

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**İKAME ÜRÜN DAĞITIM AĞLARINDA STOK
OPTİMİZASYONU VE OPTİMAL DAĞITIM
POLİTİKALARI**

DOKTORA TEZİ

End.Y.Müh. Mustafa Ahmet Beyazıt OCAKTAN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Tez Danışmanı : Yrd.Doç.Dr.Ufuk KULA

Şubat 2012

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İKAME ÜRÜN DAĞITIM AĞLARINDA STOK
OPTİMİZASYONU VE OPTİMAL DAĞITIM
POLİTİKALARI

DOKTORA TEZİ

End. Y. Müh. Mustafa Ahmet Beyazıt OCAKTAN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 13/02/2012 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Prof. Dr.
Ramazan YAMAN
Jüri Başkanı



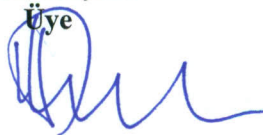
Yrd.Doç.Dr.
Ufuk KULA
Üye



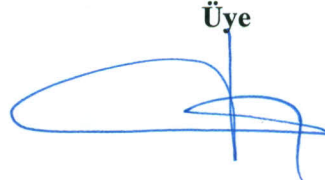
Prof. Dr.
Alparslan FIĞLALI
Üye



Prof. Dr.
Harun TAŞKIN
Üye



Doç. Dr.
Mehmet FIRAT
Üye



TEŞEKKÜR

Tez çalışmamın tüm aşamalarında gerek bilgi ve gerekse tecrübesiyle yol gösteren, ilgi ve desteğini esirgemeyen danışman hocam Yrd.Doç.Dr.Ufuk KULA'ya sonsuz teşekkürlerimi sunuyorum. İhtiyaç duyduğum bilgi eksikliğimin tamamlanması için verdiği derslerde ve özel çalışmalarımızda danışman hocamın sabrı ve öğreticiliği olmasa, bu çalışmayı tamamlayabilmem mümkün olmazdı.

Tez izleme jürimde yer alan Prof.Dr.Harun TAŞKIN ve Doç.Dr.Mehmet FIRAT hocalarıma da yorum ve yönlendirmeleri ile çalışmama yaptıkları katkılardan dolayı çok teşekkür ederim.

Danışman hocamla gerçekleştirdiğimiz hafta sonu ve gece çalışmalarımız için gösterdiği sabır ve manevi destek nedeniyle Sayın Sibel KULA'ya en içten teşekkürlerimi sunmayı borç biliyorum.

Ayrıca çalışmam süresince verdikleri manevi destekleri için değerli meslektaşlarım Arş.Gör.Elçin KABELOĞLU GÜNAY'a ve Yrd.Doç.Dr.Alparslan Serhat DEMİR'e de çok teşekkür ederim.

Son olarak, maddi ve manevi desteklerini üzerimden hiçbir zaman eksik etmeyen sevgili anneme, babama ve kardeşlerime tüm kalbimle teşekkür ediyorum.

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ	vii
TABLolar LİSTESİ.....	ix
ÖZET.....	x
SUMMARY.....	xi
BÖLÜM 1.	
GİRİŞ.....	1
1.1. Motivasyon.....	2
1.2. Problem Tanımı.....	5
1.2.1. Ürün kombinasyonu dağıtım problemi.....	7
1.2.2. Stok yenileme problemi.....	8
1.3. Literatür Taraması.....	9
1.3.1. Ürün ikamesiyle ilgili çalışmalar.....	9
1.3.2. Eşgüdümlü stok yenileme politikaları ile ilgili çalışmalar.....	15
BÖLÜM 2.	
MODEL.....	20
2.1. Ürün Talep Süreci.....	20
2.1.1. Çok ürünlü stok sistemlerinin ürün talep süreci.....	20
2.1.2. Çok ürünlü stok sistemindeki ürün çeşitlerinin talep süreci....	21
2.1.3. Tedarik süresince gelen birleşik poisson talebinin dağılımı....	22
2.1.4. Birleşik poisson talep sürecine eşdeğer poisson talep süreci...	23

