

T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI



**İKİ MÜŞTERİ SINIFLI M/M/1 KUYRUK SİSTEMİNDE OPTİMAL
KUYRUK SIRA POLİTİKASININ BELİRLENMESİ**

SELİN CİĞER

YÜKSEK LİSANS TEZİ

Jüri Üyeleri : Dr. Öğr. Üyesi M.A. Beyazıt OCAKTAN (Tez Danışmanı)
Prof. Dr. Aslan Deniz KARAOĞLAN
Doç. Dr. Alparslan Serhat DEMİR

BALIKESİR, OCAK - 2024

ETİK BEYAN

Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak tarafımda hazırlanan “**İki Müşteri Sınıflı M/M/1 Kuyruk Sisteminde Optimal Kuyruk Sıra Politikasının Belirlenmesi**” başlıklı tezde;

- Tüm bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Kullanılan veriler ve sonuçlarda herhangi bir değişiklik yapmadığımı,
- Tüm bilgi ve sonuçları bilimsel araştırma ve etik ilkelere uygun şekilde sunduğumu,
- Yararlandığım eserlere atıfta bulunarak kaynak gösterdiğimi,

beyan eder, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ederim.

Selin CİĞER

ÖZET

İKİ MÜŞTERİ SINIFLI M/M/1 KUYRUK SİSTEMİNDE OPTİMAL KUYRUK SIRA POLİTİKASININ BELİRLENMESİ

YÜKSEK LİSANS TEZİ

SELİN CİĞER

BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ M. A. BEYAZIT OCAKTAN)

BALIKESİR, OCAK - 2024

Bu tez çalışmasında normal (Tip 1) ve öncelikli (Tip 2) olmak üzere, iki farklı müşteri tipini içeren M/M/1 kuyruğu ele alınmıştır. İki tür müşteri hizmete farklı bağımsız Poisson süreçlerine göre gelerek tek bir ortak kuyruk oluşturur ve kuyruқта birim zaman başına farklı elde bulundurma maliyetlerine sahiptir. Tek sunucuya sahip olan sistemde öncelikli müşteri kuyruğa girdiğinde ya geliş sırasına göre hizmet alır ya da normal müşterileri atlayarak hizmet alır.

Tezde ele alınan problem, Markov Karar Süreci (MDP) olarak modellenmiş ve Simülasyon Tabanlı Optimizasyon yöntemlerinden olan ödüllü öğrenme algoritmalarıyla çözülerek, birim zamandaki ortalama maliyeti yaklaşık en küçükleyen kuyruk sıra politikası belirlenmesi amaçlanmıştır. İki farklı sayısal örnek tasarlanarak ödüllü öğrenme algoritmasıyla elde edilen politikaların ortalama maliyetleri miyopik politikalarla kıyaslanmıştır. Bu miyopik politikalar; öncelikli müşterilerin geliş sırasına göre kuyruğa girmesi, öncelikli müşterilerin normal müşterileri atlayarak kuyruk sırasında öne alınması ve öncelikli müşterilerin kuyruğa gelişlerinde hizmet sırası rastgele karara bağlanarak kuyruқта öne al ya da alma şeklindedir. Gerçekleştirilen sayısal örneklere göre, ödüllü öğrenme algoritmasıyla elde edilen kuyruk politikalarının ortalama maliyetleri, miyopik politikaların ortalama maliyetlerinden daha düşüktür.

ANAHTAR KELİMELELER: M/M/1 kuyruk sistemi, Öncelikli kuyruk disiplini, Markov karar süreçleri, Ödüllü öğrenme.

Bilim Kod / Kodları : 90610, 90612, 90618

Sayfa Sayısı : 71

ABSTRACT

DETERMINATION OF OPTIMAL QUEUE POLICY IN M/M/1 QUEUE SYSTEM WITH TWO CUSTOMER CLASSES

MSC THESIS

SELİN CİĞER

**BALIKESİR UNIVERSITY INSTITUTE OF SCIENCE
INDUSTRIAL ENGINEERING**

(SUPERVISOR: ASSIST. PROF. DR. M. A. BEYAZIT OCAKTAN)

BALIKESİR, JANUARY - 2024

In this thesis, we consider an M/M/1 queue with two different types of customers normal (type 1) and priority (type 2). The two types of customers arrive according to different independent Poisson processes, form a single common queue, and have different holding costs per unit time in the queue. In the system with a single server, when a priority customer enters the queue, he/she is either served according to the order of arrival, or he/she is served by skipping the regular customers.

The problem addressed in the thesis is modeled as Markov Decision Process (MDP) and solved by using reinforcement learning algorithms, which is one of the Simulation Based Optimization methods. The goal is to determine the queue policy that approximately minimizes the average cost per unit time. Two different numerical examples are designed, the average costs of policies developed with the reinforcement learning algorithm are compared with myopic policies. These myopic policies; priority customers enter the queue according to the order of arrival, priority customers are brought to the front of the queue, bypassing normal customers, and when priority customers come to the queue, the order of service is randomly decided and they are either moved to the front of the queue or not. According to result of the numerical examples, the average costs per unit time of queuing policies obtained with the reinforcement learning algorithm are much lower than ones of myopic policies.

KEYWORDS: M/M/1 queue system, priority queue discipline, Markov decision processes, reinforcement learning.

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİL LİSTESİ	iv
TABLO LİSTESİ	v
KISALTMALAR LİSTESİ	vi
ÖNSÖZ	vii
1. GİRİŞ	1
1.1 Motivasyon	2
1.2 Literatür Taraması.....	3
2. KUYRUK SİSTEMLERİ	14
2.1 Kuyruk Sistemleri Bileşenleri.....	15
2.2 Öncelikli Kuyruk Sistemleri	18
3. SİMÜLASYON TABANLI OPTİMİZASYON	22
3.1 Markov Karar Süreçleri	23
3.2 Ödüllü Öğrenme.....	25
3.2.1 Ödüllü Öğrenmede Öğrenme Yaklaşımı.....	29
3.2.2 Q-Öğrenme.....	31
4. PROBLEM TANIMI VE MODEL	36
4.1 Problem Tanımı	36
4.2 Model	39
5. SAYISAL ÖRNEKLER	44
5.1 Sayısal Örnek 1	44
5.2 Sayısal Örnek 2	48
5.2.1 R=10 Durumunda Oluşan Maliyetler.....	49
5.2.2 R=20 Durumunda Oluşan Maliyetler.....	54
5.2.3 R=30 Durumunda Oluşan Maliyetler.....	59
6. SONUÇ VE ÖNERİLER	65
7. KAYNAKLAR	67
ÖZGEÇMİŞ	71

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Temel bir kuyruk sistemi yapısı.....	15
Şekil 2.2: Kuyruk sistemi gösterimi.....	18
Şekil 2.3: k öncelik sınıfına sahip kuyruk sisteminin şematik gösterimi.....	19
Şekil 3.1: Markov karar süreci şematiği	23
Şekil 3.2: Yapay zekâ, makine öğrenmesi ve ödüllü öğrenme	25
Şekil 3.3: Makine öğrenmesinin alt dalları	26
Şekil 3.4: RL ve DP metodolojilerindeki farklılıkları vurgulayan şematik	28
Şekil 3.5: Ödüllü öğrenme mekanizması	29
Şekil 3.6: Bir simülörde Q faktör değerlerinin güncellenmesi	33
Şekil 4.1: Öncelikli kuyruk probleminin şematik gösterimi	36
Şekil 4.2: Öncelikli müşteri geldiği anda alınan iki ayrı kararın etkisi ile oluşan yeni durumun şematik gösterimi.....	41
Şekil 4.3: Simülasyon algortiması akış diyagramı.....	43
Şekil 5.1: Sayısal örnek 1’de ödüllü öğrenme algoritmasıyla elde edilen politikaların yakınsama grafiği.....	45
Şekil 5.2: Sayısal örnek 1’de miyopik politika 1’in yakınsama grafiği.....	46
Şekil 5.3: Sayısal örnek 1’de miyopik politika 2’in yakınsama grafiği.....	46
Şekil 5.4: Sayısal örnek 1’de miyopik politika 3’in yakınsama grafiği.....	47
Şekil 5.5: Sayısal örnek 1’de ödüllü Sayısal örnek 1’in öğrenilen politikayla miyopik politika 1, 2 ve 3’ün kıyaslanması	47
Şekil 5.6: R=10 ceza maliyetli örnekte ödüllü öğrenme algoritmasıyla elde edilen politikaların yakınsama grafiği	49
Şekil 5.7: R=10 ceza maliyetli örnekte miyopik politika 1’in yakınsama grafiği	50
Şekil 5.8: R=10 ceza maliyetli örnekte miyopik politika 2’in yakınsama grafiği	50
Şekil 5.9: R=10 ceza maliyetli örnekte miyopik politika 3’in yakınsama grafiği	51
Şekil 5.10: R=10 ceza maliyetli sayısal örnek 2’in öğrenilen politika ile miyopik politika 1, 2 ve 3’ün kıyaslanması	51
Şekil 5.11: R=20 ceza maliyetli örnekte ödüllü öğrenme algoritmasıyla elde edilen politikaların yakınsama grafiği	55
Şekil 5.12: R=20 ceza maliyetli örnekte miyopik politika 1’in yakınsama grafiği	55
Şekil 5.13: R=20 ceza maliyetli örnekte miyopik politika 2’in yakınsama grafiği	56
Şekil 5.14: R=20 ceza maliyetli örnekte miyopik politika 3’in yakınsama grafiği	56
Şekil 5.15: R=20 ceza maliyetli sayısal örnek 2’in öğrenilen politika ile miyopik politika 1, 2 ve 3’ün kıyaslanması	57
Şekil 5.16: R=30 ceza maliyetli örnekte ödüllü öğrenme algoritmasıyla elde edilen politikaların yakınsama grafiği.....	60
Şekil 5.17: R=30 ceza maliyetli örnekte miyopik politika 1’in yakınsama grafiği	60
Şekil 5.18: R=30 ceza maliyetli örnekte miyopik politika 2’in yakınsama grafiği	61
Şekil 5.19: R=30 ceza maliyetli örnekte miyopik politika 3’in yakınsama grafiği	61
Şekil 5.20: R=30 ceza maliyetli sayısal örnek 2’in öğrenilen politika ile miyopik politika 1, 2 ve 3’ün kıyaslanması	62

TABLO LİSTESİ

	<u>Sayfa</u>
Tablo 1.1: Literatür taraması özet tablosu.....	9
Tablo 3.1: Q-faktör tablonun örnek yapısı.....	34
Tablo 4.1: Örnek simülasyon veri tablosu.....	37
Tablo 4.2: Örnek simülasyon tablosu (saf öncelik yok).....	37
Tablo 4.3: Örnek simülasyon tablosu (saf öncelik var).....	38
Tablo 4.4: Örnek simülasyon tablosu (karma öncelik var).....	38
Tablo 4.5: Örnek simülasyon sonuç kıyaslaması	39
Tablo 5.1: Öğrenilen politika/miyopik politika 2 (R=10) varyans eşitliğine ilişkin hipotez testi Minitab sonuçları	53
Tablo 5.2: Öğrenilen politika/miyopik politika 2 (R=10) ortalama maliyet hipotez testi Minitab sonuçları	54
Tablo 5.3: Öğrenilen politika/miyopik politika 2 (R=20) varyans eşitliğine ilişkin hipotez testi Minitab sonuçları	58
Tablo 5.4: Öğrenilen politika/miyopik politika 2 (R=20) ortalama maliyet hipotez testi Minitab sonuçları	59
Tablo 5.5: Öğrenilen politika/miyopik politika 2 (R=30) varyans eşitliğine ilişkin hipotez testi Minitab sonuçları	63
Tablo 5.6: Öğrenilen politika/miyopik politika 2 (R=30) ortalama maliyet hipotez testi Minitab sonuçları	64

KISALTMALAR LİSTESİ

MDP	: Markov Karar Süreci (Markov Decision Process)
SMDP	: Yarı Markov Karar Süreci (Semi Markov Decision Process)
RL	: Ödüllü Öğrenme (Reinforcement Learning)
DP	: Dinamik Programlama (Dynamic Programming)
CTMDP	: Sürekli Zaman Markov Karar Süreci (Continuous Time Markov Decision Process)
MGZ	: Müşteri Geliş Zamanı
SS	: Servis Süresi
QLT	: Kuyruk Uzunluğu Eşiği (Queue Length Threshold)
FIFO	: İlk Gelen İlk Çıkar (First In First Out)
LIFO	: Son Gelen İlk Çıkar (Last In First Out)
SIRO	: Gelişi Güzel Sırayla Hizmet (Service In Random Order)
RR	: Dönen Tur (Round Robin)
PRIO	: Öncelikli Kuyruk Disiplini (Priority Queue Discipline)

ÖNSÖZ

Lisansüstü eğitim hayatım boyunca çalışmalarım ve araştırmalarım boyunca beni yönlendiren, bilgi ve tecrübesiyle desteğini esirgemeyen, her fırsatta zamanını ayıran ve tez çalışmamda büyük katkı sağlayan değerli danışman hocam Sayın Dr. Öğr. Üyesi M.A. Beyazıt OCAKTAN'a sonsuz teşekkür ve şükranlarımı sunarım.

Son olarak teşekkürden fazlasını hak eden, benim için her türlü fedakarlığı yapan ve yapmaya devam edeceğinden emin olduğum canım annem Gülcan TOPAK'a, gücünü sürekli arkamda hissettiğim ve her kararında yanımda olan canım babam Bülent TOPAK'a, beni her konuda destekleyen canım abim Ali TOPAK'a ve son olarak beni motive eden, desteğini hiçbir zaman esirgemeyen sevgili eşim Atalay Tahir CİĞER'e teşekkür ediyorum.

Balıkesir, 2024

Selin CİĞER

1. GİRİŞ

Günümüzde, küresel nüfus miktarının hızla artması ve karmaşıklaşan yaşam koşullarıyla birlikte “sıra bekleme” veya “kuyruk” kavramları giderek daha çok hayatımızın bir parçası olmaya başlamıştır. Trafikteki araçlar, metro durağındaki insanlar veya acil serviste bekleyen hastalar gibi birçok günlük durumda karşılaşılan sıra bekleme, modern yaşamın kabul edilen bir gerçeği haline gelmiştir.

Bekleme, genel olarak bir süreç veya bir hizmet için talebin oluşturulduğu andan, sürecin tamamlanmasını veya talebin karşılanmasını beklerken geçen zaman aralığı olarak tanımlanabilir. Bekleme için oluşturulan hatlara kuyruk hatları, kısaca kuyruk denmektedir. Kuyruklar, hizmet sektörü başta olmak üzere birçok kuruluşun yüksek hizmet kalitesi sunabilmesi için yönetilmesi gereken hatlardır. Kuyruk hatları sunuculardan hizmet almayı bekleyen müşterilerden oluşur. Organizasyonlar kuyruk yönetimine önem vermelidir, aksi takdirde kuyruk hatlarında uzun bekleme süreleriyle karşılaşan müşteriler memnun olmayarak potansiyel bir kayıp olabilmektedir. Bu yüzden kuyruk teorisi matematiksel olarak kuyruk hatları için geliştirilmiştir. Günümüzde en çok yararlanan kuyruk sistemlerinden biri öncelikli hizmet disiplinine sahip kuyruk sistemleridir. Örneğin süpermarkette beşten az ürün içeren siparişler için yeni bir ödeme hattı açılmasıyla sıklıkla karşılaşılır. Öncelikli hizmet disiplini, farklı hizmet gereksinimlerine sahip müşterilerin sınıflara ayrılarak müşteri sınıflarına öncelik ataması yapılmasını içermektedir. Önceliği yüksek olan alan müşteri sınıfının hizmet alması diğer sınıflardan önce gerçekleştirilir.

Müşteri önceliklerin kullanılması farklı müşteri tiplerini kapsayan bankacılık, telekomünikasyon, gıda, envanter ve sağlık gibi servis sistemlerinde eşit olmayan müşteriler için hizmet kalitesinin iyileştirilmesine olanak sağlar. Çoğu kuyruk sisteminde müşterilerin farklı hizmet gereksinimine göre sınıflandırılması ve bu sınıflara öncelikler atanması gerekebilir. Ancak, bir müşteri sınıfına keskin bir öncelik verildiğinde daha düşük öncelikli sınıf uzun bir süre hizmete ulaşamayacak ve bekleme süreleri uzayacaktır.

Bu tez çalışmasında öncelikli ve normal olmak üzere iki farklı müşteri tipini içeren M/M/1 kuyruğu ele alınmaktadır. İki müşteri tipi farklı bekleme maliyetlerine sahiptir. Öncelikli müşteri sınıfına ait bir müşteri kuyruğa geldiğinde ya geliş sırasına göre hizmet almakta ya

da kuyrukta bekleyen normal müşterileri atlayarak hizmet almaktadır. Tezde ele alınan problem Markov Karar Süreci (MDP) ile modellenmiş ve Simülasyon Tabanlı Optimizasyon yöntemlerinden olan ödüllü öğrenme tekniği kullanılarak yaklaşık optimal politika geliştirilmiştir. Amaç, birim zamandaki ortalama maliyeti en küçükleyen yaklaşık optimal kuyruk sıra politikasının belirlenmesidir.

1.1 Motivasyon

Hizmet sektöründeki işletmeler arasında önemli bir rol oynayan bankalar, kuyruk teorisine en çok konu olan sistemler olarak karşımıza çıkmaktadır. Sıra beklemenin söz konusu olduğu bu sistemler zaman içerisinde maliyetlerin azaltılmasının yanı sıra müşteri memnuniyetlerinin artırılmasına da odaklanmıştır. Müşteri memnuniyetini artırmanın bir yolu, müşterilerin kuyruklarda uzun süre beklemesinin önüne geçilerek en kısa sürede hizmet almalarını sağlamaktır. Bankalarda bu sebeple müşterilerin bekleme zamanının iyileştirilmesi için çeşitli çalışmalar ve stratejiler geliştirmektedir. Bu stratejiler, en kısa servis süresine sahip müşterinin önce hizmet alması, müşterinin yapacağı işlemlerin sistem tarafından daha kârlı olarak hesaplanması ve bu işlemlere öncelik verilmesi gibi durumlar olarak karşımıza çıkmaktadır.

Bankalarda olduğu gibi birçok hizmet sektöründe işlem görmek için sıra bekleyen müşteriler arasında yapılan müşteri önceliklendirme stratejileri, çeşitli sebeplerle yaygın bir şekilde kullanılmaktadır. Müşteriler farklı hizmet gereksinimleri dolayısıyla farklı müşteri sınıflarına ayrılmakta, bazı müşteriler öncelikli olarak nitelendirilmektedir. Özellikle hizmet sektöründe öncelikli müşteri grubu şirketin kârını maksimize etmek veya sistem kullanımını artırmak amacıyla belirlenebilir. Öncelikli müşteri statüsünde yer alan kişiler ya kuyrukta hiç beklemeden hizmet alır ya da kuyrukta bekleyen normal müşterilerin önüne geçerek ilk hizmet alacakların başında olur. Bu kuyruk stratejisi birçok hizmet sektöründe de karşımıza çıktığı gibi banka sistemlerinin kuyruklarında da sıklıkla uygulanmaktadır. Banka şubelerinde hizmet almak için gelen müşteriler banka girişlerinde bulunan numaratorlerden kimlik numarasını girip, yapacağı işlemi seçerek kuyruk numarası almaktadır. Ancak, kuyruk sırası müşterilerin kendi içerisinde önceliklendirilmesine göre değişmektedir. Aslında bu durum ile günlük hayatta sıklıkla karşılaşılmaktadır. Bir banka şubesine gidildiğinde servis birimine sizden sonra gelen müşterinin aynı birimde sizden önce hizmet alması gibi ya da gişe numaralarında çok daha

ileriki bir numaranın servise çağırıldığı gibi durumlar fark edilmektedir. Bu tip durumlar bankaların öncelikli müşteri ve kuyruk politikaları ile ilgilidir.

Bu çalışmanın motivasyonu, gerçek hayatta bir şekilde sürekli karşı karşıya kaldığımız banka, hastane, vb. hizmet sistemlerindeki öncelikli kuyruk sistemi uygulamalarıdır. Öncelikli müşterilere mutlak öncelik tanımak, daha düşük öncelikli müşterilerin bekleme sürelerini uzatmakta ve müşteri memnuniyetsizliğine sebep olmakta, hatta olası bir müşteri adayın kaybedilmesi neden olabilmektedir. Günümüzde artan nüfus miktarı ve azalan kaynaklarla birleşen rekabet ortamı, işletmeler ve hizmet sağlayıcılar için büyük bir zorluk oluşturmaktadır. Bu zorluklar, özellikle bekleme sürelerinin uzamasıyla daha da karmaşık hale gelmektedir. Çağımızın hızlı tempolu yaşam tarzına alışan bireyler, beklemeye tahammül konusunda hiç olmadığı kadar duyarlı hale gelmiştir. Bekleme sürelerinin uzaması müşteri memnuniyetsizliklerini, müşteri kayıplarını ve bekleme maliyetlerini artırmaktadır.

1.2 Literatür Taraması

Literatür taraması bölümünde tezin konu kapsamı dikkate alınarak öncelikli müşteri kuyrukları ve öncelikli müşteri sınıfları üzerine yapılan optimal kuyruk kontrol politikası çalışmaları incelenmiştir.

Birden fazla müşteri sınıfına ve bu müşteri sınıflarının farklı önem derecelerine sahip olduğu öncelikli kuyruk sistemleri konusunda geniş bir literatür bulunmaktadır. Literatüre bakıldığında kuyruk sistemlerinde müşterilere öncelik ataması yapılarak öncelik sınıflarına ayrılmasını sağlayan çalışmalar Cobham [1] ile başlamıştır. Ardından öncelikli sıraları konu alan başlıca klasik kitaplar [2, 3, 4] ortaya konmuştur. İlerleyen yıllarda [5, 6, 7]'deki çalışmalarla sınırlı önceliğe sahip M/M/1 kuyruk sistemi konu alınmış, performans analizi yapılmıştır.

Geleneksel, statik öncelik şemaları, önceliğin önceden atandığını ve sistemdeki kuyruk uzunluklarına bağlı olmadığını gösterir. Bu nedenle, bazen yüksek öncelikli müşterilerin kuyruğu kısayken, düşük öncelikli müşterilerin çok uzun kuyruğu olması mümkündür. Bu, düşük öncelikli müşteriler açısından adil olmayabilir [8]. Rykov ve arkadaşlarının [9] yaptığı çalışmadan bu yana literatürde sabit öncelik yerine, öncelikleri dinamik olarak sağlamaya yönelik farklı stratejiler önerilmiştir.

Kuyruk yönetimi için optimal politika bulma, önceliklendirilmiş sistemlerde özellikle ön plana çıkmaktadır ve bu konu hakkında dikkate değer derecede çok çalışma mevcuttur. Naor'un 1969'daki [10] çalışmasıyla günümüze kadar devam eden bir kuyruğa girişin optimal kontrolü üzerine önemli bir literatür koleksiyonu gözlenmektedir.

Kofman [11], öncelikli ve normal olarak adlandırılan iki tür müşteri içeren M/M/1 kuyruk problemini ele almıştır. Müşteriler servis tesisine iki bağımsız Poisson Sürecine göre gelirler ve geldikleri sıraya göre tek bir sıra oluştururlar. İki müşteri tipinde, kuyrukta bulunduğu birim zaman başına elde bulundurma maliyetleri farklıdır. Sunucu, müşterilere ya geldikleri sıraya göre hizmet verebilir ya da sabit bir R ücreti katlanarak öncelikli müşteriyi önündeki müşterilerin önüne atlatarak sıranın başına terfi ettirebilir. Sunucu, servise alacağı müşterileri seçerken sistemi çalıştırmanın birim zaman başına beklenen ortalama maliyetini en aza indirmeyi amaçlayarak, müşteri terfiinin optimal olduğu bir algoritma geliştirmiştir.

Rue [12], müşteri sınıfını M sayıda müşteri sınıfına genişletmişlerdir ve kuyruğa girişin optimal kontrolünü üzerine çalışmıştır. M tane farklı müşteri sınıfına sahip M/M/1 tipi kuyruk sisteminde her sınıf kendi ödül yapısına sahiptir. Farklı sınıflara ait müşteriler servis için R_m ödülü alırken, sistemde geçirilen zaman başına C_m ödeme yapılır. Bu kuyruk problemi için yarı-Markov karar süreci tekniği uygulanarak bireysel optimum ve sosyal optimum olmak üzere iki tür optimum politika ele alınmaktadır. Bireysel optimum politikada müşterilerin katılma veya vazgeçme kararları, varış anındaki bireysel müşterinin beklenen kazancına dayanmaktadır. Eğer müşterinin beklenen kazancı negatif değil ise müşteri kuyruğa katılır, aksi takdirde vazgeçer. Sosyal optimum politikada ise tüm müşteri sınıflarından gelen varışlar dikkate alınarak birim zamanda beklenen kazancın maksimize edilmesine yönelik katılma veya vazgeçme kararı verilir.

Siddharta ve arkadaşları [13] makalelerinde acil tıp hizmetlerindeki bekleme sürelerini azaltmak için bir öncelikli kuyruk sistemi sunar. Yazarlar, hastaların acil durumlarına göre öncelik vererek kaynaklarını bu şekilde dağıtmayı sunar. Hastaları aciliyet durumlarına göre farklı kategorilere ayırarak bir öncelik sistemi oluştururlar ve daha sonra her kategori için bir ilk gelen ilk çıkar prensibi önerirler. Öncelikli ve öncelikli olmayan prensibin bekleme süreleri karşılaştırıldığında, öncelikli kuyruk modelinde tüm hastalar için ortalama bekleme sürelerini azalttığını görmüşlerdir; ancak yüksek öncelikli hastalar için bekleme

süresi kısaldığı halde, düşük öncelikli hastalar daha uzun ortalama bekleme süresi ile karşı karşıya kalmaktadırlar. Model, bekleme sürelerini minimize etmeyi, hasta sonuçlarını iyileştirmeyi ve acil tıp sistemindeki verimliliği artırmayı amaçlamıştır.

Brouns ve Van Der Wal [14] makalelerinde sınırlı öncelikli M/M/1 kuyruk modelini ele almışlardır. Elde bulundurma ve değiştirme maliyetlerine sahip iki tip müşteri sınıfının kuyruğa kabulü ve servisteki işleminin sonlandırılması kararı için optimum politika oluşturulması ile ilgili çalışma yapmışlardır. Elde tutma ve değiştirme maliyetlerini en az indirecek şekilde dinamik programlama kullanılarak, bu iki kararın yeni işleri kabul etme veya reddetme ve bir işi kaldırma veya sürdürmenin mevcut olduğu temel bir model geliştirilmiştir.

