öğrenme birimi ayırık ve sonlu uzaylarla çalıştığı halde sonuçta elde edilen davranış uzayı sürekli özelliklere

sahip olmaktadır. Benzer bir başka çalışmada, hareket uzayı bulanık mantık yaklaşımları ile temsil edilerek öğrenme ayırık yapılar üzerinde gerçekleştirilmekte fakat sonuçta sürekli hareketler elde edilebilmektedir (Hwang et al., 2012).

Sürekli durum ve hareket uzaylarının ayırık olarak temsil edilmesinin mümkün olabildiği durumlarda, ayırıklaştırma işleminden kaynaklanan temsil hassasiyeti sorunları ve ortamın biliniyor olması gereksinimi ortaya çıkmaktadır. Durum ve hareket uzaylarını ayırıklaştırma sürecinde temsil hassasiyetinin gerekli doğrulukla belirlenmesi oldukça önemlidir. Doğru temsil hassasiyetinin belirlenmesi sistemin ihtiyaçlarının ve ortamın yeteri kadar biliniyor olması zaruretinin de beraberinde getirmektedir. Düşük temsil hassasiyeti, farklı niteliklerin aynı durum içerisinde temsil edilmesi sonucunu doğuracağı için gerekli bilgiyi taşıma anlamında yetersiz kalacaktır ve bu durum öğrenme performansını olumsuz etkileyecektir. Diğer taraftan yüksek temsil hassasiyeti ise durum ve hareket uzayı boyutlarının çok büyümesine yol açacaktır. Bunun sisteme getireceği büyük hesaplama yükü de yine öğrenme süreci için çeşitli zorlukları beraberinde getirmektedir. Aynı durum ile ifade edilen farklılıkları yakalamak için hareketlerin ve ödüllerin olasılıksal olarak belirlendiği bir öğrenme algoritması önerilmiştir (Hwang, et al., 2009b). Ayırıklaştırma sürecinde uygun ayırıklaştırma hassasiyetinin bulunabilmesi için ortam hakkında yeteri kadar bilgi sahibi olmak gerekmektedir. Birçok gerçek ortam uygulamalarında ise bu mümkün değildir. Bu amaçla, durum uzayı ayırıklaştırma işleminin öğrenme sırasında gerçekleştirildiği çeşitli çalışmalar bulunmaktadır (Hamagami and Hirata, 2003; Lee and Lau, 2004). Bu çalışmalarda ayırık durum uzayının boyutu da sistem gereksinimine göre ayarlanabilmektedir.

BÖLÜM 4

ÖNERİLEN YAKLAŞIM

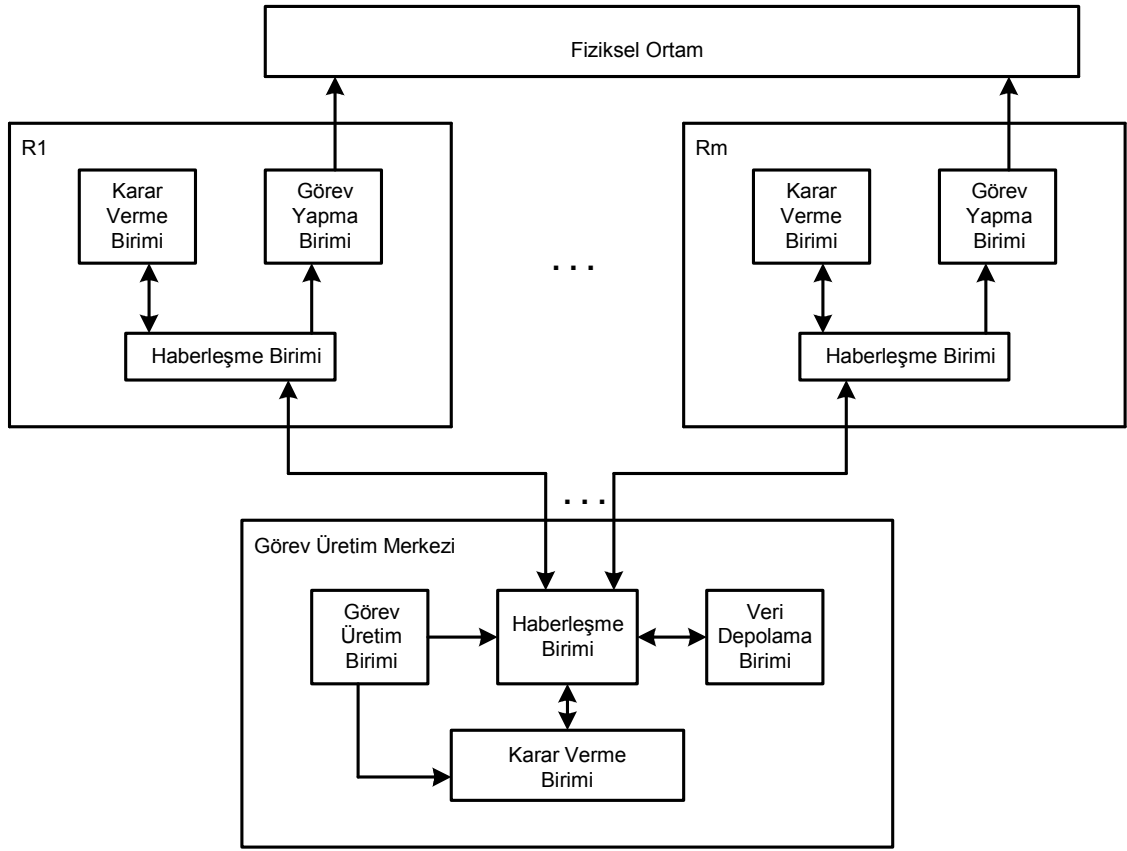
Bu çalışmada, temel anlamda çok robotlu bir sistemde öğrenme yöntemleri tabanlı yaklaşımlar kullanılarak robotlar arası görev atama problemine daha etkin çözümler getirilmesi amaçlanmıştır. Robotların geçmiş görev atama/yapma tecrübelerinden gelecek uygulamalarda yararlanmalarının sağlanması ile sistem kaynaklarının daha verimli kullanılması ve sonuç olarak toplam sistem performansının artırılması hedeflenmektedir.

Yapılan çalışmalarda, gerçek ortam uygulamalarında kullanılan çok robotlu sistem modeli ele alınmıştır. Bu sebeple ortam hakkında aşağıdaki kabullenmeler yapılmıştır:

- Çok robotlu sistem heterojen yapıya sahiptir. Robotlar farklı nitelik ve yeteneklere sahiptir.
- Çalışma sürecinde çeşitli sebeplere bağlı olarak robotlar kısmen veya tamamen devre dışı kalabilmektedir. Kısmen devre dışı kalmış robotların sürecin ilerleyen aşamalarında tekrar sisteme dahil olabilmeleri söz konusudur.
- Robotlar arası açık haberleşme yapısı mevcuttur.
- Robotlar dürüst ve işbirlikçidir.
- Robotlar tarafından yapılması gereken görevler de birbirinden farklı niteliklere sahiptir. Belli nitelikteki görevler ancak belirli yeteneklere sahip robotlar tarafından yapılabilmektedir.
- Görevlerin gelme zamanları ve sıraları sistem tarafından önceden bilinmemektedir. Görevler çalışma süreci içerisinde rassal şekilde ve aniden ortaya çıkmaktadır.

Sistemde robotlar arası görev paylaşımı problemi, Bölüm 2.4'te anlatıldığı üzere market tabanlı bir yöntem olan çok ürünlü müzayede yapısı ile çözülmektedir. Bu yapıda, kaynak tarafından üretilen görevler ortamda bulunan tüm robotlara anons edilir. Anonsu alan robotlar o esnada bir başka görevi yapmıyorsa ve anons edilen görevi yapabilme yeteneğine sahip ise o görev için bir bedel belirleyerek müzayedeye katılmak üzere görevi anons eden kaynağa gönderir. Bu bedel, anons edilen görevin kaynak tarafından belirlenen ücreti ve o görevi yaptığı takdirde robota olan maliyeti kullanılarak hesaplanmaktadır. Kaynak, robotların müzayedeye katılmaları için tanıdığı süre dolduktan sonra robotlardan gelen teklifleri değerlendirerek görevlerin robotlara atamalarını gerçekleştirir. Görev atama işlemi her seferinde en fazla görevin yapılmasını sağlayacak şekilde ve görevi yapacak her bir robotun kazançlarını en büyüleyecek şekilde yapılmaktadır. Robotlar birden fazla sayıda görev için müzayedeye katılabilmektedir ancak bir seferinde bir robota sadece bir görev atanmaktadır ve bir görev sadece bir robota verilmektedir. Atama tamamlandıktan sonra, kazanan robotlar kaynak tarafından bilgilendirilmekte ve görevlerini yapmaktadır.

Ele alınan bu sistemin temel mimarisi Şekil 4.1'de verilmiştir. Sistemde $R_1, \dots, R_m$ ile gösterilen m adet robot ve bir adet Görev Üretici Merkez (GÜM) bulunmaktadır. Her bir robot kendi içinde görev yapma birimi, karar verme birimi ve haberleşme birimi olmak üzere üç temel birim barındırmaktadır. Robotlar, GÜM tarafından üretilen görevlere ilişkin bilgilere ve müzayedeye katılma bilgilerine dair veri alışverişini haberleşme birimi üzerinden gerçekleştirmektedir. Yeni bir görev anons edildiğinde müzayede için gerekli bedelin hesaplanması karar verme birimi tarafından yapılmaktadır. Görev yapma birimi ise robotların fiziksel ortam ile irtibatlarını sağlamaktadır. GÜM ise dört temel birimden oluşmaktadır: görev üretme birimi, haberleşme birimi, karar verme birimi ve veri depolama birimi. Üretilen görevlerin robotlara anonsu ve müzayede katılımları haberleşme birimi üzerinden gerçekleştirilirken, müzayede sonunda hangi görevin hangi robota atanacağı karar verme birimi tarafından belirlenmektedir. Veri depolama birimi ise sistem ve ortam hakkında gerekli bilgilerin tutulduğu bir veri tabanıdır.



Şekil 4.1. Temel Çok Robotlu Sistem Kontrol Mimarisi

Bu çalışmada, çok robotlu sistemlerde görev atama sürecinde karşılaşılan problem ve öğrenme yeteneğinin sisteme getireceği katkı ele alınmıştır. Öğrenme yöntemi olarak robot sistemlerinde kullanımı sıklıkla tercih edilen Q-öğrenme yöntemi kullanılmıştır. Ancak Q-öğrenme yöntemi teorik olarak ayrık ve sonlu durum ve hareket uzayları üzerinde tanımlanmıştır. Oysa gerçek ortamlı robot uygulamalarında, ortamların doğası gereği sürekli değişkenlerle ifade edilen durum ve hareket uzaylarının ayrık değişkenlerle ifade edilmesi temel bir problem olarak ortaya çıkmaktadır. Bu bölümde ilk olarak bu problemler açıklanmaktadır.

Q-öğrenme yönteminin çok robotlu sistemlerde uygulanmasında ortamın uygun şekilde modellenmesi kullanılan yaklaşımın başarıya ulaşması için önemli rol

oyunmaktadır. Çok robotlu sistem için robotlar ve görevler temel alınarak durum ve hareket uzayları ve ödül değeri tanımlamaları yapılmaktadır.

Q-öğrenme yönteminin sistemi en iyileyen çözüme yakınsadığı tek erkinli sistemler üzerinde teorik olarak ispatlanmıştır (Watkins and Dayan, 1992). Ancak, genelde çok erkinli ve özelde çok robotlu sistemler üzerinde başarısı kanıtlanmış belirli bir yaklaşım bulunmamaktadır (Yang and Gu, 2004). Literatürde bu amaçla kullanılan iki temel yaklaşım bulunmaktadır: merkezi öğrenme ve dağıtık öğrenme yaklaşımları. Bu çalışmada, bu iki yaklaşımın avantajlarını birleştiren bir yaklaşım olan Strateji Planlamalı Dağıtık Öğrenme yaklaşımı önerilmektedir.

Q-öğrenme yönteminin başarısı durum uzayının sistem için en uygun şekilde belirlenebilmesiyle doğrudan ilişkilidir. Burada ele alınan sistemin durum uzayı sürekli niteliktedir. Bu çalışmada, biri sürekli durum uzayını yine sürekli değerlerle tanımlayan Dağılım Fonksiyonlu Sürekli Durum Uzayı (DFSDU) yaklaşımı ve diğerleri sürekli durum uzayını ayrıklaştırma işlemine dayanan Sabit Aralıklı Ayrık Durum Uzayı (SAADU) ve Değişken Aralıklı Ayrık Durum Uzayı (DAADU) yaklaşımları olmak üzere üç farklı yaklaşım önerilmiştir. Takip eden alt kısımlarda bu yaklaşımlar ayrıntılı bir şekilde ele alınmaktadır.

4.1 Problemin Tanımlanması

Gerçekleştirilen çalışmaların temel amacı, çok robotlu sistemlerde öğrenme yaklaşımları desteği ile robotların geçmiş görev atama deneyimlerini gelecek uygulamalarda kullanmalarını sağlayarak sistem performansının artırılması olarak tanımlanmıştır.

Önerilen yöntemlerle çözüm getirilen problemleri iki ana başlık altında incelemek mümkündür. İlk problem, çok robotlu sistemlerde robotların öğrenme yeteneğine sahip olmalarının görev atama problemlerine sağlayacağı katkıyı açıklamaktadır. İkinci

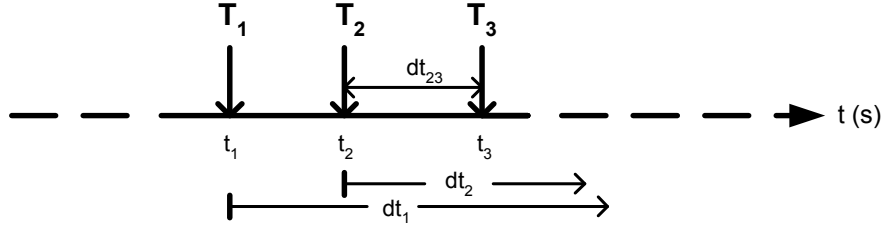
problem ise doğrudan Q-öğrenme yönteminin robot sistemlerine uygulanması esnasında ortaya çıkmaktadır. Her iki probleme ilişkin açıklamalar aşağıda yer almaktadır.

4.1.1 Problem I: Öğrenme ihtiyacı

Robotların yapmaları gereken görevlerin önceden bilinmesi durumunda hangi robotun hangi görevi hangi sırayla yapacağını başlangıçta ve sistem kazancını en iyileyecek şekilde planlanması mümkün olmaktadır. Ancak, çok robotlu sistemlerin yer aldığı birçok uygulamada görevlerin gelme zamanları ve sıraları önceden bilinmemektedir. Bu sebeple gerekli koordinasyonun önceden yapılması söz konusu değildir. Görevlerin robotlar tarafından öngörülemez zamanlarda ortaya çıkması ve görev paylaşımlarının anlık olarak ve o esnada başka bir görev atanmamış robotlar arasında yapılabilecek olması robot grubunun performansını olumsuz yönde etkilemektedir. Özellikle bu çalışmada ele alınan sistem gibi pek çok sistemde görevler arası bir hiyerarşi mevcut olduğunda öncelikle ve mutlaka yapılması istenen görevler, bu görevi yapabilecek olan robotların daha düşük öncelikli görevlerle meşgul olmaları durumunda yapılamayacaktır. Bu sistemlerde, anons edilen görevlerin ücretleri görevi duyuran kaynak tarafından belirlenmektedir ve görevin öncelik durumuna bağlı olarak değişmektedir. Görevin öncelik derecesi yükseldikçe ücreti de artmaktadır. Bu ücret farkı robotların aynı anda anons edilen görevlerin arasında yüksek öncelikli olanlara daha fazla bedel belirlemelerine neden olmaktadır. Bunun sonucu olarak da yüksek öncelikli görevlerin yapılma oranı artmaktadır. Ancak, robotların meşgul olduğu sırada gelen görevler yüksek önceliğe sahip olmalarına rağmen sistem kaynaklarının yetersizliği nedeniyle yapılamayacaklardır. Bu durumu Şekil 4.2’de gösterilen basit bir görev akış şeması ile örnekleyerek açıklamak mümkündür.

Şekil 4.2’de gösterilen örneğe göre sistem gelen her görevi yapma yeteneğine sahip olan iki robottan, R_1 ve R_2 , oluşsun. t_1 , t_2 ve t_3 değerleri sırasıyla T_1 , T_2 ve T_3 görevlerinin gelme anlarını gösterebilir. t_1 anının hemen öncesinde her iki robot da boşta olduğu için t_1 anında gelen T_1 görevini R_1 robotu ve daha sonra t_2 anında gelen T_2 görevini de R_2 robotu yapmaya başlamaktadır. T_1 görevinin yapılma süresi dt_1 s ve T_2

görevinin yapılma süresi dt_2 s olsun. Bu durumda, T_2 görevinden t_{23} s sonra yani t_3 anında anons edilen T_3 görevinin yapılması o esnada her iki robotun da meşgul olmaları nedeniyle mümkün olamayacaktır. Eğer T_1 ve T_2 görevleri düşük öncelikli görevler iken T_3 görevi yüksek öncelikli bir görev ise sistemin elde edeceği kazançta büyük azalma meydana gelmektedir. Bu noktada sistemin kazancını artırmak ancak robotlardan birinin ilk görev için müzayedeye katılmayıp sonraki görevi beklemesi ile sağlanacaktır.



Şekil 4.2. Örnek Bir Görev Akış Şeması

Genel anlamda robotlar, bir görev anons edildiğinde o anda bir başka işle meşgul olmadıkları takdirde ve o görevi yapabilme yeteneğine sahip oldukları durumda o görev için bir bedel belirleyerek müzayedeye katılmaktadır. Robotlar bu süreçte geleceğe dair herhangi bir beklentiye yani bir süre sonra daha yüksek öncelikli dolayısıyla kazancı daha yüksek bir görevin geleceği öngörüsüne sahip değildir. Eğer böyle bir öngörüye sahip olsalardı yukarıda yer alan örnekteki robotlardan en az bir tanesinin gelen düşük öncelikli görevi yapmayarak daha sonra gelecek yüksek öncelikli T_3 görevini beklemesi mümkün olacaktı. Robotların müzayedeye katılacakları görevler için bedel belirlerken devam eden süreçte anons edilecek görevlere ilişkin beklentilerini de hesaba katmaları için öncelikle görevlerin anons edilme zamanları arasındaki ilişkiye dair bilgi sahibi olmaları gerekmektedir. Robotların öğrenme yetenekleri sayesinde, geçmişte edindikleri tecrübeleri bilgiye dönüştürerek gelen görevler için müzayedeye katılmadan önce bu bilgiden yararlanmaları mümkün olduğu takdirde, sistemin toplam kazancında artış sağlanabilecektir.

4.1.2 Problem II: Durum uzayının yapılandırılması

Q-öğrenme yöntemi girdi-çıkı setlerine ve bir öğreticiye ihtiyaç duymadan yapılan hareketin sonucuna göre ortamdan alınan geri besleme bilgisine bağlı olarak öğrenme sürecini gerçekleştiren bir yöntemdir. Bu yapısı nedeniyle robot sistemlerinde kullanımı uygundur. Bu çalışmada Bölüm 4.1.1’de anlatılan probleme çözüm olması amacıyla Q-öğrenme yönteminin kullanımı tercih edilmiştir.

Q-öğrenme yöntemi, durum ve hareket uzaylarının sonlu ve ayrık olarak belirlendiği MKS olarak nitelendirilen ortamlar üzerinde tanımlanmıştır (Watkins, 1989). Ancak, robot sistemlerinin kullanıldığı bir çok gerçek ortam uygulamasında, çalışma ortamının doğası gereği durum ve/veya hareket uzayları sonsuz ve sürekli nitelikte olabilmektedir. Sürekli değişkenler olan durum ve/veya hareket uzaylarının ayrık değerlerle ifade edilebilmesi için gerçekleştirilen ayrıklaştırma süreci büyük problem teşkil etmektedir. Ayrıklaştırma süreci sonunda ortam hakkında hassas, yeterli ve doğru bilgiyi taşıyan ayrık durum ve/veya hareket uzayı bileşenlerinin elde edilebilmesi için uygun temsil hassasiyetinin kullanılması oldukça önemli rol oynamaktadır. Bu da sistem hakkında tam bilgi sahibi olunmasını gerektirmektedir.

Temsil hassasiyetinin düşük tutulması farklı nitelikteki verilerin aynı grupta değerlendirilmesine neden olacağı için sistem hakkında gerekli ve ayırt edici nitelikteki bilgilerin kaybolması sonucunu doğurabilmektedir. Bu da öğrenme sürecini olumsuz olarak etkilemektedir. Diğer taraftan, yüksek temsil hassasiyeti ise benzer nitelikte değerlendirilebilecek verilerin farklı ayrık değerlerle ifade edilmesine neden olmaktadır. Durum ve/veya hareket uzaylarının boyutunun aşırı derecede büyümesi öğrenme sürecini uzatmakta ve bazen tüm durum-hareket çiftlerinin yeteri kadar öğrenilememesi ile sonuçlanmaktadır. İlaveten, sisteme büyük hesaplama yükü getirmektedir. Özetle, ayrıklaştırma temsil hassasiyetinin gerekli doğrulukla belirlenmesi öğrenme sürecinin performansını etkileyen önemli bir problemdir.

Bu çalışmada ele alınan yapıda, hareket uzayı ayrık bileşenlerden oluşacak şekilde belirlenmiştir. Ancak, durum uzayı bileşenleri görevlerin gelme zamanları arasında

geçen süre temel alınarak düzenlenmektedir. Süre gibi sürekli bir değişkene ilişkin bilginin ayrık durum uzayı bileşeni olarak kullanılması çeşitli tanımlama ve temsil etme problemlerine yol açmaktadır. Bu probleme çözüm olarak önerilen yaklaşımlar takip eden alt bölümlerde ayrıntılı olarak anlatılmaktadır.

4.2 Çok Robotlu Bir Sistemin Q-Öğrenme Yöntemi Tabanlı Tanımlanması

Bu çalışmanın sonucunda bir robotun anons edilen görev için müzayedeye katılıp katılmayacağı kararını vermesinde geleceğe dair beklentilerinin etkin rol oynaması amaçlanmaktadır. Bu amaca ulaşabilmek için robotların geçmiş görev atama deneyimlerini geleceğe yönelik beklentiye dönüştürebilmeleri gerekmektedir. Bu da ancak robotların öğrenme yeteneğine sahip olmaları ile sağlanabilecektir. Burada, Q-öğrenme yönteminin çok robotlu bir sistemde kullanılmasını sağlamak üzere robot ortamlarının, durum ve hareket uzaylarının ve ödül değerlerinin modellenmesi yapılmaktadır.

4.2.1 Robot ortamı

n adet farklı tipteki görevi yapması gereken ve m adet robota sahip çok robotlu bir sistem için robot temelli ortam modellenmesi aşağıda verilmektedir.

Tanım 4.1. Çok robotlu bir sistemde farklı yeteneklere sahip m adet robot bulunsun. Bu sistem için robot kümesi $R = \{R_j \mid j = 1, \dots, m\}$ ile tanımlanmaktadır.

Tanım 4.2. Çok robotlu bir sistemde yapılması gereken farklı nitelikte n adet görev tipi bulunsun. $T = \{T_i \mid i = 1, \dots, n\}$ görevler kümesini göstermektedir.

Tanım 4.3. w_j , R_j robotunun yapabildiği görevlere ilişkin indis kümesi olmak üzere, R_j robotunun görev kümesi $I_j = \{T_z \mid \forall z \in w_j \text{ ve } T_z \in T\}$ ile tanımlanmaktadır.

Tanım 4.4. $S^z, \forall T_z \in \Gamma_j$ için bağımsız durum uzayını göstermek üzere R_j robotunun durum uzayı yapabildiği görevlere ilişkin bağımsız durum uzaylarının bileşimi olarak tanımlanmaktadır.

$$S_j = \cup_{\forall z \in W_j} S^z \quad (4.1)$$

Tanım 4.5. $A^z, \forall T_z \in \Gamma_j$ için bağımsız hareket uzayını göstermek üzere R_j robotunun hareket uzayı yapabildiği görevlere ilişkin bağımsız hareket uzaylarının kartezyen çarpımı olarak tanımlanmaktadır.

$$A_j = \prod_{\forall z \in W_j} A^z \quad (4.2)$$

Tanım 4.6. Herbir robot R_j için robot ortamı $O_j^R = \langle \Gamma_j, S_j, A_j \rangle$ seti ile belirtilmek üzere çok robotlu sistem için robot ortamı $O^R = \cup_{j=1}^m O_j^R$ ile tanımlanmaktadır.

4.2.2 Durum uzayı

Öğrenme sürecinin amacı, robotun anons edilen görev için müzayedeye katılıp katılmama kararının belirlenmesine ilişkin olduğu için durum uzayı bileşenlerinin anons edilen görevler ve bunların anons edilme zamanları arasındaki süreye ilişkin bilgi taşımaları gerekmektedir. Çok robotlu sistemler için görev atama problemlerinde sistemin toplam kazancının en büyüklmesi istendiği için robotlardan kazanç getirisi yüksek olan görevleri tercih etmeleri beklenmektedir.