2.2. Ürün Kombinasyonu Dağıtım Politikasının Modellenmesi.....	25
2.3. Eşgüdümlü Stok Yenileme Politikasının Belirlenmesi.....	30
2.3.1. (S,c,s) “Can Order” eşgüdümlü stok politikası.....	31
2.3.2. Stok pozisyonu dağılımının belirlenmesi.....	34
2.3.3. Eksiklik pozisyonu dağılımının belirlenmesi.....	37
2.3.4. (S,c,s) kontrol parametrelerinin belirlenmesi.....	39
2.3.4.1. Tedarik süresinin ihmal edildiği durumda (S,c,s) parametrelerinin belirlenmesi.....	41
2.3.4.2. Sabit tedarik süresi için (S,c,s) parametrelerinin belirlenmesi.....	44
BÖLÜM 3.	
ALGORİTMA VE ÖDÜLLÜ ÖĞRENME.....	51
3.1. Ürün Kombinasyonu Dağıtım ve Stok Yenileme Algoritması.....	51
3.1.1. Algoritmanın adımları.....	52
3.2. Ürün Kombinasyonu Dağıtım Probleminin Çözümü.....	56
3.2.1. Ödüllü öğrenme.....	56
3.2.2. Küçük ölçekli Markov karar süreci problemlerinin çözümü..	62
3.2.2.1.SMART algoritması.....	63
3.2.2.2.Politika ortalama maliyetlerinin simülörde hesaplanması.....	65
3.2.3.Orta ve büyük ölçekli Markov karar süreci problemlerinin çözümü.....	66
3.2.3.1.Q faktör değer fonksiyonuna yaklaşım.....	66
3.2.3.2. Yapay sinir ağları.....	68
3.2.3.3. NeuroSMART algoritması.....	76
BÖLÜM 4.	
UYGULAMALI ÖRNEKLER.....	79
4.1. Uygulamalı Örneklerde Kullanılan Ortak Parametreler.....	80
4.2.(S,c,s) Stok Politikası Altında SMART ve NeuroSMART Algoritmalarının Performansları.....	83

4.2.1. Uygulamalı örnek1	84
4.2.1.1. Uygulamalı örnek 1'in birleşik poisson ürün talebi parametreleri.....	85
4.2.1.2. Uygulamalı örnek 1'in eşdeğer poisson ürün talebi parametreleri.....	87
4.2.1.3. Uygulamalı örnek 1'de SMART ve NeuroSMART algoritmalarının çözümlerinin karşılaştırılması.....	89
4.3. Ürün Kombinasyonu Dağıtım ve Stok Yenileme Probleminin Çözümü.....	97
4.3.1. Uygulamalı örnek 2 (orta ölçekli problem).....	99
4.3.2. Uygulamalı örnek 3 (büyük ölçekli problem).....	107
BÖLÜM 5.	
SONUÇLAR VE ÖNERİLER.....	118
KAYNAKLAR.....	121
ÖZGEÇMİŞ.....	125

SİMGELER VE KISALTMALAR LİSTESİ

ATM	: Otomatik para çekme makinesi (automated teller machine)
ANN	: Yapay sinir ağları (artificial neural networks)
Bkz.	: Bakınız
cl	: Santilitre
CTMDP	: Sürekli zaman Markov karar süreci (continous time Markov decision process)
DP	: Dinamik programlama (dynamic programming)
ESM	: Ekonomik sipariş miktarı
GB	: Giga byte
Ghz	: Giga hertz
MDP	: Markov karar süreci (Markov decision process)
MSE	: Hata kareleri ortalaması (mean square of errors)
NeuroSMART	: Yapay sinir ağı yaklaşımli yarı Markov ortalama ödül tekniđi
RAM	: Hafıza kartı (random access memory)
SMART	: Yarı Markov ortalama ödül tekniđi (semi Markov avarage reward technique)
SMDP	: Yarı Markov karar süreci (semi Markov decision process)
SSE	: Hata kareleri toplamı (sum of the squares of errors)
TL	: Türk lirası
vb.	: Ve benzeri

ŞEKİLLER LİSTESİ

Şekil 1.1.	Problemin bileşenleri.....	6
Şekil 2.1.	Dört ürünlü stok sisteminde (S,c,s) stok politikası.....	32
Şekil 3.1.	Ürün kombinasyonu dağıtım ve stok politikalarının etkileşimi....	52
Şekil 3.2.	Yaklaşık optimal ürün kombinasyonu dağıtım ve stok politikası belirleme algoritması.....	54
Şekil 3.3.	Yarı Markov karar süreci problemlerinin çözüm yöntemleri.....	57
Şekil 3.4.	Simulatörde Q faktör değerlerinin güncellenmesi.....	61
Şekil 3.5.	Doğrusal yapay sinir ağı mimarisi.....	69
Şekil 3.6.	Doğrusal transfer fonksiyonu	70
Şekil 3.7.	Dört katmanlı yapay sinir ağı mimarisi.....	70
Şekil 3.8.	Log-sigmoid transfer fonksiyonu.....	73
Şekil 3.9.	Tan-sigmoid transfer fonksiyonu.....	73
Şekil 4.1.	Örnek 1'in SMART algoritması yakınsama grafiği.....	90
Şekil 4.2.	Örnek 1'in 20 gizli katmanlı NeuroSMART algoritması yakınsama grafiği.....	91
Şekil 4.3.	Örnek 1'in SMART ve NeuroSMART algoritmaları yakınsama grafikleri.....	93
Şekil 4.4.	Örnek 1'in 2 gizli katmanlı NeuroSMART algoritması yakınsama grafiği.....	94
Şekil 4.5.	Örnek 1'in 5 gizli katmanlı NeuroSMART algoritması yakınsama grafiği.....	94
Şekil 4.6.	Örnek 1'in 5 gizli katmanlı NeuroSMART algoritması yakınsama grafiği (tekrar çözüm).....	95
Şekil 4.7.	Örnek 1'in 10 gizli katmanlı NeuroSMART algoritması yakınsama grafiği.....	96