Cao ve Xie'nin [15] çalışmasında çok sınıflı bir kuyruk sisteminin optimal kontrolünü incelenmiştir. İki müşteri tipi, düşük öncelikli ve yüksek öncelikli, farklı Poisson sürecine göre gelir ve servis süreleri üstel olarak dağılır. Düşük öncelikli müşteriler üstel olarak dağıtılmış bir oran sonunda yüksek öncelikli müşterilere dönüşebileceği durumu ele alarak, kuyrukta bekleyen müşteri sayısı ve farklı müşteri tiplerinin hizmet edildiği oranları en iyi şekilde yönetmeyi amaçlar. İki müşteri tipinin tutma maliyeti ve düşük öncelikliden yüksek önceliğe tip değiştirme maliyeti bulunmaktadır. Sistem sürekli zaman Markov Karar Süreci olarak modellenmiştir. Çalışma, uzun vadeli ortalama maliyeti en aza indirecek optimal kontrol politikasını belirlemek için bir dinamik programlama modeli önerir. Sonuçlar, müşteri bekleme süreleri ve hizmet edilen müşteri tiplerinin oranlarının optimize edilebileceğini gösterir.

Xie ve arkadaşlarının [16] ele aldıkları problem, Cao ve Xie'nin [15] çalışmasına benzemektedir. Cao ve Xie'den [15] farklı olarak Xie ve arkadaşlarının [16] modelinde iki farklı transfer maliyeti vardır. Bu maliyetler; düşük öncelikli müşteri kuyruğundan yüksek öncelikli müşteri kuyruğuna transfer maliyeti ve yüksek öncelikli müşteri kuyruğundan başka bir işleme transfer maliyetidir. Transferler üstel olarak dağıtılmış θ_1 ve θ_2 parametresi ile gösterilen rastgele zamandan sonra gerçekleşir. Bir başka fark ise bir müşteri düşük öncelikli kuyruktan yüksek öncelikli kuyruğa transfer edilirse servis süresi yeniden belirlenerek değişmektedir. Sistem sürekli zaman Markov Karar Süreci olarak modellenmiştir. Amaç, servis kontrol probleminde ortalama maliyet oranını en aza indirecek en uygun politikayı bulmaktır.

Down ve arkadaşları [17], iki sınıflı bir kuyruk sisteminde müşteri vazgeçmelerinin olduğu kabul edilerek, dinamik servis kontrolü tartışmıştır. Çalışmada iki farklı durum karşılaştırılmıştır. Birinci durum, servisi tamamlanan her müşteri için sınıfa bağlı ödülün olduğu sistem modelidir; ikinci durum, her sınıfın kuyrukta beklediği birim zaman başına elde bulundurma maliyetinin ve ayrılan her müşteri için sınıfa bağlı cezanın olduğu sistem modelidir. Her durumda, sonsuz bir ufukta ortalama ödülleri maksimuma çıkarma veya ortalama maliyetleri minimuma indirme sorunu ele alınmıştır. Sistem sürekli zaman Markov Karar Süreciyle formüle edilmiştir. Sistemde iki sınıfın olduğu, ancak gelecekteki çalışmalarda üç sınıfa genişletilmesi durumunda MDP formülasyonun işlenmesinin daha zor hale geleceğinden bahsedilmiştir.

Bispo [18], çok müşteri sınıflı ve tek servisli sistemin elde bulundurma maliyeti için sınıfların kuyrukta bekleyen müşteri sayısı ile doğru orantılı dışbükey maliyet fonksiyonu gösterilmiştir. Sistemin maliyetinin minimuma indirilmesi için servisin hangi sınıftan müşteriyi servis birimine alacağı Markov Karar Süreci olarak modellenmiş ve dinamik programlama yardımıyla çözülmüştür. Servis kontrol politikası olan servis çizelgelemesine alternatif çözüm geliştirilmiştir.

Groenevelt ve arkadaşlarının [19] yazdığı makalede tutma ve değiştirme maliyetleriyle birlikte iki sınıflı öncelik kuyruğunda uzun dönemli ortalama maliyetin en aza indirilmesini incelemişlerdir. Optimal servis kontrolü için diğer çalışmalardan farklı olarak sapma vektörü kullanılmış ve politika yineleme algoritması uygulanmıştır. Politika yineleme algoritmasının uygulanması sonucu çıkan politikanın optimal politikaya yakın olduğu gösterilmiştir.

Klimenok ve arkadaşları [20], yayımlanan makalede iki müşteri tipinin önceliğinin zamanla dinamik olarak değiştiği durumda sistemin davranışını analiz etmiştir. Markov geliş süreci ile işaretlenmiş bir partiler halinde iki tür müşterinin geldiği (batch marked Markov arrival process) sonlu bir arabelleğe sahip tek servisli ve ortak bir kuyruk sistemi kurulmuştur. 1. tip müşteriler 2. tip müşterilere göre sınırlı önceliğe sahiptir. Düşük öncelikli müşteriler, PH dağılımına sahip rastgele bir süreden sonra yüksek öncelik alabilirler. Ara belleğe alınan her öncelikli olmayan müşteri için, faz tipi dağılımına sahip rastgele bir zamanı sayan bir zamanlayıcı açılır. Zamanlayıcı sona erdiğinde, belirli bir olasılıkla müşteri sisteme bırakılır ve tamamlayıcı olasılıkla yüksek öncelik kazanır.

Matematiksel modelleme ve benzetim kullanarak deęişen önceliklerin, kuyrukta bekleyen müşteri sayısı, ortalama bekleme süresi gibi sistem performansı üzerindeki etkisini incelemişlerdir. Önceliklerin dinamik olarak deęişimi sistem performansı üzerinde önemli bir etkiye sahip olduęu ve müşteri tiplerinin öncelięinin en iyi performansı sağlamak için dikkatli bir şekilde yönetilmesi gerektięi bulunmuştur. Bu kuyruk sistemi ile bir hastanedeki acil servis, iletişim merkezi, çabuk bozulan gıdaların bulunduęu envanter vb. işlerin modellenmesinde yardımcı olabilmesi amaçlanmıştır.

Çelik [21], tez çalışmasında öncelikli kuyruk disiplini ile hizmet veren kuyruk sistemleri ele alınmış olup, özel olarak sınırsız öncelikli ve sınırlı öncelikli kuyruk sistemlerinin analizi yapılmıştır. Sınırlı öncelikli ve sınırsız öncelikli şartlar altında M/M/1 kuyruk sisteminin sayısal bir uygulaması yapılmıştır. Bu çalışmada iki sınıflı öncelikli bir kuyruk sistemi analiz edilmiş ve Little formülü kullanılarak, sistemdeki ortalama müşteri sayısı ve sistemdeki ortalama bekleme süresi gibi performans deęerleri bulunmuştur. Ayrıca bu kuyruk sisteminin bir simülasyonu hazırlanmıştır. Yapılan simülasyon sonucu elde edilen performans ölçümleri araştırılmıştır.

Gong ve Rajan [22] çalışmasında iki farklı öncelik sınıfının ve tek servis sağlayıcının bulunduęu sınırsız öncelikli bir M/M/1 kuyruk modelini, bir afet meydana geldikten sonra afetzedelerinin nasıl kurtarılacağını belirlemek amacıyla bir kuyruk uzunluęu kesme yöntemini dikkate almışlardır. Bu disiplin ile ilgilenilen sınıfın kuyruk uzunluęu belirlenen L eşik deęerini aşana kadar servis sağlayıcı sadece yüksek öncelikli sınıfa hizmet verir. Eşik deęeri aşıldıktan sonra sistem düşük öncelikli kuyruęa hizmet etmek için geçiş yapar ve düşük öncelikli sınıfın kuyruk uzunluęu belirlenen eşik deęerinin altına düşünceye kadar düşük öncelikli hastalara servis sağlanır. Bu sistem için her öncelik sınıfının ortalama müşteri sayısını bulunabilmesi için iki boyutlu türeten fonksiyonlar elde etmişlerdir. Son olarak kesme politikasında (cut off) kullanmaya yarayan eşik deęeri için en uygun deęeri belirlemek amacıyla iki öncelik sınıfı kuyruęuyla ilişkili ağırlıklar kullanarak metodolojiler geliştirmişlerdir.

Paköz [23] çalışmasında k sunuculu sınırlı öncelikli kuyruk için kuyruk kontrol eden bir karar mercini ele almıştır. Sisteme gelen müşteriler p olasılıkla yüksek öncelikli ya da düşük öncelikli olarak tanımlanmıştır. Karar mercinin görevi, bekleyen müşterilerin ne zaman hizmet alacağıdır. Bunu yaparken kuyruk uzunluęu sabit tutulurken yüksek

öncelikli müşterilerin bekleme sürelerini en aza indirmeyi amaçlamıştır. Optimal politika belirlemek için doğrusal yaklaşım oluşturulmuştur.

Bai ve ark. [24] bu çalışmada, ağ sistemleri üzerinde ortalama iş gecikmesinin (veya eşdeğer olarak ortalama kuyruk birikiminin) en aza indirilmesi amacıyla kuyruk ağlarının optimal kontrol politikasını öğrenmek için model tabanlı takviyeli öğrenmeyi (RL) kullanmışlardır. Markov karar süreci (MDP) çerçevesi tarafından formüle edilebilecek genel bir ayrık zamanlı kuyruk ağı sisteminin ortalama kuyruk biriktirme listesini optimize etmeyi hedeflenmiştir.

Bedford ve ark. [25] birden fazla müşteri sınıfı için tekli kuyruk sistemlerinden farklı olarak birincil ve ikincil olmak üzere ikili sonlu kuyruk yapısı önermişlerdir. İki tür müşteri sınıfı olup sınıf 1 olarak tanımlanan müşteriler, sınıf 2 olarak tanımlanan müşterilerden önceliklidir. Önerilen ikili kuyruk yapısında gelen bir müşteri birincil kuyruğun dolu bulunduğu, herhangi bir hizmet olanağı bulunmayan ve yalnızca aşırı yüklenen birincil kuyruk için bekleme kuyruğu görevi gören ikincil kuyrukta (dolu olmadığı sürece) bekler. Her iki kuyruğun da dolu olması durumunda gelen müşteri sistem tarafından kaybedilir. İkincil kuyrukta ise müşteriler, sınıf 1 müşteriler sınıf 2 müşterilerin önünde olacak şekilde öncelik sırasına göre sıralanırlar. Birincil kuyrukta bir yer boşaldığında, ikincil kuyruktaki satırın başı birincil kuyruğa katılır. Yinelemeli ve temel matris tekniklerinin bir kombinasyonunu kullanan bir hesaplama algoritması kullanılmıştır ancak algoritmanın sürecin çeşitli aşamalarında karmaşık olduğu vurgulanmıştır. Karmaşık matris işlem yeteneğine sahip bilgisayar kodları kullanılarak çözülebileceğine dair çerçeve sunulmuştur.

Charles ve ark. [26] iki müşteri tipi olan ve bağımsız Poisson süreçlere gelen tek sunuculu sistem üzerinde çalışmış ve öncelik sırası kuyruk uzunluğu eşiği (QLT, Queue Length Threshold) çizelgelemesi tarafından belirlenmiştir. Bu kuyruk sistemine, QLT zamanlamasına sahip dinamik (veya hibrit) öncelik kuyruğu adı verilir. Müşteriler iki tipe (tip-1 ve tip-2) sınıflandırılır ve sınıfların hizmet sırası tip-1 kuyruğun kuyruk uzunluğuna bağlıdır. Servis önceliği yüksek öncelikli sınıfta olup ilk olarak yüksek öncelik sınıfa hizmet verilir. Düşük öncelikli sınıfın kuyruk uzunluğu L eşliğini aşarsa kuyruk uzunluğu L eşğine düşene kadar düşük öncelikli sınıfa hizmet verilir ardından tekrar yüksek öncelikli sınıfa hizmet sunulmaya devam edilir. Çalışmada iki müşteri sınıfının kuyruk uzunlukları

için ortak olasılık üretme fonksiyonunu elde edilmiştir. Bu, ortak kuyruk uzunluğu dağılımını bir integral olarak verilmiş ve daha sonra ortak dağılım için ayrıntılı asimptotik sonuçlar elde edilmiştir.

Li ve ark. [27] çalışmalarında sağlık sistemlerinde kaynak çizelgelemesi yapmak için kuyruk sisteminden yararlanmışlardır. Çoklu tipte hasta ve birden fazla sunucu (doktor) içeren randevu kuyruğunda, belirlenen maliyetleri (bekleme süresi maliyeti, tıbbi maliyet ve ceza maliyeti gibi) en aza indirmek için Markov karar modeli geliştirilmiştir. Üç aşamalı karar verme modeli içeren problemin yaklaşık optimal çözümünü elde edebilmek için geliştirilmiş Q-öğrenme algoritması kullanılmıştır. Çalışmanın sonucunda Q-öğrenme tabanlı çizelgeleme algoritmasının iki geleneksel çizelgeleme algoritmasından (ilk gelen ilk servis görür ve sabit öncelik politikası) önemli ölçüde daha iyi performans gösterdiği verilmiştir.

Li ve arkadaşları [28], son 60 yılda kuyruklarda ve ağlarda Markov Karar Süreçleri üzerine yapılan araştırmalara genel bir bakış sağlamaktır. Farklı kuyruk kontrol problemleri özelinde öncelikli kuyruk konusunda daha fazla çalışma için Li ve arkadaşlarına [28] bakınız. Literatür taramasına ilişkin özet, Tablo 1.1’de verilmiştir.

Tablo 1.1: Literatür taraması özet tablosu.

Çalışma	Konu	Sınıf sayısı	Kuyruk gösterimi	Yöntem	Amaç
[7]	Öncelikli ve önceliksiz müşteri sınıfları kuyruk uzunluklarının incelenmesi	2	M/M/1	Markov Vektörü	Öncelikli ve önceliksiz müşteri sınıfları performans analizi
[11]	Farklı bekleme maliyetli ve servis ödüllü kuyruk sisteminde optimal kontrol	2	M/M/1	MDP	Bekleme maliyetini minimize eden optimal kuyruk kontrolünün belirlenmesi

Tablo 1.1 (devam)

Çalışma	Konu	Sınıf sayısı	Kuyruk gösterimi	Yöntem	Amaç
[12]	Bekleme maliyetli ve servis tamamlanma ödüllü N tane farklı sınıf için bireysel ve sosyal optimum kontrol politikası	N	M/M/1	SMDP	Birim zamanda beklenen kazancın en büyüklüğüne yönelik kontrol politikasının belirlenmesi
[13]	Acil tıp departmanındaki hastaların önceliklendirilmesi	N	M/M/1	MDP	Öncelikli ve öncelikli olmayan hastaların ortalama bekleme sürelerinin minimizasyonu
[14]	Sınırsız öncelikli iki sınıf için kuyruğa işlerin kabul/reddedilmesi	2	M/M/1	Dinamik Programlama	Beklenen elde bulundurma ve değiştirme maliyetlerinin toplamını minimize edecek şekilde sınıflar arasında geçiş politikasının belirlenmesi
[15]	Farklı maliyetlere sahip iki müşteri tipinin üstel bir süre sonra tip 1 müşterinin, tip 2 müşteriye dönüşmesi	2	M/M/1	CTMD	Ortalama maliyeti minimize eden en uygun planlama kuralının belirlenmesi
[16]	Acil serviste durumu kritik ve kritik olmayan iki sınıflı kuyrukta, kritik olmayan hastanın belirli bir süre sonra durumu kritik hale geleceği baz alınıp müşteri transferi yapılması	2	M/M/1	MDP	Müşteri elde bulundurma maliyetleri ve değiştirme ücretlerinin toplamını minimize eden en uygun politikanın belirlenmesi

Tablo 1.1 (devam)

Çalışma	Konu	Sınıf sayısı	Kuyruk gösterimi	Yöntem	Amaç
[17]	Başarıyla tamamlanan her iş için ödül, doğrusal elde bulundurma maliyeti ve ayrılan her müşteri için ceza yapısının incelenmesi	2	M/M/1	CTMDP	Beklenen ödülü maksimizasyonu, ortalama maliyetlerin minimizasyonu ve en uygun sunucu atama politikasının belirlenmesi
[18]	Çok sınıflı kuyrukta, bekleyen müşteri sayısı ile doğru orantılı dışbükey elde bulundurma maliyet fonksiyonu kullanılması	M	M/M/1	Dinamik Programlama	Minimum maliyet için servis kontrol politikası belirlenmesi
[19]	İki müşteri sınıflı kuyruk sisteminde elde bulundurma ve servis verilecek sınıfı değiştirme maliyeti incelenmesi	2	M/M/1	Politika yenileme	Minimum maliyet için servis kontrol politikası belirlenmesi
[20]	Kuyruk uzunluğu dikkate alınarak dinamik önceliklendirme	2	BMM/PH/1	Markov Chain	Değişken önceliğin performans çıktıları üzerindeki etkisinin incelenmesi
[21]	Sınırlı öncelik ve Sınırsız öncelik kuyrukların karşılaştırılması	2	M/M/1	Little Yasası	Performans ölçümleri karşılaştırılması
[22]	Düşük öncelikli sınıfın kuyruk uzunluğu L eşiğini geçene kadar yalnızca yüksek öncelikli sınıfa hizmet verilmesi	2	M/M/1	Cutoff	Kesme politikası için en uygun eşik değerinin belirlenmesi
[23]	Yüksek ve düşük öncelikli k sunuculu bir sistemde hangi sınıfın hizmete alınacağına kararının verilmesi	N	M/M/k	Linear Program	Sabit kuyruk uzunluğu ile öncelikli müşterilerin bekleme sürelerini minimizasyonu

Tablo 1.1 (devam)

Çalışma	Konu	Sınıf sayısı	Kuyruk gösterimi	Yöntem	Amaç
[24]	İki sınıfa sahip ağ sistemlerinde ortalama kuyruk birikiminin en aza indirilmesi	2	M/GE/1	Ödüllü Öğrenme	İş gecikmesini etkili şekilde minimize eden optimal kuyruk kontrol politikasının belirlenmesi
[25]	Birden fazla müşteri sınıfı için tekli kuyruk yerine birinci ve ikincil ikili sonlu kuyruk yapısı	2	M1+M2 /M/1/ci	Yinelemeli ve Temel Matris	Sonlu bir tampona sahip tüm kuyruklar için kararlı durum çözümü bulunması
[26]	Öncelik sırası Kuyruk uzunluğu eşiği çizelgelemesiyle öncelik sırasının belirlenmesi	2	M/M/1	Kuyruk uzunluğu eşiği çizelgelemesi (QLT)	İki müşteri sınıfının kuyruk uzunlukları için ortak kuyruk uzunluğu dağılımının belirlenmesi
[27]	Hasta ve doktor içeren randevu kuyruğunda üç aşamalı karar verme modeli oluşturulması	N	M/M/N	Q-öğrenme	Bekleme maliyeti, tıbbi maliyet ve ceza maliyetinin minimizasyonu

Literatür taraması incelendiğinde iki ya da daha fazla farklı öncelik sınıflarına sahip önceliklendirilmiş kuyruk problemleri ile ilgili birçok çalışma olduğu görülmektedir. İlk çalışmalar 1955’li yıllarda olmasına karşın, konu halen güncelliğini korumaktadır. Yapılan ilk çalışmalar müşterilere öncelik ataması yapılarak sınıflara ayrılması hakkındadır [1]. Ardından öncelikli sınıflara sahip kuyruk sistemleri kuyruk sistemlerinin yoğun olarak çalışıldığı bir alan haline gelmiştir. M/M/1 öncelik kuyruğu olarak bilinen, üstel dağılımlı hizmete sahip tek sunuculu Poisson varışlı model, öncelikli kuyruklar arasında özellikle önemlidir. M/M/1 öncelikli kuyruk, sağlık hizmetleri, bilgisayar sistemleri gibi çok çeşitli uygulamalarda yaygın olarak kullanılmaktadır. Birden fazla öncelik sınıfı içeren kuyruk sistemlerinde servis biriminin hangi sınıfa hizmet edeceği sorusu doğduğu için optimal kuyruk kontrol politikasını hakkında çalışmalar yapılmıştır ve hale yapılmaya devam edilmektedir. Optimal kuyruk kontrol politikası bulmak için yapılan çalışmalarda MDP

yöntemi sıklıkla kullanılmıştır [11, 16]. Kullanılan diğer çözüm yöntemleri SMDP, CTMDP, dinamik programlama ve politika yineleme algoritmasıdır [12, 15, 18, 19].

Tezde ele alınan problem iki öncelikli müşteri sınıfı ve tek servise sahip olan M/M/1 kuyruk sistemi olması yönünden Kofman [11], Brouns vd. [14], Cao vd. [15] ve Down vd [17]'in çalışmalarına benzerdir. Literatürdeki çalışmalara benzer olarak her sınıf birim zaman başına farklı elde bulundurma maliyetine sahiptir. Tezde ele alınan problem literatürdeki çoğu çalışma gibi Markov Karar Süreci olarak modellenecektir ancak literatürden farklı olarak çözüm yöntemi olarak simülasyon tabanlı optimizasyon tekniği kullanılacaktır. Çözüm tekniği ile Li ve arkadaşlarının [27] yaptığı çalışma benzemekte olup Q-öğrenme kullanılmıştır, ancak [27]'de yapılan çalışmalardan farklı olarak üçlü karar aşaması kullanılmıştır.

Literatür taraması incelendiğinde ele alınan çalışma yıllardır yoğun bir şekilde çalışılan bir konu olup farklı çözüm teknikleri kullanılmaktadır. Son yıllarda hızlı bir şekilde hayatımıza giren yapay zekâ teknikleri bu konuda da çözüm yöntemi olarak optimal sonuçlara neredeyse yakın sonuçlar verdiği ve karmaşık matematiksel işlemleri gerektirmediği için tercih edilmeye başlanmıştır, ancak kısıtlı sayıda çalışma mevcuttur. Bu bağlamda ele alınan problem iki müşteri sınıflı, tek sunuculu M/M/1 öncelik kuyruk sisteminin optimal politika belirleme için kullandığı çözüm yöntemi yönüyle literatürde az sayıda olan çalışmalardan biridir.

2. KUYRUK SİSTEMLERİ

Kuyrukların veya bekleme hatlarının oluşması, bir hizmete yönelik mevcut talebin, o hizmeti sağlamaya yönelik mevcut kapasiteyi aştığı durumlarda ortaya çıkan yaygın bir olgudur. Bir müşteri hizmeti hemen alamıyor, ancak beklemek zorunda kalıyor hizmet tesisleri boшта duruyor veya hizmete ihtiyaç duyan müşteri sayısı hizmet tesislerinin toplam sayısını aşıyorsa, bekleme kuyrukları oluşmaktadır. Bekleme kuyrukları ile hem günlük yaşamımızda hem de çeşitli iş ve endüstriyel durumlarda sık karşılaşılmaktadır [29].

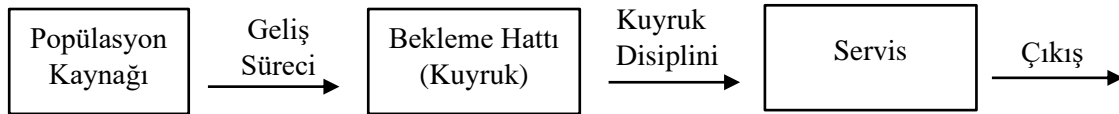
Kuyruk teorisi, rastgele ortaya çıkan taleplere hizmet sağlamaya çalışan sistemlerin davranışını tahmin etmeye yönelik modeller sağlamak için geliştirilmiştir. Bu konuda ilk çalışmaları 1909'da telefon çağrı merkezleri hakkında çalışma yayınlayan Danimarkalı matematikçi A. K. Erlang yapmıştır. Teorinin telefona uygulanmasına yönelik çalışmalar Erlang'dan sonra da devam etmiştir. Kuyruk sisteminin teorik analiz çalışmaları, 1940'ların sonlarında ve 1950'lerin başlarında Yöneylem Araştırması bilim dalının ortaya çıkmasıyla önemli ölçüde artmıştır. 1950 yıllarına gelindiğinde kuyruk teorisi ile ilgili yapılan çalışmalar hızlanmıştır [30]. Günümüze geldiğimizde ise Yöneylem Araştırması kapsamı içinde kuyruk teorisi ve kuyruk sistemleri hakkında çok fazla kitap, makale ve çalışma yapılmıştır ve yapılmaya devam etmektedir.

Kuyruk sistemi, müşterilerin hizmet için gelmeleri, hizmeti beklemeleri, hizmeti beklemişlerse hizmet alındıktan sonra sistemden ayrılmaları olarak tanımlanabilir. "Müşteri" terimi genel anlamda kullanılmaktadır ve mutlaka bir insan müşteriye ima etmemektedir. Örneğin müşteri, cıalanmayı bekleyen bir bilye, havalanmak için sırada bekleyen bir uçak veya çalıştırılmayı bekleyen bir bilgisayar programı olabilir [30]. Bir kuyruk sistemi, hizmet bekleyen müşteriler için sunucuları, müşterileri ve kuyrukları içerir.

Kuyrukta bekleyen veya hizmeti alan kişiye müşteri, müşteriye hizmet veren kişiye ise sunucu adı verilir. Müşteriler belirli bir olasılık dağılımına göre gelir ve gelen müşterilere belirli bir prensibe göre bir veya birden fazla sunucu tarafından hizmet verilir. Servis süreleri belirli bir olasılık dağılımına göre yönetilen rastgele değişkenlerdir. Hizmet aldıktan sonra müşteri sistemden ayrılır. Kuyruk modellerinin amacı; sunucu sayısı, hizmet hızı ve sipariş sırası gibi belirli faktörleri değiştirerek toplam hizmet maliyetini ve

müşterilerin bekleme süresini en aza indirecek şekilde hizmetin müşterilere nasıl sağlanacağını belirlemektir [29].

Sistemde hizmet tesisinin aynı anda hizmet edebileceğinden daha fazla müşteri olduğunda, bir kuyruk (bekleme hattı) gelişir. Bekleyen müşteriler önceden belirlenmiş bir kurala göre servis sırasını alırlar ve servis sonrası sistemden ayrılırlar. Böylece sistemin girdisi hizmet talep eden müşterilerden, çıktısı ise hizmet verilen müşterilerden oluşmaktadır. Kuyruk sistemlerinin temel yapısı Şekil 3.1’de gösterilmiştir [29].



Şekil 2.1: Temel bir kuyruk sistemi yapısı [29].

2.1 Kuyruk Sistemleri Bileşenleri

Sistemin yapısı incelendiğinde genel olarak kuyruk sistemleri bileşenleri aşağıda yer almaktadır [29];

- Müşteri popülasyonunun kaynağı,
- Geliş süreci,
- Kuyruk,
- Kuyruk disiplini,
- Sistem kapasitesi.