Bir robot herhangi bir görev anons edildiğinde bu görevden daha yüksek kazançla sahip bir görevin o andan ne kadar önce geldiğini bildiği takdirde yüksek kazançlı bir görev için bekleyip beklemeyeceğine karar verme yeteneğine sahip olacaktır.

Tanım 4.7. Çok robotlu bir sistemde her bir görev için yüksek öncelikli ve düşük öncelikli olma niteliği tanımlı olsun. u ve v üst indisleri sırasıyla daha düşük ve daha

yüksek öncelik niteliğine sahip olma durumunu temsil etmek üzere ve $Price(\cdot)$ ifadesi verilen görevin kazancını göstermek üzere düşük öncelikli T_p^u görevi ile yüksek öncelikli T_q^v görevi arasında aşağıdaki ilişki geçerlidir.

$$Price(T_p^u) < Price(T_q^v) \quad (4.3)$$

Tanım 4.8. $\Delta t_{(T_q^v|T_p^u)}$ ifadesi düşük öncelikli T_p^u görevi anons edildiğinde, yüksek öncelikli T_q^v görevinin en son o andan ne kadar önce anons edildiği bilgisini taşımaktadır. Buna göre, R_j robotu için durum uzayı aşağıda verilen biçimde ifade edilmektedir.

$$S_j = \{s^{pq} \mid s^{pq} = \{T_p^u, \Delta t_{(T_q^v|T_p^u)}\}, Price(T_p^u) < Price(T_q^v) \text{ ve } \forall p, \forall q \in w_j\} \quad (4.4)$$

4.2.3 Hareket uzayı

Öğrenme sürecinin sonunda robotun anons edilen düşük öncelikli görev için müzayedeye katılıp bu görevi yapmak ile bu görevi yapmaktan vazgeçerek yüksek öncelikli bir diğer görevi beklemek arasındaki kararı verecek bilgiye sahip olması hedeflenmektedir.

Tanım 4.9. Bir R_j robotunun hareket uzayı, her bir $s^{pq} = \{T_p^u, \Delta t_{(T_q^v|T_p^u)}\}, \forall s \in S_j$ durumu bileşeninde anons edilen T_p^u görevi için müzayedeye katılıp katılmama kararını vermek üzere aşağıdaki şekilde düzenlenmiştir.

$$A_j = \left\{ a_{j,1}^{pq}, a_{j,2}^{pq} \left| \begin{array}{l} a_{j,1}^{pq}: T_p^u \text{ görevi için müzayedeye katıl} \\ a_{j,2}^{pq}: T_p^u \text{ için müzayedeye katılma} \\ T_q^v \text{ görevi için bekle} \end{array} \right. , \forall p, \forall q \in w_j \right\} \quad (4.5)$$

4.2.4 Ödül değerinin hesaplanması

Öğrenme sürecini etkileyen önemli faktörlerden biri de her bir durum-hareket çifti için uygun ödül değerinin belirlenmesidir. Ödül değeri, görevlerin yapılması durumunda kazanılan ve yapılmaması/yapılamaması durumunda kazanılamayan kazanç değerleri üzerinden hesaplanmaktadır.

Tanım 4.10. dt_u düşük öncelikli T_p^u görevinin R_j robotu tarafından tamamlanma süresi ve $t = t_{v'}$ değeri $t_{v'} > t_u$ koşulu sağlanmak üzere düşük öncelikli T_p^u görevinden sonra gelen ilk yüksek öncelikli $T_q^{v'}$ görevinin anons anı olsun. Herhangi bir R_j robotunun $s^{pq} = \{T_p^u, \Delta t_{(T_q^{v'}|T_p^u)}\}$ durumu ve $\{a_{j,1}^{pq}, a_{j,2}^{pq}\}$ hareketlerine dair ödül değerleri aşağıdaki ifadelerle hesaplanmaktadır.

$$Rev(s^{pq}, a_{j,1}^{pq}) = \begin{cases} \mu_{11} \left(Price(T_p^u) - Price(T_q^{v'}) \right), & dt_u > t_u - t_{v'} \\ \mu_{12} \left(Price(T_p^u) + Price(T_q^{v'}) \right), & dt_u < t_u - t_{v'} \end{cases} \quad (4.6)$$

$$Rev(s^{pq}, a_{j,2}^{pq}) = \begin{cases} \mu_{21} \left(Price(T_q^{v'}) \right), & dt_u > t_u - t_{v'} \\ \mu_{22} \left(Price(T_q^{v'}) - Price(T_p^u) \right), & dt_u < t_u - t_{v'} \end{cases} \quad (4.7)$$

μ_{11} , μ_{12} , μ_{21} ve μ_{22} değerleri ödül değerlerinin ağırlığını belirleyen pozitif keyfi sabitlerdir.

4.2.5 Örnek bir sistem

Burada, önceki bölümde anlatılan modelleme yapısı basit bir ortam üzerinde örneklenmektedir. Örnek ortam, farklı yeteneklere sahip iki adet robottan oluşsun ve yapılması gereken farklı nitelikte iki görev bulunsun. Ortamın robot kümesi

$$R = \{R_1, R_2\} \quad (4.8)$$

ve görev kümesi

$$T = \{T_1, T_2\} \quad (4.9)$$

şeklindedir. R_1 robotunun T_1 ve R_2 robotunun hem T_1 hem de T_2 görevlerini yapabildiği kabulüne göre robot ortamı modelleri Çizelge 4.1’de verilmektedir.

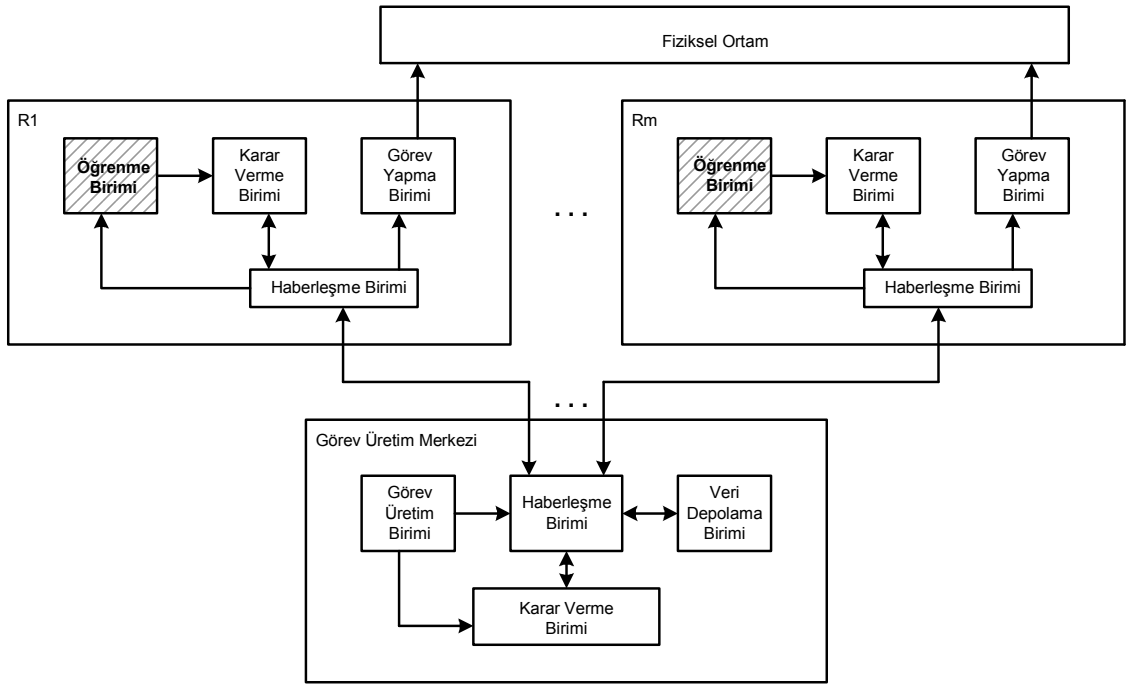
Çizelge 4.1. R_1 ve R_2 Robotları için Robot Ortamları

	R_1 Robotu	R_2 Robotu
Robot Ortamı (O_j^R)	$O_1^R = \langle \Gamma_1, S_1, A_1 \rangle$	$O_2^R = \langle \Gamma_2, S_2, A_2 \rangle$
Görev İndis Kümesi (w_j)	$w_1 = \{1\}$	$w_2 = \{1,2\}$
Robot Görev Kümesi (Γ_j)	$\Gamma_1 = \{T_1\}$	$\Gamma_2 = \{T_1, T_2\}$
Durum Uzayı (S_j)	$S_1 = \{s^{11}\}$ $ S_1 = 1$ $s^{11} = \{T_1^u, \Delta t_{(T_1^v T_1^u)}\}$	$S_2 = \{s^{11}, s^{12}, s^{21}, s^{22}\}$ $s^{11} = \{T_1^u, \Delta t_{(T_1^v T_1^u)}\}$ $s^{12} = \{T_1^u, \Delta t_{(T_2^v T_1^u)}\}$ $s^{21} = \{T_2^u, \Delta t_{(T_1^v T_2^u)}\}$ $s^{22} = \{T_2^u, \Delta t_{(T_2^v T_2^u)}\}$ $ S_2 = 4$
Hareket Uzayı (A_j)	$A_1 = \{a_{1,1}^{11}, a_{1,2}^{11}\}$ $ A_1 = 2$	$A_2 = \{a_{2,1}^{11}, a_{2,2}^{11}, a_{2,1}^{12}, a_{2,2}^{12}, a_{2,1}^{21}, a_{2,2}^{21}, a_{2,1}^{22}, a_{2,2}^{22}\}$ $ A_2 = 8$

4.3 Q-Öğrenme Yönteminin Çok Robotlu Sistemlerde Uygulanması

Bölüm 4.1’de tanımlanan probleme çözüm olmak üzere robotların öğrenme yeteneğine sahip olacakları bir yaklaşım önerilmektedir. Böylece robotların geçmiş deneyimlerinden görev atama sürecinde yararlanmaları sağlanarak sistem performansının artırılması amaçlanmaktadır. Temel olarak robotlar öğrenme süreci ile

elde ettikleri bilgileri yeni görev anons edildiğinde müzayedeye katılıp katılmama kararını alırken kullanmaktadır. Öğrenme tabanlı çok robotlu sistemin kontrol mimarisi Şekil 4.3'te gösterilmektedir. Bu yapıda robotlar ilaveten öğrenme birimine sahiptir. Yeni görev anons edildiğinde karar verme birimi öğrenme birimi ile etkileşim halinde müzayedeye karar vermekte ve eğer müzayedeye katılacaksa gerekli bedeli hesaplamaktadır.



Şekil 4.3. Öğrenme Tabanlı Çok Robotlu Sistem Kontrol Mimarisi

Öğrenme yaklaşımları açısından ele alındığında tek erkinli ve çok erkinli sistemler arasındaki fark ortamın durağan ve tamamen gözlemlenebilir olarak tanımlanması noktasında ortaya çıkmaktadır. Çok erkinli sistemlerde, erkinlerin diğerlerinden bağımsız olarak hareket edebilme yeteneğine sahip olmaları ve hareketlerinin sonucunda ortam durumlarında meydana gelen değişikliklerin tüm erkinler tarafından tam olarak kestirilememesi sebebiyle ortamı durağan olarak tanımlamak mümkün

değildir. Bu durum, pekiştirmeli öğrenme yöntemlerinin temel varsayımı olan ortamın MKS olarak tanımlanması gerekliliği ile çelişmektedir. Q-öğrenme algoritması tek erkinli sistemlerde için başarıyla uygulandığı halde, çok erkinli sistemlerdeki doğrudan uygulamalarında optimal sonuca yakınsama garanti edilememektedir (Matignon et al., 2007).

Literatürde, Q-öğrenme yönteminin çok robotlu sistemlerde kullanımına ilişkin iki temel yaklaşım bulunmaktadır. Bu yaklaşımlar tüm robotların davranışlarını birlikte öğrendikleri merkezi öğrenme yaklaşımı ve her bir robotun tamamen bağımsız olarak öğrendiği dağıtık öğrenme yaklaşımlarıdır (Wang and de Silva, 2006b). Bölüm 3.3'te anlatıldığı üzere farklı çalışmalarda bu iki yaklaşımın özelliklerini birleştiren ancak genellikle uygulamalara özel olarak geliştirilmiş yöntemler de bulunmaktadır.

Bu çalışmada, merkezi ve dağıtık öğrenme yaklaşımları incelenmekte ve bu iki temel yaklaşımın avantajlarını birleştiren Strateji Planlamalı Dağıtık Öğrenme yaklaşımı önerilmiştir.

4.3.1 Merkezi öğrenme yaklaşımı

Merkezi öğrenme yaklaşımında, robotlar kendileriyle ortak bir amaca sahip diğer robotlarla birlikte genelleştirilmiş durum uzayı için tümleşik hareketlerinin birlikte değerlendirildiği ortak Q değerlerini birlikte öğrenmektedir. Öğrenme süreci Tanım 4.6. ile verildiği üzere $O^R = \cup_{j=1}^m O_j^R$ ortamı üzerinde gerçekleştirilmektedir. Buna göre, tüm ortama ilişkin genelleştirilmiş durum uzayı tüm robotların durum uzaylarının bileşim uzayından

$$S = \cup_{j=1}^m S_j \quad (4.10)$$

ve genelleştirilmiş hareket uzayı da tüm robotların hareket uzaylarının Kartezyen çarpımından

$$A = \prod_{j=1}^m A_j \quad (4.11)$$

oluşmaktadır. Tüm sistemin durum uzayı boyutu

$$|S| \leq \sum_{j=1}^m |S_j| \quad (4.12)$$

ve hareket uzayının boyutu

$$|A| = \prod_{j=1}^m |A_j| \quad (4.13)$$

şeklinde hesaplanmaktadır.

Tüm sistem tek bir ortam olarak tanımlandığı ve tüm robotların ortamda meydana gelen tüm değişiklikleri bildiği varsayımı dikkate alındığında ortamın MKS gibi tanımlanabildiği kabul edilmektedir (Matignon et al., 2007). Dolayısıyla optimal çözümlere yakınsaması beklenmektedir.

Ancak, tümleşik hareket uzayı kullanımı nedeniyle öğrenme uzayının boyutunun çok büyük olması sisteme büyük işlemsel yük getirmektedir. Ayrıca merkezi öğrenme yaklaşımları ya harici bir öğrenme veya koordinasyon merkezine ya da işbirlikçi robotlar arası güvenilir haberleşme ağına ihtiyaç duymaktadır. Oysa robotların çalışma ortamlarının doğası nedeniyle robotlar arası güvenilir haberleşme her zaman garanti edilememektedir.

Örnek olarak; Bölüm 4.2.5'te yer alan örnek ortamının çalışması ele alınsın. Merkezi öğrenme için tüm sistemin durum uzayı $S = S_1 \cup S_2$ ve hareket uzayı $A = A_1 \times A_2$ olmaktadır. Çizelge 4.1'de yer alan bilgiler ışığında durum uzayının boyutu $|S| = 4$ ve hareket uzayı boyutu $|A| = |A_1| \times |A_2| = 2 \times 8 = 16$ olmaktadır. Bu durumda öğrenme uzayının boyutunun $|S||A| = 4 \times 16 = 64$ olduğu görülmektedir. Bu değer sadece iki robotlu ve iki tip görevin bulunduğu bir ortam olduğu halde oldukça yüksektir.

4.3.2 Dağıtık öğrenme yaklaşımı

Dağıtık öğrenme yaklaşımında robotlar kendi durum ve hareket uzaylarına ilişkin Q değerlerini diğer robotların hareketlerinden ve öğrenme süreçlerinden bağımsız olarak öğrenmektedir. Herhangi bir R_j robotu öğrenme sürecini kendi O_j^R ortamı üzerinde gerçekleştirmektedir.

Bu yaklaşımın en büyük avantajı öğrenme uzayının boyutunun çok büyük olmaması dolayısıyla işlemsel yükün az olmasıdır. Bununla birlikte, robotların hareketlerini birbirlerinden bağımsız seçmeleri davranış çakışmalarına yol açmaktadır. Bu da optimal sonuca yakınsamanın önündeki en büyük engeldir (Wang and de Silva, 2006b).

Örnek olarak; Bölüm 4.2.5'te yer alan örnek ortamının çalışması ele alınsın. Çizelge 4.1'de yer alan bilgiler ışığında öğrenme uzaylarının boyutları incelendiğinde R_1 robotu için $|S_1||A_1| = 1 \times 2 = 2$ ve R_2 robotu için $|S_2||A_2| = 4 \times 8 = 32$ sonuçlarına ulaşılmaktadır. Bu değerler özellikle merkezi öğrenme yaklaşımı göz önüne alındığında oldukça makul seviyelerdedir. Bununla birlikte, her iki robot öğrenme sürecini birbirlerinin hareketlerini dikkate almadan gerçekleştirdikleri için T_1 görevinin yapılması veya yapılmaması konusunda her iki robot arasında davranış çakışmalarının ortaya çıkmasının kaçınılmaz olduğu görülmektedir.

4.3.3 Strateji planlamalı dağıtık öğrenme yaklaşımı

Bu çalışmada, robotların bireysel öğrenmelerine dayalı fakat davranış çakışmalarını engellemeye yönelik Strateji Planlamalı Dağıtık Öğrenme Yaklaşımı önerilmiştir. Bu yaklaşımda, öğrenme dağıtık olarak gerçekleştirilmektedir. Ancak her robot işbirliği içerisinde olduğu diğer robotların öğrenme davranışlarını dikkate almaktadır. İşbirlikçi robotlara farklı öğrenme stratejileri atanmıştır. Anlaşılır olma amacıyla, bu yaklaşım iki robotlu bir yapı ele alınarak açıklanmaktadır.

Tanım 4.11. $R_f, R_g \in R$ çok robotlu sistemde bulunan iki robot olsun. Bu iki robot için işbirliğine tabi görev kümesi

$$\Gamma_{i\bar{s}b} = \Gamma_f \cap \Gamma_g \quad (4.14)$$

olsun. Bu kesişim kümesi için

$$\Gamma_{i\bar{s}b} \neq \emptyset \quad (4.15)$$

şartı sağlanıyorsa R_f ve R_g robotları $T_z \in \Gamma_{i\bar{s}b}$ görevleri için işbirlikçi robotlardır.

Tanım 4.12. Herhangi bir T_z görevi için Bölüm 4.2’de açıklanan amaca uygun öğrenme yaklaşımı temel öğrenme stratejisi kabul edilmek üzere temel öğrenme stratejisi

$$H_{baz}^z := \begin{cases} T_z \in \Gamma_j & \text{görevleri için} \\ \text{yüksek kazançlı görevleri seçme} \\ \text{davranışını öğrenme stratejisi} \end{cases} \quad (4.16)$$

olarak tanımlanmaktadır.

Tanım 4.13. $|\Gamma_f| > |\Gamma_g|$ şartı sağlanmak koşuluyla R_f robotu için $T_z \in \Gamma_{i\bar{s}b}$ görevine ilişkin öğrenme stratejisi temel öğrenme stratejisine denk seçilmektedir.

$$H_f^z = H_{baz}^z \quad (4.17)$$

Tanım 4.14. $|\Gamma_f| > |\Gamma_g|$ şartı sağlanmak koşuluyla R_g robotu için $T_z \in \Gamma_{i\bar{s}b}$ görevine ilişkin öğrenme stratejisi temel öğrenme stratejisinin tümleyeni olarak seçilmektedir.

$$H_g^z = (H_{baz}^z)' \quad (4.18)$$

temel öğrenme stratejisinin tümleyeni, $(H_{baz}^z)'$,

$$(H_{baz}^z)' := \begin{cases} T_z \in \Gamma_j & \text{görevleri için } R_f \text{ robotunun} \\ \text{yapmaktan vazgeçtiği düşük kazançlı görevleri} \\ \text{seçme davranışını öğrenme stratejisi} \end{cases} \quad (4.19)$$

olarak tanımlanmaktadır.

Temel olarak, işbirlikçi iki robotlu bir sistem için verilen Tanım 4.13 ve Tanım 4.14 stratejik planlamalı dağıtık öğrenme yaklaşımını açıklamaktadır. Bu yaklaşım sayesinde hem davranış çakışmaları engellenmekte hem de sistemin toplam kazancı artırılmış olmaktadır. Burada basit bir anlatım sağladığı için iki robot üzerinden verilen bu yaklaşımın çok robotlu yapılara da uygulanması mümkündür.

Yaklaşımın uygulamasına örnek olarak Bölüm 4.2.5'te yer alan örnek ortamının çalışması ele alınsın. Buna göre işbirliğine tabi görev kümesi

$$\Gamma_{isb} = \{T_1\} \quad (4.20)$$

olmaktadır. Her iki robotun da yapma yeteneğine sahip olduğu T_1 görevi için temel öğrenme stratejisi

$$H_{baz}^1 := \begin{cases} \text{Yüksek öncelikli } T_1 \text{ görevlerini} \\ \text{seçme davranışını öğrenme stratejisi} \end{cases} \quad (4.21)$$

olarak belirlenmektedir. Tanım 4.13 gereğince R_2 robotunun T_1 görevleri için öğrenme stratejisi

$$H_2^1 := H_{baz}^1 \quad (4.22)$$

temel öğrenme stratejisine denk seçilmektedir. R_1 robotunun T_1 görevleri için öğrenme stratejisi ise Tanım 4.14 gereğince

$$H_1^1 := (H_{baz}^1)' \quad (4.23)$$

Denklem (4.24) ile verildiği üzere temel öğrenme stratejisinin tümleyeni olarak seçilmektedir.

$$(H_{baz}^1)' := \begin{cases} R_2 \text{ robotunun yapmaktan vazgeçtiği} \\ \text{düşük öncelikli } T_1 \text{ görevlerini} \\ \text{seçme davranışını öğrenme stratejisi} \end{cases} \quad (4.24)$$

Bu yaklaşımda öğrenme uzayının boyutu yine Bölüm 4.3.2’de verilen dağıtık öğrenme yaklaşımı ile aynıdır. Böylece işlemsel yükü dağıtık öğrenme ile aynı olan fakat merkezi öğrenme yaklaşımına benzer şekilde davranış çakışmalarının engellenebilmesi bir yaklaşım ortaya konulmuştur.

4.4 Durum Uzayının Belirlenmesi

Q-öğrenme yönteminin çok robotlu sistemlere uygulamasında karşılaşılan en büyük problem ayırık durum ve hareket uzayı kullanımı zorunluluğunun getirdiği ayırıklaştırma işleminden kaynaklanmaktadır. Q- öğrenme yöntemi, durum ve hareket uzaylarının sonlu ve ayırık olarak belirlenmiş olduğu MKS olarak nitelendirilebilen ortamlar üzerinden tanımlanmıştır (Yang and Gu, 2004). Çok robotlu sistemlerin kullanıldığı birçok gerçek ortam uygulamalarında ise sistemin doğasından kaynaklanan nedenlerle durum ve hareket uzaylarının sürekli değişkenler olduğu görülmektedir. Bu durum ve hareket uzaylarının ayırıklaştırılması zorunluluğu büyük bir problem olarak ortaya çıkmaktadır.

Bu çalışmada gerçekleştirilen uygulamada, hareket uzayı, robotun anons edilen görevi yapıp yapmama kararına ilişkin olduğu için ayırık olarak tanımlanmaktadır. Durum uzayı ise, görevlerin anons edilmeleri arasında geçen süre temel alınarak belirlenmektedir. Süre gibi sürekli bir değişkenin ayırık olarak kullanılması çeşitli tanımlama ve temsil etme problemlerine yol açmaktadır.

Bu çalışmada, sürekli durum uzayını sürekli değerlerle temsil eden ve ayrıklaştırma sürecine tabi tutarak ayrık değerlerle temsil eden üç farklı yaklaşım önerilmiştir. Bunlar; Dağılım Fonksiyonlu Sürekli Durum Uzayı (DFSDU), Sabit Aralıklı Ayrık Durum Uzayı (SAADU) ve Değişken Aralıklı Ayrık Durum Uzayı (DAADU) yaklaşımlarıdır. Ayrıntılar aşağıda verilmektedir.

4.4.1 Sabit aralıklı ayrık durum uzayı (SAADU) yöntemi

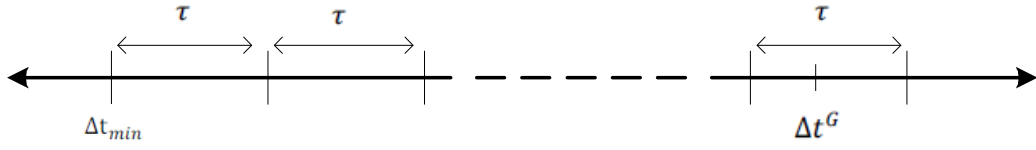
Bu yaklaşımda, sürekli durum uzayı sabit bir ayrıklaştırma temsil hassasiyeti kullanılarak ayrık değerler halinde temsil edilmektedir. Tüm durum uzayı öğrenme süreci öncesinde eşit aralıklı olarak gruplara ayrılmakta ve her bir grup durum uzayının farklı bir bileşenini oluşturmaktadır. Burada, öğrenilmesi gereken sürekli durum uzayı bilgisinin dahil olacağı ayrık grup tespit edilmekte ve o grubun temsil ettiği durum uzayı bileşeninin verisi olarak değerlendirilmektedir.