Şekil 4.8.	Örnek 1'in 2, 5, 10, 20 gizli katmanlı NeuroSMART algoritması yakınsama grafikleri.....	96
Şekil 4.9.	Örnek 2 ve 3'te kullanılan yapay sinir ağı mimarisi.....	98
Şekil 4.10.	Örnek 2'nin yaklaşık optimal banknot kombinasyonu dağıtım ve stok yenileme politikası yakınsama grafiği.....	101
Şekil 4.11.	Örnek 2'deki yapay sinir ağının ortalama hata kareleri ve korelasyon katsayıları.....	102
Şekil 4.12.	Örnek 2'deki yapay sinir ağının performans grafiği.....	103
Şekil 4.13.	Örnek 2'deki yapay sinir ağının eğitim durumu grafiği.....	104
Şekil 4.14.	Örnek 2'deki yapay sinir ağının regresyon grafikleri.....	104
Şekil 4.15.	Örnek 2'de miyopik politika 1'in yakınsama grafiği.....	105
Şekil 4.16.	Örnek 2'de miyopik politika 2'nin yakınsama grafiği.....	106
Şekil 4.17.	Örnek 2'de öğrenilen politikayla miyopik politika 1 ve 2'nin kıyaslanması.....	107
Şekil 4.18.	Örnek 3'ün yaklaşık optimal banknot dağıtım ve stok yenileme politikası yakınsama grafiği.....	110
Şekil 4.19.	Örnek 3'teki yapay sinir ağının ortalama hata kareleri ve korelasyon katsayıları.....	111
Şekil 4.20.	Örnek 3'teki yapay sinir ağının performans grafiği.....	111
Şekil 4.21.	Örnek 3'teki yapay sinir ağının eğitim durumu grafiği.....	112
Şekil 4.22.	Örnek 3'teki yapay sinir ağının regresyon grafikleri.....	113
Şekil 4.23.	Örnek 3'te miyopik politika 1'in yakınsama grafiği.....	113
Şekil 4.24.	Örnek 3'te miyopik politika 2'nin yakınsama grafiği.....	114
Şekil 4.25.	Örnek 3'te öğrenilen politikayla miyopik politika 1 ve 2'nin kıyaslanması.....	115

TABLolar LİSTESİ

Tablo 2.1.	($L=0$) ve ($L>0$) durumlarında (S,c,s) parametreleri.....	40
Tablo 4.1.	Müşterilerin bireysel para talebi olasılık dağılımı.....	80
Tablo 4.2.	Müşteri talebine göre oluşan kararlar.....	80
Tablo 4.3.	Bankamatığın başlangıç banknot dağıtım politikası.....	81
Tablo 4.4.	Müşterilerin kesikli ve özdeş fayda fonksiyonu.....	83
Tablo 4.5.	Örnek 1'in (S,c,s) stok politikası parametreleri.....	89
Tablo 4.6.	Örnek 2'nin başlangıç eşdeğer birleşik poisson banknot talebi parametreleri.....	99
Tablo 4.7.	Örnek 2'nin başlangıç eşdeğer poisson banknot talebi parametreleri.....	100
Tablo 4.8.	Örnek 2'nin başlangıç stok yenileme politikası.....	100
Tablo 4.9.	Örnek 2'nin yaklaşık optimal stok yenileme politikası.....	100
Tablo 4.10	Örnek 2'de miyopik politika 2'nin stok yenileme politikası.....	106
Tablo 4.11	Örnek 3'ün başlangıç eşdeğer birleşik poisson banknot talebi parametreleri.....	108
Tablo 4.12	Örnek 3'ün başlangıç eşdeğer poisson banknot talebi parametreleri.....	108
Tablo 4.13	Örnek 3'ün başlangıç stok yenileme politikası.....	109
Tablo 4.14	Örnek 3'ün yaklaşık optimal stok yenileme politikası.....	109

ÖZET

Anahtar kelimeler: (S,c,s) Stok Politikası, Ürün Kombinasyonu Dağıtım Politikası, Ödüllü Öğrenme, Yapay Sinir Ağları

Çok ürünlü stok sistemlerinde ürün stoklarının eşgüdümlü olarak yenilenmesi, özellikle ulaştırma maliyetlerinde oluşan ölçek ekonomileri nedeniyle önemli maliyet kazançları sağlar. Bunun yanı sıra bu tür stok sistemlerinde ürün ikamesi yapılabilmesi, stok maliyetlerini önemli derecede düşürür.

Tezde, ürünlerin birbirileri ile ikame edilebildiği çok ürünlü stok sistemlerinde ürün kombinasyonu dağıtım ve stokastik stok yenileme problemi incelenmiştir. Ele alınan problemde ürün stokları eşgüdümlü olarak yenilenmekte ve müşteri talebi firma kaynaklı olarak stoklardaki ürünlerin herhangi bir kombinasyonuyla karşılanabilmektedir. Amaç, sonsuz zaman ufkunda birim zaman başına ortalama maliyeti yaklaşık olarak en küçükleyen ürün kombinasyonu dağıtım ve stok yenileme politikasının belirlenmesidir.