Müşteri Popülasyonunun Kaynağı: Sisteme gelen birimlerin kaynağını ifade etmektedir. Her kuyruk sistemi için popülasyon kaynağının büyüklüğü değişmektedir. Kaynaktaki birimlerin sayısının sonlu ya da sonsuz olmasına bağlı olarak kaynak sonlu ya da sonsuz olarak adlandırılır. Kaynak sonlu ancak büyükse, genellikle sonsuz olarak nitelendirilir. Sonsuz kaynak popülasyonu çok sayıda olası müşteriye sahip teorik bir sistem modelini temsil eder. Örneğin bir çağrı merkezini arayabilecek müşteri kaynağı sonsuz olarak alınabilir.

Geliş Süreci: Hizmet için gelen müşterilerin geliş akışı anlamındadır. Müşteri geliş süreci veya varış süreci olarak adlandırılır. Genellikle gelişler arası sürelerin bağımsız ve ortak

bir dağılıma sahip olduğu varsayılır. Müşteri genel olarak sistemde hizmet için bekleyen birim olarak tanımlanır. Müşteriler sisteme tek tek veya gruplar halinde gelebilir [31].

Kuyruk: Sisteme hizmet ya da iş almak için gelen müşteriler, sistemin kapasitesinin talebi karşılayamaması sonucu beklemek zorunda kalırlar. Bu bekleme nedeniyle bir sıra oluşturulur ve bu sıraya kuyruk, diğer bir söyleyiş ile bekleme hattı denir.

Kuyruk Disiplini: Kuyruk disiplini, kaynaktan gelen müşterilerin hangi kurallara göre seçileceği ve servise alınacağını belirlediği yöntemdir. Servis disiplini olarak da karşımıza çıkar ve kuyruk sistemlerindeki önemli bileşenlerinden biridir. Kuyruk disiplini, mevcut müşterinin hizmeti tamamlandığında eğer kuyrukta bekleyen başka müşteriler varsa, onlardan hangisinin sıradaki hizmeti alacağını seçmek için kullanılır. Hizmet için gelen bir müşterinin veya müşterilerin nasıl seçileceğini belirleyen birçok kural vardır. Bu disiplinlerin seçimi, kuyruk sisteminin amacına, iş yüküne, müşteri beklentilerine ve kaynakların (kapasitesi, işlem hızı vb.) durumuna bağlı olarak yapılır. Kuyruk disiplinin seçimi bekleme maliyetlerini doğrudan etkileyen bir faktördür. Birçok kuyruk disiplininden söz edilebilir. Bu disiplinlerden en çok karşımıza çıkan şunlardır [10, 32]:

(i) İlk Gelen Hizmeti İlk Alır (First In First Out-FIFO): Bu disipline göre müşterilere geliş sırasına göre hizmet verilir. İlk gelen ilk çıkar olarak da bilinir.

(ii) Son Gelen Hizmeti İlk Alır (Last In First Out-LIFO): En son gelen birimlerin ilk hizmet edildiği disiplindir.

(iii) Gelişi Güzel Sırayla Hizmet (Service In Random Order-SIRO): Müşteriler sisteme gelişlerine bakılmaksızın rastgele hizmet verilir.

(iv) Dönen Tur (Round Robin-RR): Müşteri, servis birimi tarafından önceden tanımlanmış hizmet periyodu kadar hizmet alır. Bu periyot sürecince hizmeti tamamlanırsa sistemden çıkış yapar ancak bu periyot süresince servis tamamlanmazsa servis biriminden çıkarak kuyruğa geri döner. Müşteri kalan hizmet süresi için servis birimine alınmayı bekler.

(v) Öncelikli Disiplin (Priority Scheduling): Bazı müşterilere varış sıraları dikkate alınmaksızın diğer müşteri sınıflarından önce hizmet verilir; yani bazı müşterilere öncelikli olarak hizmet verilmektedir [29]. Gelen müşteriler belirli kriterlere göre öncelik sınıflarına ayrılır. Daha sonra öncelik sıralarına göre en yüksek öncelikli ilk servis alacak şekilde

disiplin uygulanır. Birçok kuruluştta müşterilere hizmet verme sırası müşterinin "tipine" bağlıdır. Örneğin hastanelerin acil servisleri, acil olmayan hastalara hizmet vermeden önce genellikle ciddi hastalara hizmet verir. Ayrıca birçok bilgisayar sisteminde daha uzun işler, kuyruktaki tüm kısa işler tamamlanana kadar hizmete girmez. Müşteri türünün, müşterilerin hizmet alacağı sırayı belirlediği modellere öncelikli kuyruk modelleri denir [33]. Öncelikli kuyruk sistemleri bölüm 2.2'de detaylandırılmıştır.

Sistem Kapasitesi: Bazı durumlarda kuyruk sistemi aynı anda gerekli sayıda müşteriyi barındıramayabilir. Yeni müşterilerin konaklaması için yer müsait hale gelinceye kadar başka müşterinin sisteme girmesine izin verilmez. Bu tür bir duruma sonlu kaynak kuyruğu denir [29].

Yukarıda tanımlanan bileşenler kuyruk sistemlerinin çoğunda farklı biçimlerde yer almaktadır. Kuyruk sistemlerini ayırt etmek için Kendall tarafından tanımlanan aşağıdaki simgeleme biçimi önerilmiştir [30].

Bu simgeleme sistemini şu şekilde gösterilir:

$a / b / c / d / e / f$

a: Gelişler arası sürelerin olasılık fonksiyonu,

b: Servis sürelerinin olasılık fonksiyonu,

c: Sunucu sayısı,

d: Kuyruk disiplini,

e: Sunucu kapasitesi,

f: Müşteri popülasyon kaynağının büyüklüğü

Bu simgeleme biçiminde ilk gösterim olan a, müşteri gelişlerinin dağılımını; ikinci gösterim olan b ise servis sürelerinin dağılımını ifade etmektedir ve aşağıdaki semboller geliş ve servis sürelerinin dağılımları için kullanılan kısaltmaları göstermektedir:

M: Poisson/üstel dağılım,

D: Servis süreleri sabit ve deterministiktir,

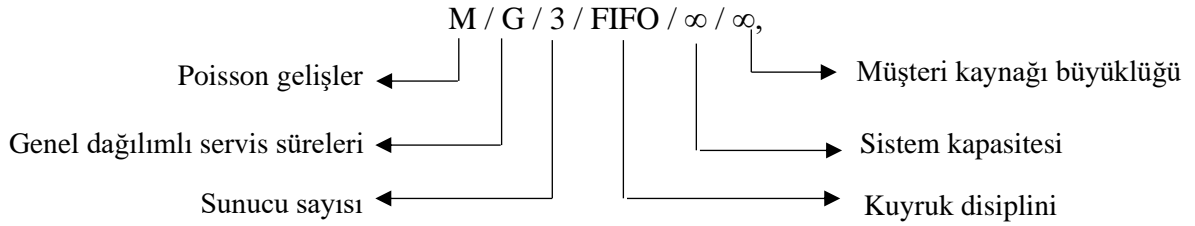
Ek: Erlang dağılım,

G: Genel dağılım,

N: Normal dağılım.

Üçüncü gösterim olan c , sunucu sayısını ifade eder. Dördüncü gösterim d , kuyruk disiplinini ifade eder ve FIFO, LIFO, SIRO ve PRIO gibi kuyruk disiplinini tanımlayan kısaltmalar kullanılır. Beşinci gösterim olan e , sistemde izin verilen en çok müşteri sayısını ifade edilir. Altıncı gösterim f , müşterilerin çekildiği popülasyonun büyüklüğünü verir. Potansiyel müşteri sayısı sunucu sayısı ile aynı büyüklükte olmadığı sürece popülasyon büyüklüğünün sonsuz olduğu kabul edilir. Birçok kuyruk sisteminde $d/e/f$ çoğunlukla FCFS/ ∞/∞ 'dur. Durum böyleyse $d/e/f$ kuyruk sistemi gösteriminde sıklıkla atlanır. Kuyruk sistemleri ilk 3 gösterimle ifade edilir: $M/G/3$, $G/Ek/1$ vb.

Bu gösterimin bir örneği olarak aşağıdaki şekil gibi açıklayabilir.



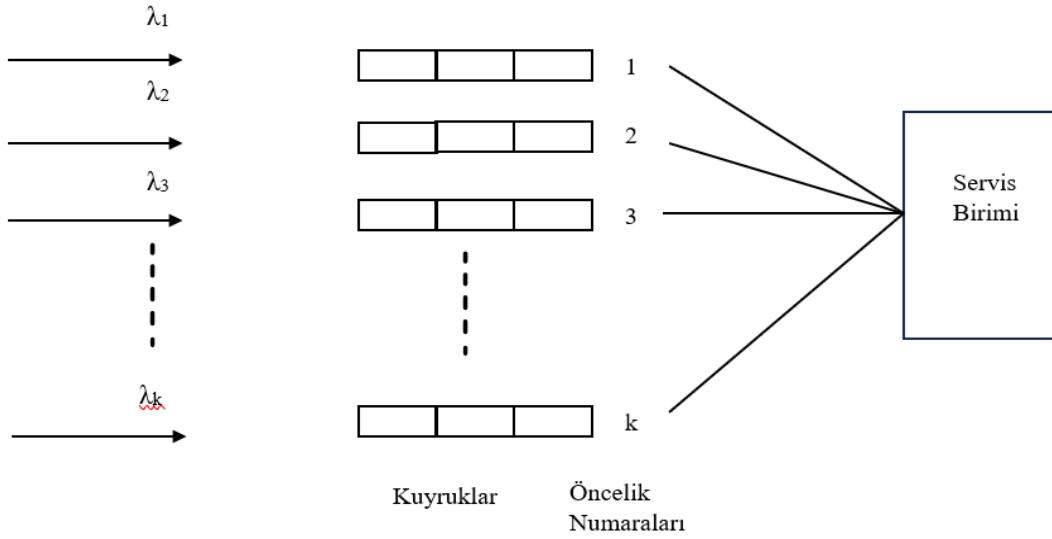
Şekil 2.2: Kuyruk sistemi gösterimi [29].

2.2 Öncelikli Kuyruk Sistemleri

Öncelikli kuyruk sistemleri, kuyruk sistemleri içerisinde önemli bir yer kaplamaktadırlar. Öncelikli kuyruk disiplini içeren sistemlerde farklı önem kategorilerine sahip müşteri sınıfları için öncelik sıraları atanır ve sistemden alacakları hizmet bu öncelik sırasına göre gerçekleştirilir. Son yıllarda öncelikli kuyruk disiplininin hizmet sektöründe yoğun bir şekilde kullanıldığı görülmektedir. Örneğin; konserlerde VIP biletleri olan müşteriler normal sıraları atlayabilir, hastane acil servislerinde durumu daha acil olan hastalar diğer hastalara göre önce tedavi alır ya da bir uygulamanın ücretli aboneliğini satın alan müşteriler ücretsiz sürümüne sahip olanlardan daha fazla iletişim hızına sahip olabilir [34].

Kuyruk sistemlerinde müşteri önceliklendirme birçok farklı sebep için yapılabilmektedir. Önceliklendirmenin üç ana motivasyonu vardır. İlk motivasyon, farklı müşterilerin aynı ürün için farklı ödeme istekliliğine (veya değerlemelerine) sahip olabilmesidir. İkinci motivasyon, müşterilerin farklı ürün veya hizmetlere ihtiyaç duyabilmesi ve bu ürünlerden bazılarının diğerlerinden daha kârlı olmasıdır. Üçüncü motivasyon, farklı hizmet düzeylerine sahip olmanın uzun vadeli kârlılığı önemli ölçüde etkileyebileceğidir; örneğin

ilk kez mükemmel hizmet alan müşterilerin sadık müşteri olma olasılıkları daha yüksektir [34].



Şekil 2.3: k öncelik sınıfına sahip kuyruk sisteminin şematik gösterimi [35].

Şekil 2.2’de görüleceği üzere bir hizmet tesisinin bazı "önem ölçülerine" göre ayırt edilebilen $k (> 1)$ birim sınıfı tarafından kullanıldığı durumlar sıklıkla ortaya çıkar. Farklı müşteri sınıfları olan öncelikli kuyruk sistemlerinde her sınıfının farklı geliş oranları vardır. Önem ölçülerini belirtmek için her sınıf, bir öncelik indeksi $i (1 < i < k)$ ile ilişkilendirilir; burada $i=1$ en yüksek önem ölçüsüne sahip sınıfı ve k en düşük sınıfı belirtir. Geleneksel olarak öncelik numarası ne kadar küçük olursa sınıfın önceliği de o kadar yüksek olur. Öncelik numarası atanan sınıflar bu öncelik sırasına göre servis biriminde hizmet alırlar. Öncelik numarası 1 olarak atanan sınıf ilk hizmet alacak olan sınıftır [35, 2]. Sunucunun bir sonraki hizmet alacak sınıfı seçtiği disiplin “öncelikli disiplin” olarak adlandırılır. Bu nedenle herhangi bir öncelik disiplini aşağıdaki iki kararın alınmasına ilişkin kuralları belirtmelidir [2]:

- (i) Sunucu bir sonraki müşteriyi almak için serbest kaldığında hizmet için hangi sınıfın seçileceği.
- (ii) Hizmet verilen birimin hizmetine devam edilip edilmeyeceği ya da son verilip verilmeyeceği.

Öncelikli müşteriler daha az önemli müşterilere göre hizmete erişimde avantajlı konumda olurlar. Öncelikli kuyruk sistemlerinin amacı, yüksek öncelikli olarak nitelendirilen müşteri gruplarına birinci sınıf hizmet vermektir ancak bu öncelikli sistem, düşük öncelikli müşteriler için olumsuz sonuçlar getirebilmektedir. Bu olumsuz sonuçlar çoğunlukla düşük öncelikli müşteriler için oldukça uzun ve değişken hizmet süreleri olabilmektedir.

Öncelikli kuyruk sistemlerinde iki tür öncelik söz konusudur [29]:

(a) Sınırsız öncelik (preemptive priority)

Sınırsız bir kuyruk sisteminde, daha düşük öncelikli bir müşteri, daha yüksek öncelikli bir müşteri geldiğinde servis hizmeti kesintiye uğratılır ve servis biriminden çıkarılarak yüksek öncelikli müşteriye hizmet verilir. Yüksek öncelikli müşteri bulunmadığında, düşük öncelikli müşteri türü hizmete yeniden girer [33]. Yeniden hizmet alındığında, kesintiye uğratılan hizmet süreci ya kaldığı yerden sürdürülür ya da hizmet süreci baştan başlatılır. Örneğin, sınırsız öncelik kuralında tip 1 müşterilerin, tip 2 müşterilere göre önceliği olsun. Sınırsız öncelik disiplini, 2. tip müşteri hizmetteyken 1. tip müşteri geldiğinde 2. tip müşterinin hizmetinin kesilmesi ve sunucunun 1. tip müşteriyle devam etmesi anlamına gelir. Sistemde 1. tip müşteri kalmadığında sunucu, 2. tip müşteri hizmetine kesintiye uğradığı yerden devam eder.

(b) Sınırlı öncelik (non-preemptive priority)

Sınırlı öncelik disiplini, sınırsız öncelikten farklı olarak öncelikli sınıfa mutlak öncelik tanımaz. Yüksek öncelikli müşteriler, düşük öncelikli müşteri servis birimindeyken hizmet alımını kesintiye uğratamaz, düşük öncelikli müşterinin hizmet süresi tamamlanana kadar beklerler. Sınırlı hizmette müşterinin hizmeti bir kez başladığında, yalnızca hizmet tamamlandığında veya müşteri vazgeçtiğinde sona erer, daha sonra sunucu bu noktalarda hizmet için yeni bir müşteri seçer.

Geleneksel öncelikli kuyruk sistemlerinin dezavantajı, düşük öncelikli ya da öncelikli olmayan müşterilere karşı olası adaletsizliğidir. Standart öncelikli kuyruk sistemlerinde bekleme hattında olası bir yığılma meydana geldiğinde öncelikli müşterilere hizmet verilirken, öncelikli olmayan müşterilerin elverişli bir süre içerisinde hizmet alma olasılığı çok azdır. Bu dezavantajın üstesinden gelmek için, statik önceliklerde çeşitli iyileştirmeler önerilebilir. Örneğin bazıları şunlardır:

- Öncelikli müşterilerin çok hızlı olan erişiminin kısıtlanması.
- Belirli sayıda öncelikli müşteriye hizmet sonrasında öncelikli olmayan bir müşteriye zorunlu hizmet verilmesi.
- Bir diğer popüler gelişme ise müşterinin kuyrukta kaldığı süre boyunca önceliğinin artırılması olanağıdır. Bu disiplin altında müşteriler, bekleme sürelerinin doğrusal bir fonksiyonu olarak bir öncelikli kredi kazanırlar [36].
- Eşik değeri belirlenmesi, düşük öncelikli müşteri sınıfının kuyruk uzunluğu belirlenen eşik değerini aştığında düşük öncelikli müşteri kuyruğuna hizmet verilmesi [26, 37].
- Müşteri transferi yapılarak düşük öncelikli müşteri sırasından yüksek öncelikli müşteri sırasına transfer olması [16, 20].
- Ödül ve ceza yapısının belirlenmesi, öncelikli kuyrukların daha verimli yönetilebilmesi için her sınıf için bekleme maliyeti ve müşterinin hizmetinin tamamlanmasının ardından alınacak bir servis ödülü belirlenmesiyle servis kontrolünün yapılması [12].

3. SİMÜLASYON TABANLI OPTİMİZASYON

Simülasyona dayalı optimizasyon, rastgele bir sistemden elde edilen net ödüllerin (veya maliyetlerin) en büyüklenmesini (veya en küçüklenmesi) gerektiren yöntemlerin etrafında döner [38]. Simülasyon tabanlı optimizasyon, optimizasyon tekniklerini, simülasyon analizine entegre eden bir alan olup, karmaşık ve gerçek dünya problemlerini çözmek için simülasyon modellerini ve optimizasyon algoritmalarının birleşimini kullanarak en iyi çözümü bulmayı amaçlayan bir yaklaşımdır [45].

Sistem, birbirleriyle etkileşime giren varlıklar (insanlar ve makineler gibi) topluluğu olarak tanımlanır. Dinamik bir sistem, sistemin durumunun zamana bağlı bir şekilde değiştiği sistemdir. Sistemdeki değişiklikleri tespit etmek için sistemi durum adı verilen sayısal bir nitelik kullanarak tanımlanır. Bu niteliğin değerindeki bir değişiklik, sistemdeki bir değişiklik olarak yorumlanabilir. Bu bilgiler ışığında stokastik sistem, durumun rastgele değiştiği dinamik bir sistem demektir. Örneğin bir süpermarkette tezgâhın önünde oluşan kuyruk ele alınsın ve sistemin durumu kuyruktaki bekleyen kişi sayısı ile ifade edilsin. O halde, sıranın stokastik bir sistem olduğu açıktır, çünkü kuyruktaki kişi sayısı rastgele değişmektedir. Kuyruktaki kişi sayısının rastgele değişmesi, müşteriler arası geliş süresi ve servis hizmeti süresi dağılımlarının rastgeleliğinden kaynaklanır [38]. Bu bağlamda bu tür kuyruk sisteminin durumu dinamik olarak değiştiğinden, stokastik olarak tanımlanır.

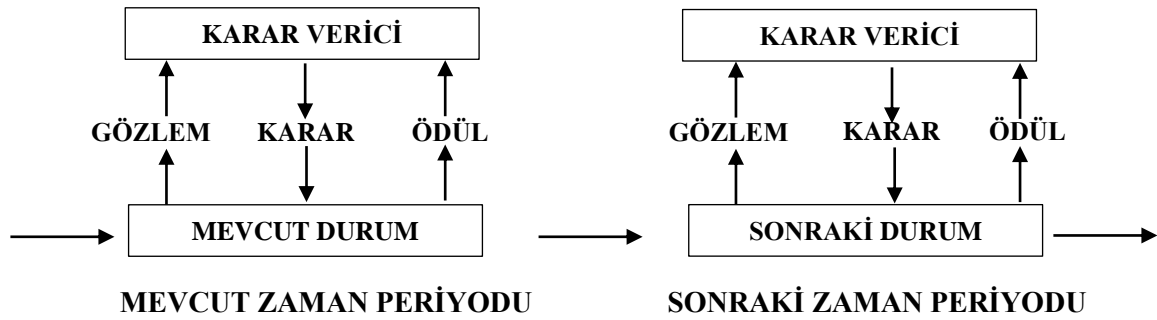
Tezde ele alınan stokastik kontrol optimizasyonu probleminde simülasyon tabanlı optimizasyon tekniklerinden yararlanılacaktır.

Bir sistemin ziyaret ettiği farklı durumlarda gerçekleştirebileceği bir dizi eylemin bulunması durumu, kontrol optimizasyonu anlamına gelir. Böylece seçilen eylemler, sistemin performans ölçümlerini optimize eder. Seçilen eylemler bir maliyeti en aza indirir veya bir ödülü en üst düzeye çıkarır. Kontrol optimizasyonu genellikle dinamik programlamayla, bazen de matematiksel programlamayla gerçekleştirilir. Ancak, çok sayıda rastgele değişkenin olduğu birçok karmaşık problemde bu zor bir iş olabilir. Bu nedenle simülasyon tabanlı optimizasyon tekniklerinden olan ödüllü öğrenmede gerçek dünya problemlerinin yarattığı hesaplama zorluğu azaltılır ve geçiş fonksiyonlarını

hesaplamak zorunda kalmadan her bir durum için değer fonksiyonları simülatör içinde hesaplanır [38].

3.1 Markov Karar Süreçleri

Bu tezdeki çalışma, Markov karar süreci (Markov decision process-MDP) adı verilen belirsizlik altında dinamik karar vermenin temel bir modeliyle ilgilidir. Tezde ele alınan problemin modellenmesinde Markov karar süreçlerinden yararlanılması ve çözüm yöntemi olarak seçilen ödüllü öğrenme tekniğinin temeli Markov karar süreçlerinin matematiksel çerçevesine dayanmasından dolayı bu bölümde Markov Karar süreçlerinden bahsedilmiştir. Markov karar süreci, ardışık karar verme problemlerinin modellenmesinde ve çözümünde kullanılan matematiksel bir çerçevedir. Markov karar süreçlerinin amacı, belirlenmiş olan amaç fonksiyonu için optimal politikayı bulmaktır. Markov karar süreçleri aynı zamanda Markov karar problemleri olarak da bilinmektedirler.



Şekil 3.1: Markov karar süreci şematığı [46].

Stokastik bir süreç genellikle birden fazla durumu ziyaret eder. MDP bir kontrol optimizasyonu problemi olup, her durumda seçilecek en uygun kararı bulma problemleridir [38]. Maliyet ve sonraki durum yalnızca mevcut duruma ve seçilen eyleme bağlıdır. Markov süreci, gelecek durumların tahmini için geçmiş durumlara ihtiyaç duyulmaz. Bu duruma Markov özelliği denmektedir [33].

Markov karar süreci, Markov karar problemini çözmek için tasarlanmıştır. Bir Markov karar süreci dört temel unsurdan oluşur [33]:

- Durum uzayı,
- Karar seti,

- Geçiş olasılıkları,
- Beklenen ödül.

Durum uzayı: Markov karar sürecinin her periyodun başında i durumundadır. Bu i durumu, S ile gösterilen durum uzayının bir elemanıdır, $i \in S = \{1, 2, 3, \dots, N\}$.

Karar seti: Her i durumu için, sonlu bir dizi izin verilebilir kararlar setidir. $A(i)$ olarak tanımlanabilir.

Geçiş olasılıkları: Bir periyodun i durumunda başladığı varsayılarak i durumunda karar uzayından bir a kararı $a \in A(i)$, alınırsa, $p(j | i, a)$ olasılıkla j durumuna geçecektir. Bir i durumda izin verilen kararlar arasından birine geçme olasılığına geçiş olasılığı denir. Bir sonraki dönemin durumu yalnızca içinde bulunulan dönemin durumuna ve içinde bulunulan dönemde alınan karara bağlıdır, önceki durum ve kararlara değildir (Markov özelliği). Bu nedenle Markov karar süreci terimini kullanılmaktadır.

Örneğin, Markov karar süreci olarak modellenebilecek M/M/1 kuyruk sistemi ele alınsın. Simülasyon için geçiş olasılıklarına ihtiyaç yoktur, gelişler arası süre ve hizmet süresi dağılımlarını bilmek yeterlidir. Temelinde Markov zinciri bulunan stokastik bir sistemi simüle etmek için sistemin geçiş olasılıklarına ihtiyaç yoktur. Simülasyon için girdi değişkenlerine yani sistem davranışını yöneten dağılımlara ihtiyaç vardır [38].

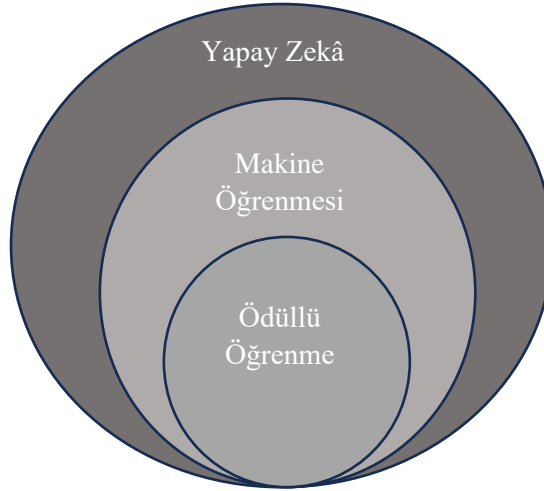
Beklenen Ödül: Durumun i olduğu ve $a \in A(i)$ kararının seçildiği bir dönemde, r_{ia} beklenen ödülü alınır.

Markov karar süreçlerinde, bir durumdan diğer duruma geçerken iki durum arası bir geçişin süresi sabit bir birim ya da üstel bir dağılımdır. Yarı Markov karar süreçlerinde (Semi Markov decision process, SMDP) ise iki durum arasındaki geçen bu süre bir rassal değişkendir ve genel dağılıma uyar. Yarı Markov karar problemi veya süreci, her durumda harcanan zamanın modelin önemli bir parçası olduğu Markov karar probleminin (veya sürecinin) daha genel bir versiyonudur. SMDP, MDP'ye çok benzerdir; aradaki fark, her geçişin 1 birim veya üstel dağılım olmayan bir süre almasıdır [38]. Bu tez çalışmasında ele alınan kuyruk probleminde müşteri gelişleri Poisson sürece göre olup, geçiş süreleri üstel dağılımlı olduğundan Markov karar süreci kullanılmıştır.