Tanım 4.15. Çok robotlu sistemin $t = t_s$ başlangıç anından $t = t_f$ ($t_s < t_f$) anına kadar çalıştığı kabul edilsin. Sistemde herhangi bir yüksek öncelikli T_q^v görevi $t = t_v$ anında ve bir sonraki düşük öncelikli T_p^u görevi $t = t_u$ anında gelmiş olsun. Buna göre, düşük öncelikli görev anons edildiğinde, yüksek öncelikli görevin bir önceki anonsundan itibaren geçen gerçek süre $\Delta t^G = t_u - t_v$ ile tanımlanmaktadır. Bu gerçek süre değeri ait olduğu ve $\{T_p^u, \Delta t_{(T_q^v|T_p^u)}\}$ ile temsil edilen ayrık durum uzayı bileşenine ait bilgi taşımaktadır.

Tanım 4.16. Gerçek süre değeri, Δt^G , için olabilecek en küçük değer Δt_{min} olsun. Gerçek süre değeri, Δt^G aşağıdaki şartı sağlıyorsa ζ 'nci ayrık durum uzayı ile temsil edilen gruba aittir.

$$\Delta t_{min} + \sum_{i=0}^{\zeta-1} \tau < \Delta t^G < \Delta t_{min} + \sum_{i=0}^{\zeta} \tau \quad (4.25)$$

Burada τ değeri ayrık grup büyüklüğünü temsil etmektedir ve sürecin başında belirlenen bir sabittir. Bu çalışmada kullanılan SAADU yönteminde, sürekli durum uzayı bilgisi süre verisi tabanlı ve tek boyutlu bir değişkendir. Ayrıklaştırılmış durum uzayını temsil eden gösterim Şekil 4.4'te yer almaktadır.



Şekil 4.4. Sabit Aralıklı Ayrıklaştırılmış Durum Uzayı

SAADU yöntemi planlama ve kullanım kolaylığı açısından etkin çözümler sunmaktadır. Ancak, en büyük problemi uygun τ değerinin belirlenmesidir. Sistem hakkında önceden ve ayrıntılı bilgi sahibi olunmayı gerektirir. Çözünürlüğün düşük tutulması işlemsel kolaylık sağlarken sistem hassasiyetinin kaybolmasına neden olmaktadır. Çözünürlüğün yüksek olması ise özellikle çok robotlu sistemler için büyük işlemsel yük getirmektedir. τ değeri, ne gereksiz durum bileşeni oluşturacak kadar küçük, ne de farklı nitelikteki bilgilerin aynı durum bileşeni ile temsil edilmesine sebep olacak kadar büyük olmalıdır.

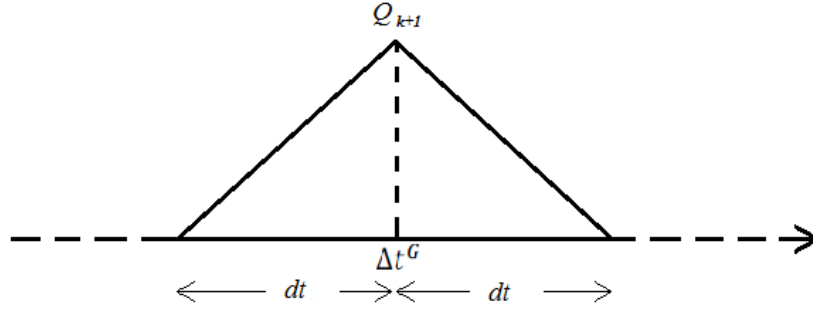
4.4.2 Dağılım fonksiyonlu sürekli durum uzayı (DFSDU) yöntemi

DFSDU yönteminde gerçek süre bilgisi, Δt^G , değerleri durum uzayı bileşenleri olarak kullanılmaktadır. Bu yöntemle göre öğrenilen Q değerleri her bir durum için tek bir değerle değil bu değer etrafında üçgen formülü bir dağılım fonksiyonu ile ifade edilmektedir. Bu yaklaşımın temeli, Δt^G durum değeri için öğrenilen Q değerinin Δt^G 'nin belirli bir $[-dt, dt]$ komşuluğu için de belli oranda etkili olabileceği kabulüne dayanmaktadır.

Tanım 4.17. Herhangi bir k adımında $s_k = \Delta t^G$ durumu için olası hareketlerden biri a_k olsun. Durum–hareket çifti (s_k, a_k) için bu adımda elde edilen Q değeri Q_{k+1} olmak üzere dağılım fonksiyonu $Q_f(t)$ aşağıdaki eşitlikle verilmektedir.

$$Q_f(t) = \begin{cases} 0 & , \quad t \leq \Delta t^G - dt \\ \frac{Q_{k+1}(t - (\Delta t^G - dt))}{dt} & , \quad \Delta t^G - dt \leq t \leq \Delta t^G \\ \frac{Q_{k+1}((\Delta t^G + dt) - t)}{dt} & , \quad \Delta t^G \leq t \leq \Delta t^G + dt \\ 0 & , \quad t \geq \Delta t^G + dt \end{cases} \quad (4.26)$$

Buna göre, $Q_f(t)$ dağılım fonksiyonu $t = \Delta t^G$ için Q_{k+1} değerine, $t = \Delta t^G - dt$ ve $t = \Delta t^G + dt$ için 0 değerine sahiptir. $Q_f(t)$ dağılım fonksiyonunun yapısı Şekil 4.5'te gösterilmektedir. Öğrenme süreci tamamlandıktan sonra öğrenilen $Q_f(t)$ fonksiyonları geçerli olan tüm t değerleri için toplanarak bütün durum-hareket uzayı için Q değerleri elde edilmektedir.



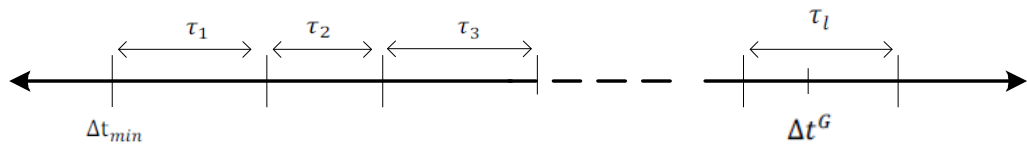
Şekil 4.5. $Q_f(t)$ Fonksiyonu

DFSDU yönteminin sağladığı en büyük avantaj ayrıklaştırma problemi yaşanmamasıdır. Ayrıca, ayrık durum uzayı yaklaşımında, öğrenme sırasında oluşmayan durum-hareket çiftlerine ilişkin Q -değerlerinin kullanılması gerektiğinde istenmeyen sonuçlar oluşabilmektedir. Oysa bu yöntemde, öğrenme sırasında bilgi edinilmeyen zaman aralıklarında bile bu anların yakın komşuluğunda bulunan değerler

aracılığıyla bilgi sahibi olunabilmektedir. DFSDU yönteminin en büyük dezavantajı ise öğrenme sürecine getirdiği daha yüksek bellek ihtiyacı ve artan işlemsel yüküdür.

4.4.3 Değişken aralıklı ayırık durum uzayı (DAADU) yöntemi

Bu çalışmada, ayırık durum uzayı yapısının kullanıldığı ancak durum uzayının önceden belirlenmek yerine öğrenme sırasında kümeleme yaklaşımı ile uyarlamalı olarak belirlendiği bir yöntem olan DAADU yöntemi ele alınmaktadır. Daha önce ele alınan ayırık durum uzayı tabanlı olan SAADU yönteminde sistem için gerekli hassasiyeti sağlayacak ayırıklaştırma çözünürlüğünün tespit edilmesi büyük bir problem olmaktaydı. Çözünürlük değerinin yüksek veya düşük olmasının getirdiği dezavantajlar bir tarafa, sistemin özelliklerine uygun bir çözünürlük değerinin tespit edilmesi de başlı başına bir problem olmaktadır. Önerilen yeni yöntemde ise durum uzayı sabit aralıklarla önceden belirlenmek yerine öğrenme sürecinde sistemin gereksinimlerine göre kümeleme yaklaşımı ile düzenlenmektedir. Bu yaklaşım sonunda elde edilen ayırıklaştırılmış durum uzayını temsil eden gösterim Şekil 4.6’da verilmektedir.



Şekil 4.6 Değişken Aralıklı Ayırıklaştırılmış Durum Uzayı

Takip eden alt bölümlerde, önce kısaca kümeleme yöntemleri ve ardışıl kümeleme algoritmaları anlatılacaktır. Daha sonra ardışıl kümeleme algoritmaları kullanılarak durum uzayının yapılandırılması yer almaktadır.

4.4.3.1 Ardışıl kümeleme

Kümeleme (clustering), verileri aralarındaki benzerlik veya farklılıkları temel alarak uygun gruplara ayırmayı hedefleyen öğreticisiz öğrenme yöntemlerinden biridir. Birçok uygulamada, çok fazla sayıdaki verilerin tek tek değerlendirilmesi mümkün olamamaktadır. Tüm verilerin ayrı ayrı işlenmesi yerine amaca uygun özelliklerine göre gruplara ayrılması ve her bir grubun kendine özgü bir öznitelikle sıfatlandırılarak kullanılması büyük kolaylık sağlamaktadır. Grupları oluştururken kullanılan özelliklerin seçimi, benzerlik/farklılık ölçütleri, kümeleme kriterleri ve kümeleme algoritması elde edilecek kümelerin yapısını ve verimliliğini belirlemede önemli rol oynamaktadır. Aynı veri çokluğu için farklı yaklaşımlarla farklı kümelemelerin elde edilmesi olasıdır.

$\mathcal{X} = \{x_1, x_2, \dots, x_N\}$, N elemanlı veri seti olmak üzere $\mathcal{X}$ 'in ℓ kümelemesi, $C_1, C_2, \dots, C_\ell$, aşağıdaki koşulları sağlamalıdır (Theodoridis and Koutroumbas, 2003):

- $C_i \neq \emptyset, i = 1 \dots \ell$
- $\bigcup_{i=1}^{\ell} C_i = \mathcal{X}$
- $C_i \cap C_j = \emptyset, i \neq j$ ve $i, j = 1 \dots \ell$

Yukarıdaki koşullar, kümeleme işleminin sonunda hiçbir kümenin boş kalmaması ve kümeleme sonucunda herhangi bir kümeye atanamamış bir veri bulunmaması gerektiğini söylemektedir. Ayrıca bu kümeleme tanımına göre, bir veri ancak bir kümenin elemanı olmak zorundadır, aynı anda başka bir kümenin daha elemanı olamaz. Ancak bu şarta uymayan bulanık kümeleme yaklaşımları da bulunmaktadır (Theodoridis and Koutroumbas, 2003).

Bir kümedeki veriler birbirlerine “daha benzer” ve diğer bir kümeyi tanımlayan öznitelik verisiyle “daha az benzer” olmak özelliğine sahiptir (Carvalho, 2006). Bir farklılık ölçütü (dissimilarity measure), $d: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, $\mathcal{X}$ üzerinde tanımlı bir fonksiyondur ve aşağıdaki şartları sağlamaktadır (Theodoridis and Koutroumbas, 2003).

- $\exists d_0 \in \mathbb{R} : -\infty < d_0 \leq d(x, y) < +\infty, \forall x, y \in \mathcal{X}$
- $d(x, x) = d_0, \forall x \in \mathcal{X}$
- $d(x, y) = d(y, x), \forall x, y \in \mathcal{X}$
- $d(x, y) = d_0 \Leftrightarrow x = y$
- $d(x, z) \leq d(x, y) + d(y, z), \forall x, y, z \in \mathcal{X}$

Bazı durumlarda uzaklık ifadesi de farklılık ölçütü ifadesinin yerine kullanılmaktadır. Örneğin, Öklid uzaklığı $d_0 = 0$ olan bir farklılık ölçütüdür. Benzer şekilde Manhattan normu ve sonsuz normu da farklılık ölçütü olarak kullanılabilir. Bunlara ilaveten bazı özel ölçütler de tasarlanmıştır (Spath, 1980).

Ardışıl kümeleme (sequential clustering) algoritmaları uygulaması kolay ve hızlı çalışan, sonuç olarak tek bir kümeleme yapısı oluşturan algoritmalar. Bu algoritmalar verileri çoğu kez bir defa veya bazen birkaç defa işleyerek kümeleme işlemini gerçekleştirmektedir. Ancak, elde edilecek kümeleme yapısı verilerin algoritmada işlenme sırasına bağlıdır. Verilerin farklı sıralarla kullanılması ile farklı kümelemelerin elde edilmesi olasıdır. Çoğu kümeleme algoritmasının ortak özelliği kümeleme sürecinin sonunda elde edilecek toplam küme sayısının başlangıçta bilinmesini gerektirmeleridir (Amador, 2005). Ancak bu, veri kümesinin yapısının ve özelliklerinin iyi bilinmesi gibi bir dezavantajı da beraberinde getirmektedir. Oysa ardışıl kümeleme algoritmaları için elde edilecek küme sayısının başlangıçta bilinmesi şart değildir, algoritma ilerledikçe verilerin yapısına bağlı olarak şekillenmektedir.

$\mathcal{X} = \{x_1, x_2, \dots, x_N\}$, N elemanlı veri seti ve $d(x_i, C_k)$ x_i verisiyle C_k kümesi arasındaki farklılık ölçütü olsun. Bu farklılık ölçütü C_k kümesinin tüm elemanları kullanılarak hesaplanabileceği gibi kümeyi temsil eden bir öznitelikle de bulunabilir. l_i mevcut küme sayısı, N toplam veri sayısı ve θ farklılık ölçütü eşik değeri olmak üzere temel bir ardışıl kümeleme algoritması aşağıda verilmektedir (Theodoridis and Koutroumbas, 2003).

Bu çalışmada önerilen yöntemde bu temel ardışıl kümeleme algoritması baz alınmış ve kullanım amacına uygun değişiklikler yapılmıştır.

Temel Ardışıl Kümeleme Algoritması

```

 $l_i = 1$ 
 $C_{l_i} = [x_1]$ 
for  $j = 2 \dots N$ 
    eşleşen küme bul  $C_i: d(x_j, C_i) = \max_{1 \leq k \leq l_i} d(x_j, C_k)$ 
    if  $d(x_j, C_i) > \theta$ 
         $l_i = l_i + 1$ 
         $C_{l_i} = [x_j]$ 
    else
         $C_i = C_i \cup [x_j]$ 
        küme parametrelerini güncelle
    end
end

```

4.4.3.2 Durum uzayının ardışıl kümeleme algoritmaları ile yapılandırılması

Önerilen yeni yöntemde, durum uzayı önceden belirlenmiş sabit aralıklarla ayrıklaştırılmak yerine öğrenme sürecinde kümeleme yaklaşımı ile düzenlenmektedir. Süre tabanlı olarak tanımlanmış olan sürekli yapıdaki durum uzayı yine ayrıklaştırılmakta fakat bu ayrıklaştırma önceden belirlenmiş sabit aralıklarla değil sistemin ihtiyaçlarına göre değişken aralıklarla yapılmaktadır.

Sabit aralıklı ayırık durum uzayı yapısında robotun o anki durum değişkeni Δt^G değerinin dahil olduğu aralık $\Delta t_{(T_q^v|T_p^u)}$ olarak kabul edilmekteydi. Bu yöntemde de robotun durum değişkeni olarak yine Δt^G değeri kullanılmaktadır. Kümeleme algoritmasında girdi olarak kullanılacak olan $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$, N elemanlı veri seti için her bir veri

$$x_j = \langle \Delta t_j^G, r_j^d \rangle, j = 1 \dots N \quad (4.27)$$

şeklinde iki parametre ile tanımlanmaktadır. r_j^d , robotun Δt^G anında yapabileceği olası iki hareketin sonucunda alması muhtemel ödül değerlerinin farkı olarak tanımlanmaktadır. Kümeleme algoritmasının sonunda elde edilecek her bir C_i kümesi $x_k = \langle \Delta t_k^G, r_k^d \rangle$, $\forall x_k \in C_i$ verilerine ilişkin bilgi taşımaktadır.

Tanım 4.18. Bir C_i kümesi için küme parametreleri aşağıdaki şekilde tanımlanmış olsun:

- $\min\Delta_i$: C_i kümesinde bulunan Δt^G değerlerinin en küçüğü

$$\min\Delta_i = \min(\Delta t_k^G) , \forall x_k \in C_i \quad (4.28)$$

- $\max\Delta_i$: C_i kümesinde bulunan Δt^G değerlerinin en büyüğü

$$\max\Delta_i = \max(\Delta t_k^G) , \forall x_k \in C_i \quad (4.29)$$

- $\text{mean}\Delta_i$: C_i kümesinde bulunan Δt^G değerlerinin ortalaması

$$\text{mean}\Delta_i = \frac{\sum_{\forall x_k \in C_i} \Delta t_k^G}{|C_i|} \quad (4.30)$$

- $\min\text{Limit}_i$: C_i kümesinin izin verilen en küçük Δt^G değeri θ_0 küme sınırları için tanımlanmış eşik değeri olmak üzere:

$$\min\text{Limit}_i = \text{mean}\Delta_i - \theta_0 \quad (4.31)$$

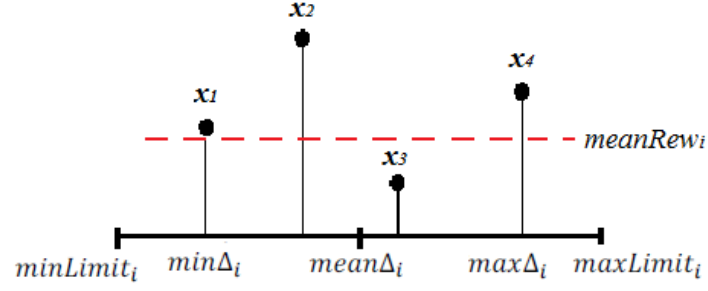
- $\max\text{Limit}_i$: C_i kümesinin izin verilen en büyük Δt^G değeri

$$\max\text{Limit}_i = \text{mean}\Delta_i + \theta_0 \quad (4.32)$$

- meanRew_i : C_i kümesinde bulunan r^d değerlerinin ortalaması

$$\text{meanRew}_i = \frac{\sum_{\forall x_k \in C_i} r_k^d}{|C_i|} \quad (4.33)$$

Örnek olarak; dört elemanlı bir kümeye ait küme parametreleri Şekil 4.7’de görülmektedir.



Şekil 4.7. Dört Elemanlı Bir C_i Kümesi için Küme Parametreleri

Bu çalışmada önerilen yöntemde kümeleme kriteri olarak iki aşamalı farklılık testi önerilmiştir.

Tanım 4.19. İlk farklılık ölçütü d^1 aşağıdaki eşitlik ile verilmektedir.

$$d^1 = \begin{cases} 1, & \min\Delta_i < \Delta_j^G < \max\Delta_i \\ 0, & \text{diğer} \end{cases}, \quad i = 1 \dots l_i, j = 1 \dots N \quad (4.34)$$

l_i mevcut küme sayısını göstermektedir. d^1 farklılık ölçütü ikili tabanda çalışan bir fonksiyondur. x_j verisinin ilk parametresi olan Δ_j^G değerinin eşleştiği mevcut küme yoksa '0', eşleşen en az bir küme varsa '1' değeri döndürmektedir. Eğer $d^1 = 0$ ise yeni bir C_{l_i+1} kümesi oluşturularak x_j verisi o kümeye dahil edilir.

Tanım 4.20. Eğer $d^1 = 1$ ise x_j verisinin eşleşen kümeye uyumunu test etmek için Denklem (4.34) ile tanımlanan ikinci bir farklılık ölçütü d^2 kullanılır. İşaret test fonksiyonu

$$\text{signComp}(x, y) = \begin{cases} 1, & \text{eğer } x.y > 0 \\ -1, & \text{eğer } x.y < 0 \end{cases} \quad (4.35)$$

olmak üzere, d^2 farklılık ölçütü Denklem (4.36) ile tanımlanmaktadır.

$$d^2 = \text{signComp}(r_j^d, \text{meanRew}_i) \quad (4.36)$$

d^2 farklılık ölçütü, x_j verisinin ikinci parametresi olan r_j^d değeri ile eşleştiği kümenin ortalama r^d değeri aynı yönlü ise '1', değilse '-1' değerine sahip olmaktadır. $d^2 = 1$ olduğunda x_j verisi eşleşen kümeye dahil edilir ve küme parametreleri güncellenir. Eğer $d^2 = -1$ ise x_j verisi yine eşleşen kümeye dahil edilir, fakat bu defa küme parametreleri güncellenmez. Önerilen algoritma için bu tip kümeler "karışık küme" olarak adlandırılmıştır. Temel anlamda x_j verisinin kümeye atanma işlemi bu adımla tamamlanmaktadır. Ancak bundan sonra "karışık küme" olarak adlandırılan kümeler için aykırı veri tespit algoritması uygulanmalıdır. $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ N elemanlı veri seti, d^1 ve d^2 farklılık ölçütleri ve l_i mevcut küme sayısı olmak üzere ayrık durumların tespiti için önerilen ardışıl kümeleme algoritması aşağıda verilmektedir.

Önerilen Ardışıl Kümeleme Algoritması

```

 $l_i = 1$ 
 $C_{l_i} = [x_1]$ 
for  $j = 2 \dots N$ 
     $d^1$  farklılık ölçütü ile eşleşen küme bul ( $C_i$ )
    if  $d^1 = 0$ 
         $l_i = l_i + 1$ 
         $C_{l_i} = [x_j]$ 
    elseif  $d^1 = 1$ 
         $C_i = C_i \cup [x_j]$ 
         $d^2$  farklılık ölçütü ile eşleşen kümeye uyumu bul
        if  $d^2 = 1$ 
            küme parametrelerini güncelle
        elseif  $d^2 = -1$ 
             $C_i$  : "karışık küme" olarak işaretle
        end if
    end if
    "aykırı veri tespit algoritması" çalıştır
end for

```

“Karışık küme” olarak adlandırılan kümelerin elemanlarından aykırı (outlier) olan verilerin tespit edilmesi amacıyla aşağıdaki prosedür tanımlanmıştır.

Tanım 4.21. C_i^+ ve C_i^- sırasıyla karışık küme C_i 'nin $meanRew_i$ parametresiyle aynı yönlü olan ve aynı yönlü olmayan elemanlarını içeren altkümeleri olsun. Bu alt kümeler aşağıdaki şartları sağlamaktadır:

$$C_i^+ \cup C_i^- = C_i \quad ve \quad C_i^+ \cap C_i^- = \emptyset \quad (4.37)$$

e_i^+ ve e_i^- sırasıyla C_i^+ ve C_i^- altkümelerinin eleman sayıları olmak üzere Denklem (4.38) ve Denklem (4.39) ile verilen yoğunluk test fonksiyonları tanımlansın.

$$y_i^+ = \frac{e_i^+}{max\Delta_i - min\Delta_i} \quad (4.38)$$

$$y_i^- = \frac{e_i^-}{max\Delta_i - min\Delta_i} \quad (4.39)$$

Bu fonksiyonlar altkümeleri oluşturan elemanların küme içerisindeki yoğunluklarını göstermektedir. Eğer y_i^+ ile y_i^- arasındaki fark belli bir θ_1 eşik değerinden büyük ise C_i^- altkümelerini oluşturan elemanlar C_i kümesi için aykırı eleman kabul edilir. Aksine y_i^+ ile y_i^- arasındaki fark belli bir θ_2 eşik değerinden küçük ise C_i^+ altkümelerini oluşturan elemanlar C_i kümesi için aykırı eleman kabul edilerek C_i küme parametreleri C_i^- altkümeleri elemanlarına göre yeniden düzenlenir. Eğer y_i^+ ile y_i^- arasındaki bu iki eşik değerinin arasında ise küme ayırma algoritması olarak adlandırılan bir prosedür uygulanarak C_i^- altkümeleri elemanlarının yoğunlaştığı bölgeler tespit edilir ve yeni kümeler oluşturulur.

l_k mevcut karışık küme sayısı, y_i^+ ile y_i^- Denklem (4.38) ve Denklem (4.39) ile tanımlanan yoğunluk fonksiyonları ve θ_1 ile θ_2 yoğunluk fonksiyonu alt ve üst eşik değerleri olmak üzere aykırı veri tespit algoritması aşağıda yer almaktadır.