Tezde, stok yenileme politikası olarak sürekli gözden geçirme esasına dayanan eşgüdümlü (S,c,s) “can order” politikası kullanılmış ve stok kontrol parametrelerinin belirlenmesinde literatürdeki çalışmalardan faydalanılmıştır. Ürün kombinasyonu dağıtım problemi ise yarı Markov karar süreci olarak modellenmiş ve çözüm için Q faktör fonksiyonunun çok katmanlı yapay sinir ağlarıyla tahmin edildiği ödüllü öğrenme algoritmaları kullanılmıştır. Çalışmada, ürün kombinasyonu dağıtım ve stok yenileme problemini sıralı ve yinelemeli olarak çözen bir algoritma geliştirilmiş ve değişik ölçeklerdeki uygulamalı denemelerde, geliştirilen algoritmanın performansı miyopik politikalarla kıyaslanmıştır. Gerçekleştirilen uygulamalı denemelere göre, geliştirilen algoritmayla elde edilen ürün kombinasyonu dağıtım ve stok yenileme politikalarının ortalama maliyetleri, miyopik politikaların ortalama maliyetlerinden çok daha düşüktür.

INVENTORY OPTIMIZATION AND OPTIMAL DISTRIBUTION POLICIES IN SUBSTITUTABLE PRODUCT DISTRIBUTION NETWORKS

SUMMARY

Key Words: (S,c,s) Inventory Policy, Order Fulfillment Policy, Reinforcement Learning, Artificial Neural Networks

In multi-product inventory systems, considerably savings in specially transportation costs may be achieved by coordination of replenishment orders for groups of products. Moreover, product substitutions in these systems reduce significantly inventory costs.

In this thesis, a fully substitutable multi product system is considered in which customer demands may be satisfied by delivering any combination of products. In the studied system, product substitution is driven by firm and product inventories may be replenished by coordinated control. The aim of the thesis is to determine order fulfillment and replenishment policies minimizing approximately average cost in unit time in infinite horizon.

In this thesis, a specific control system is considered which is known as a (S,c,s) type policy, and works in the literature are used to determine the control parameters. The order fulfillment problem is modeled as a semi Markov decision process, and reinforcement learning algorithms with approximation to Q factor function by artificial neural network are used to solve this problem. In the thesis, an algorithm is developed which determines iteratively order fulfillment and (S,c,s) policies for each product in a multi-product inventory system. In numerical examples, it is seen that order fulfillment and replenishment policies determined by the developed algorithm outperform myopic policies.

BÖLÜM 1. GİRİŞ

Çok ürünli stok sistemlerinde eşgüdümlü (koordineli) bir stok kontrol politikası uygulanması sonucu ortaya çıkan ölçek ekonomisi nedeniyle, özellikle ulaştırma maliyetlerinde önemli maliyet avantajları oluşur. Bunun yanı sıra bu tür stok sistemlerinde ürün ikamesinin yapılabilmesi, stok maliyetlerinin düşürülmesinde önemli bir faktördür. Özellikle ürün ikamesine firmaların karar verdiği sistemlerde kararların hem stok maliyeti, hem de müşteri faydası göz önünde bulundurularak verilmesi gereklidir. Çünkü firmalar sadece kendi stoklarını azaltacak şekilde kararlar aldıklarında, orta ve uzun vadede kaybedecekleri müşteriler nedeniyle daha büyük maliyetlere katlanmak zorunda kalabilir.

Tezde, ürün çeşitlerinin birbirileri ile ikame edilebildiği çok ürünli stok sistemlerinde stokastik stok yenileme ve ürün kombinasyonu dağıtım problemi incelenmiştir. Burada stok sisteminde yer alan ürün çeşitlerinin stoklarının birbirileri ile eşgüdümlü olarak yenilendiği durum ele alınmıştır. Tezin amacı, sonsuz zaman ufunda birim zaman başına beklenen (ortalama) maliyeti yaklaşık en küçükleyecek ürün kombinasyonu dağıtım ve stok yenileme politikasının belirlenmesidir. Ele alınan problem, Bölüm 1.1’de verilen endüstri örneklerinden de görüleceği üzere büyük ölçekli ve karmaşık bir gerçek yaşam problemidir.

Ele alınan problemin doğası gereği stok yenileme ve dağıtılacak ürün kombinasyonu kararları belirsizlik altında verilmektedir ve bu nedenle çalışmada stokastik modelleme yaklaşımına odaklanılmıştır. Büyük ölçekli, karmaşık ve stokastik yapıya sahip gerçek yaşam problemlerinin, boyutun laneti (curse of dimensionality) ve modellemenin laneti (curse of modeling) nedenleriyle klasik dinamik programlama algoritmalarıyla çözülebilmesi mümkün değildir. Ancak, günümüz bilgisayarlarının giderek artan hesaplama ve depolama gücü sayesinde simülasyon tabanlı ve fonksiyon yaklaşımli yeni algoritmalar, büyük ölçekli ve karmaşık gerçek yaşam

problemlerine optimale yakın çözümler bulabilmeyi mümkün hale getirmiştir. Bu nedenle çalışmanın diğer bir odak noktası, stokastik dinamik program olarak modellenebilen büyük ölçekli ve karmaşık problemlerin çözüm yöntemlerine katkı sağlamaktır.