3.2 Ödüllü Öğrenme

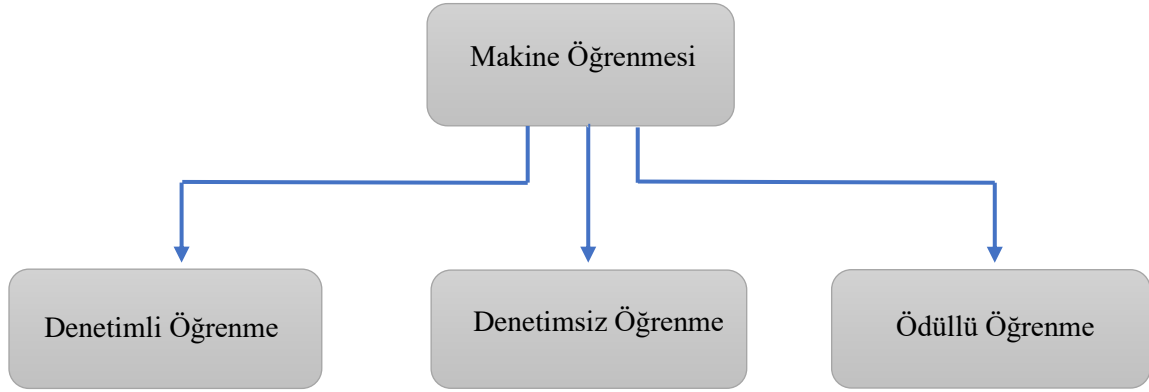
Ödüllü öğrenme temel olarak simülasyon tabanlı optimizasyon tekniği olup makine öğrenme dalıdır. Ödüllü öğrenme için Türkçe kaynaklara bakıldığında literatürde takviyeli öğrenme, pekiştirmeli öğrenme ya da destekleyici öğrenme olarak da karşımıza çıktığı görülmektedir. Ödüllü öğrenme, dinamik programlamaya dayalı tekniklerin geliştirilmesiyle meydana geldiği için neuro dinamik, dinamik stokastik veya yaklaşık dinamik programlama isimlerinin de kullanıldığı görülmektedir [38, 39].

Ödüllü öğrenme, makine öğrenmesi yöntemi olarak yapay zekâ (artificial intellegent) topluluğundaki araştırmacılar tarafından geliştirilmiştir. Yapay zekâ topluluğunda ödüllü öğrenme, bir makine öğrenme tekniği olarak görülüyordu. Yapay zekâ, makine öğrenmesi ve ödüllü öğrenme arasındaki ilişkiyi göstermek için Şekil 3.2 kullanılmıştır. Makine öğrenmesi dallarından biri olan ödüllü öğrenme, diğer makine öğrenimi, yapay zekâ ve sinir ağı gibi araştırma alanları arasında giderek en aktif araştırma alanlarından biri haline gelmektedir [38].



Şekil 3.2: Yapay zekâ, makine öğrenmesi ve ödüllü öğrenme [41].

Makine öğrenmesi, bilgisayarın geçmiş verilerden öğrenip gelecekteki oluşacak benzer olayları tahmin edebilmesini ve problemlere çözüm getirebilmesini sağlar. Makine öğrenmesi, deneyim ve veri yığınları üzerinden otomatik olarak öğrenme yeteneğine sahip, bilgisayar algoritmalarının geliştirilmesini içeren bir yapay zekâ alanıdır. Makine öğrenme yaklaşımları Şekil 3.3'te gösterildiği gibi denetimli, denetimsiz ve ödüllü olmak üzere üç grupta incelenmektedir [40].



Şekil 3.3: Makine öğrenmesinin alt dalları [41].

Denetimli öğrenmede, bir dizi girdi ölçüsüne dayalı olarak bir sonuç ölçüsünün değerini tahmin edilir; denetimsiz öğrenmede ise herhangi bir sonuç ölçümü yoktur ve bir dizi girdi ölçümü arasındaki ilişkiler ve örüntüler tanımlanır [41].

Ödüllü öğrenme ise diğer öğrenme yaklaşımlarından biraz farklıdır. Sisteme hangi eylemleri gerçekleştireceği söylenmez bunun yerine sistemin hangi eylemin en çok ödül getirdiğini deneyerek keşfetmesi gerekir. Sistemde etmen ya da ajan adı verilen aracı, bulunduğu çevre ile etkileşime girerek deneme-yanılma davranışı gerçekleştirir, yapacağı ve yapmayacağı seçimleri ödül-ceza yöntemiyle öğrenir. Amaç alacağı ödülü en üst düzeye çıkarmaktır [41]. Ödüllü öğrenme hakkında daha detaylı bilgiler Bölüm 3.4’ de yer almaktadır.

Ödüllü öğrenme algoritmasıyla Markov/yarı Markov problemlerinde yaklaşık optimal çözüm elde edilmesi amaçlanmaktadır. Diğer yandan dinamik programlama kontrol optimizasyonu problemleri olan Markov karar süreçleri problemlerinde optimum çözümü verir, ancak bu problemlerde dinamik programlamayı kullanabilmek için geçiş olasılıklarının hesaplanmış olması gerekmektedir. Geçiş olasılıklarını, ödülleri ve zamanları elde etmek büyük ölçekli sistemler için karmaşık matematik içeren zor bir süreçtir. Geçiş olasılıklarının hesaplanması çok katlı integral hesaplamaları içerdiğinden dolayı büyük ve karmaşık problemlerde rassal değişkenlerin sayısı fazlalaştıkça, bu hesaplamalar daha güç hale gelmektedir. Literatürde dinamik programlamanın bu durumu modellemenin laneti (curse of modeling) olarak isimlendirilmektedir. Hatta karmaşık teorik modellere sahip birçok sistem yerine nispeten daha küçük gerçek hayat problemleri

için bile geçiş olasılıklarının hesaplanması hantal bir süreç olabilmektedir [38]. Bu yüzden modellemenin laneti küçük problemler için bile geçerli olabilmektedir. Ödüllü öğrenme geçiş olasılıklarına ihtiyaç duymadığı için modellemenin lanetiyle başa çıkabilmektedir.

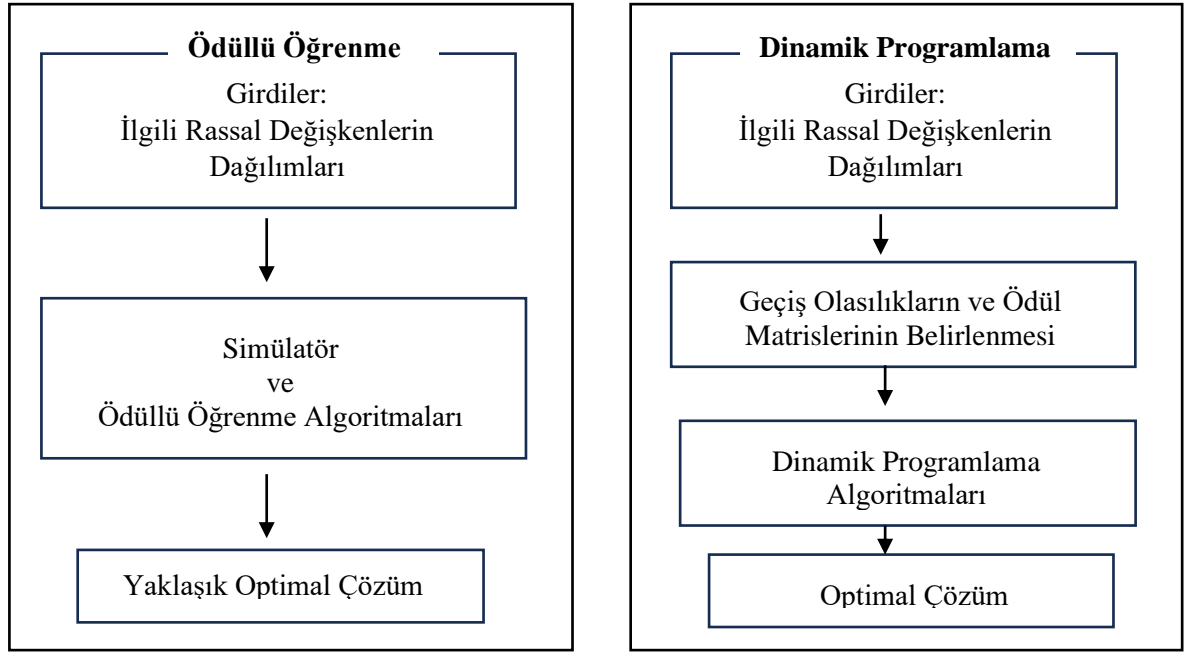
Dinamik programlamada bir diğer problem ise geçiş olasılıklarının depolanması sorunudur. Örnek olarak 100 adet durum ve her bir durum (state) için 3 kararı (actions) olan bir Markov karar problemi ele alınabilir. Bu durumda her kararın geçiş olasılık matrisi $100^3=10^6$ (1 milyon) eleman içerecektir ve bu matrisler dinamik programlamanın algoritmaları tarafından kullanılması için bilgisayar tarafından depolanması gerekecektir. Durum sayısı, durum değişkenlerinin sayısı ile birlikte üstel olarak arttığı için eleman sayısı çok büyük matrisler ile karşılaşmaktadır. Bu kadar büyük matrisleri bilgisayarlarımızda saklamak imkânsız olmasa da zordur. Bu durum günümüz bilgisayarlarımızda bile depolanması zor olduğundan çözüm üretilmesi amacıyla işlenmesi de zor olacaktır [38]. Bu zorluk Bellman [42] tarafından boyutun laneti (curse of dimensionality) olarak adlandırılmıştır. Bu nedenle dinamik programla, modellemenin laneti ve boyutun laneti tarafından iki kez lanetlenmiştir [38].

Ödüllü öğrenme bu iki lanetten de bir çıkış yolu sağlayabilir [38]:

i) Ödüllü öğrenme geçiş olasılığı matrisleri hakkında önceden bilgiye ihtiyacı yoktur. Ödüllü öğrenmede model gerektirmediği için modelleme lanetinden kaçınabildiği söylenmektedir.

ii) Ödüllü öğrenmede dinamik programlamanın kullandığı değer fonksiyonunun elemanlarını kullanılmasına gerek yoktur. Ödüllü öğrenme algoritmaları, dinamik programlamada durumlar için kullanılan değer fonksiyonu elemanlarını Q faktör olarak adlandırılan terimlerle tanımlar ve depolar. MDP'nin milyonlarca durumu olduğunda, milyonlarca Q faktörüne dönüşür. Böylece ödüllü öğrenmede de boyutun laneti problemi ile karşı karşıya kalınır. Q faktör değerleri, fonksiyon yaklaşımı teknikleri yoluyla az sayıda skaler biçiminde depolanabilir. Başka bir deyişle, durum-karar uzayındaki herhangi bir Q faktörünün değerini sağlayabilen nispeten az sayıda skaler depolanır. Böylece ödüllü öğrenme, fonksiyon yaklaşımı metoduyla dinamik programlamanın yaşadığı boyutluluk laneti probleminden kaçınabilir.

Ödüllü öğrenme ve dinamik programlama algoritmaları arasındaki farklar Şekil 3.4'te karşılaştırma yapılarak gösterilmektedir. Şekil 3.4'te gösterildiği üzere dinamik programlamadan farklı olarak ödüllü öğrenme algoritmaları için geçiş olasılıkları, geçiş ödülleri ve geçiş sürelerinin hesaplanması gerekmez, çünkü bütün bu hesaplamalar simülasyon aracılığıyla yapılır. Dinamik programlama algoritmaları tarafından optimal çözüm sunulurken, ödüllü öğrenme algoritmalarıyla yaklaşık optimal çözüme ulaşılabilmektedir [24, 39].



Şekil 3.4: RL ve DP metodolojilerindeki farklılıkları vurgulayan şematik [38].

Sonuç olarak simülasyonlarda çalışan ödüllü öğrenme, bize Markov özelliği tarafından yönetilen karmaşık ve büyük ölçekli kontrol problemlerini geçiş olasılıklarını oluşturmadan çözebilecek bir mekanizma sağlar. Böylece, ödüllü öğrenme yoluyla, geçiş olasılıklarını elde etmenin imkânsız olmasa da zor olduğu karmaşık problemleri çözme becerisine sahip olunur [38].

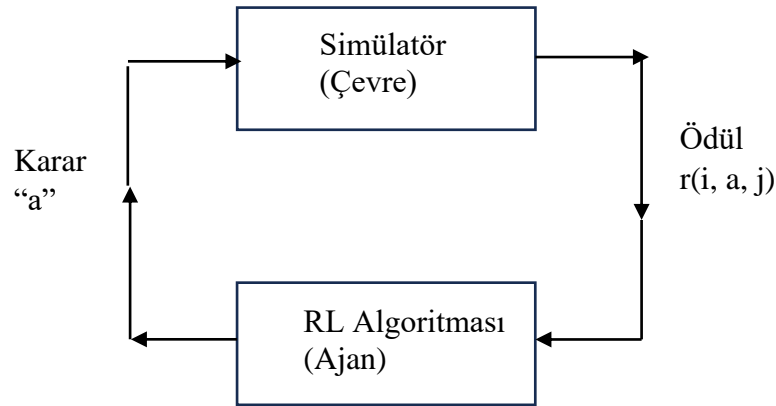
Kuyruk sistemlerinde optimal kontrol politikasını bulmak genellikle zorlu problemler olmuştur. Dinamik programlama algoritmaları, Markov karar süreci olarak modellenmiş bu tip kontrol problemlerinin optimal çözümünü bulmayı hedefler. Ancak, dinamik programlama yöntemleriyle sorunun çözümü boyutun ve modellenmenin laneti nedeniyle çoğu zaman çok zorlu bir süreçtir. Yukarıda anlatılan zorluklar nedeniyle tezde ele alınan

Markov karar problemi olarak modellenen problemin çözümünde ödüllü öğrenme teknikleri kullanılmıştır.

3.2.1 Ödüllü Öğrenmede Öğrenme Yaklaşımı

Çevre ile etkileşim sırasında, bir etmen (ajan) bir eylemde bulunur ve her adımda bir gözlem ve ödül alır [41]. Bu eylemler karşısında ödüllü öğrenme algoritmaları, aldığı toplam ödülü en üst düzeye çıkarmayı veya maliyetini en aza indirmeyi amaçlamaktadır.

Ödüllü öğrenme algoritması iki ana bileşenden oluşur: ajan ve çevre. ‘Ajan’ denilen sistem çevreyi gözlemleyerek, çeşitli seçenekler arasından seçimler yapmaktadır. ‘Çevre’ terimi ajanın yaşadığı ve etkileşimde bulunduğu alan için kullanılır. Ödüllü öğrenme algoritması çevre ile etkileşim halindedir. Çevreden aldığı sinyallerle kendine hareket dizisi belirlemektedir. Bu seçimleri karşısında ödül ve cezalar almakta ve zaman içinde daha çok ödül almak için en iyi ilkeleri öğrenmektedir. Tüm bu deneyimler sonunda, belirli durumlarda doğru seçimleri yapabilme yetisini kazanabilmektedir.



Şekil 3.5: Ödüllü öğrenme mekanizması [38].

Şekil 3.5’te gösterilen ödüllü öğrenme mekanizmasında şu olaylar gerçekleşmektedir; RL ajanı (algoritma) tarafından seçilen eylem, simülatöre beslenir. Simülasyon, kararı simüle eder ve sonuçta elde edilen geri bildirim (anında ödül ya da ceza), ajanın bilgi tabanı (Q faktörleri) geri beslenir. Ajan, Q faktörleri güncellemek için RL algoritmasını kullanır, süreçte daha akıllı hale gelir ve ardından daha iyi bir eylem seçer. Amaç, iyi kararları ödüllendirmek, kötü kararları cezalandırmak ve genel olarak simülasyondaki deneme yanılma yoluyla öğrenmektir.

Simülasyonun çalışma süresi boyunca öğrenme aracısının simülatör içinde öğrenip öğrenemeyeceğini (davranışını geliştirip geliştiremeyeceğini) simülatör çalışırken belirlemek zordur. Algoritmanın performansı en iyi şekilde öğrenme tamamlandıktan sonra değerlendirilir. Algoritmanın performansının ölçülebilmesi için ortalama ödül ölçütü kullanılır. Performansı değerlendirmek için öğrenilen politikayı kullanarak sistem yeniden simüle edilir. Böylece öğrenilen politikanın ortalama ödülü, algoritmanın öğrenip öğrenmediği hakkında bilgi verir.

Ödüllü öğrenme algoritmalarının birçoğu dinamik programlamanın değer fonksiyonunu hesaplar, ancak ödüllü öğrenme değer fonksiyonunu Q faktörleri olarak adlandırılan formda saklar. Dinamik programlamada değer fonksiyonunun bir elemanı belirli bir durumla ilişkilendirilirken, ödüllü öğrenmede ise Q-faktörü vektörü olarak adlandırılan vektörün bir elemanını belirli bir durum-karar çiftiyle ilişkilendirilir. Ödüllü öğrenmede bir durumun Q faktör ilişkisi anlamak için MDP’de optimal politikayı bulmaya yarayan Bellman optimallik denklemini (Denklem 3.1) bilmek gerekir [38].

$$j^*(i) = \max_{a \in A(i)} \left(\sum_{j=1}^{|S|} p(i, a, j) [r(i, a, j) + \lambda J^*(j)] \right) \quad (3.1)$$

Bu denklemde $J^*(i)$ optimal politikayla ilişkili değer fonksiyon vektörünün i inci elemanını; S , sistemdeki durumlar kümesini; $A(i)$, i durumunda alınabilecek kararlar kümesini; $p(i, a, j)$, i durumundayken a kararı verildiğinde j durumuna geçiş olasılığını; a , i durumundan j durumuna geçmek için alınan kararı; $r(i, a, j)$, i durumundayken a kararı verildiğinde j durumuna geçmenin anlık maliyetini belirtir. λ , iskonto faktörünü temsil eder. Bu olayı anlamak için örnek olarak bir Markov karar süreci problemi ele alınsın. Her bir durumda iki kararın alınabildiği 3 durumlu bir örnek temel alındığında, dinamik programlama ile j^* değer fonksiyon vektörü Denklem 3.2’de gösterildiği gibi üç elemana sahip olur. Ödüllü öğrenme algoritmasında ise Q faktör vektörü Denklem 3.3’teki görüldüğü üzere 6 durum-karar ikilisine sahip olur [38].

$$j^* = \{j^*(1), j^*(2), j^*(3)\} \quad (3.2)$$

$$Q = \{Q(1,1), Q(1,2), Q(2,1), Q(2,2), Q(3,1), Q(3,2)\} \quad (3.3)$$

MDP problemlerinde her bir durum karar ikilisinin Q faktör değer vektörü elemanları, Denklem 3.4’ten hesaplanır [38].

$$Q(i, a) = \sum_{j=1}^{|S|} p(i, a, j)[r(i, a, j) + \lambda J^*(j)] \quad (3.4)$$

Denklem 3.1 ve Denklem 3.4'ün kombinasyonunun sonucunda ortaya çıkan bir durumun değer fonksiyonu ve Q faktör ilişkisi, Denklem 3.5'te verilmiştir [38].

$$j^* = \min_{a \in A(i)} Q(i, a) \quad (3.5)$$

Yukarıdaki denklemler bir durumun değer fonksiyonu ile durumla ilişkili Q faktörleri arasındaki önemli ilişkiyi kurar. Eğer Q faktörleri biliniyorsa belirli bir durumun değer fonksiyonu, Denklem 3.5'ten elde edilebilir. Örneğin iki eylemli bir i durumunun Q faktörü, $Q(i,95)$ ve $Q(i,100)$ ise $j^*(i)$, değer fonksiyonun cevabı 95 olur.

$$j^*(i) = \min\{95,100\} = 95 \quad (3.6)$$

Dinamik programlamada mutlaka bulunması gereken geçiş olasılıklarının hesaplanmasına gerek duyulmadan Q faktör değerleri Denklem 3.6'da verildiği şekilde simülatörde güncellenir [38,39].

$$Q(i, a)^{k+1} \leftarrow (1 - \alpha^{k+1})Q^k(i, a) + \alpha^{k+1} \left[r(i, a, j) - \rho^* + \min_{b \in A(j)} Q^k(j, b) \right] \quad (3.7)$$

Bu denklemde α , öğrenme oranı; k, iterasyon sayısı veya yineleme sayısı; $r(i, a, j)$, mevcut durum i iken a kararı verildiğinde durum j'ye geçmenin anlık maliyeti; ρ^* ise sonsuz zaman ufkunda birim zaman başına optimal ortalama maliyettir. Yukarıdaki denklemde bulunan öğrenme oranı olan α için 1'den küçük olacak şekilde sabit bir değer kullanılabilir ya da simülasyon boyunca her bir $Q(i,a)$ ikilisini yani i durumu altında a kararının kaç defa ziyaret edildiği bilgisi tutularak dinamik olarak hesaplanabilir [38, 39].

3.2.2 Q-Öğrenme

Q-öğrenme, ödüllü öğrenmenin bir alt dalıdır. Q-öğrenmede ajan yaptıkları eylemlerin sonuçlarını deneyimleyerek Markov etki alanlarında en iyi şekilde hareket etmeyi öğrenme yeteneği sağlar. Öğrenme yöntemi, bir etmen belirli bir durumda bir eylemi denemesi ve sonuçlarını anında aldığı ödül veya ceza ile yeni durumun değerine ilişkin tahminin değerlendirilmesidir. Tüm durumlardaki tüm eylemleri tekrar tekrar deneyerek, uzun vadeli iskontolu ödüllere göre değerlendirip hangilerinin en iyi olduğunu öğrenir. Q-öğrenme uygun koşullar altında yaklaşık optimal çözümler üretmektedir [43].

Çalışmada Göreceli (relative) Q-öğrenme algoritması kullanılmıştır. Algoritmanın adımları şu şekilde sıralanır [38]:

Adım 1: Tüm durum karar ikilileri olan (s,a) için Q faktör değeri sıfır olarak alınarak başlanır; $s \in S$ ve $a \in A(s)$, $Q(s,a)=0$. Herhangi bir durum-karar ikilisini ayırt edici (i^*,a^*) ikilisi olacak şekilde seçin. Durum geçişlerin sayısı olan k sayısı 0 olarak ayarlanır. Algoritma k_{max} sayılı yineleme için çalıştırılır; burada k_{max} yeterince büyük bir sayı olacak şekilde seçilir.

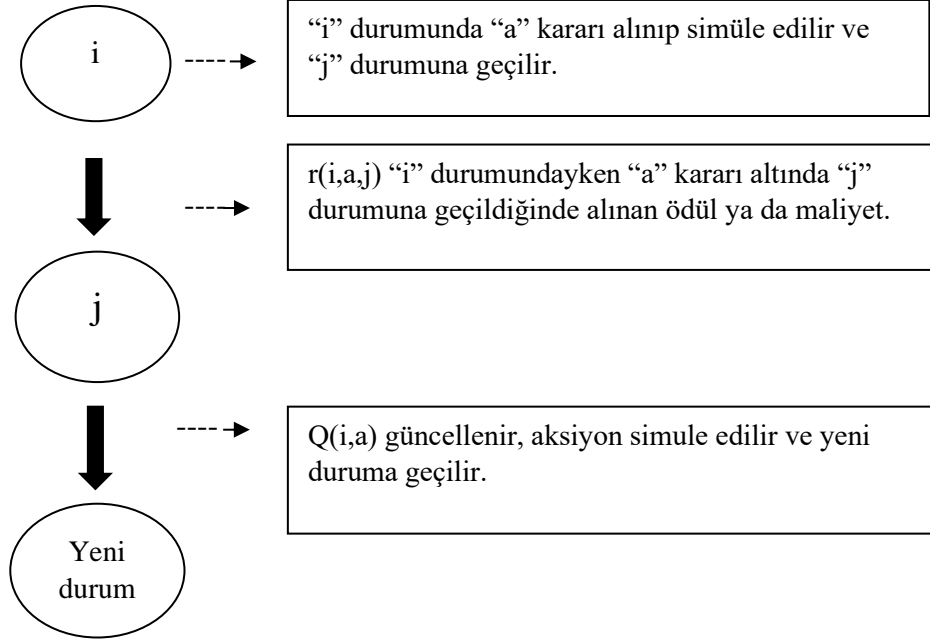
Adım 2: Mevcut durum i olsun ve a kararı $1/|A(i)|$ olasılığı ile seçilir.

Adım 3: “a” kararı simüle edilir. Bir sonraki durum j olsun. “i” durumunda “a” eylemi yapıldığında, bir sonraki durum olan “j” durumuna geçildiğinde ortaya çıkan anlık maliyet (immediate cost) ya da ödül $r(i, a, j)$ olsun. $r(i, a, j)$ miktarı simülatör tarafından belirlenecektir. Diğer iterasyona geçilir. Daha sonra α güncellenir.

Adım 4: $Q(i, a)$ aşağıdaki Denklem 3.8’ten faydalanılarak güncellenir.

$$Q(i, a) \leftarrow (1 - \alpha)Q(i, a) + \alpha \left[r(i, a, j) + \max_{b \in A(j)} Q(j, b) - Q(i^*, a^*) \right] \quad (3.8)$$

Bu denklemde bir simülatörde Q faktörlerin güncellenmesi Şekil 3.4’te gösterilmiştir. Her ok, simülatördeki bir durum geçişini belirtir. j durumuna gidildikten sonra, önceki i durumu ve i’de seçilen a kararı, yani $Q(i, a)$ için Q faktörü güncellenir [38].



Şekil 3.6: Bir simülatörde Q faktör değerlerinin güncellenmesi [38].

Adım 5: Eğer $k < k_{\max}$ ise $i \leftarrow j$ olarak ayarlanır ve **Adım 2**'ye gidilir aksi takdirde, **Adım 6**'ye gidilir.

Burada k, iterasyon sayısı veya yineleme sayısı; k_{\max} , algoritmanın çalışacağı toplam yineleme sayısıdır.

Adım 6: Her bir $l \in S$ için $d(l) \in \arg \max_{b \in A(l)} Q(l, b)$ seçilir. Algoritmayla üretilen politika, d'dir.

Q-öğrenmede, tüm durum ve alınabilecek kararları tutan bir tablo kullanılır. Öğrenme işlemi Şekil 3.6'da gösterildiği şekilde yapılır ve her bir adımında durum-karar çiftlerini ve Q-faktör değerlerini tutan tabloda değerler güncellenir. Q-öğrenme işleminde, Q-faktör tablosu başlangıçta sıfır değerleri ile doldurulur. Bu tablo her bir s durumu ve bu s durumunda alınabilecek bir a karar için Q-faktör değerleri tutulur. Tablo 3.1'de Q-faktör tablosunun genel yapısı gösterilmiştir.

Tablo 3.1 Q-faktör tablonun örnek yapısı.

Durum		Karar (a)	Q (s ₁ ,s ₂ ,a)	Q-Faktör Değeri <Başlangıç
s ₁	s ₂			
0	0	a	Q (0, 0, a)	0
1	0	a	Q (1, 0, a)	0
..	..	a	Q (., ., a)	0
m	n	a	Q (m, n, a)	0
..	..	a	Q (., ., a)	0

Algoritmanın çözümü tarafından üretilen ortalama ödülün değeri, öğrenme tamamlandıktan sonra öğrenilen politikayı kullanarak simülörün yeniden çalıştırılmasıyla bir simülör içinde kolayca ölçülebilir [38].