Aykırı Veri Tespit Algoritması

```

for  $jj = 1 \dots l_k$ 
   $C_{jj}^+$  ve  $C_{jj}^-$  altkümelerini oluştur
   $y_{jj}^+$  ve  $y_{jj}^-$  değerlerini bul
  if  $y_{jj}^+ - y_{jj}^- > \theta_1$ 
     $C_{jj}^-$  altkümesi elemanlarını aykırı veri olarak işaretle
  elseif  $y_{jj}^+ - y_{jj}^- < \theta_2$ 
     $C_{jj}^+$  altkümesi elemanlarını aykırı veri olarak işaretle
     $C_{jj}$  küme parametrelerini  $C_{jj}^+$  altkümesi
      elemanlarına göre güncelle
  else
    “küme ayırma algoritması” çalıştır
  end if
end for

```

Küme ayırma algoritması karışık küme olarak nitelendirilen kümelerdeki aykırı verilerin yoğunlaştığı bölgeleri tespit ederek bu bölgelerin ayrı bir küme olarak adlandırılması amacıyla kullanılmaktadır. L^+ ve L^- küme ayırma alt ve üst sınırlarını gösteren değişkenler, $m\Delta^+$ ve $m\Delta^-$ aykırı verilerin en küçük ve en büyük Δt^G değerlerini gösteren değişkenler olmak üzere küme ayırma algoritması aşağıdaki şekildedir. Bu yöntem sonucunda tüm $x_j = \langle \Delta t_j^G, r_j^d \rangle$, $j = 1 \dots N$ verilerinin dahil olduğu durum kümeleri belirlenmektedir. Böylece, durum uzayının ayırıklaştırma işlemi sona ermektedir.

Küme Ayırma Algoritması

```

 $L^+ = \max \Delta_{jj}$ 
 $L^- = \min \Delta_{jj}$ 
 $y_{jj}^-(0) = y_{jj}^-$ 
for  $ii = 1 \dots e_{jj}^-$ 
   $C_{jj}^- = C_{jj}^- - \{C_{jj}^-(ii)\}$ 
   $y_{jj}^-(k) = \frac{e_{jj}^- - ii}{m\Delta^+ - m\Delta^-}$ 
  if  $y_{jj}^-(k) > y_{jj}^-(k-1)$ 
     $L^- = m\Delta^-$ 
  end if
  if  $y_{jj}^-(k) < y_{jj}^-(k-1)$ 
     $L^+ = m\Delta^+$ 
  end if
end for
 $C_{jj}^-$  altkümesinin  $L^- < \Delta t_{jj}^G < L^+$  olan elemanları için
  yeniküme tanımla. Bu elemanları  $C_{jj}$  kümesinden çıkar

```

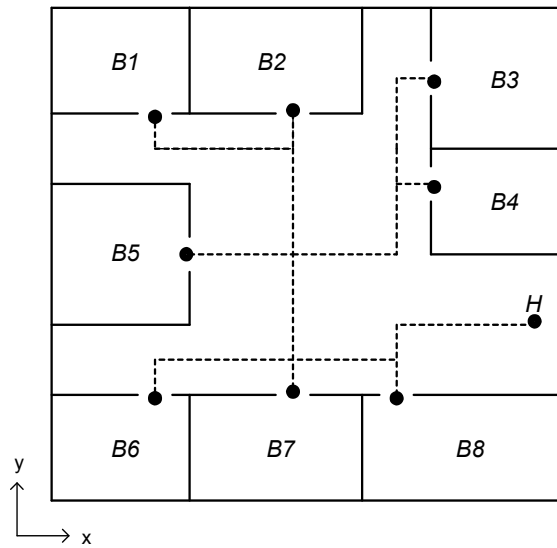
BÖLÜM 5

UYGULAMA

Bu bölümde, çok robotlu sistemlerde görev atama performansını artırmak amacıyla önerilen ve Bölüm 4'te ayrıntıları verilen yaklaşımların uygulama koşullarına ilişkin bilgiler yer almaktadır. Aşağıda öncelikle uygulama ortamının yapısı ve özellikleri sonra da çok robotlu sistemin ve görevlerin nitelikleri verilmektedir.

5.1 Uygulama Ortamı

Önerilen yaklaşımların uygulamaları için Şekil 5.1'de verilen deneysel ortam kullanılmıştır. Bu deneysel ortam şekilde de görüldüğü üzere $B1, \dots, B8$ ile adlandırılan 8 adet odadan, odalar arası bağlantıları sağlayan koridorlardan ve H olarak adlandırılan şarj bölümünden oluşmaktadır. Yapılması gereken görevler sadece odalarda oluşmaktadır. Robotların görevleri yapacakları odalara ulaşimleri Şekil 5.1'de noktalı çizgilerle gösterilen güzergahlar üzerinden sağlanmaktadır.



Şekil 5.1 Deneysel Uygulama Ortamı

Uygulama ortamının fiziksel yapısını belirleyen odaların metre cinsinden en (x) ve boy (y) ölçüleri Çizelge 5.1’de ve yine metre cinsinden odalar arası uzaklık bilgileri Çizelge 5.2’de verilmektedir.

Çizelge 5.1 Uygulama Ortamının Fiziksel Ölçüleri

	B1	B2	B3	B4	B5	B6	B7	B8
x (m)	4	5	4	4	4	4	5	6
y (m)	3	3	4	3	4	3	3	3

Çizelge 5.2 Odalar Arası Mesafeler

	B1	B2	B3	B4	B5	B6	B7	B8	H
B1	—	6	17	14	11	16	12	15	7
B2	6	—	13	10	7	12	8	11	13
B3	17	13	—	5	12	17	13	10	12
B4	14	10	5	—	9	14	10	7	9
B5	11	7	12	9	—	11	7	10	12
B6	16	12	17	14	11	—	6	15	17
B7	12	8	13	10	7	6	—	11	13
B8	15	11	10	7	10	15	11	—	6
H	7	13	12	9	12	17	13	6	—

5.2 Çok Robotlu Sistemlerin ve Görevlerin Yapısı

Etkinliğini göstermek üzere, önerilen yöntemlerin uygulamaları iki farklı sistem üzerinde gerçekleştirilmiştir. Her iki sistem de farklı yeteneklere sahip altı adet robottan (R_1, R_2, R_3, R_4, R_5 ve R_6) oluşmaktadır. Robotlardan farklı nitelikteki beş tip görevi (T_1, T_2, T_3, T_4 ve T_5) yapmaları beklenmektedir. Bu görevler farklı donanımsal özelliklere sahip robotlar tarafından yapılmak zorundadır. Görevlerin ve robot sistemlerinin yapısı aşağıda ele alınmaktadır.

5.2.1 Görevlerin özellikleri

Uygulamada kullanılan her iki robot sistemi için de görevlerin özellikleri aynıdır. Görevlerin nitelikleri Çizelge 5.3'te verilmektedir.

Çizelge 5.3 Görevler ve Nitelikleri

Görev Tipi	Görev Niteliği
T_1	Nesne Taşıma
T_2	Hassas Nesne Taşıma
T_3	Temizlik
T_4	Gözlemlleme
T_5	Nesne Arama

Bu görevler, farklı zamanlarda ve farklı sıralarla merkezi bir kaynak tarafından anons edilmektedir. Her bir görev için o görevi tanımlayan çeşitli parametreler mevcuttur. Bu parametreler; nesne taşıma görevleri için nereden nereye taşınacağını, temizlik/gözlemlleme görevleri için nerenin temizleneceğini/gözlemleneceğini, yine gözlemlleme görevi için görev süresini ve nesne arama görevi için nerede kaç tane nesne aranacağını tanımlamaktadır. Bu parametreler Çizelge 5.4'te özetlenmektedir.

Çizelge 5.4 Görevler ve Tanımlama Parametreleri

Görev Tipi	Parametre-1	Parametre-2	Parametre-3	Parametre-4
T_1	Nereden	Nereye	—	—
T_2	Nereden	Nereye	—	—
T_3	Nerede	—	—	—
T_4	Nerede	—	Süre	—
T_5	Nerede	—	—	Nesne Adeti

Görevler çalışma süresince rassal olarak ve tüm görev tipleri için hemen hemen eşit sayıda üretilmektedir. Görev üretim merkezinin bir seferde anons ettiği görev sayısı iki ila beş arasında değişmektedir. Görevlerin anons edilme anları ve sıralarının önceden bilinmemesi sebebiyle robotların görev atamaları ancak görevler anons edildikten sonra anlık olarak yapılabilir. Yapılacak görevler görev üretim merkezi tarafından tüm robotlara anons edilmektedir. Görevler anons edildiğinde robotlar o esnada bir başka görevi yapmıyorsa ve anons edilen görevi yapabilme yeteneğine sahip ise o görevi yapmak üzere görev üretim merkezini bilgilendirmektedir. Görev atama işlemi görevi anons eden kaynak tarafından gerçekleştirilmektedir. Sistemin yapısı bir robotun birden fazla göreve talip olması imkânı tanımaktadır. Ancak görev atama süreci tamamlandığında bir robot sadece bir iş için görevlendirilmektedir ve bir görev sadece bir robot tarafından yapılabilir.

Üretilen her bir görev, düşük ve yüksek öncelikli görevler olmak üzere iki farklı öncelik derecesine sahiptir. Karakteristik olarak, düşük ve yüksek öncelik dereceli görevler tüm görevlerin içinde sırasıyla %65 ve %35'lik dilimlere sahiptir.

5.2.2 Çok robotlu sistemlerin yapısı

Görevlerin birbirinden farklı niteliklere sahip olması görevleri yapacak olan robotların da farklı özelliklere ve yeteneklere sahip olmasını zorunlu kılmaktadır. Robotlar farklı donanım özelliklerine sahiptir dolayısıyla her bir robotun yapabildiği görevler yeteneklerine bağlı olarak değişmektedir. Aşağıda farklı iki sistem için robotların yapabildikleri görevler ve robot parametreleri anlatılmaktadır.

5.2.2.1 Çok robotlu sistem-I (CRS-I)

Sistem tamamen heterojen yapıya sahip olduğu için her robotun yapma yeteneğine sahip olduğu görevler farklıdır. İlk sistem olan CRS-I için sistemi oluşturan robotlar ve

yapabildikleri görevler Çizelge 5.5’te gösterilmektedir. Robotların yapabildiği görevler " + " ile işaretlenmiştir.

Çizelge 5.5 ÇRS-I için Robotlar ve Yapabildikleri Görevler

Robotlar	Görev Tipi				
	T_1	T_2	T_3	T_4	T_5
R_1	+				
R_2	+	+			
R_3			+		
R_4			+	+	
R_5				+	+
R_6				+	

Robotlar arası farklılık sadece sahip oldukları donanım özellikleri değildir aynı zamanda başlangıç parametreleri (başlangıç enerjileri veya başlangıç konumları), yol alma ve tarama hızları ile şarj harcama süreleri gibi çeşitli kriterler açısından da farklı nitelikte olmaları mümkündür. ÇRS-I için robotların ilgili parametreleri Çizelge 5.6 ile verilmektedir.

Çizelge 5.6 ÇRS-I için Robot Parametreleri

Parametreler	R_1	R_2	R_3	R_4	R_5	R_6
İlerleme Hızı (m/s)	0,3	0,2/0,15	0,2	0,3	0,25	0,3
Tarama Hızı (m/s)	0,2	0,15	0,1	0,15	0,15	0,2
Başlangıç Şarjı	%80	%30	%50	%80	%60	%75
Deşarj Süresi (s)	4000	2500	3200	2800	4250	3000
Başlangıç Konumu	H	H	H	H	H	H

Bu parametrelerden ilerleme hızı robotların odalar arası ortalama ilerleme hızını göstermektedir. R_2 robotunun iki adet ilerleme hızı değeri bulunmaktadır. İlki T_1 görevini yaparken, ikincisi de T_2 görevini yaparken yol alma hızını göstermektedir. T_2 görevi hassas nesne taşıma görevi olduğu için robotun daha yavaş hareket etmesi uygun olmaktadır. Tarama hızı da temizlik ve nesne arama görevleri için robotların ortamda dolaşma hızlarını vermektedir. Başlangıç şarjı sistem ilk çalışmaya başladığında robotların mevcut enerji durumlarını ifade etmektedir. Deşarj süresi tam dolu konumda olduklarında şarjları tükenmeden önceki görev yapma sürelerini göstermektedir. Tüm robotların başlangıçta Şekil 5.1’de H ile işaretli olan şarj ünitesi bölümünde bulundukları varsayılmıştır.

5.2.2.2 Çok robotlu sistem-II (CRS-II)

İkinci çok robotlu sistem bir önceki sisteme göre daha karmaşık bir yapıya sahiptir. Bu karmaşıklık robotların daha fazla tip görevi yapma yeteneğine sahip olmalarından ve her bir görevin daha fazla sayıda robot tarafından yapılabilmesinden kaynaklanmaktadır. ÇRS-II için sistemi oluşturan robotlar ve yapabildikleri görevler Çizelge 5.7’de verilmektedir. Robotların yapabildiği görevler "+" ile işaretlenmiştir.

Çizelge 5.7 ÇRS-II için Robotlar ve Yapabildikleri Görevler

Robotlar	Görev Tipi				
	T_1	T_2	T_3	T_4	T_5
R_1			+	+	+
R_2				+	
R_3	+				+
R_4	+	+			
R_5			+	+	
R_6	+			+	+

Robotlar birbirlerinden farklı başlangıç parametreleri (başlangıç enerjileri veya başlangıç konumları), yol alma ve tarama hızları ile şarj harcama süreleri gibi değerlere sahiptir. ÇRS-II için robotların ilgili parametreleri Çizelge 5.8 ile verilmektedir.

Çizelge 5.8 ÇRS-II için Robot Parametreleri

Parametreler	R_1	R_2	R_3	R_4	R_5	R_6
İlerleme Hızı (m/s)	0,3	0,3	0,25	0,3	0,2	0,25
Tarama Hızı (m/s)	0,25	0,20	0,15	0,15	0,2	0,2
Başlangıç Şarjı	%75	%80	%60	%80	%55	%85
Deşarj Süresi (s)	4200	4750	3900	3500	4250	3000
Başlangıç Konumu	H	H	H	H	H	H

5.3 Uygulama ve Hesaplamalar

Önerilen yaklaşımlar çok robotlu bir sistem modeli üzerinde yukarıda yer alan deneysel uygulama ortamında yine yukarıda yer alan parametre değerleri kullanılarak gerçekleştirilmiştir. Bu gerçeklemeler temel olarak üç aşamada gerçekleştirilmiştir:

- 1- Sistem öncelikle hiçbir öğrenme algoritması kullanılmadan test edilmiştir. Bu aşamada robotlar arası görev atama problemi algoritmaları çalıştırılarak öğrenme süreci için gerekli veriler elde edilmiştir.
- 2- Öğrenme aşamasında önerilen çeşitli yaklaşımlar kullanılarak mevcut veriler üzerinden çevrimdışı olarak öğrenme süreci gerçekleştirilmiştir.
- 3- Öğrenilen bilgiler kullanılarak robotlar arası görev atama algoritmaları tekrar işletilerek sonuçlar elde edilmiştir.

Uygulamaların görev atama aşamasında müzayede süreci için gerek görev üretim merkezi tarafından görev ücretlerinin hesaplanması gerekse robotların görevlere talip olmak için gerekli görev bedellerinin hesaplanması amacıyla görevler için harcanacak süre tabanlı bir yaklaşım kullanılmıştır. Görevler için harcanacak sürelerin hesaplanma yöntemi görev tipine bağlı olarak değişmektedir.

Robotların görevi yapmak üzere aldıkları yolu iki farklı tipte ele almak mümkündür. İlki robotların bulundukları konumdan görev alanına gitmek için aldıkları yol (yol_1) iken ikincisi ise görevi yapmak için aldıkları yoldur (yol_2). yol_2 değişkeni nesne taşıma görevleri (T_1 ve T_2) için nesnenin bulunduğu konum ile taşınacağı konum arasındaki yoldur. Nesne taşıma göre için harcanacak toplam süre:

$$t = \frac{yol_1}{ilerleme\ hızı_1} + \frac{yol_2}{ilerleme\ hızı_2} \quad (5.1)$$

ifadesiyle hesaplanmaktadır. T_1 görevi için $ilerleme\ hızı_1$ değeri $ilerleme\ hızı_2$ değeri ile aynıdır. Fakat T_2 görevi için $ilerleme\ hızı_1$ değeri $ilerleme\ hızı_2$ değerinden farklıdır.

Temizlik görevi (T_3) ve nesne arama görevi (T_5) için yol_2 değişkeni taranacak ortamın alanı olarak alınmalıdır. Bu sebeple toplam görev süresi aşağıdaki eşitlikle bulunmaktadır.

$$t = \frac{yol_1}{ilerleme\ hızı_1} + \frac{alan}{tarama\ hızı} \quad (5.2)$$

Gözlemleme görevi (T_4) için harcanacak gözlemleme süresi üretici kaynak tarafından verilmektedir. Bu durumda, toplam görev süresi Denklem (5.3) ile verilmektedir.

$$t = \frac{yol_1}{ilerleme\ hızı_1} + Gözlemleme\ süresi \quad (5.3)$$

BÖLÜM 6

UYGULAMA SONUÇLARI

Bu çalışmada, çok robotlu bir sistemde robotlara öğrenme yeteneği kazandırılarak robotlar arası görev atama problemine etkin çözümler getirilmesi amaçlanmıştır. Robotların geçmiş görev atama deneyimlerinden gelecek uygulamalarda yararlanmaları ile sistem performansının artırılması beklenmektedir. Görev atama problemlerinin temel hedefi sistem kaynaklarının en uygun şekilde kullanılmaları ile mümkün olduğunca çok sayıda görevin yapılmasının sağlanmasıdır. Gerçekleştirilen uygulamalarda sistem performansı görevlerin yapılma oranları üzerinden değerlendirilmektedir. Bu bölümde öncelikle, uygulamada kullanılan ve Bölüm 5'te anlatılan çok robotlu sistemlerin Bölüm 4.2'de anlatıldığı üzere modellenmesi verilmektedir. Daha sonra, Q-öğrenme yönteminin çok robotlu sistemlerde kullanımına dair yaklaşımlar görevlerin yapılma oranları ve yöntemlerin sisteme getirdiği işlemsel yük kriterlerine göre değerlendirilerek kullanımı en uygun yaklaşım belirlenmektedir. Son olarak da önerilen durum uzayı yaklaşımlarına dair sonuçlar ele alınmakta ve görevlerin yapılma oranları dikkate alınarak sonuçlar değerlendirilmektedir.

6.1 Uygulamada Kullanılan Çok Robotlu Sistemlerin Modellenmesi

Uygulamalar, ayrıntıları Bölüm 5'te verilen iki farklı ortam kullanılarak gerçekleştirilmiştir. Her iki ortam da farklı yeteneklere sahip altı robotlu sistemler olup bu robotlardan farklı nitelikteki beş tip görevi yapmaları beklenmektedir. Bu ortamlar için görev nitelikleri aynı olmakla birlikte sistemleri oluşturan robotların özellikleri ve yetenekleri farklıdır. ÇRS-I ve ÇRS-II ortamları için robot kümesi ve görev kümesi sırasıyla aşağıdaki şekilde tanımlıdır:

$$R = \{R_1, R_2, R_3, R_4, R_5, R_6\} \quad (6.1)$$

$$T = \{T_1, T_2, T_3, T_4, T_5\} \quad (6.2)$$

6.1.1 ÇRS-I modeli

ÇRS-I’de bulunan her bir robot için ortam modellemesi görev kümeleri, durum ve hareket uzayları bağlamında Çizelge 6.1’de verilmektedir.

Çizelge 6.1. ÇRS-I’de Bulunan Robotlara İlişkin O_j^R Bilgileri

	w_j ve Γ_j	S_j	A_j	$ S_j $ ve $ A_j $
R_1	$w_1 = \{1\}$ $\Gamma_1 = \{T_1\}$	$S_1 = \{s^{11}\}$ $s^{11} = \{T_1^u, \Delta t_{(T_1^v T_1^u)}\}$	$A_1 = \{a_{1,1}^{11}, a_{1,2}^{11}\}$	$ S_1 = 1$ $ A_1 = 2$
R_2	$w_2 = \{1,2\}$ $\Gamma_2 = \{T_1, T_2\}$	$S_2 = \{s^{11}, s^{12}, s^{21}, s^{22}\}$ $s^{11} = \{T_1^u, \Delta t_{(T_1^v T_1^u)}\}$ $s^{12} = \{T_1^u, \Delta t_{(T_2^v T_1^u)}\}$ $s^{21} = \{T_2^u, \Delta t_{(T_1^v T_2^u)}\}$ $s^{22} = \{T_2^u, \Delta t_{(T_2^v T_2^u)}\}$	$A_2 = \left\{ \begin{matrix} a_{2,1}^{11}, a_{2,2}^{11}, \\ a_{2,1}^{12}, a_{2,2}^{12}, \\ a_{2,1}^{21}, a_{2,2}^{21}, \\ a_{2,1}^{22}, a_{2,2}^{22} \end{matrix} \right\}$	$ S_2 = 4$ $ A_2 = 8$
R_3	$w_3 = \{3\}$ $\Gamma_3 = \{T_3\}$	$S_3 = \{s^{33}\}$ $s^{33} = \{T_3^u, \Delta t_{(T_3^v T_3^u)}\}$	$A_3 = \{a_{3,1}^{33}, a_{3,2}^{33}\}$	$ S_3 = 1$ $ A_3 = 2$
R_4	$w_4 = \{3,4\}$ $\Gamma_4 = \{T_3, T_4\}$	$S_4 = \{s^{33}, s^{34}, s^{43}, s^{44}\}$ $s^{33} = \{T_3^u, \Delta t_{(T_3^v T_3^u)}\}$ $s^{34} = \{T_3^u, \Delta t_{(T_4^v T_3^u)}\}$ $s^{43} = \{T_4^u, \Delta t_{(T_3^v T_4^u)}\}$ $s^{44} = \{T_4^u, \Delta t_{(T_4^v T_4^u)}\}$	$A_4 = \left\{ \begin{matrix} a_{4,1}^{33}, a_{4,2}^{33}, \\ a_{4,1}^{34}, a_{4,2}^{34}, \\ a_{4,1}^{43}, a_{4,2}^{43}, \\ a_{4,1}^{44}, a_{4,2}^{44} \end{matrix} \right\}$	$ S_4 = 4$ $ A_4 = 8$
R_5	$w_5 = \{4,5\}$ $\Gamma_5 = \{T_4, T_5\}$	$S_5 = \{s^{44}, s^{45}, s^{54}, s^{55}\}$ $s^{44} = \{T_4^u, \Delta t_{(T_4^v T_4^u)}\}$ $s^{45} = \{T_4^u, \Delta t_{(T_5^v T_4^u)}\}$ $s^{54} = \{T_5^u, \Delta t_{(T_4^v T_5^u)}\}$ $s^{55} = \{T_5^u, \Delta t_{(T_5^v T_5^u)}\}$	$A_5 = \left\{ \begin{matrix} a_{5,1}^{44}, a_{5,2}^{44}, \\ a_{5,1}^{45}, a_{5,2}^{45}, \\ a_{5,1}^{54}, a_{5,2}^{54}, \\ a_{5,1}^{55}, a_{5,2}^{55} \end{matrix} \right\}$	$ S_5 = 4$ $ A_5 = 8$
R_6	$w_6 = \{4\}$ $\Gamma_6 = \{T_4\}$	$S_6 = \{s^{44}\}$ $s^{44} = \{T_4^u, \Delta t_{(T_4^v T_4^u)}\}$	$A_6 = \{a_{6,1}^{44}, a_{6,2}^{44}\}$	$ S_6 = 1$ $ A_6 = 2$

Çizelge 6.1’de yer alan veriler incelendiğinde ÇRS-I’in Denklem (6.3) ve Denklem (6.4) ile verilen şartları sağlayan iki alt ortamın, O^{R1} ve O^{R2} , bileşimi olarak modellenebildiği görülmektedir.

$$O^{R1} \cup O^{R2} = O^R \quad (6.3)$$

$$O^{R1} \cap O^{R2} = \emptyset \quad (6.4)$$

O_j^R, R_j robotu için robot ortamını göstermek üzere O^{R1} ve O^{R2} alt ortamları aşağıdaki şekilde ifade edilmektedir:

$$O^{R1} = O_1^R \cup O_2^R \quad (6.5)$$

$$O^{R2} = O_3^R \cup O_4^R \cup O_5^R \cup O_6^R \quad (6.6)$$

6.1.2 ÇRS-II modeli

ÇRS-II ortamında bulunan robotlar için modellemeler Çizelge 6.2’de verilmektedir.