Bölüm 1.1’de teze motivasyon sağlayan endüstri örnekleri verilmiş ve Bölüm 1.2’de ele alınan problem ayrıntılı bir şekilde tanımlanmıştır. Bölüm 1.3’te ise literatürde yer alan ilgili çalışmalar ve yapılan çalışmanın literatürden farkı ve katkısı yer almaktadır.

1.1. Motivasyon

Tezde ele alınan problemin birincil motivasyon kaynağı, müşterilerin bir bankanın Otomatik Para Çekme Makinesinden (Bankamatik, ATM) nakit para taleplerini karşılamaları probleminden gelir. Bankamatiklerin stok sistemleri, birbirileri ile ikame edilebilen farklı banknot çeşitlerinden oluşmaktadır. Örneğin bir bankamatiğin stok sisteminde 10 TL, 20 TL, 50 TL ve 100 TL olmak üzere 4 farklı banknot çeşidi bulunduğu kabul edilsin. Bankamatiğe gelen müşteri talepleri, birbiri ile ikame edilebilen bu banknot çeşitlerinin farklı kombinasyonlarıyla karşılanabilir. Örneğin bankamatiğe gelen 50 TL’lik müşteri talebi,

- 5 adet 10 TL,
- 3 adet 10 TL ve 1 adet 20 TL,
- 1 adet 10 TL ve 2 adet 20 TL,
- 1 adet 50 TL,

kombinasyonlarıyla karşılanabilir. Eğer bankamatik stoklarından müşteri talebini tam olarak karşılayacak bir banknot kombinasyonu oluşturulamıyorsa, müşteri talebi kaybolur (lost sales). Ele alınan problemde her bir ürün -bankamatik için diğer bir deyişle her bir banknot çeşidinin- stoklarının aynı dağıtım merkezinden yenilendiği

kabul edilmektedir. Bu varsayım altında stok sisteminde yer alan ürün çeşitlerinin stoklarının yenilenmesi kararlarının birbiri ile eşgüdüm halinde verilmesi, oluşacak ölçek ekonomisi nedeniyle önemli maliyet kazançları yaratır.

Tedarik zinciri ağında doğrudan sevkiyatı kullanan bilgisayar üreticilerinin hafıza kartları (RAM) farklılaştırmasını geciktirebilmesi, çalışmanın diğer bir motivasyon kaynağıdır. Bilgisayar ana kartlarında farklı kapasitelerde RAM'lerin takılabildiği birden çok RAM yuvası (slot) mevcuttur. Bilgisayar üreticileri değişik kapasitelerdeki RAM'leri stoklarında hazırda bulundurmakta ve müşteri talebi açığa çıktıktan sonra istenen toplam kapasiteyi sağlayacak şekilde ana karttaki yuvalara RAM'leri takarak talebi karşılamaktadır. Örneğin, 3 RAM yuvasının bulunduğu bir anakart düşünölsün. Bir müşterinin 4 GB RAM içeren bilgisayar talebi, üretici tarafından ana karta takılacak,

- 1 adet 4 GB RAM,
- 2 adet 2 GB RAM,
- 1 adet 2 GB ve 2 adet 1 GB RAM,

ile karşılanabilir. Bankamatik problemine benzer şekilde burada da genellikle RAM çeşitlerinin ürün stokları aynı merkezden (tedarikçiden) yenilenir ve stokların yenilenmesinde eşgüdümlü bir politika uygulanması önemli maliyet avantajları sağlar.

Ele alınan probleme benzer bir diğer endüstri örneği, tedarik zincirinde imalatçı ve dağıtımıcının mülkiyetinin aynı olduğu durumlarda açığa çıkar. Örneğin bir meşrubat firmasının, ürünlerini 3 farklı ebatta (500 cl, 1000 cl, 2000 cl) şişelere doldurarak sattığı düşünölsün. Burada dağıtımıcı, imalatçıdan her bir meşrubat şişe grubu için ayrı ayrı ve farklı miktarlarda sipariş vermesine karşın, imalatçı kendi stoklarının durumunu ve diğer dağıtımıcıların olası taleplerine göz önüne alarak, dağıtımıcının toplam talebini karşılayacak şekilde meşrubat şişe grubunun talebini ikame edebilir. Örneğin dağıtımıcıdan gelen 500 adet 2000 cl'lik meşrubat talebi imalatçı tarafından,

- 1000 adet 1000 cl,
- 2000 adet 500 cl,
- 1000 adet 500 cl ve 500 adet 1000 cl

vb. daha birçok kombinasyonla karşılanabilir. Benzer durumlar birçok sektörde imalatçı-dağıtımçı ya da dağıtımçı-bayi arasında yaşanabilir. Örneğin otomobil imalatçısıyla, bayileri arasındaki ilişki ele alınsın. Her ne kadar otomobil imalatçısı tedarik zincirinin tamamının mülkiyetini elinde bulundurmasa da diğer bileşenler üzerinde hakim konumundadır. Bu nedenle otomobil bayilerinin bir otomobil modelinin farklı opsiyonları (renk, özellik vb.) için verdiği sipariş, otomobil imalatçısı tarafından üretim planı, stokların durumu ve diğer bayilerin olası talepleri göz önüne alınarak farklı opsiyonlarla ikame edilebilir.