Çalışmada ödüllü öğrenme algoritmasıyla öğrenilen politikanın simülörde performansı aşağıdaki algoritma kullanılarak incelenmiştir. Ödüllü öğrenme algoritması, belirlenen aralıklarla simülör içinde yeniden çalıştırılır ve o süreye dek öğrenilen politikanın birim zamandaki ortalama maliyeti hesaplanarak ödüllü öğrenmenin performansı değerlendirilir. Ödüllü öğrenme algoritmasının yineleme sayıları ilerledikçe, hesaplanan ortalama maliyetler bir değere yakınsar [38].

Ortalama ödül hesaplanma adımları aşağıdaki gibidir [38];

Adım 1: Geçiş sayısı ve toplam ödül sıfır (0) yapılır.

Adım 2: Mevcut durum i olsun. i durum için minimum Q faktörüyle ilişkili olan “a” kararı seçilir.

$$a \in \arg \min_{u \in A(i)} Q(i, u) \quad (3.9)$$

Adım 3: Seçilen “a” kararı simüle edilir ve a kararı seçildiğinde gidilen durum j olsun. $r(i, a, j)$, i durumdayken a kararı verildiğinde j durumuna geçiş sonucu kazanılan anlık maliyeti gösterir. Toplam maliyet Denklem 3.10’u kullanarak güncellenir.

$$\text{Toplam Maliyet} \leftarrow \text{Toplam Maliyet} + r(i, a, j). \quad (3.10)$$

$k \leftarrow k+1$

Adım 4: Eğer $k < k_{\max}$ ise, $i \leftarrow j$ yapılır ve Adım 2'ye dönülür. $k < k_{\max}$ değilse, Adım 5'e gidilir.

Adım 5: Öğrenilen politikanın Denklem 3.11'i vasıtasıyla birim süredeki ortalama maliyet hesaplanır.

$$\rho = \frac{\text{Toplam maliyet}}{\text{Toplam süre}} \quad (3.11)$$

Bu tez çalışmasında ortalama ödül problemi, maliyet açısından değerlendirilmiştir. Ödüllü öğrenme algoritmasının etkinliğinin değerlendirilmesi için belli aralıklarda başlangıçtan itibaren belirlenen süreye kadar öğrenilen politika simüle edilmiş ve birim zamanda içinde ortaya çıkan ortalama maliyet hesaplanmıştır. Ödüllü öğrenme algoritmasını içeren simülasyonun yinleme sayısı arttıkça öğrenilen politikanın ortalama maliyeti azalır ve Q faktör değerleri değişmemeye başladığı görülür böylece elde edilen politikanın ortalama maliyeti bir değere yakınsamaktadır [38].

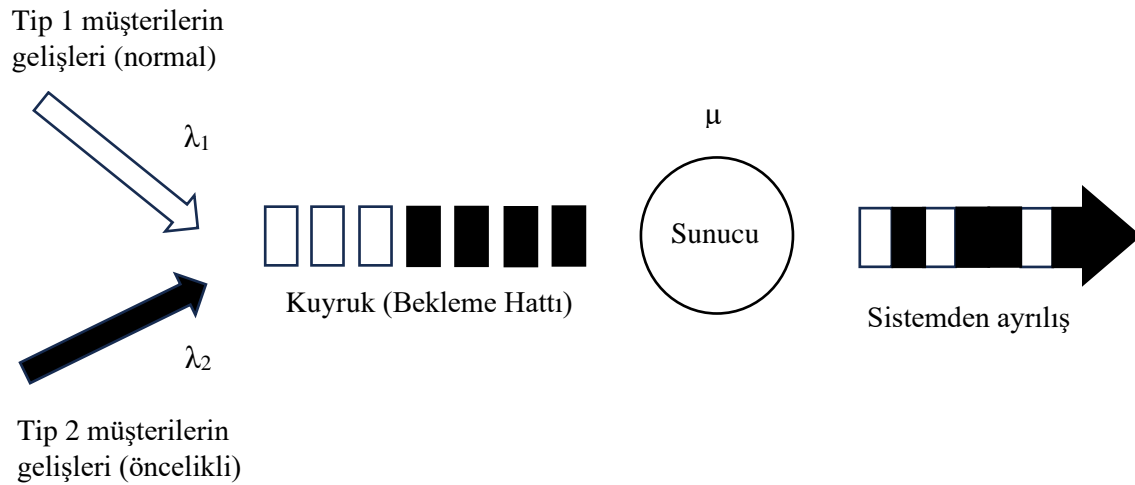
4. PROBLEM TANIMI VE MODEL

4.1 Problem Tanımı

Tezde ele alınan problem öncelikli ve normal olmak üzere iki farklı müşteri tipini içeren tek sunucuya sahip bir M/M/1 kuyruğudur (Bkz Şekil 4.1). Bu müşteriler Tip 1 ve Tip 2 müşteriler olmak üzere ikiye ayrılmıştır. Tip 1 müşteriler normal, Tip 2 müşteriler ise önceliklidir. Öncelik disiplini sınırlı önceliklidir (non-preemptive priority). Bu iki tür müşteri sınıfı sisteme bağımsız Poisson Sürecine göre λ_i oranıyla gelir ve ortak tek bir kuyruk oluşturarak bir sunucudan ortak üstel dağılımla, $1/\mu$ oranıyla hizmet alır. Her tip müşteri, kuyrukta birim zaman başına farklı bekleme maliyetine sahiptir. Bu maliyet h_i birim zaman/müşteri/TL ile tanımlanır, $i= 1, 2$. Tip 1 müşterilerin kuyrukta bekleme maliyetleri h_1 daha düşük, Tip 2 müşterilerin bekleme maliyeti ise h_2 daha yüksektir ($h_2 > h_1$). Sisteme, Tip 2 öncelikli bir müşteri geldiğinde karar vericinin (sunucunun) vermek mecburiyetinde olduğu iki tür karar vardır:

- 1) Tip 2 müşterinin önceliklendirilmesi ve normal müşterileri atlayarak öncelikli hizmet alması.
- 2) Gelen Tip 2 müşterinin önceliklendirilmemesi ve geldiği sıraya göre kuyruğa girmesi.

Problem, birim zaman başına beklenen (ortalama) maliyeti yaklaşık en küçükleyecek optimal kuyruk kontrol politikasının belirlenmesidir. Markov karar süreciyle matematiksel modeli ile temsil edilen problem, ödüllü öğrenme algoritması kullanılarak yaklaşık optimal sonuç bulunmaya çalışılmıştır.



Şekil 4.1: Öncelikli kuyruk probleminin şematik gösterimi.

Problemin simülasyon modelinin daha iyi anlaşılabilmesi için bir küçük örnek tasarlanmıştır. Bu örnekte 10 müşterinin geldiği bir sistem için müşterilerin geliş zamanları olarak $MGZ = \{2, 3, 5, 7, 10, 12, 16, 19, 20, 23\}$ ‘inci dakikalarda müşterilerin gelişleri olarak kabul edilsin. Müşteri tipleri sırasıyla $i = \{1, 2, 1, 2, 1, 2, 1, 1, 2, 2\}$ ve müşterilerin servis süreleri sırasıyla $SS = \{2, 3, 3, 4, 2, 7, 6, 4, 5, 4\}$ dakikadır. Örnek probleme ilişkin veriler Tablo 4.1’ de özetlenmiştir.

Tablo 4.1: Örnek simülasyon veri tablosu.

Müşteri No	Müşteri Geliş Zamanı	Müşteri Tipi	Müşteri Servis Süresi (dakika)
1	2	1	2
2	3	2	3
3	5	1	3
4	7	2	4
5	10	1	2
6	12	2	7
7	16	1	6
8	19	1	4
9	20	2	5
10	23	2	4

Her bir 2. tip müşterinin geldiği sıraya göre hizmet alması durumunda oluşan simülasyon tablosu Tablo 4.2’de verilmiştir.

Tablo 4.2: Örnek simülasyon tablosu (saf öncelik yok).

Müşteri Numarası	Müşteri Geliş Zamanı	Müşteri Tipi	Servis Başlama Zamanı	Servis Süresi	Servis Tamamlanma Zamanı	Kuyruқта Bekleme Süresi
1	2	1	4	2	6	0
2	3	2	6	3	9	3
3	5	1	9	3	12	4
4	7	2	12	4	16	5
5	10	1	16	2	18	6
6	12	2	18	7	25	6
7	16	1	25	6	31	9
8	19	1	31	4	35	12
9	20	2	35	5	40	15
10	23	2	40	4	44	17

Tablo 4.2’de görüldüğü üzere tüm müşterilerin geldikleri sıraya göre hizmet almaları durumunda 1. Tip müşterilerin toplam kuyrukta bekleme süresi 31 dk.; 2. Tip müşterilerin kuyrukta bekleme süresi 46 dk. olarak gerçekleşmiştir. Her gelen 2. Tip müşterinin önceliklendirildiği durumu gösteren simülasyon tablosu, Tablo 4.3’te verilmiştir.

Tablo 4.3: Örnek simülasyon tablosu (saf öncelik var).

Müşteri Numarası	Müşteri Geliş Zamanı	Müşteri Tipi	Servis Başlama Zamanı	Servis Süresi	Servis Tamamlanma Zamanı	Kuyrukta Bekleme Süresi
1	2	1	4	2	6	0
2	3	2	6	3	9	3
4	7	2	9	4	13	2
6	12	2	13	7	20	1
9	20	2	20	5	25	0
10	23	2	25	4	29	2
3	5	1	29	3	32	24
5	10	1	32	2	34	22
7	16	1	34	6	40	18
8	19	1	40	4	44	21

Tablo 4.3’te görüldüğü üzere 2. Tip müşterilerin saf önceliklendirilerek hizmet almaları durumunda 1. Tip müşterilerin toplam kuyrukta bekleme süresi 85 dk.; 2. Tip müşterilerin kuyrukta bekleme süresi 8 dk. olarak gerçekleşmiştir.

Örnekte sadece 4. ve 9. öncelikli 2. Tip müşterilerin önceliklendirildiği durumu gösteren simülasyon tablosu, Tablo 4.4’te verilmiştir.

Tablo 4.4: Örnek simülasyon tablosu (karma öncelik var).

Müşteri Numarası	Müşteri Geliş Zamanı	Müşteri Tipi	Servis Başlama Zamanı	Servis Süresi	Servis Tamamlanma Zamanı	Kuyrukta Bekleme Süresi
1	2	1	4	2	6	0
2	3	2	6	3	9	3
4	7	2	9	4	13	2
3	5	1	13	3	16	8
5	10	1	16	2	18	6
6	12	2	18	7	25	6
9	20	2	25	5	30	5
7	16	1	30	6	36	14
8	19	1	36	5	41	17
10	23	2	41	4	45	18

Sadece 4. Ve 9. Müşterilerin önceliklendirildiği durumda 1. Tip müşterilerin toplam kuyrukta bekleme süresi 45 dk; 2. Tip müşterilerin kuyrukta bekleme süresi 34 dk. olarak gerçekleşmiştir.

Verilen örnek için 3 politikanın kıyaslaması Tablo 4.5'te görülmektedir.

Tablo 4.5: Örnek simülasyon sonuç kıyaslaması.

Politika	Kuyrukta Bekleme Süresi (Tip 1)	Kuyrukta Bekleme Süresi (Tip 2)	Tip 1 müşterilerin birim zaman başına bekleme maliyeti (h_1)	Tip 2 müşterilerin birim zaman başına bekleme maliyeti (h_2)	Toplam Bekleme Maliyeti (TL)
Saf Öncelik yok	31	46	1 TL	10 TL	$1 \times 31 + 10 \times 46 = 491$
Saf öncelik var	85	8	1 TL	10 TL	$1 \times 85 + 10 \times 8 = 165$
Karma öncelik	45	34	1 TL	10 TL	$1 \times 45 + 10 \times 34 = 385$

Tablo 4.5'te görüldüğü üzere simülasyonda kullanılan verilerle müşteri sınıfları arasında saf önceliğin olmaması ve geldikleri sırayla hizmet alınmasının bekleme maliyeti 491 TL dir. 2.Tip müşterilere saf önceliğin olması durumunda oluşan maliyet 165 TL dir. Son olarak karma öncelik kullanılması sadece seçilen müşterilere öncelik tanınması halinde maliyeti 385 TL dir. Bu politikalarda oluşan en düşük maliyet saf öncelik var politikasıdır ve Tip 2 müşterilerin toplam kuyrukta bekleme süreleri 8 dakikadır, ancak tip 1 normal müşterilerin toplam kuyrukta bekleme süresi, saf öncelik yok politikasında 31 dakikayken 85 dakikaya çıkmıştır.

4.2 Model

Bu bölümde ele alınan kuyruk kontrol politikası optimizasyonu problemi Markov karar süreci olarak modellenmiştir. Problem modelinde Bölüm 3.1'de bahsedildiği gibi bir durumdan diğer duruma geçiş süreleri üstel dağılıma uyduğundan dolayı SMDP yerine, MDP kullanılmıştır.

- **Durum Uzayı S:**

Sistem durumu (state) s ile gösterilmiş ve $(s_1, s_2) \in s$ demeti ile tanımlanmıştır. Sistem durumu kuyruğa tip 2 müşteri sınıfından bir müşteri geldiği anda kuyrukta bekleyen tip 1 ve tip 2 müşteri sayısı olarak ifade edilmiştir. Tanımlanan s_1 , tip 2 sınıfına ait müşteri geldiği anda kuyrukta bekleyen tip 1 müşteri sayısı, s_2 ise tip 2 sınıfına ait müşteri geldiği anda kuyrukta bekleyen tip 2 müşteri sayısını ifade eder. Sistemin durumu $s = (s_1, s_2)$ şeklinde tanımlanmıştır.

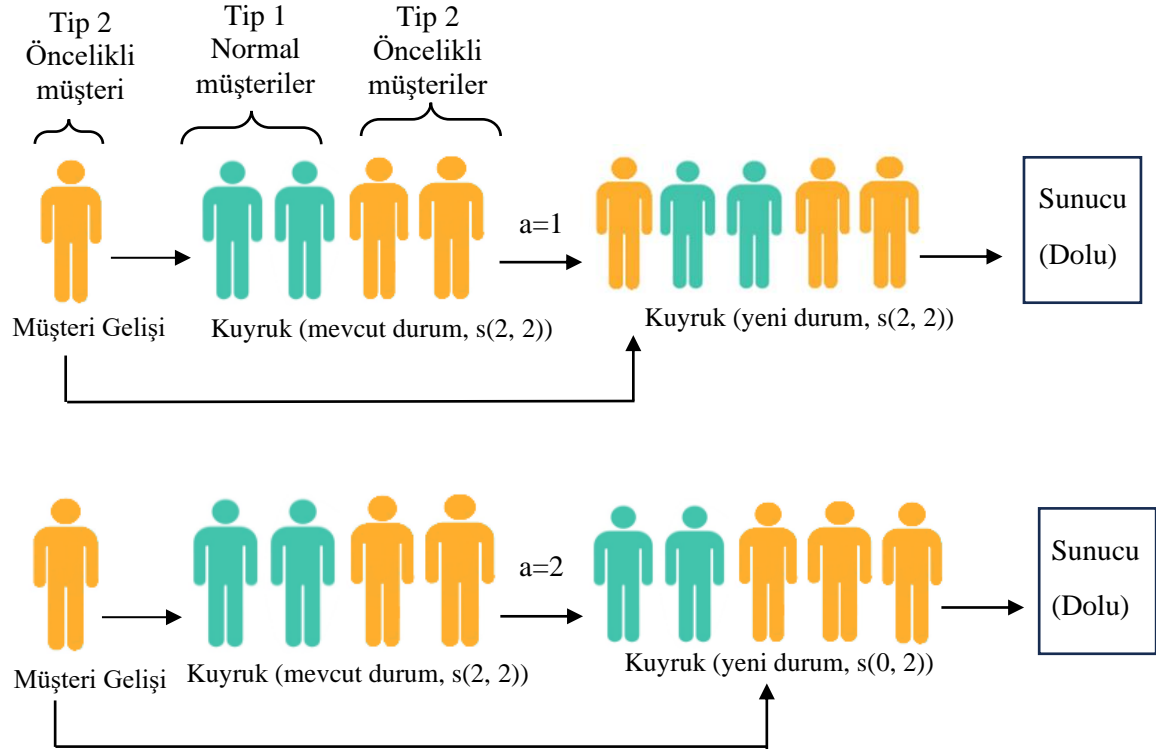
- **Karar Uzayı A:**

Verilen bir $s = (s_1, s_2)$ durumu için geçerli kararlar (actions) kümesi $A(s)$, gelen tip 2 müşterisinin öncelik verilip sırada tip 1 müşterilerinin önüne geçirilip geçirilmeyeceğine dair kararları içerir. Modelde tip 2 müşteri sınıfından müşteri geldiği anda sistemin durumuna göre karar vericinin alabileceği iki tür karar vardır. Bu kararlar; $a=1$ yeni gelen tip 2 müşterisini önceliklendirme yapılmadan kuyruğun sonuna sıraya alma kararı, $a=2$ ise yeni gelen tip 2 müşterisini önceliklendirerek kuyruktaki tip 1 müşterilerinin önüne alma kararı ifade eder. Karar kümesi $A = \{1, 2\}$ ve $a \in A$ olarak ifade edilir.

Örneğin şekil 4.2’de gösterildiği şekilde $s_1 = 2, s_2 = 2$ olsun. Sistemin durumu $(2, 2)$ durumundayken $a=1$ yani yeni gelen tip 2 müşterisini önceliklendirme yapılmadan kuyruğun sonuna sıraya alma kararı seçilirse, gelecek durum yine $s = (2, 2)$ olur.

$s = (2, 2)$ durumundayken $a=2$ yani yeni gelen tip 2 müşterisini önceliklendirerek kuyruktaki tip 1 müşterilerinin önüne alma kararı seçilirse, gelecek durum $s'=(0, 2)$ olur.

$$S = s(2, 2) \begin{cases} a = 1 \text{ kararı seçilirse } s'(2, 2) \text{ olur,} \\ a = 2 \text{ kararı seçilirse } s'(0, 2) \text{ olur.} \end{cases}$$



Şekil 4.2: Öncelikli müşteri geldiği anda alınan iki ayrı kararın etkisi ile oluşan yeni durumun şematik gösterimi.

- **Maliyet Fonksiyonu:**

Sistem (s_1, s_2) durumundayken a kararı altında bir sonraki durum olan $(s_1, s_2)'$ geçildiğinde ortaya çıkan anlık maliyet $r(i, a, j)$ olmaktadır. Karar verici bir a kararını (action) seçtiğinde yani yeni gelen tip 2 müşterisinin önceliklendirip, önceliklendirilmemesine göre kuyrukta bekleyen tip 1 ve tip 2 müşterilerinin kuyrukta bekleme süreleri değişir. Tip 2 müşterinin $a=2$ kararı altında tip 1 müşterilerinin kuyrukta önüne geçmesi durumunda, tip 1 müşterilerinin kuyrukta bekleme süresi uzayacak, tip 2 müşterilerinin kuyrukta bekleme süresi kısılacaktır. Ortaya çıkan bu yeni durumda kuyrukta bekleyen tip 1 ve tip 2 müşterilerinin bekledikleri süre boyunca birim zaman başına sisteme bir maliyeti vardır. h_1 : Tip 1 müşterilerinin birim zaman başına kuyrukta bekleme maliyetini, h_2 : Tip 2 müşterilerinin birim zaman başına kuyrukta bekleme maliyetlerini göstermektedir. w_1 : a kararı alındıktan sonra oluşan kuyruktaki tip 1 müşterilerin bekleme süresi, w_2 : a kararı alındıktan sonra oluşan kuyruktaki tip 2 müşterilerin bekleme süresidir. Tip 2 müşteri geldiğinde $a=2$ kararı altında tip 1 müşterilerin önüne geçmesi durumunda R ceza maliyeti oluşmaktadır. Sistem s durumundayken a kararını verildiğinde oluşan anlık maliyeti Denklem 4.1'de verilmiştir.

$$G(s, a) = w_1 \cdot h_1 + w_2 \cdot h_2 + (a - 1) \cdot R \quad (4.1)$$

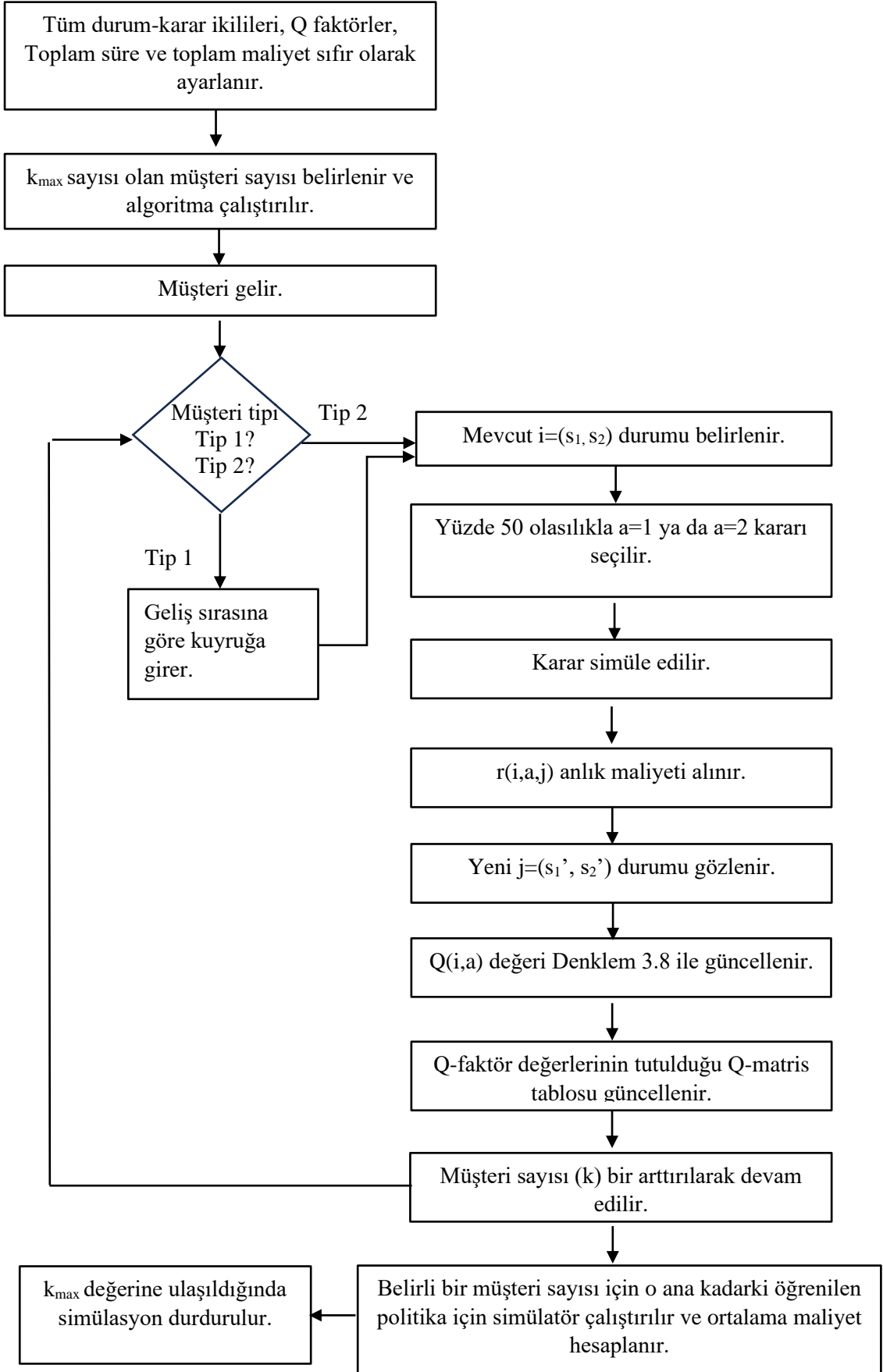
Problemde her bir Tip 1 (normal) müşteri geliş sırasına göre hizmet almaktadır. Tip 2 (öncelikli) müşteriler ise kuyrukta birinci ve ikinci tip müşteri sayısına bağlı olarak uzun dönemde ortalama maliyete göre Tip 1 müşterilerin önüne ya da geliş sırasına göre hizmet alabilmektedir. Probleme ilişkin oluşturulan simülasyonun olay sırası aşağıdaki gibidir:

- 1) Müşteri gelir.
- 2) Müşteri tipi belirlenir.
- 3) Sunucu boşsa, müşteri tipine bağlı olmaksızın müşteri servise alınır.

Sunucu meşgulse;

- i) Müşteri 1. tip (normal) ise geliş sırasına göre kuyruğa girer.
 - ii) Müşteri 2. tip (öncelikli) ise ya geliş sırasına göre kuyruğa girer (karar 1) ya da 1.tip müşterilerin önüne geçer (karar 2).
- 4) Hizmet alan müşteri sistemden ayrılır.

Problemın çözümüne yönelik oluşturulan simülasyonun simülatör olarak kullanıldığı ödüllü öğrenme algoritmasının akış diyagramı Şekil 4.3'te verilmiştir. Ödüllü öğrenme algoritması kullanılarak öncelik kontrol politikası oluşturulmuştur.



Şekil 4.3: Problem çözüm akış diyagramı.