Çizelge 6.2. ÇRS-II’de Bulunan Robotlara İlişkin O_j^R Bilgileri

	w_j ve Γ_j	S_j	A_j	$ S_j $ ve $ A_j $
R_1	$w_1 = \{3,4,5\}$ $\Gamma_1 = \{T_3, T_4, T_5\}$	$S_1 = \begin{Bmatrix} s^{33}, s^{34}, s^{35} \\ s^{43}, s^{44}, s^{45} \\ s^{53}, s^{54}, s^{55} \end{Bmatrix}$ $s^{33} = \{T_3^u, \Delta t_{(T_3^v T_3^u)}\}$ $s^{34} = \{T_3^u, \Delta t_{(T_4^v T_3^u)}\}$ $s^{35} = \{T_3^u, \Delta t_{(T_5^v T_3^u)}\}$ $s^{43} = \{T_4^u, \Delta t_{(T_3^v T_4^u)}\}$ $s^{44} = \{T_4^u, \Delta t_{(T_4^v T_4^u)}\}$ $s^{45} = \{T_4^u, \Delta t_{(T_5^v T_4^u)}\}$ $s^{53} = \{T_5^u, \Delta t_{(T_3^v T_5^u)}\}$ $s^{54} = \{T_5^u, \Delta t_{(T_4^v T_5^u)}\}$ $s^{55} = \{T_5^u, \Delta t_{(T_5^v T_5^u)}\}$	$A_1 = \begin{Bmatrix} a_{1,1}^{33}, a_{1,2}^{33} \\ a_{1,1}^{34}, a_{1,2}^{34} \\ a_{1,1}^{35}, a_{1,2}^{35} \\ a_{1,1}^{43}, a_{1,2}^{43} \\ a_{1,1}^{44}, a_{1,2}^{44} \\ a_{1,1}^{45}, a_{1,2}^{45} \\ a_{1,1}^{53}, a_{1,2}^{53} \\ a_{1,1}^{54}, a_{1,2}^{54} \\ a_{1,1}^{55}, a_{1,2}^{55} \end{Bmatrix}$	$ S_1 = 9$ $ A_1 = 18$
R_2	$w_2 = \{4\}$ $\Gamma_2 = \{T_4\}$	$S_2 = \{s^{44}\}$ $s^{44} = \{T_4^u, \Delta t_{(T_4^v T_4^u)}\}$	$A_2 = \{a_{2,1}^{44}, a_{2,2}^{44}\}$	$ S_2 = 1$ $ A_2 = 2$

Çizelge 6.2. ÇRS-II’de Bulunan Robotlara İlişkin O_j^R Bilgileri (devam)

	w_j ve Γ_j	S_j	A_j	$ S_j $ ve $ A_j $
R_3	$w_3 = \{1,5\}$ $\Gamma_3 = \{T_1, T_5\}$	$S_3 = \{s^{11}, s^{15}, s^{51}, s^{55}\}$ $s^{11} = \{T_1^u, \Delta t_{(T_1^v T_1^u)}\}$ $s^{15} = \{T_1^u, \Delta t_{(T_5^v T_1^u)}\}$ $s^{51} = \{T_5^u, \Delta t_{(T_1^v T_5^u)}\}$ $s^{55} = \{T_5^u, \Delta t_{(T_5^v T_5^u)}\}$	$A_3 = \begin{Bmatrix} a_{3,1}^{11}, a_{3,2}^{11}, \\ a_{3,1}^{15}, a_{3,2}^{15}, \\ a_{3,1}^{51}, a_{3,2}^{51}, \\ a_{3,1}^{55}, a_{3,2}^{55} \end{Bmatrix}$	$ S_3 = 4$ $ A_3 = 8$
R_4	$w_4 = \{1,2\}$ $\Gamma_4 = \{T_1, T_2\}$	$S_4 = \{s^{11}, s^{12}, s^{21}, s^{22}\}$ $s^{11} = \{T_1^u, \Delta t_{(T_1^v T_1^u)}\}$ $s^{12} = \{T_1^u, \Delta t_{(T_2^v T_1^u)}\}$ $s^{21} = \{T_2^u, \Delta t_{(T_1^v T_2^u)}\}$ $s^{22} = \{T_2^u, \Delta t_{(T_2^v T_2^u)}\}$	$A_4 = \begin{Bmatrix} a_{4,1}^{11}, a_{4,2}^{11}, \\ a_{4,1}^{12}, a_{4,2}^{12}, \\ a_{4,1}^{21}, a_{4,2}^{21}, \\ a_{4,1}^{22}, a_{4,2}^{22} \end{Bmatrix}$	$ S_4 = 4$ $ A_4 = 8$
R_5	$w_5 = \{3,4\}$ $\Gamma_5 = \{T_3, T_4\}$	$S_5 = \{s^{33}, s^{34}, s^{43}, s^{44}\}$ $s^{33} = \{T_3^u, \Delta t_{(T_3^v T_3^u)}\}$ $s^{34} = \{T_3^u, \Delta t_{(T_4^v T_3^u)}\}$ $s^{43} = \{T_4^u, \Delta t_{(T_3^v T_4^u)}\}$ $s^{44} = \{T_4^u, \Delta t_{(T_4^v T_4^u)}\}$	$A_5 = \begin{Bmatrix} a_{5,1}^{33}, a_{5,2}^{33}, \\ a_{5,1}^{34}, a_{5,2}^{34}, \\ a_{5,1}^{43}, a_{5,2}^{43}, \\ a_{5,1}^{44}, a_{5,2}^{44} \end{Bmatrix}$	$ S_5 = 4$ $ A_5 = 8$
R_6	$w_6 = \{1,4,5\}$ $\Gamma_6 = \{T_1, T_4, T_5\}$	$S_6 = \begin{Bmatrix} s^{11}, s^{14}, s^{15}, \\ s^{41}, s^{44}, s^{45}, \\ s^{51}, s^{54}, s^{55} \end{Bmatrix}$ $s^{11} = \{T_1^u, \Delta t_{(T_1^v T_1^u)}\}$ $s^{14} = \{T_1^u, \Delta t_{(T_4^v T_1^u)}\}$ $s^{15} = \{T_1^u, \Delta t_{(T_5^v T_1^u)}\}$ $s^{41} = \{T_4^u, \Delta t_{(T_1^v T_4^u)}\}$ $s^{44} = \{T_4^u, \Delta t_{(T_4^v T_4^u)}\}$ $s^{45} = \{T_4^u, \Delta t_{(T_5^v T_4^u)}\}$ $s^{51} = \{T_5^u, \Delta t_{(T_1^v T_5^u)}\}$ $s^{54} = \{T_5^u, \Delta t_{(T_4^v T_5^u)}\}$ $s^{55} = \{T_5^u, \Delta t_{(T_5^v T_5^u)}\}$	$A_6 = \begin{Bmatrix} a_{6,1}^{11}, a_{6,2}^{11}, \\ a_{6,1}^{14}, a_{6,2}^{14}, \\ a_{6,1}^{15}, a_{6,2}^{15}, \\ a_{6,1}^{41}, a_{6,2}^{41}, \\ a_{6,1}^{44}, a_{6,2}^{44}, \\ a_{6,1}^{45}, a_{6,2}^{45}, \\ a_{6,1}^{51}, a_{6,2}^{51}, \\ a_{6,1}^{54}, a_{6,2}^{54}, \\ a_{6,1}^{55}, a_{6,2}^{55} \end{Bmatrix}$	$ S_6 = 9$ $ A_6 = 18$

6.2 Q-Öğrenme Yönteminin Çok Robotlu Sistemlerde Kullanım Yaklaşımları

Gerçekleştirilen çalışmanın temel amacı, grup içi görev paylaşımının market tabanlı olarak gerçekleştirildiği çok robotlu bir sistemde öğrenme yöntemleri kullanılarak sistem performansının artırılmasıdır. Daha özel olarak, sistemin toplam kazancını en büyükleyecek şekilde getirisi fazla olan yüksek öncelikli görevlerin yapılma oranlarının artırılması amaçlanmaktadır. İlaveten, düşük öncelikli görevlerin yapılma oranlarında meydana gelecek azalmaların da en aza indirilmesi istenmektedir. Öğrenme yaklaşımının kullanılmadığı durumda robotlar anons edilen görevler için yapabilme yeteneğine sahip oldukları takdirde müzayedeye katılmaktadırlar. Öğrenme yaklaşımı sisteme dâhil edildiğinde robotlar geçmiş anonslara ilişkin tecrübelerini kullanarak gelecek anonslara dair beklentilerini de hesaba katmakta ve buna göre müzayedeye katılma veya katılmama kararı vermektedirler.

Çok robotlu sistemlerde genellikle, robotların bağımsız karar verme ve hareket etme yeteneklerine sahip olmaları ve hareketlerinin sonucunda ortam durumlarında meydana gelen değişimlerin diğer robotlar tarafından öngörülemedebilecek olması ortamların durağan olarak nitelendirilememesiyle neticelenmektedir. Bu durum, pekiştirmeli öğrenme yöntemlerinin temel varsayımı olan ortamın MKS olarak tanımlanması şartı ile çelişmektedir. Bu sebeple, geleneksel Q-öğrenme algoritması tek erkinli sistemlerde başarı ile kullanılabilirken, çok erkinli sistemlerdeki uygulamalarda optimal çözümlere ulaşmak her zaman mümkün olamamaktadır.

Gerçekleştirilen uygulamalarda öncelikle, kullanılacak Q-öğrenme yönteminin seçimiyle ilgili çalışmalar yapılmış ve bu amaçla Bölüm 4.3'te anlatılan üç farklı yaklaşım irdelenmiştir. Bu yaklaşımlar, durum uzaylarının SAADU yöntemiyle belirlendiği ortamlar üzerinde gerçekleştirilmiş ve elde edilen sonuçlar görevlerin yapılma oranları ve işlemsel yük bazında incelenmiştir.

Önerilen yaklaşımları incelemek üzere ÇRS-I'de her biri 50 adet görev içeren 45 deney grubu ve ÇRS-II'de yine her biri 50 adet görev içeren 75 deney grubu kullanılmıştır. Her bir deney grubu tanımlanan beş tip görevi içermektedir ve tüm

görev tiplerinin gelme olasılığı yaklaşık olarak eşittir. Görevlerin sisteme gelme süreci rassal olarak belirlenmektedir. Her iki ortam için de deney gruplarının ilk 2/3'lük kısmı eğitim grubu ve son 2/3'lük kısmı test grubu olarak belirlenmiştir. ÇRS-II, daha karmaşık bir sistem olduğu ve öğrenme uzayının boyutu ÇRS-I'e göre çok daha yüksek olduğu için başarılı bir öğrenme süreci daha fazla veriye ihtiyaç duymaktadır. Bu sebeple ÇRS-II için daha fazla sayıda deney grubu kullanılmıştır. Deney süreci, sırasıyla aşağıda verilen üç adımdan oluşmaktadır:

- 1- Eğitim grubu, hiçbir öğrenme yaklaşımı kullanılmadan çalıştırılmış ve görev atama sürecine ilişkin bilgiler elde edilmiştir.
- 2- Robotlar birinci adımda elde ettikleri görev atama verilerini kullanarak öğrenme sürecini gerçekleştirmiştir.
- 3- Robotların öğrenme süreci sonunda elde ettikleri tecrübeler kullanılarak görev atama süreci tekrarlanmaktadır. Ancak bu defa test grupları kullanılmaktadır. Tüm test grupları hem öğrenme olmadığı hem de öğrenme yaklaşımlarının kullanıldığı durumlar için çalıştırılmış ve karşılaştırmalı sonuçlar elde edilmiştir.

Her bir görev tipi için düşük öncelikli olma ve yüksek öncelikli olma şeklinde iki farklı öncelik durumu tanımlıdır. Bir robotun herhangi bir görevi yapma yeteneğine sahip olması o görevin hem düşük hem de yüksek öncelikli görevlerini yapabildiği anlamına gelmektedir. Örneğin T_1 görev tipi için Bölüm 4.2'de tanımlandığı üzere düşük öncelikli T_1^u ve yüksek öncelikli T_1^v görevleri mevcuttur. Bu tanımlama tüm görev tipleri için geçerlidir.

Görev atama sürecinde bir robota atanan görevlerin tamamlandığı varsayılmaktadır. Anons edildikten sonra hiçbir robota atanmayan görevler yapılmamış olarak kabul edilmekte ve görev listesinden silinmektedir.

Çok robotlu sistemlerde görev atama algoritmalarının hedefi robotlar tarafından yapılan görev sayısını artırmaktır. Sistemlerin heterojen yapısı nedeniyle tüm görev tiplerinin tüm robotlar tarafından yapılamayacak olması, robotların öngörülemezlik

çeşitli sebeplerle zaman zaman çalışma dışı kalmaları, bir görevi yapabilecek robotların o görev anons edildiğinde meşgul olmaları gibi nedenlerle tüm görevlerin yapılması mümkün olamamaktadır. Bu çalışmada, temel merkezi ve dağıtık öğrenme yaklaşımları incelenmiş ve strateji planlamalı dağıtık öğrenme yaklaşımı önerilmiştir. Bu üç öğrenme yaklaşımının sistem performansı üzerindeki etkileri sırasıyla ÇRS-I ve ÇRS-II ortamları için görevlerin yapılma oranları ve işlemsel yük kriterleri açısından irdelenmektedir.

Görevlerin yapılma oranı kriteri robotlara atanan görev sayısının sisteme anons edilen görev sayısına oranı olarak tanımlanmıştır ve yüzde olarak verilmektedir. Örnek olarak ÇRS-I ortamındaki T_1 görevi ele alındığında toplam 292 adet T_1 görevinin sisteme anons edildiği, bunların 193 adedinin düşük öncelikli ve 99 adedinin ise yüksek öncelikli T_1 görevleri olduğu gözlenmiştir. Merkezi öğrenme yaklaşımının kullanıldığı durumda, 104 adet düşük öncelikli görev ve 91 adet yüksek öncelikli görev olmak üzere toplam 195 adet T_1 görevi tamamlanmıştır. Bu sonuçlara göre, T_1 görevi için düşük öncelikli görevlerin %54'ü ve yüksek öncelikli görevlerin %92'si tamamlanmış olmaktadır. Tamamlanan toplam T_1 görevi oranı ise %67'dir. Takip eden bölümlerde sonuçlar bu yapı üzerinden incelenmektedir.

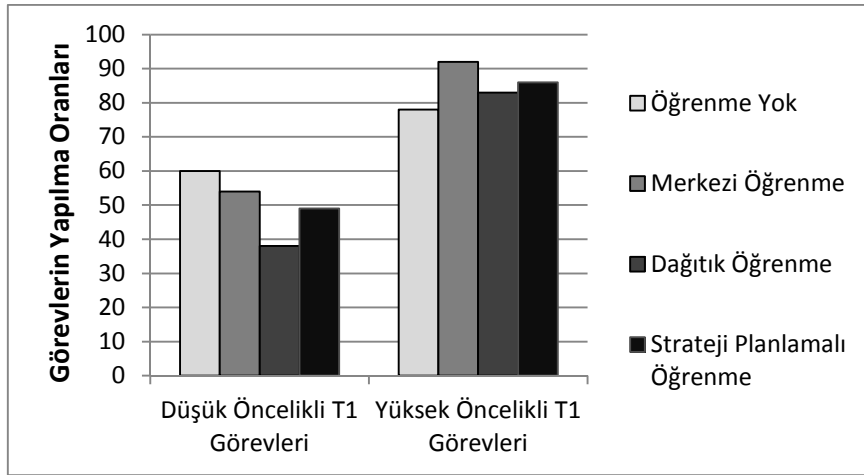
İşlemsel yük kriteri ise öğrenme uzayının boyutu ile ilişkili olarak tanımlanmıştır (Boşuniu et al., 2008) ve öğrenme yaklaşımının işlediği veri sayısı hakkında bilgi vermektedir. Ayrıntıları Bölüm 6.2.1.2'de yer almaktadır.

6.2.1 ÇRS-I için öğrenme yaklaşımlarına ilişkin sonuçlar

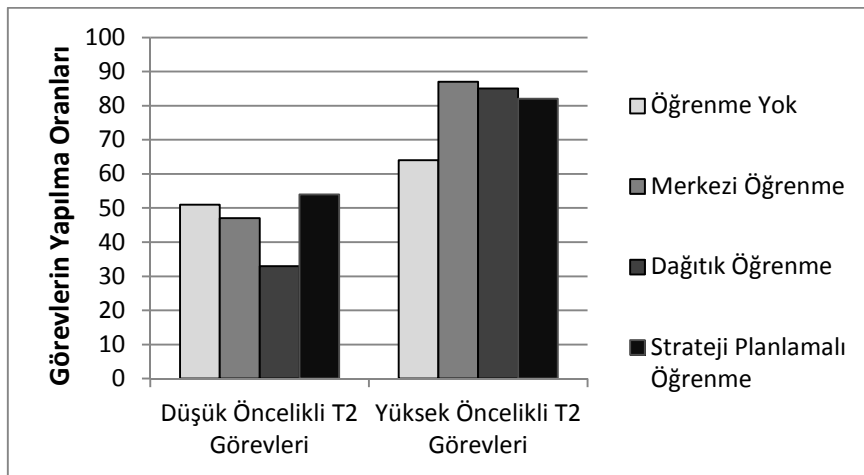
Bu bölümde, ÇRS-I ortamı için merkezi, dağıtık ve strateji planlamalı dağıtık öğrenme yaklaşımlarının kullanımına ilişkin sonuçlar yer almaktadır. Bu yaklaşımlar görevlerin yapılma oranları ve işlemsel yük kriterleri temel alınarak karşılaştırılmakta ve avantaj ve dezavantajları ortaya konulmaktadır.

6.2.1.1 Görevlerin yapılma oranları

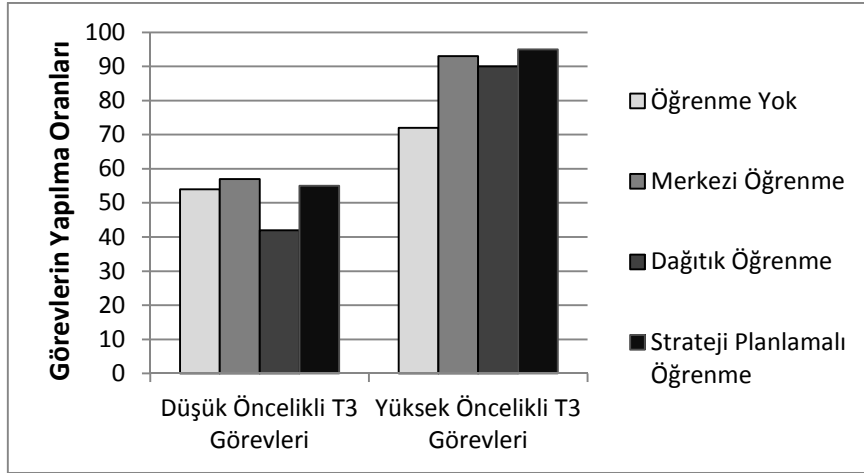
ÇRS-I ortamı için öğrenme yaklaşımlarının kullanılmadığı ve ele alınan üç farklı Q-öğrenme yaklaşımının kullanıldığı durumlarda görevlerin yapılma oranları karşılaştırmalı olarak Şekil 6.1’de grafikler halinde verilmektedir. Amaç, sistem performansını en büyükmek adına özellikle yüksek öncelikli görevlerin yapılma oranlarını artırmak olduğu için sonuçlar her bir görev tipine ait düşük ve yüksek öncelikli görevler için ayrı ayrı verilmektedir.



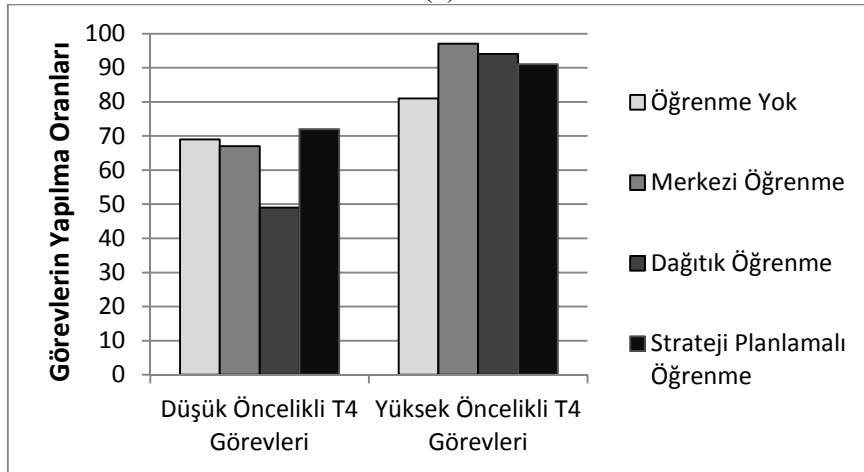
(a)



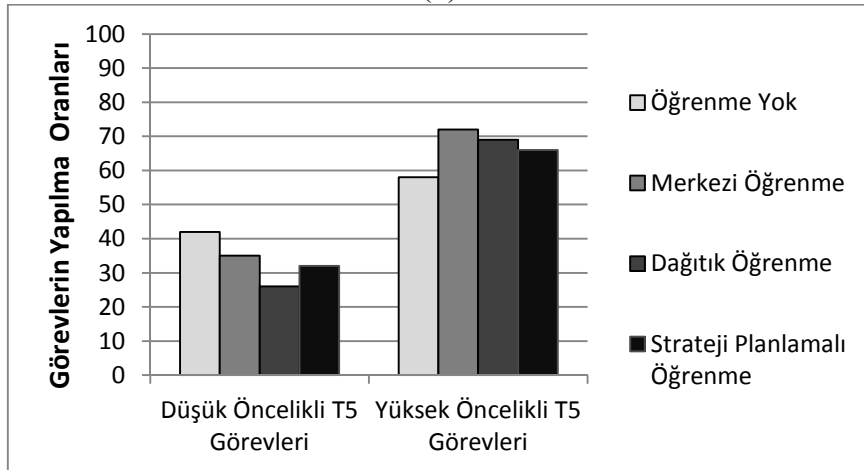
(b)



(c)



(d)



(e)

Şekil 6.1 ÇRS-I’de Tüm Öğrenme Yaklaşımları için Düşük ve Yüksek Öncelikli Görevlerin Yapılma Oranları (%) (a) T_1 (b) T_2 (c) T_3 (d) T_4 (e) T_5 Görevleri

Şekil 6.1’de yer alan grafikler incelendiğinde herhangi bir öğrenme yaklaşımının kullanıldığı durumda tüm görev tipleri için yüksek öncelikli görevlerin yapılma oranlarının arttığı görülmektedir. Bu sonuç, robotların öğrenme yetenekleri sayesinde elde ettikleri görev atama tecrübelerinden başarıyla yararlandıklarını ve hedeflendiği üzere sistem performansında beklenen iyileştirmelerin sağlandığını göstermektedir. Bununla birlikte hemen hemen tüm görev tipleri için düşük öncelikli görevlerin yapılma oranlarında azalma meydana gelmiştir. Sistem kaynaklarının yetersizliği nedeniyle robotların belirli bir görev yapma kapasiteleri bulunmaktadır. Bu sebeple, yüksek öncelikli görevlerin yapılma oranları artarken düşük öncelikli görevlerin yapılma oranlarının azalması sürecin doğası gereğidir.

Tüm öğrenme yaklaşımları için görevlerin yapılma oranlarına ilişkin sayısal değerler Çizelge 6.3 ile verilmektedir. Her bir görev tipi için düşük öncelikli görevlerin, yüksek öncelikli görevlerin ve öncelik durumu dikkate alınmaksızın tüm görevlerin yapılma oranları yüzde olarak çizelgede yer almaktadır. Toplam ifadesi o görev tipi için düşük ve yüksek öncelikli görevlerin toplamına ilişkin verileri göstermektedir.

Yüksek öncelikli görevlerin yapılma oranındaki en fazla artış, işbirliği içerisinde bulunan tüm robotların hareketlerinin birlikte değerlendirme esasına dayandığı için, merkezi öğrenme yaklaşımı ile elde edilmiştir. Yine aynı sebep dolayısıyla, düşük öncelikli görevlerin en fazla yapıldığı öğrenme yaklaşımı merkezi öğrenme yaklaşımıdır. Özellikle birden fazla robot tarafından yapılabilen görevler için bir robot yüksek öncelikli görevi beklemeyi tercih ederken diğer robotlar düşük öncelikli görevleri yapma imkanı bulmaktadır.

Dağıtık öğrenme yaklaşımında robotlar öğrenme sürecini diğer robotların hareketlerinden bağımsız olarak gerçekleştirmektedir. Tüm robotlar aynı hedefe yöneldikleri yani yüksek öncelikli görevleri daha çok yapma eğiliminde oldukları için yüksek öncelikli görevlerin yapılma oranları artarken düşük öncelikli görevlerin yapılma oranlarında azalma olmaktadır. Dağıtık öğrenme yaklaşımının en büyük dezavantajı olan robotlar arası davranış çakışması problemi bu sonucu açıklamaktadır.