Tezde ele alınan problemde müşteri talebinin karşılandığı her ürün kombinasyonunun müşteriye faydası birbirinden farklıdır. Tezin ana motivasyon kaynağı olan bankamatik örneği ele alınacak olursa, müşteriler bankamatikten düşük miktarlarda para talep ettiklerinde genellikle bankamatiğin mümkün olduğunca fazla sayıda banknot çeşidi içeren bir kombinasyon ile taleplerini karşılamasını bekler. Örneğin bankamatikten 50 TL çekmek isteyen bir müşterinin talebinin 1 adet 10 TL ve 2 adet 20 TL kombinasyonu ile karşılanmasının müşteriye faydası, talebin 1 adet 50 TL kombinasyonu ile karşılanmasından muhtemelen çok daha fazla olacaktır. Müşterilerin bankamatikten yüksek meblağlarda para çekmek istemeleri durumunda ise, banknot çeşitliliğinin yanı sıra banknot sayısı da müşteri için önem kazanır. Örneğin bir müşteri bankamatikten 500 TL çekmek istediğinde, 4 adet 100 TL, 1 adet 50 TL, 2 adet 20 TL ve 1 adet 10 TL kombinasyonunun müşteriye faydası, 50 adet 10 TL kombinasyonundan muhtemelen çok daha fazla olacaktır.

RAM örneğinde de benzer şekilde farklı ürün kombinasyonlarının müşteriye faydası birbirinden farklıdır. Müşteriler doğal olarak bilgisayarlarının ana kartlarında mümkün olduğunca çok sayıda boş RAM yuvası bulunmasını tercih ederler. Çünkü bu durumda sistemlerini daha iyi hale getirmek istediklerinde, ana kartlarındaki boş

yuvalara RAM ekleyebilme imkânları olur. Aksi halde ise düşük kapasiteli RAM'leri söküp, yerlerine yüksek kapasiteli yeni RAM'ler takmak zorunda kalacak ve ellerinde kullanamadıkları düşük kapasiteli RAM'ler artmış olacaktır.

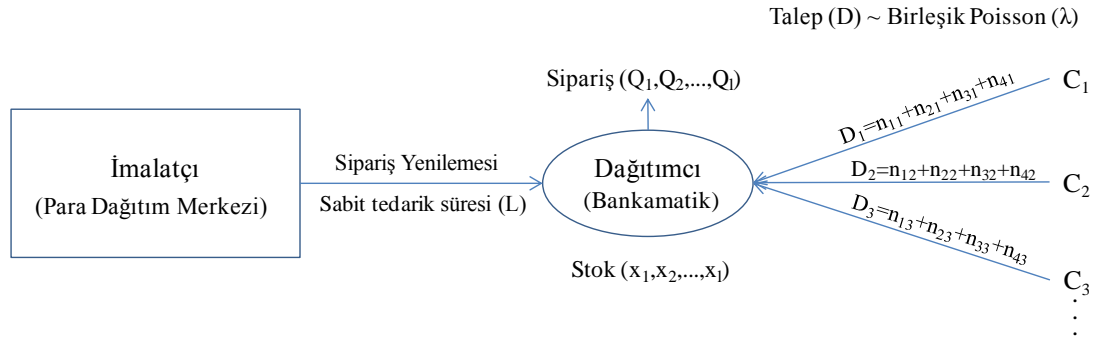
Tedarik zincirlerinde imalatçının ve dağıtımıcının mülkiyetinin tek olduğu örneklerde de benzer durumlar söz konusudur. Böyle bir durumda imalatçı ve dağıtımıcı aynı işletme olduğu yada imalatçı zincirin hakimi konumunda bulunduğu için dağıtımıcı, talebinin imalatçı tarafından düzenlenmesine karşı çıkamaz. Hatta çoğu zaman dağıtımıcının satışları ve stokları imalatçı tarafından kontrol edilir ve stoklar tamamen imalatçının kontrolünde yenilenir. Ancak, talebin imalatçı tarafından karşılandığı farklı ürün kombinasyonlarının dağıtımıcıya faydası, diğer örneklerde de olduğu gibi birbirinden farklıdır. Örneğin bir otomobil bayisinin otomobil imalatçısından siparişlerinin satışı hızlı olan renkler yerine satışı yavaş olan renkler ile karşılanması durumunda, bayinin arabaları satmak için harcayacağı çaba ve maliyet artacaktır.