5. SAYISAL ÖRNEKLER

Çalışmanın bu bölümünde modellenen ödüllü öğrenme algoritmasının performansını ve sonuçlarını değerlendirmek için iki farklı sayısal örnek tasarlanmıştır. Sayısal örnek 1’de, sadece kuyrukta bekleme maliyeti söz konusudur. Ödüllü öğrenme algoritmasının öğrenip öğrenmediğini anlamak ve öğrenilen politikaların etkinliğinin analizi için sayısal örnek 1 tasarlanmıştır. Sayısal örnek 2’de ise bekleme maliyetine ek olarak her Tip 2 müşterisinin önceliklendirilmesi durumunda bir ceza maliyeti eklenerek, ceza maliyetinin algoritmanın öğrenme performansını nasıl etkilediği analiz edilmiştir. İki sayısal örnek için ödüllü öğrenme algoritmasıyla belirlenen politikaların birim zamandaki ortalama maliyetleri 3 farklı miyopik politika uygulandığında oluşan birim zamandaki ortalama maliyetleri ile kıyaslanmıştır. Bu miyopik politikalar aşağıdaki gibidir;

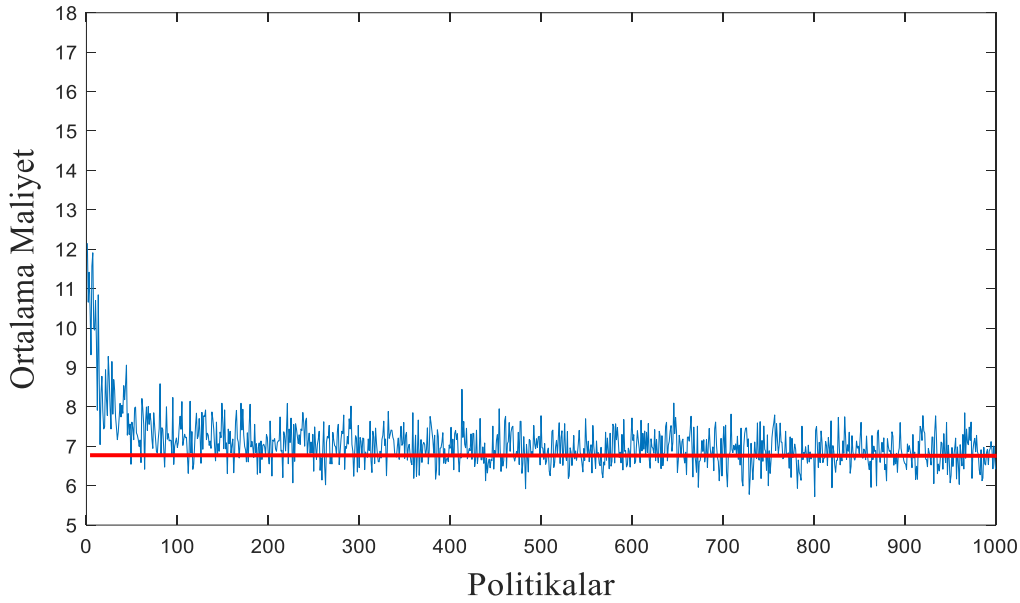
- **Miyopik Politika 1:** $a=1$ kararı uygulandığında Tip 2 müşterilere öncelik verilmemesi ve geliş sırasına göre kuyruğa girme politikası,
- **Miyopik Politika 2:** $a=2$ kararı uygulandığında Tip 2 müşterilere her zaman öncelik verilmesi politikası,
- **Miyopik Politika 3:** %50 olasılıkla $a=1$, %50 olasılıkla $a=2$ kararı: Tip 2 müşterilere yüzde 50 olasılıkla öncelik verilip verilmemesi politikası.

Oluşturulan sayısal örnekler Bölüm 4’te anlatılan problemin tanımı ve modeli etrafında şekillendirilmiştir. Sayısal örnekler için modellenen ödüllü öğrenme algoritması MATLAB R2021a sürümüyle kodlanmış ve simüle edilmiştir. Yineleme sayısı olarak 100.000 müşteri kullanılmakta ve her 100 müşteride bir simülatörde 100 defa öğrenilen politika çalıştırarak ödüllü öğrenmenin o ana kadarki öğrenilen politikaların birim zamandaki ortalama maliyeti hesaplanmaktadır. Belirtilen parametrelerle algoritma çalıştırıldığında Q-faktör matrisinde 359 durum-karar ikilisi oluşmuştur. Hesaplama için kullanılan bilgisayar, Intel(R) Xeon(R) CPU E3-1270 v3 @ 3.50GHz işlemci ve 16 GB RAM'e sahiptir.

5.1 Sayısal Örnek 1

Bu örnekte maliyet fonksiyonuna, sadece bekleme maliyeti dahil edilerek ödüllü öğrenme algoritmasıyla politika elde edilmiştir. Öğrenilen politika ile 3 farklı miyopik politikanın ortalama maliyet grafikleri kıyaslanmıştır.

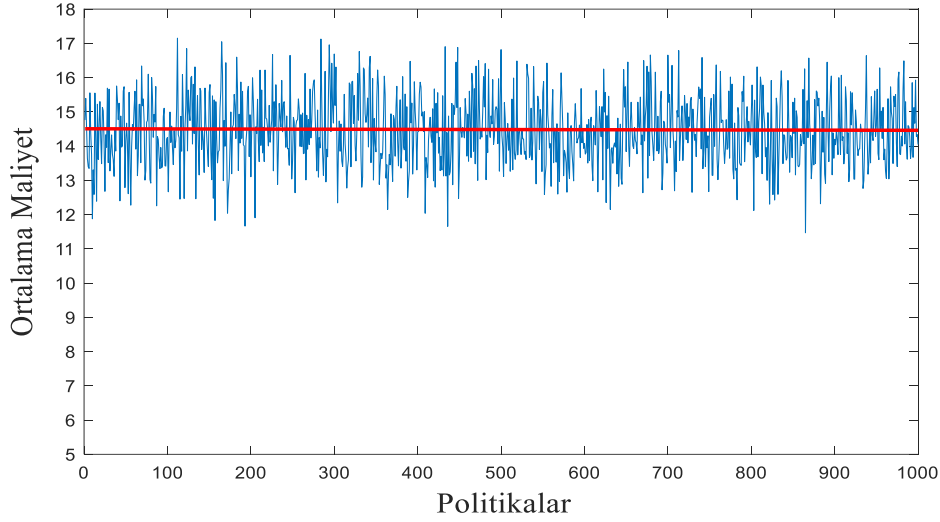
Tip 1 müşteri, normal müşteridir ve $\lambda_1=0.7$ müşteri/dakika parametrelili Poisson Sürecine göre gelmektedir. Tip 2 müşteri, öncelikli müşteridir ve lamda $\lambda_2=0.3$ müşteri/dakika parametrelili Poisson sürecine göre gelmektedir. Her iki tip müşteri türünün de servis oranı üstel dağılımlı $\mu=0.9$ müşteri/dakikadır. Tip 1 müşterilerin kuyrukta bekleme maliyetleri $h_1= 1$ TL/ad/zaman ve Tip 2 müşterilerin kuyrukta bekleme maliyetleri $h_2= 10$ TL/ad/zaman olarak belirlenmiştir. Algoritmanın her Tip 2 müşterisinin geldiği an da uygulayabileceği iki tür karar vardır: Tip 2 müşterisini geliş sırasına göre kuyruğa al, Tip 2 müşterisini kuyrukta Tip 1 müşterilerin önüne al. Ödüllü öğrenme algoritmasında kullanılan iterasyon sayısı olarak 100.000 müşteri olarak belirlenmiş ve rastgele (yüzde 50 olasılıkla) iki karardan biri seçilmiştir. Simülatörde her 100 müşteride bir ödüllü öğrenmenin o zamana kadarki öğrenilen politikanın birim zamandaki ortalama maliyeti hesaplanmıştır. Modellenen ödüllü öğrenme algoritmasının kullanılmasıyla elde edilen öğrenilen politikaların birim zamandaki ortalama maliyetlerini gösteren grafik, Şekil 5.1'de verilmiştir.



Şekil 5.1: Sayısal örnek 1'de ödüllü öğrenme algoritmasıyla elde edilen politikaların yakınsama grafiği.

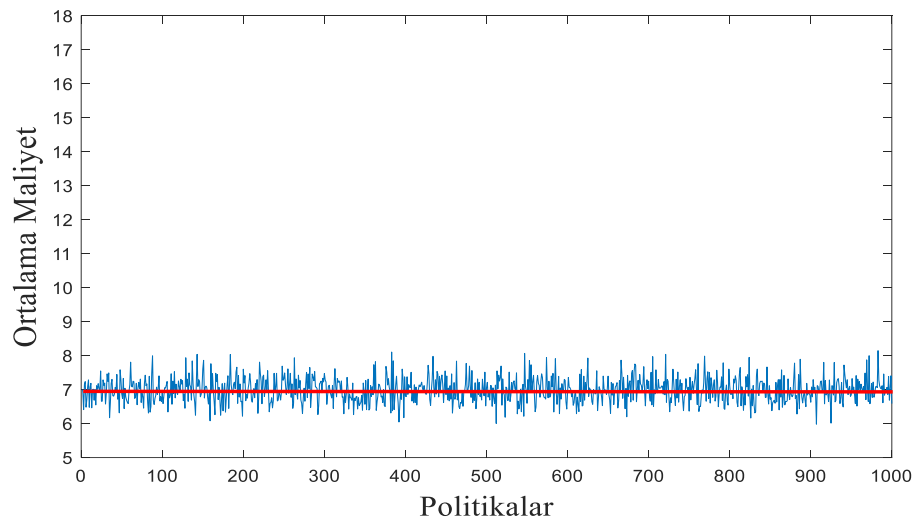
Şekil 5.1'de görüldüğü üzere öğrenilen politikaların birim zamandaki ortalama maliyeti 12'lerde başlamış ancak algoritmanın öğrenmesiyle birlikte politikaların maliyetleri çok hızlı bir biçimde yaklaşık olarak 7-8 bandına kadar düşmüş ve 7 değerine yakınsamıştır. Şekil 5.2'de ikinci tip müşterilere öncelik verilmediği, geldikleri sırada kuyruğa girme kararı olan $a=1$ kararı politikasının birim zamandaki ortalama maliyet grafiği verilmiştir.

Şekil 5.2’de görüldüğü gibi miyopik politika 1’in birim zamandaki ortalama maliyeti 14,5 TL’ye yakınsamaktadır. Tip 1 normal ve Tip 2 öncelikli müşterilerin kuyrukta birim zaman başına bekleme maliyetleri sırasıyla 1 ve 10 TL olduğu için kuyrukta Tip 2 müşterilere öncelik verilmemesi bekleme maliyetini artırmaktadır.



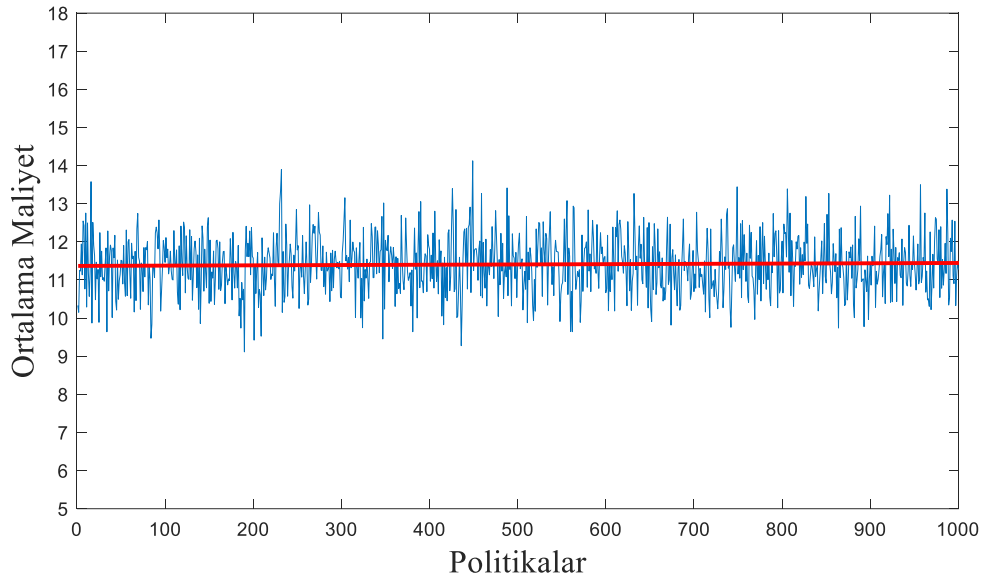
Şekil 5.2: Sayısal örnek 1’de miyopik politika 1’in yakınsama grafiği.

Şekil 5.3’te miyopik politika 2’nin birim zamandaki ortalama maliyet grafiği verilmiştir. Şekilde görüldüğü üzere miyopik politika 2’nin birim zamanda ortalama maliyeti 7 TL’ye yakınsamaktadır. Tip 2 müşterinin kuyrukta bekleme maliyeti, Tip 1 müşterisinin kuyrukta bekleme maliyetinden yüksek olduğu için algoritma, ikinci tip müşterilere öncelik vermesi durumunda miyopik politika 1’den daha düşük bir ortalama maliyete sahiptir.



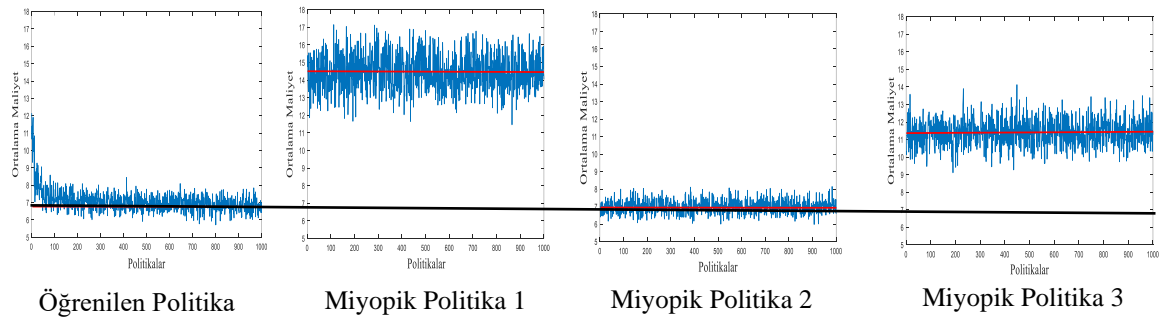
Şekil 5.3: Sayısal örnek 1’de miyopik politika 2’nin yakınsama grafiği.

Şekil 5.4'te Tip 2 müşterilere öncelik verilir verilmemesine yüzde 50 olasılıkla olan a=rassal kararı politikasının (miyopik politika 3) birim zamandaki ortalama maliyet grafiği verilmiştir. Şekilde görüldüğü üzere miyopik politika 3'ün birim zamandaki ortalama maliyeti 14,5 TL'ye yakınsamaktadır. Tip 1 normal ve Tip 2 öncelikli müşterilerin kuyrukta birim zaman başına bekleme maliyetleri sırasıyla 1 ve 10 TL olduğu için kuyrukta Tip 2 müşterilere öncelik verilmemesi politikalarının ortalama bekleme maliyetini artırmaktadır.



Şekil 5.4: Sayısal örnek 1'de miyopik politika 3'ün yakınsama grafiği.

Şekil 5.5'te ödüllü öğrenme algoritmasıyla elde edilen politikaların ve 3 miyopik politikanın ortalama maliyet grafikleri birlikte verilmiştir.



Şekil 5.5: Sayısal örnek 1'de öğrenilen politikayla miyopik politika 1, 2 ve 3'ün kıyaslanması.

Sonuç olarak politikaların birim zamandaki ortalama maliyetleri Şekil 5.5'te kıyaslanmıştır. Şekilden görüldüğü ve beklenildiği gibi miyopik politika 1 olan Tip 2 müşterilere öncelik verilmediği politika diğer politikalar arasında en yüksek ortalama maliyet değerine sahip olmuştur. Öncelik durumunun rassal karar bağlandığı miyopik politika 3 ise miyopik politika 1 nispeten daha iyi sonuç vermiştir. Öğrenilen politika ve Tip 2 müşterilerine daima öncelik verildiği miyopik politika 2'nin birim zamandaki ortalama maliyetleri birbirini çok yakındır. Tip 2 müşterilerin birim zamandaki bekleme maliyetleri olan h_2 değeri, h_1 e göre 10 katı yüksek değer verildiği için ödüllü öğrenme ile elde edilen politika ve miyopik politika 2'nin sonuçlarının benzer çıkması normal karşılanmaktadır. Tip 2 müşterinin kuyrukta bekleme maliyeti, Tip 1 müşterisinin kuyrukta bekleme maliyetinden yüksek olduğu için algoritma ikinci tip müşterilere öncelik verme eğilimindedir. Öğrenilen politikanın ve Tip 2 müşterilere her zaman öncelik verildiği miyopik politika 2'nin birim zamandaki ortalama bekleme maliyetleri yakınsamaları birbirine yakın sonuçlar çıkmış olsa da ufak bir farkla öğrenilen algoritmanın politika maliyetleri daha iyi yakınsama grafiği çıktığı görülmektedir.

5.2 Sayısal Örnek 2

Literatürde de yer alan çalışmalara bakıldığında farklı bekleme maliyetleriyle yönetilmeye çalışılan öncelikli müşteri kuyrukları olduğu gibi, farklı bekleme maliyetlerine ek olarak öncelikli her müşterinin önceliksiz müşterilerin önüne geçirilmesi ya da terfi ettirmesi için sabit bir R ücreti ödenen sistemlerde vardır. Kofman'nın [11] yaptığı çalışmada bekleme maliyetlerinin dışında iki tip müşteri arasında, öncelikli her bir müşteri öne alındığında sabit bir R maliyeti eklenmiştir. Bu sayısal örnekte de kuyrukta bekleme maliyetinin dışında her ikinci tip müşteriyi öne geçirdiğimizde sabit bir ceza maliyetinin söz konusu olduğu problem ele alınmıştır. Tasarladığımız örnekte ceza maliyetinin 10 TL, 20 TL ve 30 TL olduğu durumlar ayrı ayrı incelenmiştir.

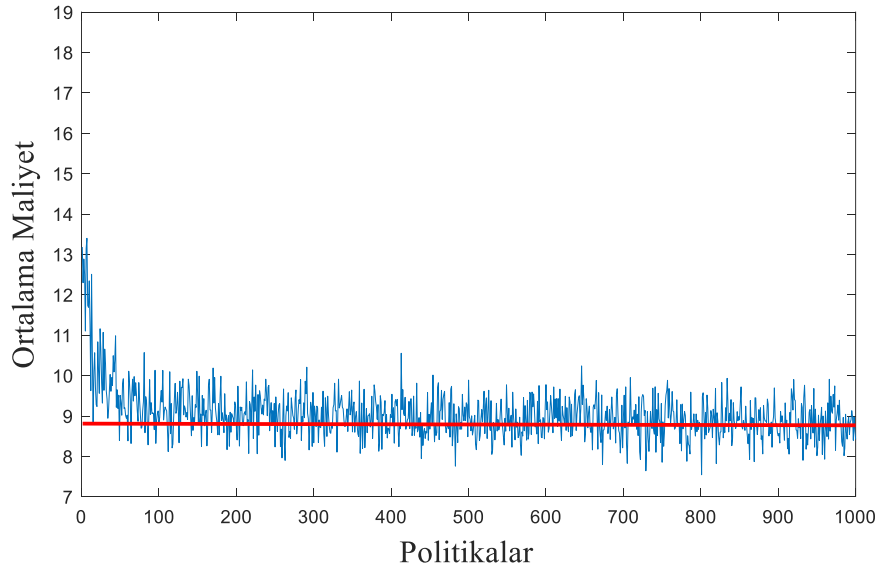
Sayısal örnek 1'deki parametreler kullanılmıştır. Tip 1 müşteri, normal müşteridir ve $\lambda_1=0.7$ müşteri/dakika parametrelili Poisson Sürecine göre gelmektedir. Tip 2 müşteri, öncelikli müşteridir ve lamda $\lambda_2=0.3$ müşteri/dakika parametrelili Poisson sürecine göre gelmektedir. Tip 1 müşterilerin kuyrukta bekleme maliyetleri $h_1= 1$ TL/ad/zaman ve Tip 1 müşterilerin kuyrukta bekleme maliyetleri $h_2= 10$ TL/ad/zaman dır. Ek olarak her Tip 2 müşterisini Tip 1 normal müşterilerinin önüne geçirmenin cezası olan sabit R TL/müşteri

maliyeti varsayılmıştır. İkinci tip müşterilerine her öncelik verildiğinde R maliyeti ödenmesi gereken bir ceza olarak düşünülmüştür. Toplam maliyet, bekleme maliyeti ve her ikinci tip öncelik verildiğinde ödenen ceza maliyetidir.

Ödüllü öğrenme algoritması her R maliyeti için ayrı ayrı çalıştırılmıştır. Her algoritma, yaklaşık optimal kuyruk kontrol politikasını ortalama 93 saatte bulmuştur. Her bir algortmada yineleme sayısı olarak 100.000 müşteri kullanılmış ve karar seçimi için rastgele yüzde 50 olasılık uygulanmıştır. Simülatörde her 100 müşteride bir ödüllü öğrenmenin o ana kadarki öğrenilen politikanın birim zamandaki ortalama maliyeti hesaplanmıştır.

5.2.1 R=10 Durumunda Oluşan Maliyetler

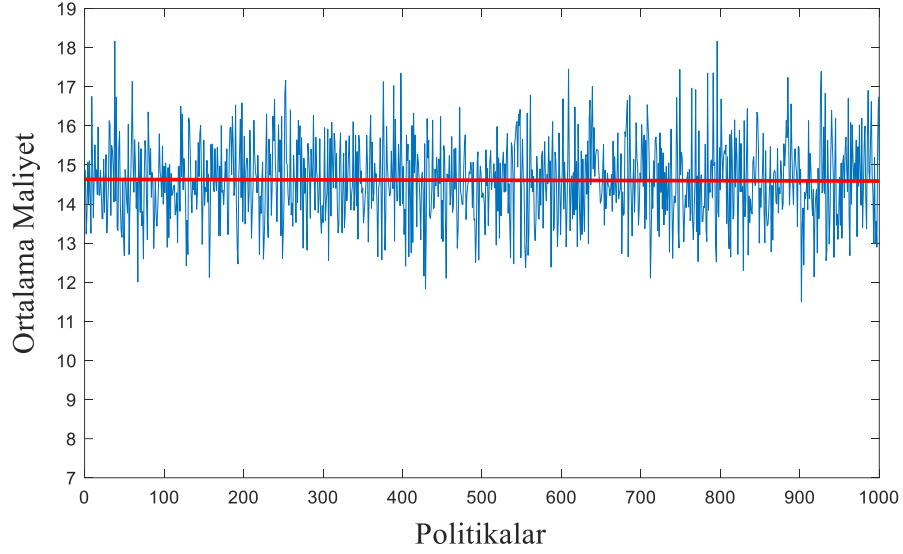
Sayısal örnek 2’de tasarlanan parametreler ile her Tip 2 müşteriye öncelik verildiğinde eklenecek ceza maliyetinin 10 TL/müşteri olduğu durumlar simüle edilmiştir. Ödüllü öğrenme algoritmasının kullanılmasıyla elde edilen öğrenilen politikaların birim zamandaki ortalama maliyet grafiği Şekil 5.6’da verilmiştir.



Şekil 5.6: R=10 ceza maliyetli örnekte ödüllü öğrenme algoritmasıyla elde edilen politikaların yakınsama grafiği.

Şekil 5.6’de görüldüğü üzere öğrenilen politikaların birim zamandaki ortalama maliyeti 13-14 TL bandında başlamış, ancak algoritmanın öğrenmesiyle algoritma tarafından üretilen politikaların birim zamandaki ortalama maliyetleri çok hızlı bir biçimde yaklaşık

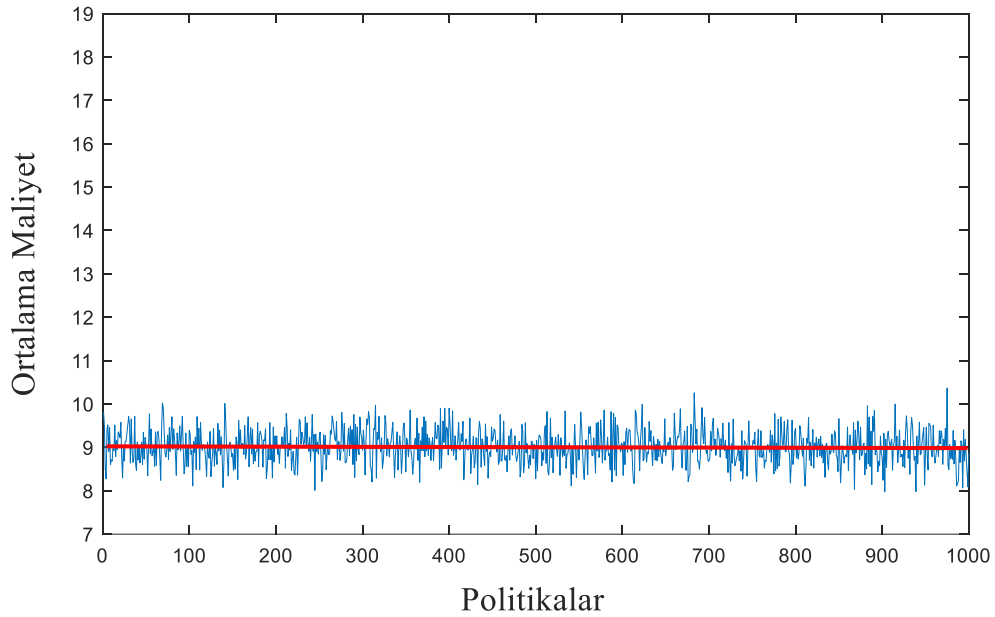
olarak 9 TL değerinin altına yakınsamıştır. Şekil 5.7’de miyopik politika 1’in birim zamandaki ortalama maliyet grafiği verilmiştir.



Şekil 5.7: R=10 ceza maliyetli örnekte miyopik politika 1'in yakınsama grafiği.

Şekil 5.7’den de görüleceği gibi miyopik politika 1’in birim zamandaki ortalama maliyeti 14,5 TL’ye yakınsamaktadır, ancak birim zamandaki ortalama maliyet yakınsama grafiği sonucunda elde edilen politikaların maliyetleri geniş bir aralıkta (11-18) salınmaktadır.

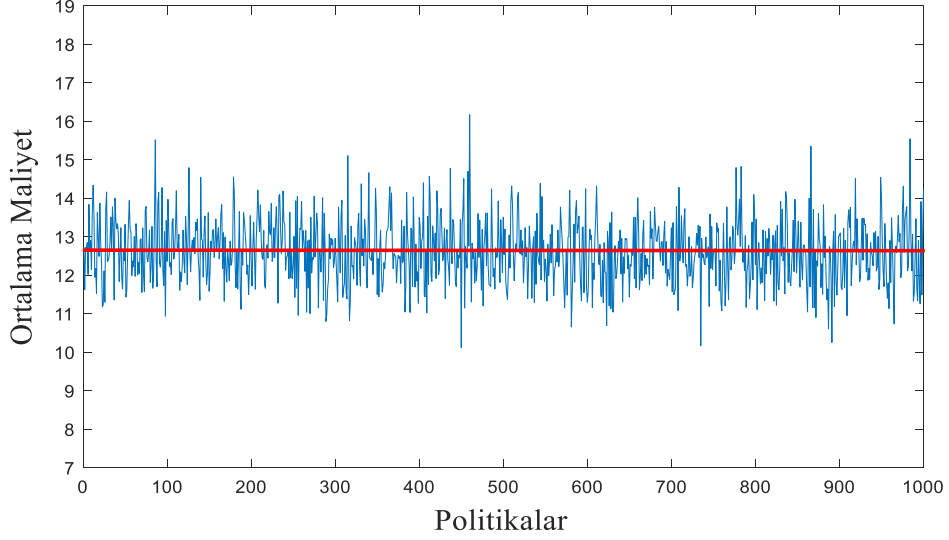
Şekil 5.8’de miyopik politika 2’nin birim zamandaki ortalama maliyet grafiği verilmiştir.