Çizelge 6.3 ÇRS-I’de Tüm Öğrenme Yaklaşımları için Görevlerin Yapılma Oranları

		Q-Öğrenme Yöntemi Yaklaşımları			
Görevler		Öğrenme Yok	Merkezi Öğrenme	Dağıtık Öğrenme	Strateji Planlamalı Dağıtık Öğrenme
T_1	Düşük Öncelikli	60	54	38	49
	Yüksek Öncelikli	78	92	83	86
	Toplam	66	67	54	62
T_2	Düşük Öncelikli	51	47	33	54
	Yüksek Öncelikli	64	87	85	82
	Toplam	56	61	51	64
T_3	Düşük Öncelikli	54	57	42	55
	Yüksek Öncelikli	72	93	90	95
	Toplam	60	70	59	69
T_4	Düşük Öncelikli	69	67	49	72
	Yüksek Öncelikli	81	97	94	91
	Toplam	73	77	65	79
T_5	Düşük Öncelikli	42	35	26	32
	Yüksek Öncelikli	58	72	69	66
	Toplam	48	48	41	44

Strateji planlamalı dağıtık öğrenme yaklaşımı ile elde edilen görevlerin yapılma oranları ise hemen hemen tüm görev tipleri için merkezi öğrenme yaklaşımı ile elde edilen oranlar kadar yüksek olmamakla beraber dağıtık öğrenme yaklaşımına göre oldukça iyi durumdadır. Yüksek öncelikli görevlerin yapılma oranı diğer öğrenme yaklaşımlarına ve özellikle merkezi öğrenme yaklaşımına göre bir miktar daha azdır ama düşük öncelikli görevlerin yapılma oranları özellikle dağıtık öğrenme yaklaşımına göre çok daha yüksek bir değere sahiptir.

Bağımlı örneklem t testi (paired-samples t-test) özellikle deneysel çalışmalarda aynı denek grubunun farklı test verileri için ortaya koydukları tepkilerin irdelenmesi amacıyla kullanılan istatistiksel bir yöntemdir. Bir denek grubu için elde edilen farklı örneklem arasındaki farkın belirli bir test istatistiğine göre anlamlı olup olmadığını ortaya koymaktadır (Zikmund and Babin, 2010). Eleman sayısı η ve ortalamaları sırasıyla olan μ_1 ve μ_2 olan iki örneklem seti arasındaki farkın ortalama değeri μ_D ve standart sapması σ_D olmak üzere test istatistiği Denklem (6.7) ile hesaplanmaktadır.

$$t_0 = \frac{\mu_D}{\sigma_D / \sqrt{\eta}} \quad (6.7)$$

Örneklem setlerinin değerlendirilmesi için önerilen iki temel hipotez sıfır hipotezi H_0 ve alternatif hipotez H_A sırasıyla aşağıdaki şekilde tanımlanmaktadır:

$$\begin{aligned} H_0: \mu_1 - \mu_2 &= 0 \\ H_A: \mu_1 - \mu_2 &\neq 0 \end{aligned} \quad (6.8)$$

Belirlenmiş bir anlamlılık değeri (α) için kritik t değerleri tablosuna (Zikmund and Babin, 2010) göre bulunan kritik t değeri (t_α) kullanılarak hipotezler hakkında karar verilmektedir. $|t_0| < t_\alpha$ H_0 hipotezi, $|t_0| > t_\alpha$ ise H_A hipotezi kabul edilmektedir. Anlamlılık değeri örneklem değerlerinin elde edilmesi esnasında meydana gelebilecek hatalara karşı yanılma payı olarak değerlendirilmektedir ve uygulamalarda sıklıkla 0,05 veya 0,01 değerleri kullanılmaktadır.

ÇRS-I ortamı için elde edilen sonuçlar incelendiğinde merkezi öğrenme yaklaşımının tüm diğer öğrenme yaklaşımları karşısındaki başarısı açıkça gözlemlenebilmektedir. Yine benzer şekilde, merkezi öğrenme ve strateji planlamalı dağıtık öğrenme yaklaşımları dağıtık öğrenme yaklaşımına göre oldukça başarılı sonuçlar ortaya koymaktadır. Strateji planlamalı dağıtık öğrenme yaklaşımının etkinliğini incelemek üzere merkezi öğrenme yaklaşımı ve strateji planlamalı öğrenme yaklaşımı ile elde edilen sonuçlar üzerinde bağımlı örneklem t testi uygulanmıştır. İlk örneklem seti merkezi öğrenme yaklaşımı ile elde edilen görevlerin yapılma oranları

verileri ve ikinci örneklem seti strateji planlamalı öğrenme yaklaşımı ile elde edilen görevlerin yapılma oranları verileri olmak üzere sıfır hipotezi ve alternatif hipotez sırasıyla aşağıdaki şekilde belirlenmiştir:

H_0 : İki öğrenme yaklaşımı ile elde edilen görev yapılma oranları arasında fark yoktur

H_A : İki öğrenme yaklaşımı ile elde edilen görev yapılma oranları arasında fark vardır

İki örneklem seti verileri arasındaki farkların ortalama değeri $\mu_D=0,7367$ ve standart sapması $\sigma_D=2,8153$ olarak bulunmuştur. Bu fark dağılımının test istatistiği Denklem (6.7) kullanılarak $t_0=1,0134$ olarak hesaplanmıştır. Kritik t değerleri tablosuna göre anlamlılık değeri $\alpha=0,05$ ve örneklem seti eleman sayısı $n=15$ için kritik t değeri $t_{\alpha}=2,145$ olduğu için sıfır hipotezi H_0 kabul edilmektedir. Bu sonuç, strateji planlamalı dağıtık öğrenme yaklaşımının da merkezi öğrenme kadar başarılı olduğunu ortaya koymaktadır.

6.2.1.2 İşlemsel yük

Literatürde yer alan çalışmalarda, Q-öğrenme yöntemi için işlemsel yükün öğrenme uzayının boyutu ile ilişkili olduğu belirtilmektedir (Hu and Wellman, 2003; Boşuniu et al., 2008). Bölüm 4.3'te öğrenme uzayının boyutunun durum uzayının boyutu ile hareket uzayının boyutunun çarpımı ile elde edildiği verilmişti. Dağıtık öğrenme yaklaşımında öğrenme, her bir robot tarafından diğer robotlardan bağımsız olarak sadece kendi durum ve hareket bilgileri dahilinde gerçekleştirildiği için öğrenme uzayı her robot için kendi durum ve hareket uzaylarının boyutları ile alakalıdır. j 'nci robot için öğrenme uzayı boyutu:

$$|\text{Öğrenme Uzayı}| = |S_j| \times |A_j| \quad (6.9)$$

Merkezi öğrenme yaklaşımında ise öğrenme sürecinde birleşik durum uzayı ve tümleşik hareket uzayı kullanıldığı için tüm ortamın öğrenme uzayının boyutu buna göre hesaplanmalıdır:

$$|\text{Öğrenme Uzayı}| = |S|x|A| \quad (6.10)$$

Burada S birleşik durum uzayını ve A ise tümleşik durum uzayını ifade etmektedir ve sırasıyla Denklem (4.10) ve Denklem (4.11) ile hesaplanmaktadır. Durum ve hareket uzaylarının boyutlarına ilişkin bilgiler ise Denklem (4.12) ve Denklem (4.13) ile verilmektedir.

ÇRS-I ortamına ilişkin merkezi, dağıtık ve strateji planlamalı dağıtık öğrenme yaklaşımlarının durum uzayı boyutlarına ilişkin sayısal değerler Çizelge 6.4’de yer almaktadır.

Çizelge 6.4 ÇRS-I için Öğrenme Uzayı Boyutları

	$ S_j $ ve $ A_j $	Dağıtık Öğrenme	Merkezi Öğrenme	Strateji Planlamalı Dağıtık Öğrenme
R_1	$ S_1 = 1$ $ A_1 = 2$	$ S_1 x A_1 = 2$	$ S^1 = 4$ $ A^1 = 8$ $ S^1 x A^1 = 64$	$ S_1 x A_1 = 2$
R_2	$ S_2 = 4$ $ A_2 = 8$	$ S_2 x A_2 = 32$		$ S_2 x A_2 = 32$
R_3	$ S_3 = 1$ $ A_3 = 2$	$ S_3 x A_3 = 2$	$ S^2 = 7$ $ A^2 = 256$ $ S^2 x A^2 = 1792$	$ S_3 x A_3 = 2$
R_4	$ S_4 = 4$ $ A_4 = 8$	$ S_4 x A_4 = 32$		$ S_4 x A_4 = 32$
R_5	$ S_5 = 4$ $ A_5 = 8$	$ S_5 x A_5 = 32$		$ S_5 x A_5 = 32$
R_6	$ S_6 = 1$ $ A_6 = 2$	$ S_6 x A_6 = 2$		$ S_6 x A_6 = 2$
TOPLAM		102	1856	102

ÇRS-I ortamı için strateji planlamalı dağıtık öğrenme yaklaşımı dağıtık bir yapıya sahip olduğu için öğrenme uzayının boyutu dağıtık öğrenme yaklaşımı ile aynıdır. Merkezi öğrenme yaklaşımında sistem birbirinden bağımsız iki alt sistem gibi ele alınabildiği için merkezi öğrenme iki ayrı alt birim tarafından gerçekleştirilmektedir. Bu sebeple, toplam öğrenme uzayı boyutu Denklem (6.5) ve Denklem (6.6) ile tanımlanan bu alt ortamların öğrenme uzayı boyutlarının toplamı olarak alınmaktadır. Çizelgenin en alt satırı toplam öğrenme uzayı boyutunu vermektedir. ÇRS-I basit yapıları bir sistem olduğu halde merkezi öğrenme yaklaşımının öğrenme uzayı boyutunun dağıtık öğrenme yaklaşımının öğrenme uzayı boyutu ile karşılaştırıldığında oldukça yüksek olduğu görülmektedir. Bu fark çeşitli dezavantajlarına rağmen neden dağıtık öğrenme yaklaşımlarının kullanımının tercih edildiğini ortaya koymaktadır.

6.2.2 ÇRS-II için öğrenme yaklaşımlarına ilişkin sonuçlar

Bu bölümde, ÇRS-II ortamı için merkezi, dağıtık ve strateji planlamalı dağıtık öğrenme yaklaşımlarının kullanımına ilişkin uygulama sonuçları görevlerin yapılma oranları ve işlemsel yük kriterleri temelinde incelenmektedir.

6.2.2.1 Görevlerin yapılma oranları

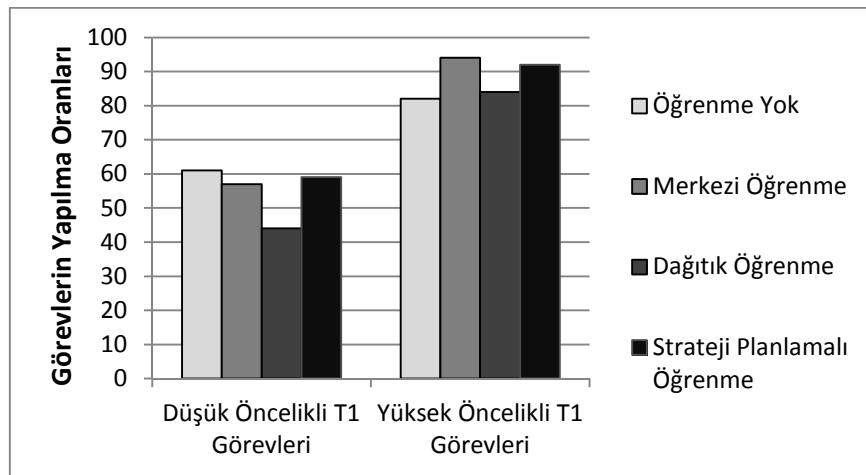
Öğrenme yaklaşımlarının kullanılmadığı durum ve üç farklı Q-öğrenme yaklaşımının kullanıldığı durumlar için ÇRS-II’de görevlerin yapılma oranları karşılaştırmalı olarak Şekil 6.2’de grafikler halinde verilmektedir. Sonuçlar her görev tipinin düşük ve yüksek öncelikli görevleri için ayrı ayrı verilmektedir.

Her iki ortam da aynı sayıda robot ve aynı sayıda görev tipi içerdiği halde robotların yeteneklerindeki farklılıklardan dolayı ÇRS-II ortamı çok daha karmaşık bir yapı sunmaktadır. Bununla birlikte, görevlerin yapılma oranları kriteri bağlamında benzer bir karakteristik ortaya çıkmıştır. Genel anlamda Şekil 6.2’de yer alan grafikler incelendiğinde herhangi bir öğrenme yaklaşımının kullanılması öğrenme yaklaşımının kullanılmadığı duruma göre tüm görev tipleri için yüksek öncelikli görevlerin yapılma

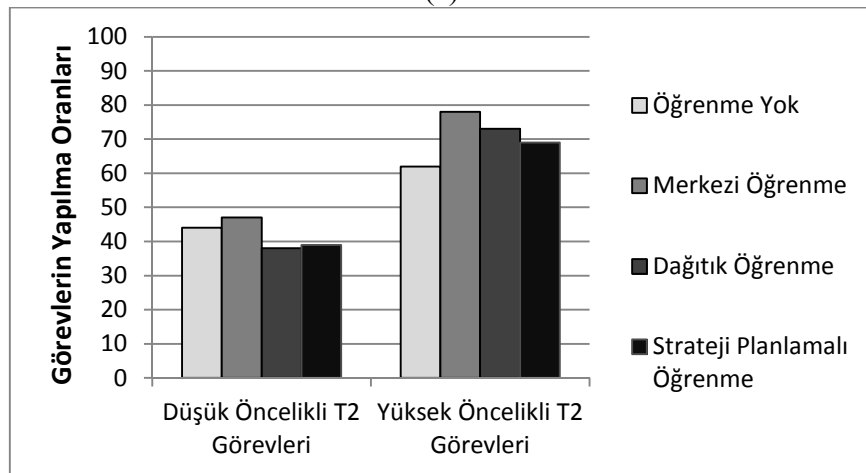
oranlarında artış sağladığı görülmektedir. Ancak, düşük öncelikli görevlerin yapılma oranları kısmen aynı değerlerde kalmakta ama genelde azalma göstermektedir.

Merkezi öğrenme yaklaşımı hemen hemen tüm görev tipleri için en fazla görevin yapılmasını sağlamaktadır. Merkezi bir koordinasyon sağladığı için hem yüksek öncelikli hem de düşük öncelikli görevlerin yapılma oranlarını artırmaktadır.

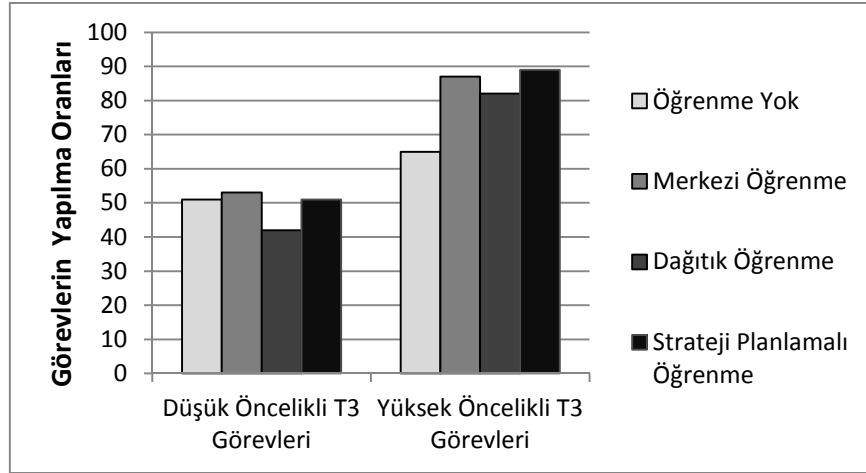
Dağıtık öğrenme yaklaşımında ise düşük ve yüksek öncelikli görevler için yapılma oranları en düşük seviyede kalmaktadır. Bu durumun robotların bağımsız öğrenme süreçlerinden kaynaklanan davranış çakışmaları nedeniyle ortaya çıktığı açıktır.



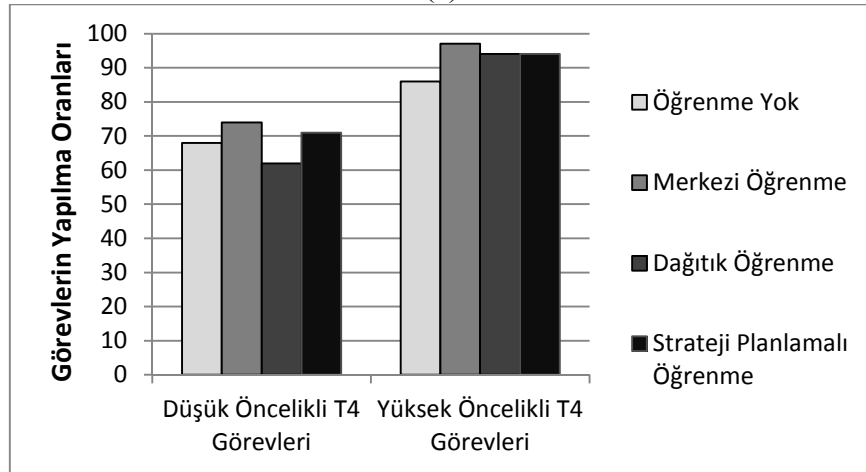
(a)



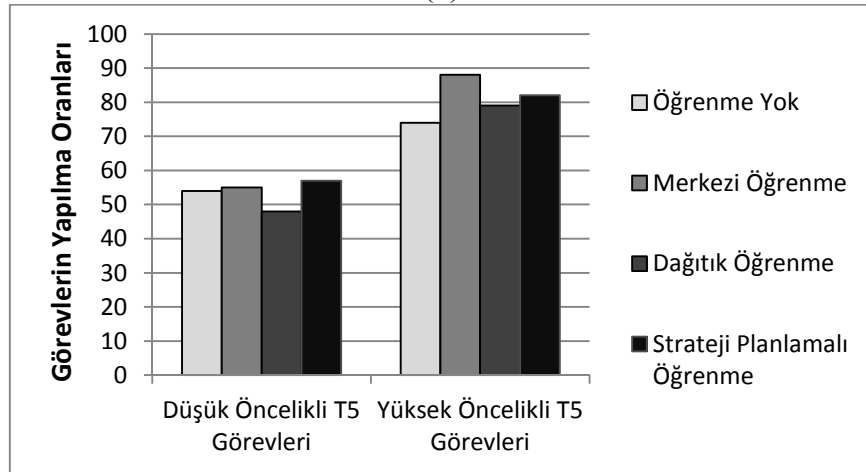
(b)



(c)



(d)



(e)

Şekil 6.2 ÇRS-II’de Tüm Öğrenme Yaklaşımları için Düşük ve Yüksek Öncelikli Görevlerin Yapılma Oranları (%) (a) T_1 (b) T_2 (c) T_3 (d) T_4 (e) T_5 Görevleri

Strateji planlamalı dağıtık öğrenme yaklaşımı yüksek öncelikli görevlerin yapılma oranları anlamında merkezi öğrenme yaklaşımına yakın fakat bir miktar daha düşük, düşük öncelikli görevlerin yapılma oranları anlamında ise hemen hemen yakın değerlerde sonuçlar ortaya koymaktadır. Elde edilen oranlar dağıtık öğrenme yaklaşımı ile elde edilen değerlere göre oldukça yüksektir

Tüm öğrenme yaklaşımları için görevlerin yapılma oranlarına ilişkin sayısal değerler Çizelge 6.5’te verilmektedir.

Çizelge 6.5 ÇRS-II’de Tüm Öğrenme Yaklaşımları için Görevlerin Yapılma Oranları

Görevler		Q-Öğrenme Yöntemi Yaklaşımları			
		Öğrenme Yok	Merkezi Öğrenme	Dağıtık Öğrenme	Strateji Planlamalı Dağıtık Öğrenme
T_1	Düşük Öncelikli	61	57	44	59
	Yüksek Öncelikli	82	94	84	92
	Toplam	68	70	58	71
T_2	Düşük Öncelikli	44	47	38	39
	Yüksek Öncelikli	62	78	73	69
	Toplam	50	58	50	50
T_3	Düşük Öncelikli	51	53	42	51
	Yüksek Öncelikli	65	87	82	89
	Toplam	56	65	56	64
T_4	Düşük Öncelikli	68	74	62	71
	Yüksek Öncelikli	86	97	94	94
	Toplam	74	82	73	79
T_5	Düşük Öncelikli	54	55	48	57
	Yüksek Öncelikli	74	88	79	82
	Toplam	61	67	59	66

Çizelgelerde, her bir görev tipi için düşük öncelikli görevlerin, yüksek öncelikli görevlerin ve öncelik durumu dikkate alınmaksızın tüm görevlerin yapılma oranlarına ilişkin sayısal değerler çizelgede yer almaktadır. Bu değerler incelendiğinde yine ÇRS-I ortamına benzer şekilde merkezi öğrenme yaklaşımının diğer tüm öğrenme yaklaşımlarına ve strateji planlamalı dağıtık öğrenme yaklaşımına karşı başarısı ortaya çıkmaktadır. Merkezi öğrenme ve strateji planlamalı dağıtık öğrenme yaklaşımları ile elde edilen görevlerin yapılma oranlarına ilişkin değerler arasında ÇRS-I ortamındakine benzer şekilde bağımlı örneklem t testi uygulandığında strateji planlamalı dağıtık öğrenme yaklaşımının da merkezi öğrenme yaklaşımı kadar başarılı olduğu sonucuna ulaşılmaktadır.

6.2.2.2 İşlemsel yük

ÇRS-II ortamı için dağıtık, merkezi ve strateji planlamalı dağıtık öğrenme yaklaşımları için öğrenme uzayı boyutları Çizelge 6.6'da yer almaktadır. ÇRS-II ortamı ÇRS-I ortamına göre oldukça karmaşık bir yapıya sahiptir. Çizelge 6.6'da verilen değerler Çizelge 6.4'te bulunan değerlerle karşılaştırıldığında bu durum görülebilmektedir.

Dağıtık ve strateji planlamalı dağıtık öğrenme yaklaşımlarında öğrenme uzayı makul seviyelerde iken merkezi öğrenme yaklaşımının öğrenme uzayı oldukça büyük bir değere ulaşmaktadır. Sistemde sadece altı robot bulunduğu ve yapılması gereken görev sayısı beş olduğu halde merkezi öğrenme yaklaşımının sisteme getirdiği işlemsel yükün ne kadar fazla olduğu düşünüldüğünde daha büyük sistemler için merkezi öğrenme yaklaşımının kullanım zorlukları ortaya çıkmaktadır. Merkezi öğrenme yaklaşımı sisteme büyük performans artışı sağlarken getirdiği işlemsel yük büyük bir dezavantaj teşkil etmektedir. Dağıtık öğrenme ve strateji planlamalı dağıtık öğrenme yaklaşımları bireysel robot davranışları değerlendirildiği için küçük öğrenme uzayı boyutuna sahiptir. Küçük öğrenme boyutunun düşük işlemsel yük anlamına geldiği ve sunduğu işlemsel kolaylık açıktır.

Çizelge 6.6 ÇRS-II için Öğrenme Uzayı Boyutları

	$ S_j $ ve $ A_j $	Dağıtık Öğrenme	Merkezi Öğrenme	Strateji Planlamalı Dağıtık Öğrenme
R_1	$ S_1 = 9$ $ A_1 = 18$	$ S_1 x A_1 = 162$	$ S = 17$ $ A = 3^4x2^{12}$ $ S x A = 17x3^4x2^{12}$	$ S_1 x A_1 = 162$
R_2	$ S_2 = 1$ $ A_2 = 2$	$ S_2 x A_2 = 2$		$ S_2 x A_2 = 2$
R_3	$ S_3 = 4$ $ A_3 = 8$	$ S_3 x A_3 = 32$		$ S_3 x A_3 = 32$
R_4	$ S_4 = 4$ $ A_4 = 8$	$ S_4 x A_4 = 32$		$ S_4 x A_4 = 32$
R_5	$ S_5 = 4$ $ A_5 = 8$	$ S_5 x A_5 = 32$		$ S_5 x A_5 = 32$
R_6	$ S_6 = 9$ $ A_6 = 18$	$ S_6 x A_6 = 162$		$ S_6 x A_6 = 162$
TOPLAM		426	$17x3^4x2^{12}$	426

6.2.3 Öğrenme yaklaşımlarının değerlendirilmesi

Bu çalışmada ele alınan üç öğrenme yaklaşımı ÇRS-I ve ÇRS-II ortamları üzerinde uygulanmış ve sonuçlar yukarıda verilmiştir. Tüm öğrenme yaklaşımları yüksek öncelikli görevlerin yapılma oranlarında artış sağlamış olduğu için öğrenme tabanlı görev atama yaklaşımlarının başarıya ulaştığı açıkça söylenebilmektedir. Ancak, öğrenme yaklaşımlarının avantaj ve dezavantajlarını değerlendirmek ve önerilen strateji planlamalı dağıtık öğrenme yaklaşımının etkinliğini görmek üzere tüm yöntemler sistem performansı, öğrenme uzayı boyutu, işlemsel yük, davranış çakışmalarına neden olma ve haberleşme ihtiyacı açılarından incelenmekte ve sonuçlar Çizelge 6.7 ile verilmektedir. Sistem performansı, öğrenme uzayı boyutu ve işlemsel yük kriterlerini değerlendirmek üzere Düşük-Orta-Yüksek-Çok yüksek şeklinde dört

kademeli bir ölçekleme kullanılmıştır. Davranış çakışmaları ve haberleşme ihtiyacı kriterleri ise Evet/Hayır şeklinde değerlendirilmiştir.