Tezde ele alınan problemde kısa vadede müşteriler taleplerinin kendileri için en az faydayı sağlayan ürün kombinasyonları ile karşılanması durumunda bile, verilen kombinasyonu kabullenmek zorundadır. Ancak, yine de dağıtılan ürün kombinasyonlarıyla beklenen müşteri faydasının olabildiğince yüksek tutulması çok önemlidir. Çünkü ele alınan problemde müşteriler, ürün kombinasyonu ikamesiyle elde edecekleri fayda ne kadar düşük olursa olsun kısa vadede bu ikameyi kabullense de, uzun vadede başka alternatiflere yönelebilirler. Örneğin bankamatik müşterilerine sürekli olarak tercih etmedikleri kombinasyonlarda banknot dağıtımı yapılması durumunda müşteriler, uzun vadede çalıştıkları bankayı değiştirebilir. Benzer şekilde kendilerine hep satışı yavaş opsiyonlu arabaların gönderilmesi durumunda otomobil bayileri, başka bir otomobil firmasıyla çalışmaya başlayabilir.

1.2. Problem Tanımı

Tezde düşünülen problemde Şekil 1.1'de verildiği üzere bir dağıtımıcı ve bir imalatçı ele alınmıştır. Müşterilerin siparişlerini verdiği dağıtımıcının stok sisteminde birbirileri ile ikame birden fazla ürün çeşidi yer almakta ve bu ürün çeşitlerinin stokları eşgüdümlü olarak sabit bir tedarik süresinde imalatçı tarafından

yenilenmektedir. Problem, sonsuz zaman ufkunda birim zaman başına beklenen (ortalama) maliyeti yaklaşık en küçükleyecek ürün kombinasyonu dağıtım ve stok yenileme politikasının belirlenmesidir.



Şekil 1.1. Problemin bileşenleri

Ele alınan sistemde imalatçı ve dağıtımçı olarak gösterilen birimler, ürünlerin tedarik zinciri dağıtım ağı tasarımlarına göre birçok farklı şey olabilir. Örneğin ele alınan ürünün tedarik zincirine göre imalatçı; dağıtımçının stoklarında yer alan tüm ürün çeşitlerinin tedarikçisi, imalatçısı yada dağıtımçısı olabilir. Tezin ana motivasyonunu sağlayan bankamatik örneği ele alınacak olursa imalatçı, bir bankanın anapara dağıtım merkezi yada belli bir banka şubesidir. Benzer şekilde dağıtımçı da bir perakendeci, depo, hatta imalatçı olabilir. Bankamatik örneği için ele alınan sistemde dağıtımçı, bankamatiklerdir. Tezin ana motivasyonu bankamatiklerin stok yenileme politikası ve banknot dağıtım problemi olduğu için tezin ilerleyen bölümlerinde dağıtımçıdan bankamatik, imalatçıdan da para dağıtım merkezi olarak bahsedilecektir.

Müşterilerin bankamatiğe λ oranlı poisson sürecine uygun olarak geldikleri ve D ile gösterilen müşterilerin bireysel para talep miktarlarının bilinen $f_D(\cdot)$ olasılık fonksiyonuna uyan kesikli bir rassal değişken olduğu varsayılmaktadır. Diğer bir deyişle bir t anına dek bankamatiğe gelen toplam para talebi, λt oranlı birleşik poisson (compound poisson) sürecidir. Bankamatiğin stokları (x_1, x_2, \dots, x_l) banknot çeşitlerinden oluşmakta ve müşterilerin bireysel para talepleri (D), l farklı banknotun farklı kombinasyonlarıyla karşılanabilmektedir. Bankamatiğe gelen bireysel müşteri talebini karşılayan mümkün banknot kombinasyonlardan hangisinin seçileceğine

bankamatik karar vermektedir. Bankamatik stoklarından bireysel müşteri talebini tam olarak karşılayacak bir banknot kombinasyonu oluşturulamazsa, müşteri talebi kaybolmaktadır.

Sistemde gerçekleşen olaylar göz önüne alındığında karar vericinin (bankamatiğin) vermek zorunda olduğu iki karar vardır:

- 1) Müşterilerin bankamatikten istedikleri bireysel para talepleri, bankamatik stoklarında yer alan banknotların hangi mümkün kombinasyonu ile karşılanacak?
- 2) Para dağıtım merkezi, bankamatik stoklarında yer alan banknot çeşitlerinden
 - a) hangilerinin stoklarını,
 - b) hangi miktarlarda,
 - c) ne zaman yenileyecek?

Diğer bir deyişle problem, ürün kombinasyonu dağıtım ve stok yenileme alt problemlerinden oluşmaktadır. Tezin ilerleyen bölümlerinde de açıklandığı üzere karar vericinin belirlemek zorunda olduğu ürün kombinasyonu dağıtım ve stok yenileme kararları birbirileri ile yakından ilişkilidir ve belirlenecek politikalar birbirini etkilemektedir. Bölüm 1.2.1’de ürün kombinasyonu dağıtım problemi ve Bölüm 1.2.2’de stok yenileme problemi kısaca tanımlanmıştır.