Şekil 5.8: R=10 ceza maliyetli örnekte miyopik politika 2'nin yakınsama grafiği.

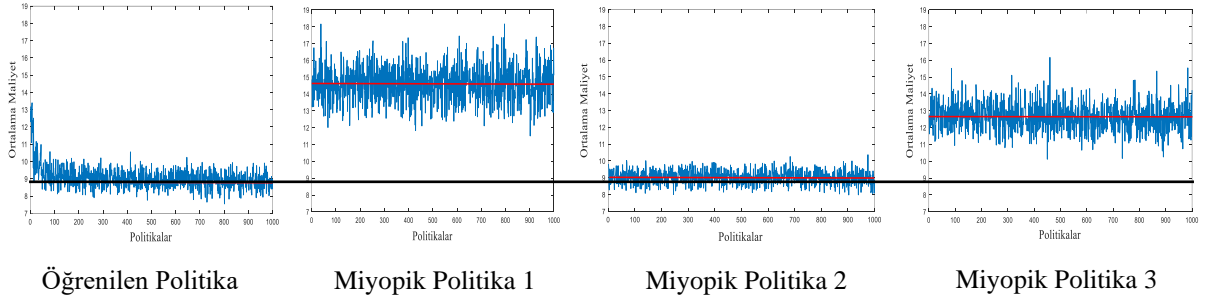
Şekil 5.8’de yer alan grafikte gözlemlendiği üzere miyopik politika 2’nin birim zamanda ortalama maliyeti ortalama 9 TL’ye yakınsamaktadır.

Şekil 5.9’da miyopik politika 3’ün birim zamandaki ortalama maliyet grafiği verilmiştir.



Şekil 5.9: R=10 ceza maliyetli örnekte miyopik politika 3’ün yakınsama grafiği.

Şekil 5.9’da gösterildiği üzere miyopik politika 3’ün birim zamandaki ortalama maliyeti 13 TL civarındadır.



Şekil 5.10: R=10 ceza maliyetli sayısal örnek 2’in öğrenilen politika ile miyopik politika 1, 2 ve 3’ün kıyaslanması.

Şekil 5.10’da R= 10 TL değeri için elde edilen öğrenilen politika ve miyopik politikaların ortalama maliyetlerini gösteren birim zamandaki ortalama maliyet yakınsama grafikleri birlikte verilerek kıyaslanmıştır. Şekilde görüldüğü üzere R=10 TL ceza maliyeti uygulanması durumunda miyopik politika 1 ve miyopik politika 3’ün birim zamandaki

ortalama maliyetleri, öğrenilen politikanın ortalama maliyetinin üstünde kalmıştır. Öğrenilen politikanın ve miyopik politika 2'nin birim zamandaki ortalama maliyetleri birbirine çok yakın değerler çıkmıştır. Öğrenilen politikanın ortalama maliyeti 9 TL'nin biraz altında, miyopik politika 2'nin ortalama maliyeti ise 9 TL civarında olduğu görülmektedir.

Şekil 5.11'de görüldüğü üzere R=10 TL ceza maliyeti uygulanması durumunda öğrenilen politikanın ortalama maliyeti ile miyopik politika 2'nin ortalama maliyeti birbirine yakındır. Öğrenilen politikanın ortalama maliyetinin miyopik politika 2'nin ortalama maliyetinden daha düşük olup olmadığını ilişkin hipotez testi yapılmıştır. Öncelikle iki ana kitlenin standart sapmalarının eşitliğine ilişkin test gerçekleştirilmiştir.

İki ana kitlenin standart sapmalarının eşitliğine ilişkin hipotezler ve Minitab'da gerçekleştirilen teste ilişkin sonuçlar Tablo 5.1'de verilmiştir.

σ_1 : Öğrenilen Politika Ana Kitle Standart Sapma (R=10)

σ_2 : Miyopik Politika 2 Ana Kitle Standart Sapma (R=10)

$$H_0: \frac{\sigma_1}{\sigma_2} = 1$$

$$H_1: \frac{\sigma_1}{\sigma_2} \neq 1$$

Tablo 5.1: Öğrenilen politika/miyopik politika 2 (R=10) standart sapma eşitliğine ilişkin hipotez testi Minitab sonuçları

Method				
σ_1 : standard deviation of Öğrenilen Politika (R=10)				
σ_2 : standard deviation of Miyopik Politika 2 (R=10)				
Ratio: σ_1/σ_2				
The Bonett and Levene's methods are valid for any continuous distribution.				
Descriptive Statistics				
Variable	N	StDev	Variance	95% CI for σ
Öğrenilen Politika (R=10)	800	0,462	0,213	(0,441; 0,485)
Miyopik Politika 2 (R=10)	800	0,376	0,141	(0,358; 0,395)
Test				
Null hypothesis	Ho: $\sigma_1 / \sigma_2 = 1$			
Alternative hypothesis	H1: $\sigma_1 / \sigma_2 \neq 1$			
Significance level	$\alpha = 0,05$			
Test				
Method	Statistic	DF1	DF2	P-Value
Bonett	26,31	1		0,000
Levene	23,01	1	1598	0,000

Tablo 5.1'den görüldüğü üzere hesaplanan P-Değeri=0'dır. Buna göre bu örneklem verisiyle H_0 hipotezi reddedildiğinde yapılan 1.Tip hata olasılığı 0'dır. Bu nedenle H_0 hipotezi reddedilir, iki ana kitle standart sapması birbirinden farklıdır. İki ana kitle ortalama kıyaslaması ana standart sapmalarının farklı olma durumuna göre gerçekleştirilecektir.

Öğrenilen politikanın ortalama maliyetinin Miyopik Politika 2'nin ortalama maliyetinden daha düşük olup/olmadığına ilişkin aşağıda verilen hipotezler kurulmuş ve Minitab test sonucu Tablo 5.2'de verilmiştir.

μ_1 : Öğrenilen politikanın ortalama maliyeti (R=10)

μ_2 : Miyopik Politika 2'nin ortalama maliyeti (R=10)

H_0 : $\mu_1 = \mu_2$

H_1 : $\mu_1 < \mu_2$

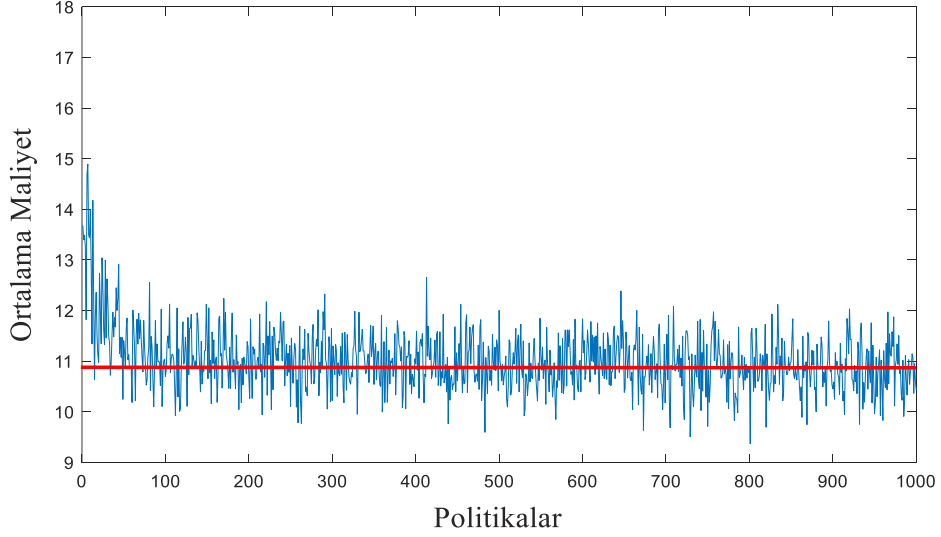
Tablo 5.2: Öğrenilen politika/miyopik politika 2 (R=10) ortalama maliyet hipotez testi
Minitab sonuçları

Descriptive Statistics				
Sample	N	Mean	StDev	SE Mean
Öğrenilen	800	8,930	0,462	0,016
Politika (R=10)				
Miyopik	800	8,996	0,399	0,013
Politika 2 (R=10)				
Test				
Null hypothesis $H_0: \mu_1 - \mu_2 = 0$				
Alternative hypothesis $H_1: \mu_1 - \mu_2 < 0$				
T-Value	DF	P-Value		
-3,13	1534	0,000		

Testin sonucunda P-Değeri=0 olarak bulunmuştur. Buna göre bu örneklem verisiyle H_0 hipotezi reddedildiğinde yapılacak 1. Tip hata olasılığı 0'dır. Bu nedenle H_0 hipotezi reddedilir, ceza maliyeti 20 TL olduğu durumda öğrenilen politikanın ortalama maliyetinin, miyopik politika 2'nin ortalama maliyetinden daha düşük olduğuna dair istatistiksel kanıt vardır.

5.2.2 R=20 Durumunda Oluşan Maliyetler

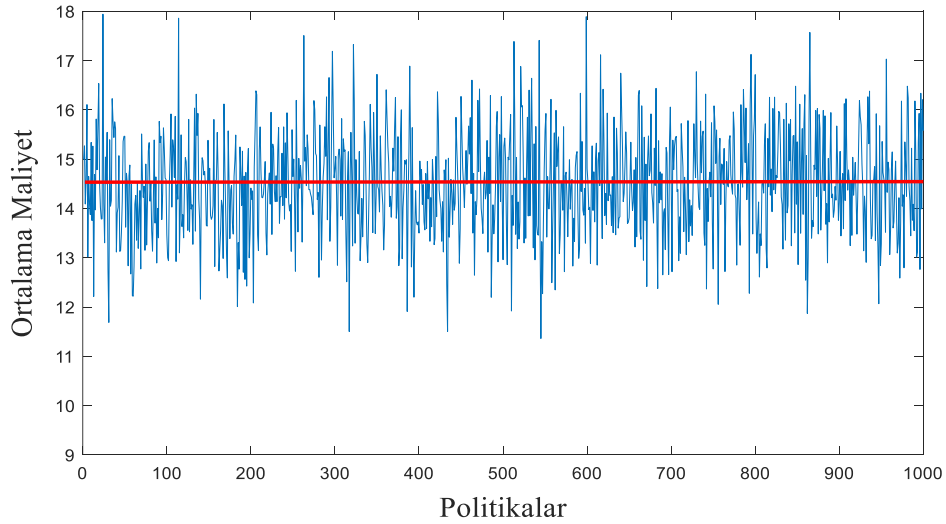
Sayısal örnek 2'de tasarlanan parametreler ile her Tip 2 müşteriye öncelik verildiğinde eklenecek ceza maliyetinin 20 TL olduğu durumlar simüle edilmiştir. Modellenen ödüllü öğrenme algoritmasının kullanılmasıyla elde edilen öğrenilen politikaların birim zamandaki ortalama maliyet grafiği Şekil 5.11'de verilmiştir.



Şekil 5.11: R=20 ceza maliyetli örnekte ödüllü öğrenme algoritmasıyla elde edilen politikaların yakınsama grafiği.

Şekil 5.11’de yer alan grafikte, öğrenilen politikaların birim zamandaki ortalama maliyeti 15 TL değerlerinde başlamış, ancak algoritmanın öğrenmesiyle algoritma tarafından üretilen politikaların birim zamandaki ortalama maliyetleri çok hızlı bir biçimde yaklaşık olarak 11 TL değerinin altına yakınsadığı görülmüştür.

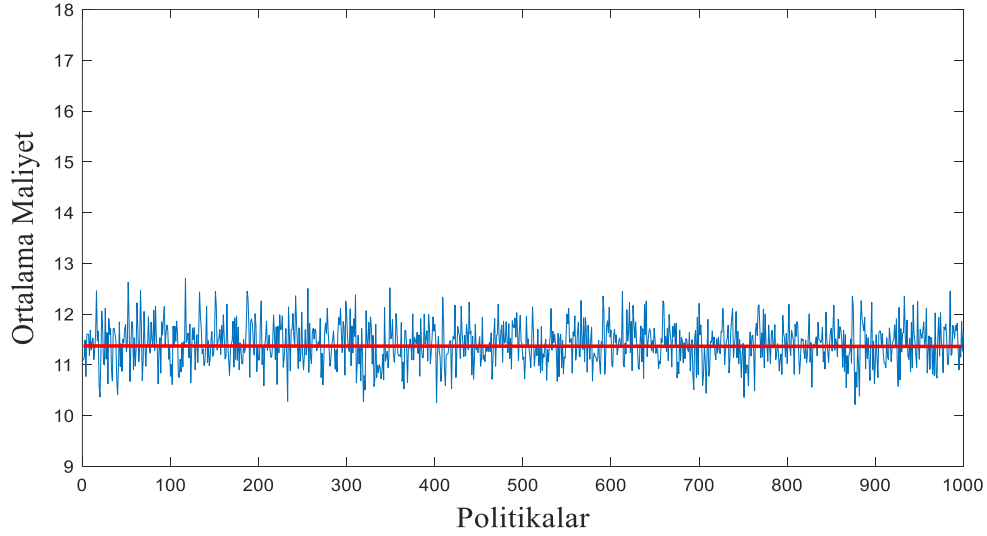
Şekil 5.12’de miyopik politika 1’in birim zamandaki ortalama maliyet grafiği verilmiştir.



Şekil 5.12: R=20 ceza maliyetli örnekte miyopik politika 1’in yakınsama grafiği.

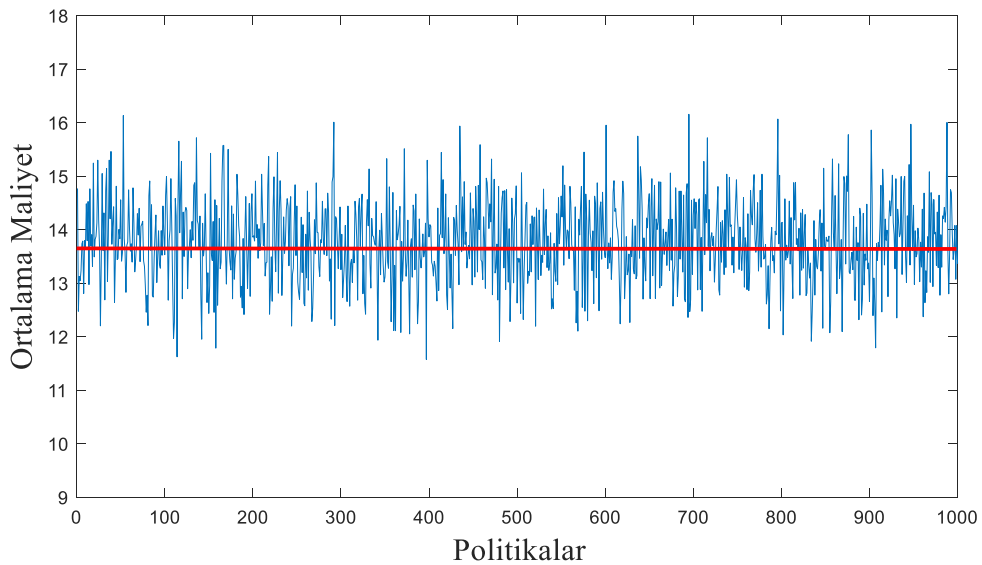
Şekil 5.12’de görüldüğü üzere miyopik politika 1’in birim zamandaki ortalama maliyeti 14,5 TL civarındadır. Elde edilen politikaların maliyetleri çok geniş bir aralıkta (11-18) salınmaktadır.

Şekil 5.13’te miyopik politika 2’nin birim zamandaki ortalama maliyet grafiği verilmiştir. Grafikte miyopik politika 2’nin birim zamanda ortalama maliyeti ortalama 11,5 TL’ye yakınsadığı görülmektedir.



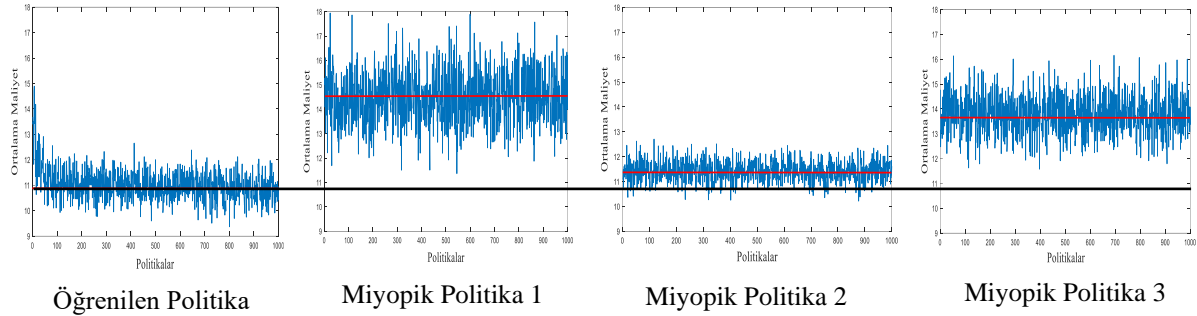
Şekil 5.13: R=20 ceza maliyetli örnekte miyopik politika 2’nin yakınsama grafiği.

Şekil 5.14’te miyopik politika 3’nin birim zamandaki ortalama maliyet grafiği verilmiştir.



Şekil 5.14: R=20 ceza maliyetli örnekte miyopik politika 3’ün yakınsama grafiği.

Şekil 5.14'te gösterildiği üzere miyopik politika 3'ün birim zamandaki ortalama maliyeti 14 TL'nin biraz altındadır. Ortalama maliyet 12-16 TL arasında salınım yapmaktadır.



Şekil 5.15: R=20 ceza maliyetli sayısal örnek 2'in öğrenilen politika ile miyopik politika 1, 2 ve 3'ün kıyaslanması.

Şekil 5.15'te R= 20 TL değeri için elde edilen öğrenilen politika ve miyopik politikaların ortalama maliyetlerini gösteren birim zamandaki ortalama maliyet yakınsama grafikleri birlikte verilerek kıyaslanmıştır. Şekilde görüldüğü üzere R=20 TL ceza maliyeti eklenmesi durumunda miyopik politika 1 ve miyopik politika 3'ün birim zamandaki ortalama maliyetleri, öğrenilen politikanın ortalama maliyetinin çok üstünde kalmıştır. Öğrenilen politikanın ve miyopik politika 2'nin birim zamandaki ortalama maliyetleri arasındaki fark belirginleşmiştir. Öğrenilen politikanın ortalama maliyeti 11 TL'nin biraz altında, miyopik politika 2'nin ortalama maliyeti ise 11,5 TL civarında olduğu görülmektedir. Birim zamandaki en düşük ortalama maliyetin öğrenilen politika ile elde edildiği Şekil 5.15'te görülmektedir.

Şekil 5.15'te görüldüğü üzere R=20 TL ceza maliyeti uygulanması durumunda öğrenilen politikanın ortalama maliyeti ile miyopik politika 2'nin ortalama maliyeti birbirine yakındır. Öğrenilen politikanın ortalama maliyetinin Miyopik Politika 2'nin ortalama maliyetinden daha düşük olup olmadığını ilişkin hipotez testi yapılmıştır. Öncelikle iki ana kitlenin varyanslarının eşitliğine ilişkin test gerçekleştirilmiştir.

İki ana kitlenin standart sapmalarının eşitliğine ilişkin hipotezler ve Minitab'da gerçekleştirilen teste ilişkin sonuçlar Tablo 5.3'te verilmiştir.

σ_1 : Öğrenilen Politika Ana Kitle Standart Sapma (R=20)

σ_2 : Miyopik Politika 2 Ana Kitle Standart Sapma (R=20)

$$H_0: \frac{\sigma_1}{\sigma_2} = 1$$

$$H_1: \frac{\sigma_1}{\sigma_2} \neq 1$$

Tablo 5.3: Öğrenilen politika/miyopik politika 2 (R=20) standart sapma eşitliğine ilişkin hipotez testi Minitab sonuçları

Method				
σ_1 : standard deviation of Öğrenilen Politika (R=20)				
σ_2 : standard deviation of Miyopik Politika 2 (R=20)				
Ratio: σ_1/σ_2				
The Bonett and Levene's methods are valid for any continuous distribution.				
Descriptive Statistics				
Variable	N	StDev	Variance	95% CI for σ
Öğrenilen Politika (R=20)	800	0,508	0,258	(0,485; 0,533)
Miyopik Politika 2 (R=20)	800	0,425	0,181	(0,407; 0,446)
Test				
Null hypothesis	H ₀ : $\sigma_1 / \sigma_2 = 1$			
Alternative hypothesis	H ₁ : $\sigma_1 / \sigma_2 \neq 1$			
Significance level	$\alpha = 0,05$			
Test				
Method	Statistic	DF1	DF2	P-Value
Bonett	26,31	1		0,000
Levene	23,01	1	1598	0,000

P-Değeri=0 olduğu için H₀ hipotezi bu örneklem verisiyle reddedildiğinde yapılan 1.Tip hata 0'dır. Bu nedenle H₀ hipotezi reddedilir, iki ana kitle standart sapması birbirinden farklıdır. İki ana kitle ortalama kıyaslaması, ana kitle standart sapmasından farklı olma durumuna göre gerçekleştirilecektir.

Öğrenilen politikanın ortalama maliyetinin Miyopik Politika 2'nin ortalama maliyetinden daha düşük olup/olmadığına ilişkin aşağıda verilen hipotezler kurulmuş ve Minitab test sonucu Tablo 5.4'te verilmiştir.

μ_1 : Öğrenilen politikanın ortalama maliyeti (R=20)

μ_2 : Miyopik Politika 2'nin ortalama maliyeti (R=20)

$$H_0: \mu_1 = \mu_2$$

$$H_1: \mu_1 < \mu_2$$

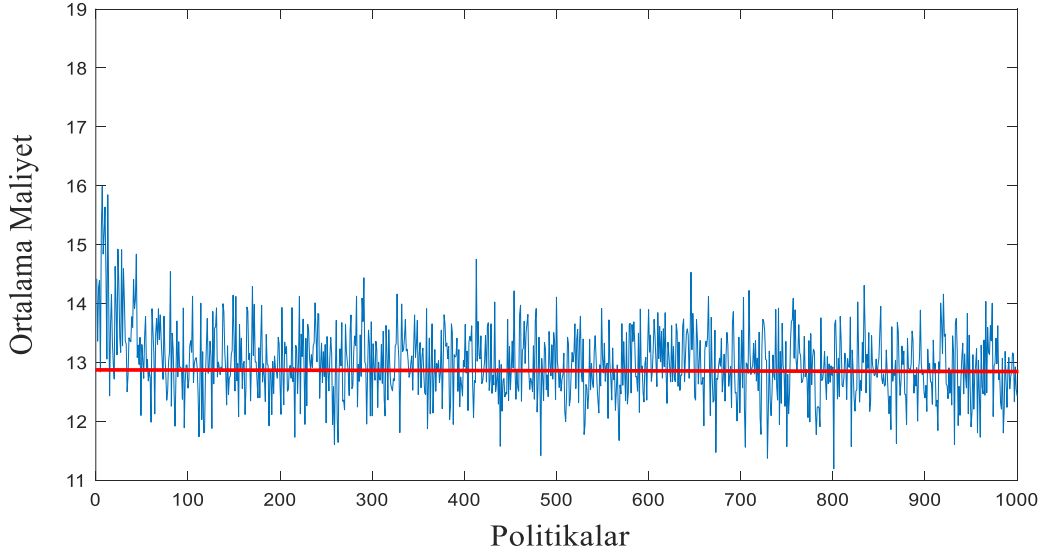
Tablo 5.4: Öğrenilen politika/miyopik politika 2 (R=20) ortalama maliyet hipotez testi Minitab sonuçları

Descriptive Statistics				
Sample	N	Mean	StDev	SE Mean
Öğrenilen Politika (R=20)	800	8,930	0,508	0,018
Miyopik Politika (R=20)	800	8,996	0,425	0,015
Test				
Null hypothesis $H_0: \mu_1 - \mu_2 = 0$				
Alternative hypothesis $H_1: \mu_1 - \mu_2 < 0$				
T-Value	DF	P-Value		
-19,23	1550	0,000		

Testin sonucunda P-Değeri=0 olarak bulunmuştur. Buna göre bu örneklem verisiyle H_0 hipotezi reddedildiğinde yapılacak 1. Tip hata olasılığı 0'dır. Bu nedenle H_0 hipotezi reddedilir, ceza maliyeti 20 TL olduğu durumda öğrenilen politikanın ortalama maliyetinin, miyopik politika 2'nin ortalama maliyetinden daha düşük olduğuna dair istatistiksel kanıt vardır.

5.2.3 R=30 Durumunda Oluşan Maliyetler

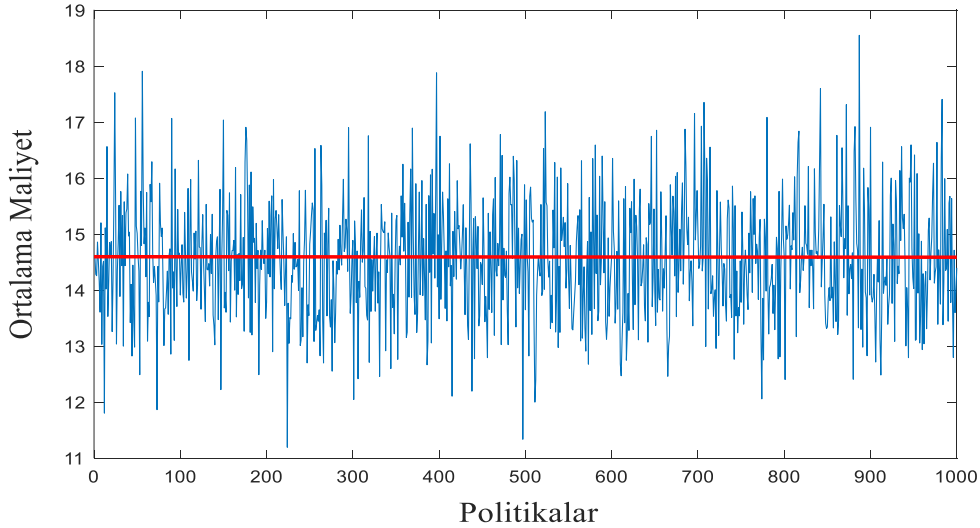
Sayısal örnek 2'de tasarlanan parametreler ile her Tip 2 müşteriye öncelik verildiğinde eklenecek ceza maliyetinin 30 TL/müşteri olduğu durumlar simüle edilmiştir. Modellenen ödüllü öğrenme algoritmasının kullanılmasıyla elde edilen öğrenilen politikaların birim zamandaki ortalama maliyet grafiği Şekil 5.16'da verilmiştir.



Şekil 5.16: R=30 ceza maliyetli örnekte ödüllü öğrenme algoritmasıyla elde edilen politikaların yakınsama grafiği.