Çizelge 6.7 Öğrenme Yaklaşımlarının Değerlendirilmesi

Kriter	Q-Öğrenme Yöntemi Yaklaşımları		
	Merkezi Öğrenme	Dağıtık Öğrenme	Strateji Planlamalı Dağıtık Öğrenme
Sistem Performansı	Çok yüksek	Düşük	Yüksek
Öğrenme Uzaı Boyutu	Çok yüksek	Düşük	Düşük
İşlemsel Yük	Çok yüksek	Düşük	Düşük
Davranış Çakışmaları	Hayır	Evet	Hayır
Haberleşme İhtiyacı	Evet	Hayır	Hayır

Her bir görev tipi için yapılan toplam görev oranları dikkate alındığında merkezi öğrenme yaklaşımında hemen hemen tüm görevler için en çok yapılma oranının yakalandığı görülmektedir. Gerçekleştirilen bağımlı örneklem t testi ile strateji planlama dağıtık öğrenme yaklaşımının sunduğu sonuçların da merkezi öğrenme yaklaşımı ile elde edilen sonuçlar kadar başarılı olduğu görülmüştür. Bu sebeple ve Çizelge 6.7’de özetlenen yüksek sistem performansı, düşük işlemsel yük, davranış çakışmalarının ve haberleşme ihtiyacının olmaması gibi avantajları nedeniyle strateji planlamalı öğrenme yaklaşımının en etkin yöntem olduğu görülmektedir. Görev tipleri arası yapılma oranlarının farklı olması ise hem görevlerin yapılarının ve yapılma sürelerinin farklı olmasından hem de her görevin farklı sayıda robot tarafından yapılmasından kaynaklanmaktadır.

6.3 Durum Uzaı Belirleme Yaklaşımları

Durum uzaının sistemin ihtiyaçlarına uygun olarak belirlenmesi, Q-öğrenme yönteminin başarısını etkileyen önemli bir faktördür. Q-öğrenme yöntemi, durum ve hareket uzaılarının ayrık ve sonlu olduğu ortamlar üzerinde tanımlanmış olduğu için

genellikle sürekli durum ve/veya hareket uzaylarına sahip robot sistemleri uygulamalarında durum ve hareket uzaylarının temsili ile ilgili problemlerle karşılaşmaktadır. Bu çalışmada, durum uzayının temsili için üç yaklaşım, SAADU, DFSDU ve DAADU, önerilmiştir.

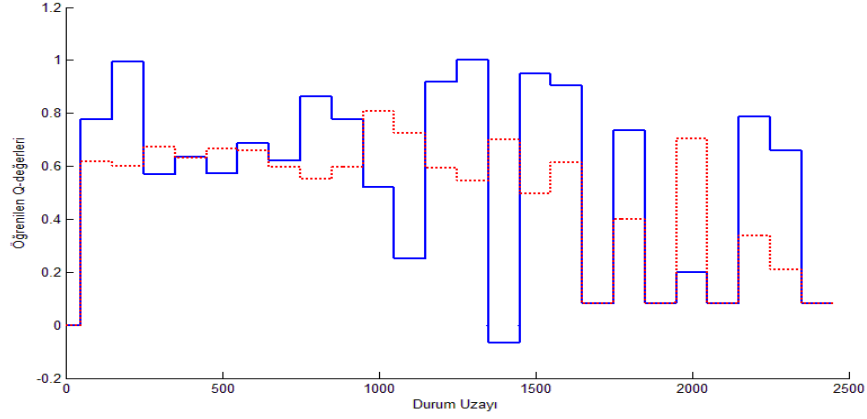
Önceki bölümde incelendiği üzere Q-öğrenme yönteminin çok robotlu sistemlere uygulanmasında Strateji Planlamalı Öğrenme Yaklaşımının en uygun sonuçları verdiği görülmüştür. Bu sebeple, durum uzayı belirleme yaklaşımlarının değerlendirilmesi esnasında bu yaklaşımın kullanımı tercih edilmiştir. Her üç durum uzayı belirleme yaklaşımı Bölüm 5'te anlatılan ÇRS-I ve ÇRS-II ortamları üzerinde uygulanmıştır. Aşağıda öncelikle farklı yaklaşımlarla elde edilen durum uzayı yapısına bir örnek verilecektir. Daha sonra ise bu yaklaşımların sistem performansı üzerindeki etkileri incelenecektir.

6.3.1 Durum uzayı belirleme yaklaşımlarının incelenmesi

Burada, öncelikle her yaklaşımın sonunda elde edilen Q-değerlerinin yapısı yer almaktadır. Gösterim ve karşılaştırma basitliği açısından ÇRS-I ortamında yer alan R_1 robotunun durum uzayı örneklenmektedir. Kullanılan sistemde R_1 robotu sadece T_1 görevini yapabilmektedir. Öğrendiği Q değerleri, robotun T_1 görevinin düşük öncelikli (T_1^u) ve yüksek öncelikli (T_1^v) görevlerini yapma tercihinine ilişkindir. R_1 robotunun biri düşük öncelikli görevi yapmak (a_1^{11}), diğeri de yüksek öncelikli görevi beklemek (a_2^{11}) olmak üzere iki terimli bir hareket uzayı ($A_1 = \{a_1^{11}, a_2^{11}\}$) bulunmaktadır.

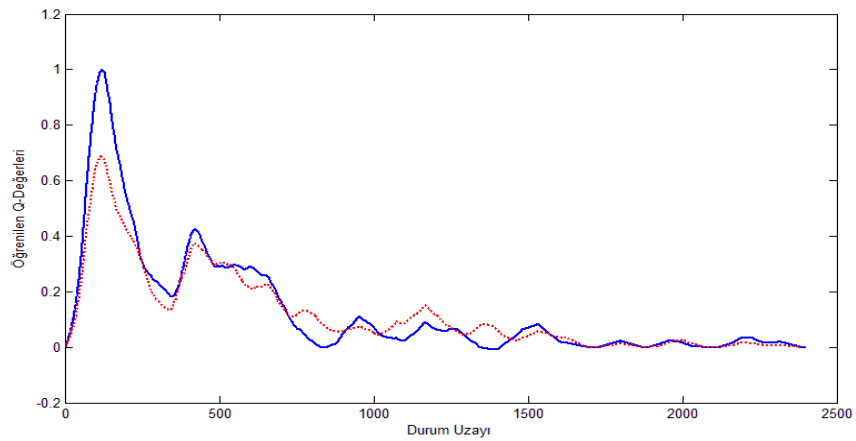
SAADU yaklaşımında, durum uzayını temsil etmek üzere ayırık zaman aralıkları $\tau = 100$ saniye olarak alınmıştır. Bu değer deneysel olarak tespit edilmiştir. İlk 0-50 s aralığında bilgi bulunmamaktadır. Ayırık durum uzayı 50 s değerinden sonra 100 s aralıklarla (51-150 s arası bir durum, 151-250 s arası ikinci durum...) ayrılmıştır. Örneğin $\Delta t^G = 273$ s değeri için R_1 robotu 251-350 s aralığının temsil ettiği duruma ilişkin Q-değerlerini öğrenmektedir. R_1 robotu için SAADU yaklaşımı kullanılarak elde

edilen Q değerleri grafiği Şekil 6.3'te gösterilmiştir. Grafikte de düz çizgi a_1^{11} hareketine ve kesikli çizgi a_2^{11} hareketine ilişkin Q değerlerini göstermektedir.



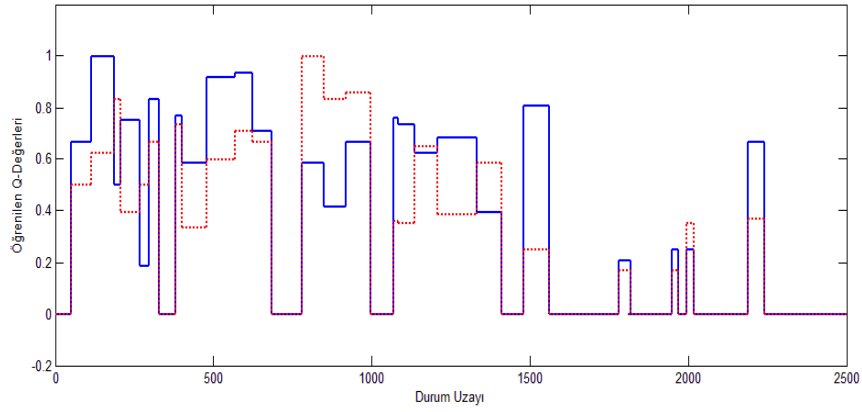
Şekil 6.3 ÇRS-I Ortamında Bulunan R_1 Robotuna ait Durum Uzayının SAADU Yaklaşımı için Elde Edilen Q Değerleri Grafiği

Durum uzayını ayrık değerler yerine sürekli değerlerle ifade eden DFSDU yaklaşımında Bölüm 4.4.2'de tanımlanan dağılım fonksiyonları kullanılmaktadır. Şekil 6.4'te R_1 robotuna ait DFSDU ile elde edilen Q değerleri grafiği görülmektedir.



Şekil 6.4 ÇRS-I Ortamında Bulunan R_1 Robotuna ait Durum Uzayının DFSDU Yaklaşımı için Elde Edilen Q Değerleri Grafiği

SAADU ve DAADU yaklaşımları için elde edilen Q-değerleri grafikleri incelendiğinde bariz farklılıklar görülmektedir. DAADU yaklaşımı ile elde edilen grafikte ayırık durum değeri aralıklarının değişken olduğu görülmektedir. Özellikle, DAADU yaklaşımında birbirine komşu ama farklı ayırık durum değerleri ile ifade edilen bazı değerlerin SAADU yaklaşımında aynı ayırık durum değeri ile ifade edildiği dikkati çekmektedir.

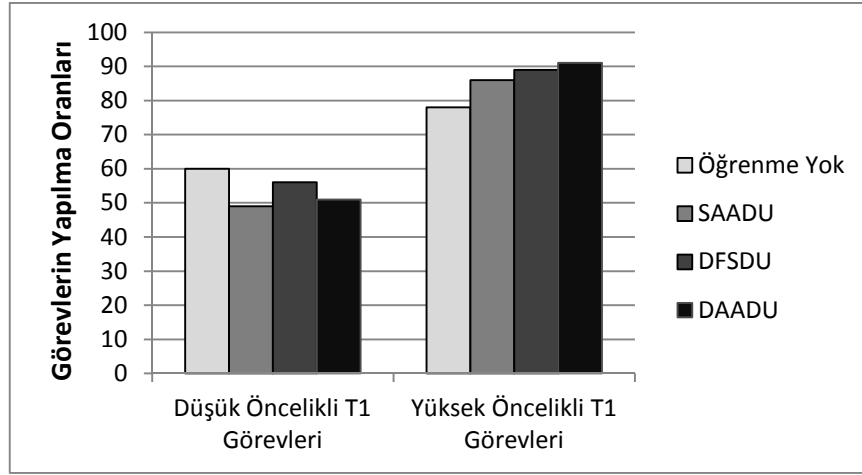


Şekil 6.5. ÇRS-I Ortamında Bulunan R_1 Robotuna ait Durum uzayının DAADU Yaklaşımı ile Elde Edilen Q Değerleri Grafiği

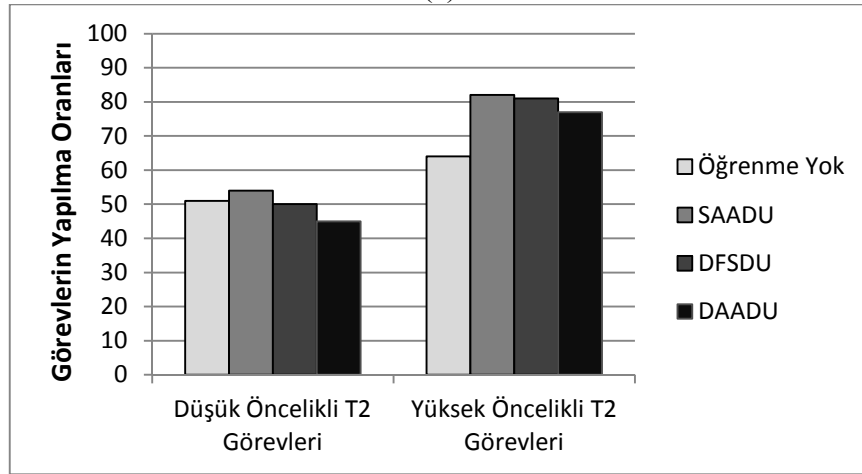
6.3.2 ÇRS-I için durum uzayı belirleme yaklaşımlarına ilişkin sonuçlar

ÇRS-I için durum uzayının SAADU, DFSDU ve DAADU yaklaşımları ile belirlenmesi halinde elde edilen sonuçlar görevlerin yapılma oranları kriteri açısından incelenmektedir. Elde edilen sonuçlar tüm görev tiplerinin yüksek ve düşük öncelikli görevleri için grafiksel olarak Şekil 6.6’da gösterilmektedir.

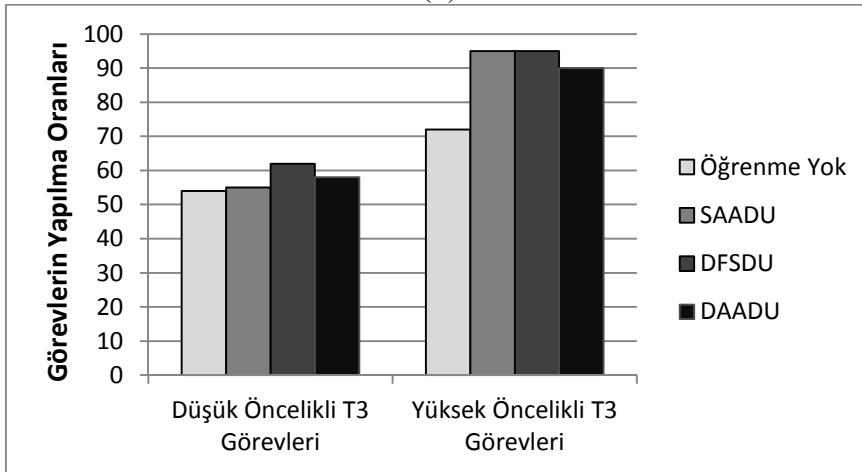
Grafikler incelendiğinde öğrenme yaklaşımlarının kullanılmasının durum uzayı belirleme yaklaşımın seçiminden bağımsız olarak yüksek öncelikli görevlerin yapılma oranlarında artış sağlandığı görülmektedir. Düşük öncelikli görevlerin yapılma oranlarında ise kullanılan yaklaşıma bağlı olmakla birlikte genellikle artış ama bazı yaklaşımlar için azalma meydana gelmektedir.



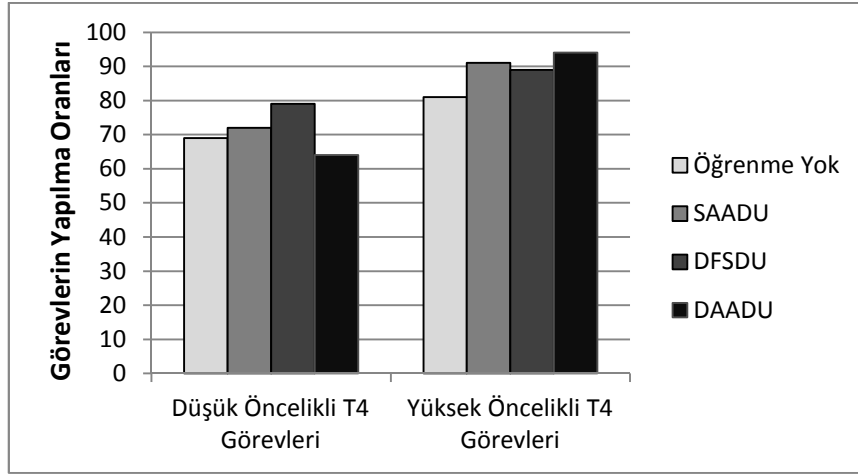
(a)



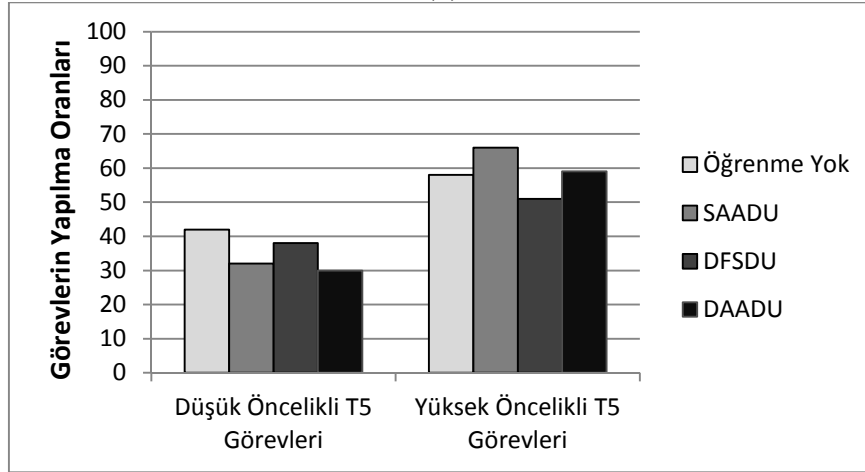
(b)



(c)



(d)



(e)

Şekil 6.6 ÇRS-II’de Durum Uzayı Belirleme Yaklaşımları için Görevlerin Yapılma Oranları (%) (a) T_1 (b) T_2 (c) T_3 (d) T_4 (e) T_5 Görevleri

Tüm durum uzayı belirleme yaklaşımları için görevlerin yapılma oranlarına ilişkin sayısal değerler Çizelge 6.8 ile verilmektedir. Her bir görev tipi için düşük öncelikli görevlerin, yüksek öncelikli görevlerin ve öncelik durumu dikkate alınmaksızın tüm görevlerin yapılma oranlarına ilişkin sayısal değerler çizelgede yer almaktadır. SAADU yaklaşımı tüm görev tipleri ve öncelik durumları için uygun sonuçlar sunmaktadır. Tüm görevler için yüksek öncelikli görevlerin yapılma oranları artmaktadır. Özellikle T_2 , T_3 ve T_4 görevleri için düşük öncelikli görevlerin yapılma oranlarında da artış sağlamaktadır.

Çizelge 6.8 ÇRS-I’de Durum Uzaýı Belirleme Yaklaşımaları için Görevlerin Yapılma Oranları

		Durum Uzaýı Belirleme Yaklaşımaları			
Görevler		Öğrenme Yok	SAADU	DFSDU	DAADU
T_1	Düşük Öncelikli	60	49	56	51
	Yüksek Öncelikli	78	86	89	91
	Toplam	66	62	68	65
T_2	Düşük Öncelikli	51	54	50	45
	Yüksek Öncelikli	64	82	81	77
	Toplam	56	64	61	56
T_3	Düşük Öncelikli	54	55	62	58
	Yüksek Öncelikli	72	95	95	90
	Toplam	60	69	74	69
T_4	Düşük Öncelikli	69	72	79	64
	Yüksek Öncelikli	81	91	89	94
	Toplam	73	79	83	75
T_5	Düşük Öncelikli	42	32	38	30
	Yüksek Öncelikli	58	66	51	59
	Toplam	48	44	43	40

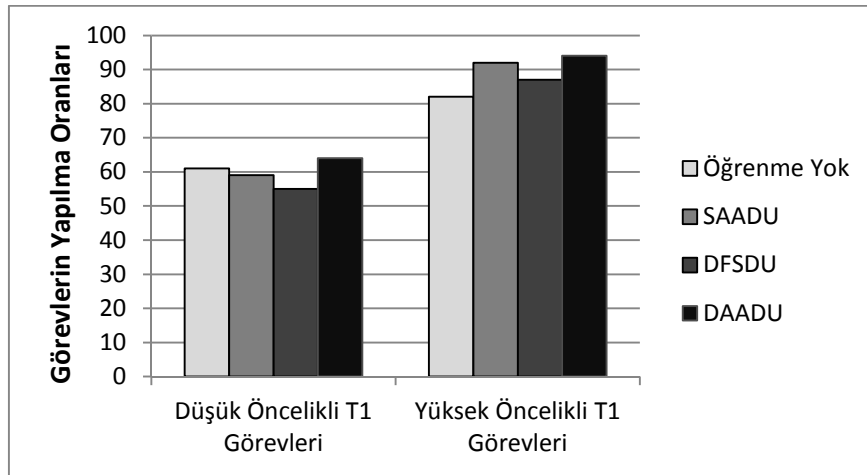
DFSDU yaklaşımı birçok görevde yüksek öncelikli görevlerin yapılma oranlarına ilişkin SAADU yaklaşımına benzer sonuçlar ve düşük öncelikli görevler için SAADU yaklaşımına göre daha yüksek yapılma oranları sağlamaktadır.

DAADU yaklaşımı ile özellikle yüksek öncelikli görevlerin yapılma oranları açısından tatmin edici sonuçlar elde edilmektedir. Düşük öncelikli görevlerin yapılma oranlarında ise genellikle azalma meydana gelmiştir. Bu değerler bazı görevler için SAADU yaklaşımına göre daha düşük yapılma oranları anlamına gelse de DAADU yaklaşımının sunduğu avantajlar kullanımının tercih edilmesini sağlamaktadır.

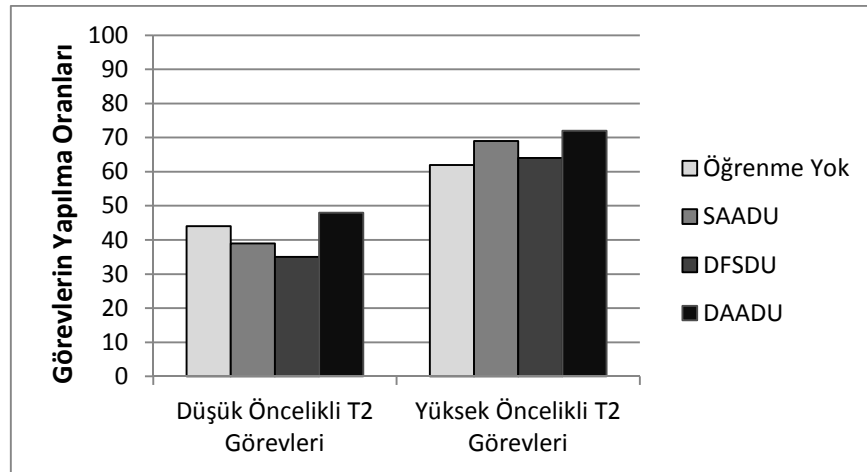
DAADU yaklaşımının avantajları, durum uzayını sistemin yapısına göre şekillendirmesi ve SAADU yaklaşımında olduğu gibi önceden tanımlı parametrelere ihtiyaç duymaması olarak belirtilebilir.

6.3.3 ÇRS-II için durum uzayı belirleme yaklaşımlarına ilişkin sonuçlar

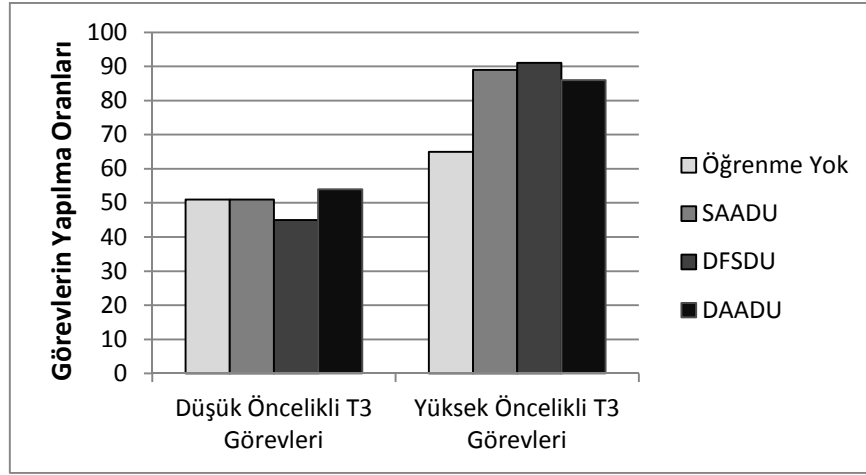
Durum uzayının önerilen SAADU, DFSDU ve DAADU yaklaşımları ile belirlendiği ÇRS-II ortamı için görevlerin yapılma oranları Şekil 6.7’de yer almaktadır. Sonuçlar her görev tipinin düşük ve yüksek öncelikli görevleri için ayrı ayrı verilmektedir.