Şekil 5.16’da grafikte öğrenilen politikaların birim zamandaki ortalama maliyeti 16 TL değerlerinde başlamış, ancak algoritmanın öğrenmesiyle algoritma tarafından üretilen politikaların birim zamandaki ortalama maliyetleri hızlı bir biçimde yaklaşık olarak 13 TL değerinin altına yakınsamıştır.

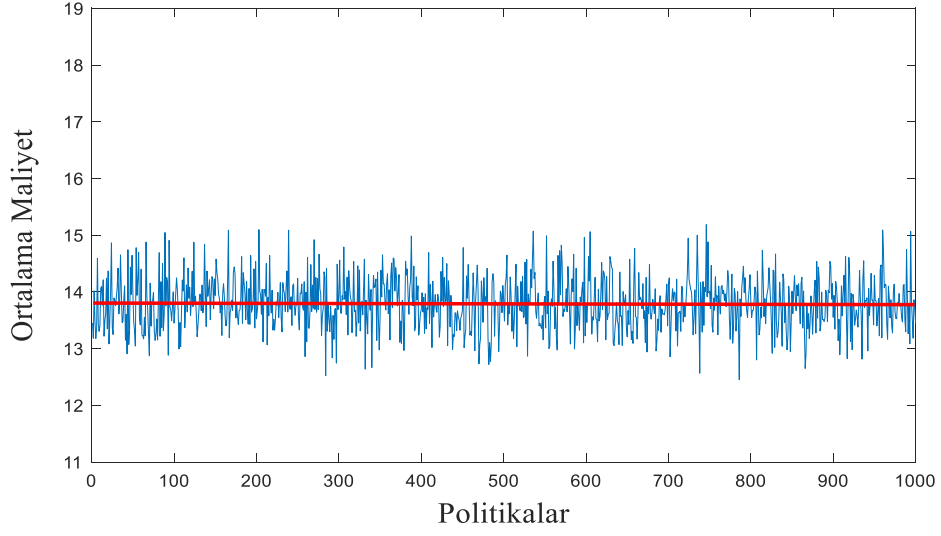
Şekil 5.17’de miyopik politika 1’in birim zamandaki ortalama maliyet grafiği verilmiştir.



Şekil 5.17: R=30 ceza maliyetli örnekte miyopik politika 1’in yakınsama grafiği.

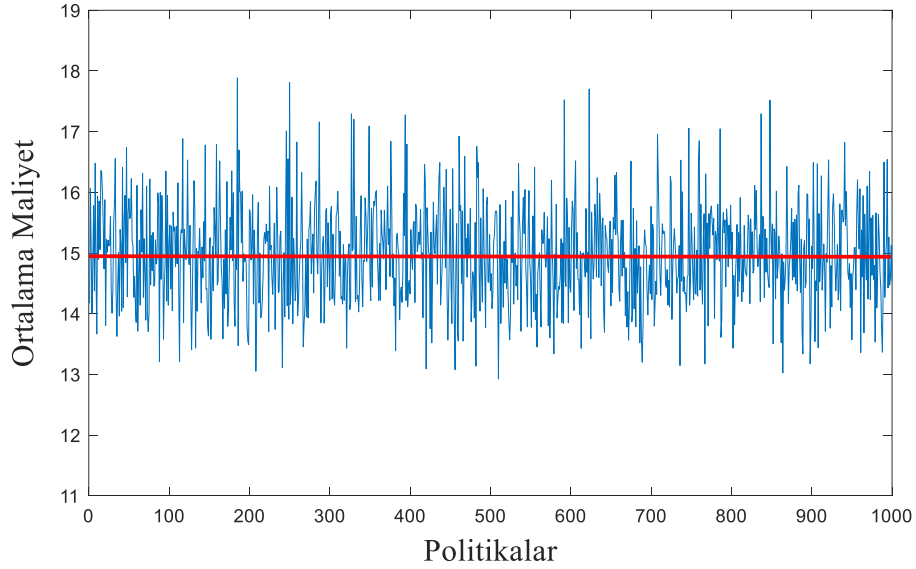
Şekil 5.17’de yer alan grafikte miyopik politika 1’in birim zamandaki ortalama maliyeti 14,5 TL civarındadır. Elde edilen politikaların ortalama maliyetlerinin çok geniş bir aralıkta (11-19) salınım yaptığı görülmektedir.

Şekil 5.18’de miyopik politika 2’nin birim zamandaki ortalama maliyet grafiği verilmiştir.



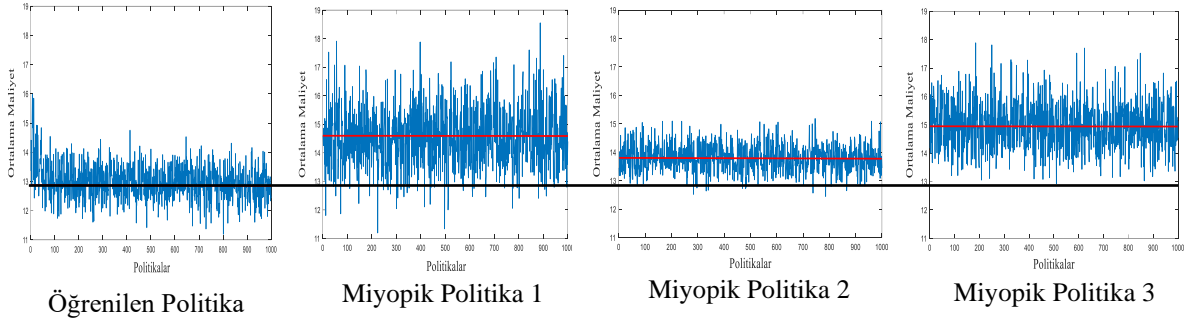
Şekil 5.18: R=30 ceza maliyetli örnekte miyopik politika 2’nin yakınsama grafiği.

Şekil 5.18’de elde edilen grafikte miyopik politika 2’nin birim zamanda ortalama maliyeti ortalama 14 TL’ye yakınsamaktadır. Şekil 5.19’da miyopik politika 3’ün birim zamandaki ortalama maliyet grafiği verilmiştir.



Şekil 5.19: R=30 ceza maliyetli örnekte miyopik politika 3’ün yakınsama grafiği.

Yukarıdaki şekilde gösterildiği üzere miyopik politika 3’ün birim zamandaki ortalama maliyeti 15 TL civarındadır. Ortalama maliyetlerin 13-18 TL arasında salınım yaptığı görülmektedir.



Şekil 5.20: R=30 ceza maliyetli sayısal örnek 2'in öğrenilen politika ile miyopik politika 1, 2 ve 3'ün kıyaslanması.

Şekil 5.20'de R= 30 TL değeri için elde edilen öğrenilen politika ve miyopik politikaların ortalama maliyetlerini gösteren birim zamandaki ortalama maliyet yakınsama grafikleri birlikte verilerek kıyaslanmıştır. Şekilde görüldüğü üzere R=30 TL ceza maliyeti eklenmesi durumunda miyopik politika 1, miyopik politika 2 ve miyopik politika 3'ün birim zamandaki ortalama maliyetleri, öğrenilen politikanın ortalama maliyetinin üstündedir. Öğrenilen politikanın ortalama maliyetine en yakın miyopik politika 2'nin ortalama maliyetidir. Öğrenilen politikanın ortalama maliyeti 13 TL'nin biraz altında, miyopik politika 2'nin ortalama maliyeti ise 14 TL'nin biraz altındadır. Birim zamandaki en düşük ortalama maliyetin öğrenilen politika ile elde edildiği Şekil 5.20'de görülmektedir.

Şekil 5.20'de görüldüğü üzere R=30 TL ceza maliyeti uygulanması durumunda öğrenilen politikanın ortalama maliyeti ile miyopik politika 2'nin ortalama maliyeti birbirine yakındır. Öğrenilen politikanın ortalama maliyetinin Miyopik Politika 2'nin ortalama maliyetinden daha düşük olup olmadığını ilişkin hipotez testi yapılmıştır. Öncelikle iki ana kitlenin standart sapmalarının eşitliğine ilişkin test gerçekleştirilmiştir.

İki ana kitlenin standart sapmalarının eşitliğine ilişkin hipotezler ve Minitab'da gerçekleştirilen teste ilişkin sonuçlar Tablo 5.5'te verilmiştir.

σ_1 : Öğrenilen Politika Ana Kitle Standart Sapma (R=30)

σ_2 : Miyopik Politika 2 Ana Kitle Standart Sapma (R=30)

$$H_0: \frac{\sigma_1}{\sigma_2} = 1$$

$$H_1: \frac{\sigma_1}{\sigma_2} \neq 1$$

Tablo 5.5: Öğrenilen politika/miyopik politika 2 (R=30) varyans eşitliğine ilişkin hipotez testi Minitab sonuçları

Method				
σ_1 : standard deviation of Öğrenilen Politika (R=20)				
σ_2 : standard deviation of Miyopik Politika 2 (R=20)				
Ratio: σ_1/σ_2				
The Bonett and Levene's methods are valid for any continuous distribution.				
Descriptive Statistics				
Variable	N	StDev	Variance	95% CI for σ
Öğrenilen Politika (R=20)	800	0,556	0,310	(0,531; 0,584)
Miyopik Politika 2 (R=20)	800	0,469	0,220	(0,448; 0,493)
Test				
Null hypothesis	$H_0: \sigma_1 / \sigma_2 = 1$			
Alternative hypothesis	$H_1: \sigma_1 / \sigma_2 \neq 1$			
Significance level	$\alpha = 0,05$			
Test				
Method	Statistic	DF1	DF2	P-Value
Bonett	23,76	1		0,000
Levene	20,58	1	1598	0,000

P-Değeri=0 olduğu için H_0 hipotezi bu örneklem verisiyle reddedildiğinde yapılan 1.Tip hata olasılığı 0'dır. Bu nedenle H_0 hipotezi reddedilir, iki ana kitlenin standart sapması birbirinden farklıdır. İki örneklem ortalama kıyaslaması ana kitle standart sapması farklı olma durumuna göre gerçekleştirilecektir.

Öğrenilen politikanın ortalama maliyetinin miyopik politika 2'nin ortalama maliyetinden daha düşük olup/olmadığına ilişkin aşağıda verilen hipotezler kurulmuş ve Minitab test sonucu Tablo 5.6'da verilmiştir.

μ_1 : Öğrenilen politikanın ortalama maliyeti (R=30)

μ_2 : Miyopik Politika 2'nin ortalama maliyeti (R=30)

$$H_0: \mu_1 = \mu_2$$

$$H_1: \mu_1 < \mu_2$$

Tablo 5.6: Öğrenilen politika/miyopik politika 2 (R=30) ortalama maliyet hipotez testi Minitab sonuçları

Descriptive Statistics				
Sample	N	Mean	StDev	SE Mean
Öğrenilen Politika (R=30)	800	12,920	0,556	0,020
Miyopik Politika (R=30)	800	13,793	0,469	0,017
	2			
Test				
Null hypothesis $H_0: \mu_1 - \mu_2 = 0$				
Alternative hypothesis $H_1: \mu_1 - \mu_2 < 0$				
T-Value	DF	P-Value		
-33,64	1553	0,000		

Testin sonucunda P-Değeri=0 olarak bulunmuştur. Buna göre bu örneklem verisiyle H_0 hipotezi reddedildiğinde yapılacak 1. Tip hata olasılığı 0'dır. Bu nedenle H_0 hipotezi reddedilir, ceza maliyeti 30 TL olduğu durumda öğrenilen politikanın ortalama maliyetinin, miyopik politika 2'nin ortalama maliyetinden daha düşük olduğuna dair istatistiksel kanıt vardır.

6. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, normal ve öncelikli olmak üzere Tip 1 ve Tip 2 müşteri sınıfını içeren tek sunuculu M/M/1 kuyruğu ele alınmıştır. Her tip müşteri sınıfı, kuyrukta birim zaman başına farklı bekleme maliyetine sahiptir. Tek sunuculu ve ortak tek kuyruklu oluşturulan kuyruk sisteminde müşteri Tip 1 (normal) ise geliş sırasına göre kuyruğa girer, müşteri Tip 2 ise sistemin alacağı iki tür karar vardır: müşteri ya geliş sırasına göre kuyruğa girer ($a=1$) ya da 1.tip müşterilerin önüne geçer ($a=2$). Bu karar verme problemi Markov karar süreci olarak modellenerek, yaklaşık optimal kuyruk sıra politikası simülasyon tabanlı bir optimizasyon tekniği olan ödüllü öğrenme ile belirlenmiştir.

Bölüm 5'te tasarlanan sayısal örneklerin ilk bölümünde kuyruk kontrol politikası ödüllü öğrenme algoritması bekleme maliyeti açısından ele alınmıştır. Ödüllü öğrenme algoritmasıyla elde edilen politikaların ortalama maliyetleri, incelenen miyopik politikaların birim zamandaki ortalama maliyetleri ile kıyaslanmıştır. Ödüllü öğrenme algoritması ile elde edilen politikaların ortalama maliyet yakınsama grafikleri 3 farklı miyopik politika ile karşılaştırılmıştır. Bu 3 politika sonucunda; her Tip 2 müşterisi için $a=1$ kararı uygulandığında oluşan ortalama maliyet grafiği, her Tip 2 müşterisi $a=2$ kararı uygulandığında oluşan ortalama maliyet grafiği ve her Tip 2 müşterisi için $a=1$ ve $a=2$ kararının yüzde elli olasılıkla rastgele seçilip uygulandığında oluşan ortalama maliyet grafikleri elde edilmiştir. Öğrenilen politikaların birim zamandaki ortalama maliyeti, algoritmanın öğrenmesiyle birlikte politikaların ortalama maliyetleri çok hızlı bir biçimde bir değere yakınsamaktadır. Ödüllü algoritma ile elde edilen politikaların ortalama maliyeti miyopik politika 1 ve 3'ten çok daha düşük. Miyopik politika 2 ve ödüllü öğrenmeyle elde edilen politikanın ortalama maliyetleri birbirine yakın değerler çıkmıştır. Ödüllü öğrenme algoritması ile Miyopik politika 2'den daha iyi çözüm bulunamamasının nedeni, bu örnekte sadece bekleme maliyetinin dikkate alınması ve Tip 2 müşterilerinin bekleme maliyeti h_2 'nin, Tip 1 müşterilerinin bekleme maliyeti h_1 'den çok daha fazla olmasıdır. Bu nedenle bu örnekte optimal politika her zaman Tip 2 müşterilere öncelik verme eğilimindedir.

Bölüm 5'te tasarlanan sayısal örneklerin ikinci bölümünde kuyruk kontrol politikası için hem bekleme maliyeti hem de Tip 2 müşterileri kuyrukta öne alma durumunda açığa çıkan

sabit maliyet bir R maliyeti dikkate alınmıştır. Maliyet fonksiyonuna her Tip 2 müşterisine öncelik verilmesi durumunda katlanılması gereken R ceza maliyeti eklenmiştir. 10 TL, 20 TL ve 30 TL olarak üç farklı R maliyetli model için ödüllü öğrenmeyle elde edilen politikaların ortalama maliyetleri üç farklı miyopik politika ile kıyaslanmıştır. Ödüllü öğrenme algoritmasıyla ile öğrenilen politikaların ortalama maliyetleri belli bir değerde başlayıp, öğrenme etkinliğinin gerçekleşmesi sonucu hızlı bir şekilde düşmüş ve ortalama maliyete yakınsamıştır. Ödüllü öğrenme algoritmasıyla elde edilen politikaların ortalama maliyetlerinin, incelenen diğer miyopik politikaların ortalama maliyetlerinden daha düşük olduğu gerçekleştirilen hipotez testleri ile gösterilmiştir.

Çalışmada ele alınan problem, ileriki çalışmalarda birden fazla öncelikli müşteri sınıfı ve sunucu sayısı eklenerek genişletilebilir. Problemdaki ortak tek kuyruk, her müşteri sınıfı için ayrı kuyruk olarak ele alınabilir. Tezde sadece öncelikli müşterileri öne aldığımız durumda ceza maliyeti ve kuyruktaki müşterilerin bekleme maliyeti dikkate alınabildiği maliyet ve kuyruktaki müşterilerin bekleme maliyeti dikkate alınarak bir maliyet fonksiyon oluşturulmuştur. Gelecekteki çalışmalarda ceza maliyetlerinin yanı sıra her müşteri sınıfına sunucu tarafından hizmet verilmesinin ardından kazanılan bir ödül yapısı eklenerek ödül ve ceza sistemi genişletilebilir. Gerçek hayatta uzun süre bekleyen müşteriler kuyruktan ayrılma eğiliminde olurlar ve bu tip müşteriler kuyruk sistemlerinde sabırsız müşteriler olarak geçmektedir. Öncelikli kuyruklardaki müşteri sınıflarına sabırlı-sabırsız müşteriler dahil edilebilir. Müşteri sınıfları için bekleme sabrı için bir eşik bekleme süresi seviyesi belirlenerek bu seviyeyi geçen müşteriler kuyruktan ayrılması eklenebilir. Sistemden ayrılan her müşteri kayıp olarak görülür ve kaybedilen her müşteri için ceza maliyeti ve servisi tamamlanan her müşteri için bir ödül tanımlanabilir.

7. KAYNAKLAR

- [1] A. COBHAM, "Priority assignments in waiting line problems", *Journal of the Operations Research Society of America*, vol. 2, no. 1, pp. 70-76, 1954, doi:10.1287/opre.2.1.70
- [2] N.K. Jaiswal, *Priority queues*. vol. 50, New York: Academic press, 1968.
- [3] H. Takagi, *Queueing analysis: a foundation of performance evaluation, volume 1: vacation and priority systems*. Elsevier: Amsterdam, The Netherlands, 1991.
- [4] B.V. Gnedenko, E.A. Danielyan, B.N. Dimitrov, G.P. Klimov and V.F. Matvejev, "Priority queueing systems" Moscow State University: Moscow, Russian, 1973.
- [5] D.R. Miller, "Computation of steady-state probability of M/M/1 priority queue", *Operations Research* vol. 29, no. 5, pp.945-958, 1981.
- [6] A. Sleptchenko, J. Selen, I. Adan and G. van Houtum, "Joint queue length distribution of multi-class, single-server queues with preemptive priorities", *Queueing Syst*, vol. 81, pp. 379–395, 2015, <https://doi.org/10.1007/s11134-015-9460-z>.
- [7] Z. Guo-xi and H. Qi-Zhou, "M/M/1 Queueing System with Non-preemptive Priority", 2009, <https://doi.org/10.48550/arXiv.0902.2086>.
- [8] K. Samouylov, O. Dudina and A. Dudin, "Analysis of Multi-Server Queueing System with Flexible Priorities", *Mathematics* vol. 11, no. 4, 2023, <https://doi.org/10.3390/math11041040>.
- [9] V.V. Rykov and E.Y. Lember. "Optimal dynamic priorities in single-line queueing systems." *Engineering Cybernetics*, vol. 5, no.1, pp. 21-30,1967.
- [10] P. Naor, "The Regulation of queue size by levying tolls", *Econometrica*, vol. 37, pp. 15-24, 1969.
- [11] E. Kofman, "An M/M/1 dynamic priority queue with optional promotion", Ph.D. dissertation, University of California, Los Angeles, 1978.
- [12] R.C Rue, "Optimal control for entry of many classes of customers to an M/M/1 queue", *Naval Research Logistics Quarterly*, Vol. 28, pp. 489-495, 1981.
- [13] K. Siddharthan, W. J. Jones and J. A. Johnson, "A priority queueing model to reduce waiting times in emergency care", *International Journal of Health Care Quality Assurance*, vol. 9, no. 5, pp. 10-16, 1996, <https://doi.org/10.1108/09526869610124993>.

- [14] G. A. J. F. Brouns and J. van der Wal, "Optimal threshold policies in a two-class preemptive priority queue with admission and termination control", SPOR-Report: reports in statistics, probability and operations research, vol. 200216, Technische Universiteit Eindhoven, 2002.
- [15] P. Cao and J. Xie, "Optimal control of a multiclass queueing system when customers can change types" *Queueing Syst* 82, pp. 285–313, 2016, <https://doi.org/10.1007/s11134-015-9466-6>.
- [16] X. Jingui, P. Cao, B. Huang and M.E.H. Ong, "Determining the conditions for reverse triage in emergency medical services using queuing theory" *International Journal of Production Research*, vol. 54, no. 11, pp. 3347-3364, 2016, doi: 10.1080/00207543.2015.1109718.
- [17] D. G. Down, G. Koole and M.E. Lewis, "Dynamic control of a single-server system with abandonments", *Queueing Syst*, vol. 67, no. 1, pp. 63–90, 2010, doi:10.1007/s11134-010-9201-2.
- [18] C.F. Bispo, "The single-server scheduling problem with convex costs", *Queueing Syst*, vol. 73, pp. 261–294, 2013, <https://doi.org/10.1007/s11134-012-9316-8>.
- [19] R. Groenevelt, G. Koole and P. Nain, "On the bias vector of a two-class preemptive priority queue", *Mathematical Methods of Operations Research*, vol. 55, pp. 107-120, 2002.
- [20] K. Klimenok, V. Dudin, A. Dudina and O. Kochetkova, "Queueing system with two types of customers and dynamic change of a priority" *Mathematics*, vol. 8, no. 5, pp. 824, 2020, <https://doi.org/10.3390/math8050824>.
- [21] A.Çelik, "İki müşteri sınıflı öncelikli kuyruk sistemleri ve sayısal bir örnek", Yüksek lisan tezi, Ondokuz Mayıs Üniv., Samsun, 2017.
- [22] Q. Gong and B. Rajan "A queue-length cutoff model for a preemptive two-priority M/M/1 system." *SIAM Journal on Applied Mathematics*, vol. 67, no. 1, pp. 99–115, 2006.
- [23] E. A. Peköz, "Optimal policies for multi-server non-preemptive priority queues", *Queueing Syst.*, vol. 42, no. 1, pp. 91-101, 2002, doi:10.1023/A:101994681216.
- [24] B. Liu, Q. Xie and E. Modiano, "Reinforcement learning for optimal control of queueing systems," in *57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, USA, Sep. 2019, pp. 663-670, doi: 10.1109/ALLERTON.2019.8919665.

- [25] A. Bedford and P. Zeephongsekul, “On a dual queueing system with preemptive priority service discipline”, *European Journal of Operational Research*, vol. 161, no. 1, pp. 224–239, 2005, <https://doi.org/10.1016/j.ejor.2003.06.024>.
- [26] C. Knessl, D.I. Choi and C. Tier, “A dynamic priority queue model for simultaneous service of two traffic types”, *SIAM Journal on Applied Mathematics*, vol. 63, no. 2, pp. 398–422, 2002,
- [27] Y. Li, H. Wang, N. Wang and T. Zhang, “Optimal scheduling in cloud healthcare system using Q-learning algorithm”, *Complex Intell. Syst.*, vol. 8, pp. 4603–4618, 2022, <https://doi.org/10.1007/s40747-022-00776-9>.
- [28] Q. Li, J. Mu, R. Fan and L. Xia, “An overview for Markov decision processes in queues and networks”, in: *International Conference of Celebrating Professor Jinhua Cao's 80th Birthday*, Singapore, Aug. 2019, pp. 44-71.
- [29] A. K. Bhunia, L. Sahoo and A. A. Shaikh, “Advanced optimization and operations research”, P. M. Pardalos Ed., FL, USA: Springer, 2019, <https://doi.org/10.1007/978-981-32-9967-2>.
- [30] D. Gross, J. F. Shortle, J. M. Thompson, C. M. Harris, “Fundamentals of queueing theory”, John Wiley & sons press, 2008.
- [31] I. Adan and R. Jacque, 2002. “Queueing theory”, Available: <https://ieeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf>
- [32] U.N. Bhat, *An introduction to queueing theory*. Beoston, Birkhäuser press, 2008, doi:10.1007/978-0-8176-4725-4.
- [33] W.L. Winston, *Operations research: applications and algorithm*. Thomson Learning, Inc., 2004.
- [34] J. Wang, O. Baron, and A. Scheller-Wolf, “M/M/c queue with two priority classes”, *Operations Research*, vol. 63, no. 3, pp. 733-749, 2015, <https://doi.org/10.1287/opre.2015.1375>.
- [35] R.C. Larson and A.R. Odoni, *Urban operations research*. Englewood Cliffs N.J: Prentice-Hall, 1981.
- [36] A.B. Sharif, D.A. Stanford, P. Taylor and I. Ziedins, “A multi-class multi-server accumulating priority queue with application to health care” *Operations Research for Health Care*, vol. 3, no. 2, pp. 73–79, 2014, doi: 10.1016/j.orhc.2014.01.002.

- [37] S.R. Chakravarthy, “A Dynamic Non-preemptive Priority Queueing Model with Two Types of Customers”, in: Ghosh, D., Giri, D., Mohapatra, R., Sakurai, K., Savas, E., Som, T. (eds) *Mathematics and Computing. ICMC 2018. Springer Proceedings in Mathematics & Statistics*, vol 253. pp. 23-42, 2018, https://doi.org/10.1007/978-981-13-2095-8_3.
- [38] A. Gosavi, *Simulation-based optimization*. Kluwer Academic Publishers, 2003, doi:10.1007/978-1-4899-7491-4 10.1.
- [39] M.A.B. Ocaktan, “İkame ürün dağıtım ağlarında stok optimizasyonu ve optimal dağıtım politikaları”, Doktora Tezi, Balıkesir Üniv., Balıkesir, 2012.
- [40] T.M. Mitchell, *Machine learning*, McGraw Hill, Boston, 1997.
- [41] R. Sutton and A.G. Barto, *Reinforcement learning: an introduction*. MIT press, 1998.
- [42] R.E. Bellman, *Dynamic programming*, Princeton University Press, Princeton, 1957.
- [43] C. J.C.H. Watkins, P. Dayan, “Q-learning”, *Machine Learning*, vol. 8, pp. 279–292, 1992, <https://doi.org/10.1007/BF00992698>.
- [44] M.A.B. Ocaktan, U. Kula, “Yarı Markov karar süreci problemlerinin çözümünde çok katmanlı yapay sinir ağlarıyla fonksiyon yaklaşımli ödüllü öğrenme algoritması”, *Sakarya Üniv. Fen Bil. Der.* 17. Cilt, 3. Sayı, s. 307-314, 2013.
- [45] G. Deng, “Simulation-based optimization” 2007. Ph. D. dissertation, University of Wisconsin-Madison, USA, 2007.
- [46] O. Madani, "Complexity Results for Infinite-Horizon Markov Decision Processes." Order No. 9983515, University of Washington, United States -- Washington, 2000.

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Selin Ciğer

Doğum tarihi ve yeri :

e-posta :

Öğrenim Bilgileri

Derece	Okul/Program	Yıl
Y. Lisans	Balıkesir Üniversitesi/Endüstri Mühendisliği	2024
Lisans		
Lise		