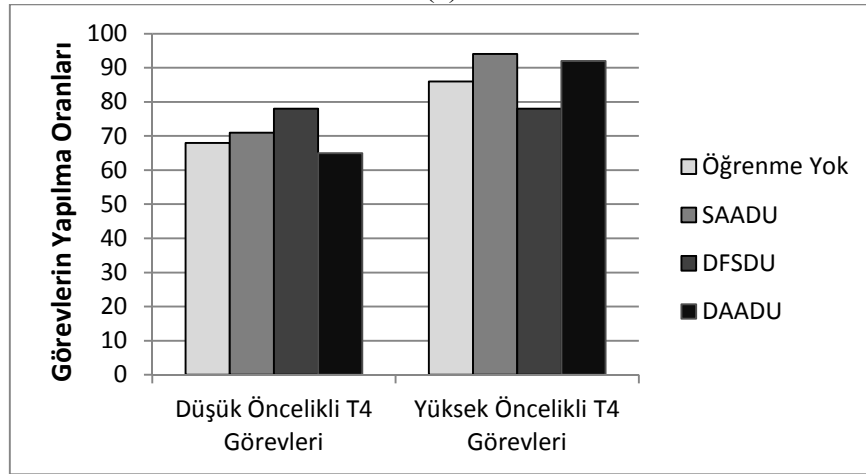
(a)



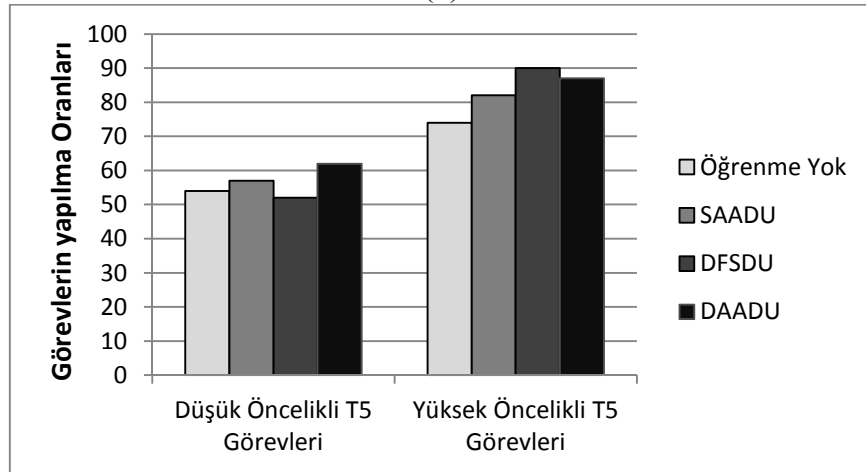
(b)



(c)



(d)



(e)

Şekil 6.7 ÇRS-II’de Durum Uzayı Belirleme Yaklaşımları için Görevlerin Yapılma Oranları (%) (a) T_1 (b) T_2 (c) T_3 (d) T_4 (e) T_5 Görevleri

Grafiklerde yer alan sonuçlar incelendiğinde kullanılan durum uzayı belirleme yönteminden bağımsız olarak öğrenme yaklaşımı kullanıldığında tüm görev tipleri için yüksek öncelikli görevlerin yapılma oranlarının arttığı görülmektedir. ÇRS-II için elde edilen sonuçlar da ÇRS-I için elde edilen sonuçları destekler niteliktedir. Özellikle DAADU yöntemi ile elde edilen görev yapılma oranları genellikle artış göstermiştir ve yöntemin kullanımının tercih edilmesi için yeteri kadar iyi sonuçlar sunmaktadır. Tüm durum uzayı belirleme yaklaşımları için görevlerin yapılma oranlarına ilişkin sayısal değerler Çizelge 6.9 ile verilmektedir. Her bir görev tipi için düşük öncelikli, yüksek öncelikli görevlerin ve toplam tüm görevlerin yapılma oranları çizelgede yer almaktadır.

Çizelge 6.9 ÇRS-II’de Durum Uzayı Belirleme Yaklaşımları için Görevlerin Yapılma Oranları

		Durum Uzayı Belirleme Yaklaşımları			
Görevler		Öğrenme Yok	SAADU	DFSDU	DAADU
T_1	Düşük Öncelikli	61	59	55	64
	Yüksek Öncelikli	82	92	87	94
	Toplam	68	71	66	75
T_2	Düşük Öncelikli	44	39	35	48
	Yüksek Öncelikli	62	69	64	72
	Toplam	50	50	45	56
T_3	Düşük Öncelikli	51	51	45	54
	Yüksek Öncelikli	65	89	91	86
	Toplam	56	64	61	65
T_4	Düşük Öncelikli	68	71	78	65
	Yüksek Öncelikli	86	94	78	92
	Toplam	74	79	78	74
T_5	Düşük Öncelikli	54	57	52	62
	Yüksek Öncelikli	74	82	90	87
	Toplam	61	66	65	71

6.3.4 Durum uzayı belirleme yaklaşımlarının değerlendirilmesi

Bu çalışmada ele alınan üç farklı durum uzayı belirleme yaklaşımı ÇRS-I ve ÇRS-II ortamları üzerinde uygulanmış ve sonuçlar yukarıda verilmiştir. Tüm yaklaşımlar yüksek öncelikli görevlerin yapılma oranlarında artış sağlamış olduğu için öğrenme tabanlı görev atama yaklaşımlarının durum uzayı belirleme yaklaşımlarından bağımsız olarak başarıya ulaştığını söylemek mümkündür. Önerilen durum uzayı belirleme yaklaşımlarının avantaj ve dezavantajlarını değerlendirmek üzere tüm yaklaşımlar sistem performansı, işlemsel yük, ön hazırlık gereksinimi, parametre bağımlılığı ve ortamı temsil etme yeteneği kriterleri açısından incelenmekte ve sonuçlar Çizelge 6.10'da verilmektedir. İşlemsel yük ve ortam temsili kriterlerini değerlendirmek üzere Düşük-Orta-Yüksek şeklinde üç kademeli bir ölçekleme kullanılmıştır. Ön hazırlık gereksinimi ve parametre bağımlılığı kriterleri ise Evet/Hayır şeklinde değerlendirilmiştir.

Çizelge 6.10 Durum Uzayı Belirleme Yaklaşımlarının Değerlendirilmesi

Kriter	Durum Uzayı Belirleme Yaklaşımları		
	SAADU	DFSDU	DAADU
İşlemsel Yük	Düşük	Yüksek	Yüksek
Ön Hazırlık Gereksinimi	Evet	Hayır	Hayır
Parametre Bağımlılığı	Evet	Evet	Hayır
Ortam Temsili	Düşük	Orta	Yüksek

Çizelge incelendiğinde önerilen üç yöntemin de çeşitli avantaj ve dezavantajlarının bulunduğu görülmektedir. Ön hazırlık gerektirmemesi, parametre bağımlılığının olmaması ve ortamın gereksinimlerine göre oluşturulması DAADU yönteminin kullanımını teşvik etmektedir. Ancak bu yöntem sisteme yüksek işlemsel yük getirmektedir.

BÖLÜM 7

DEĞERLENDİRME

Bu çalışmada, çok robotlu sistemlerde görev atama problemlerinin çözümünde sistem verimini artırmak amacıyla öğrenme yöntemleri tabanlı yaklaşımların uygulanması, uygulamada karşılaşılan problemler ve uygulamada karşılaşılan problemlerin çözüm yöntemleri araştırılmıştır. Gerçekleştirilen çalışmanın temel amacı, robotların öğrenme yeteneğine sahip olması ile geçmiş görev atama deneyimlerini daha sonraki uygulamalarda kullanmalarını sağlamaktır. Çok robotlu sistemlerde bir görev anons edildiğinde, robotlar o görevi yapma yeteneğine sahip olmaları durumunda ve o esnada başka bir görev yapmadıkları takdirde anons edilen bu görev için müzayedeye katılmaktadır. Ancak, robotların görevlere doğrudan istekli olmak yerine devam edecek süreçte anons edilecek yeni görevlere ilişkin beklentilerini de hesaba katarak o görevi yapma veya yapmama konusunda bir karar verme yapısına sahip olmaları ile sistemin toplam kazancında artış olması beklenmektedir. Yapılan çalışmalarda bu amaca ulaşmak üzere Q-öğrenme yöntemi kullanılmıştır.

Bu çalışmada, teorik olarak tek erkinli sistemler üzerinde tanımlanmış olan Q-öğrenme yönteminin çok erkinli sistemlerde ve özel olarak çok robotlu sistemlerde kullanımı araştırılmıştır. Q-öğrenme yönteminin ayrık ve sonlu durum ve hareket uzaylarına sahip MKS olarak tanımlanabilen ortamlar üzerinde optimal çözümlere yakınsadığı ispatlanmıştır. Ancak, çok robotlu sistemlerde robotların birbirlerinden bağımsız olarak hareket etme yeteneğine sahip olmaları ve çalışma ortamlarının genellikle kısmi gözlemlenebilir olması MKS olarak tanımlanma şartıyla çelişmektedir. Literatürde Q-öğrenme yönteminin çok erkinli sistemler için kullanımına dair iki temel yaklaşım bulunmaktadır. İlk yaklaşım olan dağıtık öğrenme yaklaşımı, tek erkinli Q-öğrenme yönteminin doğrudan çok erkinli yapıya uyarlanmasına dayanmaktadır. Her bir erkin diğer erkinlerin kararlarını ve hareketlerini dikkate almadan sadece kendi yerel bilgileri dahilinde öğrenme sürecini gerçekleştirmektedir. Bu yaklaşımın uygulaması tek erkinli Q-öğrenme yöntemi kadar basit olmakla birlikte erkinlerin sadece yerel

bilgilere bağılı kalmaları nedeniyle davranış çakışmaları ortaya çıkabilmektedir. Optimal sonuca ulaşma garanti edilememektedir. İkinci yaklaşım ise merkezi öğrenme yaklaşımıdır. Merkezi öğrenme yaklaşımında erkinler tümleşik hareketlerinin sonucuna göre öğrenmekte ve tüm erkinler için ortak en iyi çözüm aranmaktadır. Böylece davranış çakışmalarının önüne geçilebilmektedir. Ancak tümleşik hareket uzayı kullanımının neden olduğu öğrenme uzayı boyutunun erkin sayısına bağılı olarak üstel şekilde artması, sisteme büyük işlemsel yük getirmektedir. Bu çalışmada, bu iki temel öğrenme yaklaşımı incelenmiş ve her iki yaklaşımın avantajlarından yararlanmak üzere yeni bir yaklaşım olarak Strateji Planlamalı Dağıtık Öğrenme yaklaşımı önerilmiştir. Bu yaklaşımda, robotlar öğrenme sürecini önceden belirlenmiş stratejiler doğrultusunda dağıtık olarak gerçekleştirmektedir. Dağıtık öğrenme tabanlı olduğu için işlemsel kolaylık sağlamaktadır ve merkezi bir planlama yapısı kullanıldığı için robotlar arası davranış çakışmalarının önüne geçilmektedir. Gerçekleştirilen uygulamalarla her üç yaklaşımın da sistem performansı üzerindeki etkileri, görevlerin yapılma oranları ve işlemsel yük kriterleri açısından incelenmiş, önerilen yaklaşımın yukarıda anlatıldığı üzere etkin sonuçlar sunduğu görülmüştür.

Q-öğrenme yöntemi, ayırık ve sonlu durum ve hareket uzayları üzerinde her bir durum-hareket çiftine ilişkin Q değerlerinin öğrenilmesi esasına dayalı bir yöntemdir. Oysa, çok robotlu sistemlerin çalışma ortamlarının doğası gereği durum ve/veya hareket uzaylarının ayırık ve sonlu olması garanti edilememektedir. Bu sebeple, Q-öğrenme yönteminin çok robotlu sistemlerde uygulanmasında karşılaşılan temel problemlerden biri de durum ve/veya hareket uzaylarının ayırık olarak tanımlanmasında ortaya çıkmaktadır. Bu noktada, ayırıklaştırma temsil hassasiyetinin gerektiği gibi belirlenebilmesi önem kazanmaktadır. Yüksek temsil hassasiyeti aynı nitelikteki bilginin farklı durum bileşenleriyle ifade edilmesi dolayısıyla işlemsel yükün artmasına neden olurken, düşük temsil hassasiyeti sistem için gerekli ve ayırt edici nitelikteki verilerin kaybolması ile neticelenebilmektedir. Sistem için gerekli temsil hassasiyetinin belirlenebilmesi, sistem hakkında önceden ve ayrıntılı bilgi sahibi olmayı gerektirmektedir. Bu çalışmada ele alınan yapıda hareket uzayı ayırık olarak tanımlanmıştır. Ancak, durum uzayı bileşenleri görevlerin gelme zamanları arasında geçen süre temel alınarak düzenlenmektedir. Süre gibi sürekli bir değişkene ilişkin

bilginin ayrıık olarak kullanılması tanımlama ve temsil etme problemlerine yol açmaktadır. Bu amaçla, durum uzayını belirlemek üzere SAADU, DFSDU ve DAADU yöntemleri önerilmiştir. SAADU yönteminde durum uzayı sabit temsil hassasiyeti kullanılarak öğrenme sürecinin başında ayrıklaştırılırken, DAADU yönteminde ayrıklaştırma işlemi ardışıl kümeleme tabanlı bir yaklaşım kullanılarak öğrenme sürecinde sistemin gereklerine göre gerçekleştirilmektedir. DFSDU yöntemi ise durum uzayının sürekli değişkenlerle tanımlanmasına imkan sağlayan bir yaklaşım ortaya koymaktadır. Her üç yöntemin de sistem performansı üzerindeki etkisi gerçekleştirilen uygulamalarla görevlerin yapılma oranları kriteri açısından incelenmiş ve başarılı sonuçlar elde edildiğı görölmüştür.

Gelecek çalışmalarda, öğrenme sürecinin sistemin çalışması esnasında gerçekleştirildiğı durumların ele alınması hedeflenmektedir.

KAYNAKLAR DİZİNİ

- Amador J. J., 2005, Sequential clustering by statistical methodology, *Pattern Recognition Letters*, 26, pp. 2152-2163.
- Arkin R. C., 1998, *Behavior-based robotics*, MIT Press, Cambridge.
- Berhault M., Huang H., Keskinocak P., Koenig S., Elmaghraby W., Griffin P. and Kleyegd A., 2003, Robot exploration with combinatorial auctions, *Proceedings of IEEE/RSJ International Conference of Robots and Systems (IROS)*, pp. 1957-1962.
- Boeling M., 2000, Convergence problem of general-sum stochastic games, *Proceedings of Seventeenth International Conference on Machine Learning*, pp. 89-94.
- Boutlier C., 1996, Planning, learning and coordination in multiagent decision processes, *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge, TARK '96*, pp. 195-210.
- Busquets D., Simmons R., 2006, Learning when to auction and when to bid, *Distributed Autonomous Robotic Systems (DARS)*, vol. 7, pp. 21-30.
- Buşoniu L., Babuška R., Schutter B., 2008, A comprehensive survey of multiagent reinforcement learning, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol.38, no.2, pp. 156-172.
- Carvalho F., Souza R., Chavent M., Lechevailler Y., 2006, Adaptive hausdorff distances and dynamic clustering of symbolic interval data, *Pattern Recognition Letters*, 27, pp. 167-179.
- Chaimowicz, L., Sugar, T., Kumar, V. and Campos, M. 2001, An architecture for tightly coupled multi-robot cooperation, *Proceedings of the International Conference on Robotics and Automation*, Seoul, Korea, pp. 2992-2997.

KAYNAKLAR DİZİNİ (devam)

- Claus C., Boutilier C., 1998, The dynamics of reinforcement learning in cooperative multiagent systems, Proceedings of Fifteenth National Conference on Artificial Intelligence, pp. 235-262.
- Dias, M. B., 2004, TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments, Ph.D. Thesis, Carnegie Mellon University.
- Dias M. B., Zlot R. M., Kaltra N., Stentz A., 2006, Market-based multirobot coordination: a survey and analysis, Proceedings of the IEEE, vol. 94, no.7, July 2006, pp. 1257-1270.
- Gerkey B. P., Mataric M. J., 2002, Sold!: Auction methods for multi robot coordination, IEEE Transactions on Robotics and Automation, vol. 18, no. 5, pp. 758-768.
- Gerkey B. P., Mataric M. J., 2004, A formal analysis and taxonomy of task allocation in multi-robot systems, International Journal of Robotics Research, 23(9), pp. 939-954.
- Goldberg, D., Cicirello V., Dias, M. B., Simmons, R., Smith, S. and Stentz, A., 2003, Market-based multi-robot planning in a distributed layered architecture, Multi-Robot Systems: From Swarms to Intelligent Automata Proceedings from the International Workshop on Multi-Robot Systems, pp. 27-38.
- Hamagami T., Hirata H., 2003, An adjustment method of the number of states on Q-learning segmenting state space adaptively, IEEE International Conference on Systems, Man and Cybernetics, vol.4, pp. 3062-3067.
- Hatime H., Pendse R., Watkins J. M., 2013, A comparative study of task allocation strategies in multi-robot systems, IEEE Sensors Journal, vol. 13, no. 1, 253-262.

KAYNAKLAR DİZİNİ (devam)

- Hu J., Wellman M. P., 1998, Multiagent reinforcement learning: theoretical framework and an algorithm, Proceedings of the Fifteenth International Conference on Machine Learning ICML'98, pp. 242-250.
- Hu J., Wellman M. P., 2003, Nash Q-learning for general sum games, Journal of Machine Learning Research, 4, pp. 1039-1069.
- Hwang K. S., Tan S. W., Chen C. C., 2004, Cooperative strategy based on adaptive Q-learning for robot soccer systems, IEEE Transactions on Fuzzy Systems, vol. 12, no. 4, pp. 569-576.
- Hwang K. S., Chen Y. J., Lee C. H., 2007, Reinforcement learning in strategy selection for a coordinated multirobot system, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 37, no.6, pp. 1151-1157.
- Hwang K. S., Lo C.Y., Liu W. L., 2009a, A modular agent architecture for an autonomous robot, IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 8, pp. 2797-2806.
- Hwang K. S., Lo C.Y., Chen K. J., 2009b, Real-valued Q-learning in multi-agent cooperation, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, USA, October 2009, pp. 395-400.
- Hwang K. S., Jiang W. C., Lin Y. H., Lai L. H., 2012, Continuous Q-learning for multi-agent cooperation, Cybernetics and Systems: an International Journal, vol. 43, pp. 227-256.
- Jones E. G., Dias M. B., Stentz A., 2007, Learning-enhanced market-based task allocation for oversubscribed domains, Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, pp. 2308-2313.

KAYNAKLAR DİZİNİ (devam)

- Kaelbling L. P., Littman M. L., Moore A. W., 1996, Reinforcement learning: a survey, *Journal of Artificial Intelligence Research*, 4, pp. 237-285.
- Kaleci B., Parlaktuna O., Ozkan M., Kırılık G., 2010, Market-based task allocation by using assignment problem, *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 135-14.
- Kawakami K. I., Ohkura K., Ueda K., 1999, Adaptive role development in a homogeneous connected robot group, *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol.3, pp. 254-256.
- Lauer M., Riedmiller M., 2000, An algorithm for distributed reinforcement learning in cooperative multi-agent systems, *Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, p. 535-542.
- Lee I. S. K., Lau H. Y. K., 2004, Adaptive state space partitioning for reinforcement learning, *Engineering Application of Artificial Intelligence*, vol. 17, pp. 577-588.
- Liu Z., Ang Jr M. H., Seah W. K. G., 2005, Reinforcement learning of cooperative behaviors for multi-robot tracking of multiple moving targets, *Proceedings of IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*, pp. 1289–1294.
- Littman M. L., 1994, Markov games as a framework for multiagent learning, *Proceedings of the Eleventh International Conference on Machine Learning*, San Francisco, California, pp. 157–163.
- Littman M. L., 2001a, Value-function reinforcement learning in markov games, *Journal of Cognitive Systems Research*, 2, pp. 55-66.
- Littman M. L., 2001b, Friend-or-foe Q-learning in general-sum games, *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 322–328.

KAYNAKLAR DİZİNİ (devam)

- Martinson E., Stoytchev A., Arkin R. C., 2002, Robot behavioral selection using Q-learning, Proceedings of IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS), Lausanne, Switzerland, pp. 970-977.
- Martinson E., Arkin R. C., 2003, Learning to role-switch in multi-robot systems, Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, pp. 2727-2734.
- Mataric M. J., 1995, Issues and approaches in the design of collective autonomous agents, Robotics and Autonomous Systems, vol.16, pp. 321–331.
- Mataric M. J., 1997, Reinforcement learning in multi-robot domain, Autonomous Robots, 4(1), pp. 73-83.
- Matignon L., Laurent G. J., Le Fort-Piat N., 2007, Hysteretic Q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams, Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, pp. 64-69.
- Mitchell T. M., 1997, Machine learning, McGraw-Hill, New York.
- Mosteo A. R., Montano L., 2007, Comparative experiments on optimization criteria and algorithms for auction based multi-robot task allocation, Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3345-3350.
- Park K. H., Kim Y. J., Kim J. H., 2001, Modular Q-learning based multi-agent cooperation for robot soccer, Robotics and Autonomous Systems, 35, pp. 109-122.
- Parker, L. E., 2008, Distributed intelligence: overview of the field and its application in multi-robot systems, Journal of Physical Agents, Special Issue on Multi-Robot Systems, vol. 2(2), pp. 5-14.

KAYNAKLAR DİZİNİ (devam)

- Riedmiller M., Gabel T., Hafner R., 2009, Reinforcement learning for robot soccer, *Auton Robot*, vol. 27, pp. 55-73.
- Rossi C., Aldama L., Barrientos A., 2009, Simultaneous task subdivision and allocation for teams of heterogeneous robots, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 946-951.
- Russel S., Norvig P., 2003, *Artificial intelligence a modern approach*, Prentice Hall, New Jersey.
- Schneider, J., Apfelbaum, D., Bagnell, D. and Simmons, R., 2005, Learning opportunity costs in multi-robot market based planners, *Proceedings of the International Conference on Robotics and Automation*, Barcelona, Spain, pp. 1151-1156.
- Smart W. D., Kaelbling L. P., 2002, Effective reinforcement learning for mobile robots, *Proceedings of the IEEE International Conference on Robotics and Automation ICRA'2002*, Washington DC, pp. 3404-3410.
- Spaan M. T. J., Gonçalves N., Sequeira J., 2010, Multirobot coordination by auctioning POMDPs, *IEEE International Conference on Robotics and Automation*, Alaska, USA, pp. 1446-1451.
- Spatch H., 1980, *Cluster analysis algorithm*, Ellis Horwood.
- Stone P., Veloso M., 2000, Multiagent systems: a survey from machine learning perspective, *Autonomous Robots*, vol. 8, pp. 345-383.
- Sutton R. S., Barto A. G., 1998, *Reinforcement learning: an introduction*, MIT Press, Cambridge.
- Theodoridis S., Koutroumbas K., 2003, *Pattern recognition*, Elsevier Academic Press.

KAYNAKLAR DİZİNİ (devam)

- Tuyls K., Nowè A., 2005, Evolutionary game theory and multi-agent reinforcement learning, *The Knowledge Engineering Review*, vol. 20, no. 1, pp. 63-90.
- Uchibe E., Nakamura M., Asada M., 1998, Co-evolution for cooperative behavior acquisition in a multiple mobile robot environment, *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria,B.C., Canada, pp. 245-430.
- Wang Y., de Silva C. W., 2005, An object transportation system with multiple robots and machine learning, *Proceedings of the 2005 American Control Conference (ACC 2005)*, Portland, OR, pp. 1371-1376.
- Wang Y., de Silva C. W., 2006a, Multi-robot box-pushing: single-agent Q-learning vs. team Q-learning, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, pp. 3694-3699.
- Wang Y., de Silva C. W., 2006b, Extend single-agent reinforcement learning approach to a multi-robot cooperative task in an unknown dynamic environment. *Proceedings of the IEEE 2006 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, pp. 10098-10104.
- Wang Y., de Silva C. W., 2008, A machine-learning approach to multi-robot coordination, *Engineering Applications of Artificial Intelligence*, vol. 21, issue 3, pp. 470-488.
- Watkins C. J. C. H., 1989, *Learning from delayed rewards*, University of Cambridge, UK, PhD Thesis.
- Watkins C. J., Dayan P., 1992, Q-learning, *Machine Learning*, vol. 8, 1992.
- Wooldridge M., 2002, *An introduction to multiagent systems*, John Wiley and Sons Ltd, Sussex, England.

KAYNAKLAR DİZİNİ (devam)

Yang E., Gu D., 2004, Multiagent reinforcement learning for multi-robot systems: a survey, CSM-404, Technical Reports of the Department of Computer Science, University of Essex.

Zikmund W. G., Babin B. J., Essentials of marketing research, South Western, Cengage Learning, p:394, 2010.

Zlot, R. and Stentz, A., 2006, Market-based multirobot coordination for complex tasks, International Journal of Robotics Research, Special Issue on the 4th International Conference on Field and Service Robotics, vol. 25(1), pp. 73-101.

ÖZGEÇMİŞ

Hatice Hilal EZERCAN KAYIR 1978 yılında Denizli’de doğdu. İlk, orta ve lise eğitimini aynı şehirde tamamladı. Lisans eğitimini 1995-2000 yılları arasında Dokuz Eylül Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü’nde ve yüksek lisans eğitimini 2000-2003 yılları arasında Pamukkale Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı, Kontrol ve Kumanda Bilim Dalında tamamladı. 2014 yılında Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı, Kontrol ve Kumanda Bilim Dalında doktora eğitimini tamamlamıştır. Başlıca çalışma konuları gezgin robot sistemleri ve akıllı sistemlerdir.