1.2.1. Ürün kombinasyonu dağıtım problemi

Tezde ele alınan ürün kombinasyonu dağıtım problemi, sonsuz zaman ufkunda birim zaman başına beklenen (ortalama) maliyeti en küçükleyecek şekilde, stok sistemine (bankamatiğe) gelen müşteri talebinin, stoklarda yer alan ürün çeşitlerinin (banknotların) hangi kombinasyonu ile karşılanması gerektiğinin belirlenmesidir. Ürün kombinasyonu dağıtım politikasının belirlenmesinde en küçüklenmesi istenen maliyeti oluşturan kalemler, stok sistemindeki ürün (banknot) çeşitlerinin stokta

bulundurma maliyetleri, ürün (banknot) çeşitlerinin sabit sipariş verme maliyetleri, stok sistemine gelen talebin stok sisteminde yer alan ürün (banknot) çeşitlerinin stoklarından tam olarak karşılanamadığında oluşan stoksuz kalma maliyetidir. Çalışmada müşterilerin, dağıtımçı (bankamatik) tarafından sunulan her ürün kombinasyonunu kabul ettiği varsayılmaktadır. Bununla birlikte, bir müşterinin talebi karşılanırken almayı beklediği ürün kombinasyonu, dağıtımçının (bankamatik) sunduğu kombinasyon arasındaki farklılık arttıkça, müşterinin kazandığı fayda azalır. Bu nedenle ürün dağıtım politikasının belirlenmesi için kurulan modelde, müşterilerin taleplerinin en çok istedikleri ürün kombinasyonu karşılanamaması durumunda kaybedecekleri fayda da maliyet fonksiyonuna ceza olarak eklenmiştir.

Çalışmada ele alınan ürün kombinasyonu dağıtım problemi, Bölüm 2.2’de yarı Markov karar süreci (semi Markov decision process, SMDP) olarak modellenmiştir.

1.2.2. Stok yenileme problemi

Ele alınan problemde dağıtımçının (bankamatik) stok sistemi birden fazla ve birbiri ile ikame ürün (banknot) çeşitleri içermekte ve dağıtımçının (bankamatik) tüm ürün (banknot) çeşitlerinin stoklarını sabit bir tedarik süresinde aynı imalatçıdan (para dağıtım merkezi) yenilediği kabul edilmektedir. Çalışmada ayrıca imalatçının (para dağıtım merkezinin) ürün (banknot) kaynağının ve taşıma araçlarının kapasitelerinin sonsuz olduğu varsayılmaktadır.

Tezde ele alınan problemdeki gibi ürün çeşitlerinin stoklarının aynı imalatçıdan yenilenmesi durumlarında, stok sisteminde yer alan ürün çeşitlerinin stoklarının birbirinden bağımsız olarak yenilenmesi yerine eşgüdümlü bir stok yenileme politikası uygulanması, oluşacak ölçek ekonomisi nedeniyle önemli maliyet avantajları sağlar. Burada ölçek ekonomilerini, eşgüdüm sonucu birlikte sipariş edilen ürünlerin aynı ulaşım aracıyla -tam dolu kamyonlarla- taşınması oluşturur. Bankamatik problemde müşterilerin tüm para çekim işlemleri, elektronik ortamda ve anlık olarak takip edilmektedir. Bu nedenle banknot stoklarının “sürekli gözden geçirme” prensibine göre kontrol edilmesi mantıklıdır.

Tezde stokların yenilenmesi kararının “can order” (S_i, c_i, s_i) eşgüdümlü sürekli gözden geçirme politikasına göre verildiği kabul edilmiştir. Burada $i=1,2,\dots,l$ olmak üzere S_i , i 'nci ürünün (banknotun) stoklarının tamamlanacağı üst seviyeyi (up to level); c_i , başka bir ürün (banknot) çeşidinin stoklarının yenilenecek olması durumunda, i 'nci ürünün (banknotun) stoklarının da yenilenmesini tetikleyen stok düzeyini (can order level) ve “ s_i ”, diğer ürünlerin (banknotların) stoklarının yenilenecek olup olmamasına bakmaksızın, i 'nci ürünün (banknotun) stoklarının yenilenmesini tetikleyen stok düzeyidir (must order level). (S_i, c_i, s_i) politikasında doğal olarak $c_i \geq s_i$ 'dir.

Bölüm 2.3'te detaylandırılan (S_i, c_i, s_i) politikası basit, uygulanması kolay ve optimale yakın sonuçlar veren; ancak, kontrol parametrelerinin belirlenmesi oldukça zor olan bir politikadır. Tezde (S_i, c_i, s_i) politikasının parametrelerinin belirlemek için Silver [1, 3] ve Thompstone ve Silver [2] çalışmalarındaki yöntemler kullanılmıştır.

1.3. Literatür Taraması

Tezde ele alınan problem, ürünlerin birbirileri ile ikame edilebildiği çok ürünlü bir stok sisteminde, müşteri talebini karşılayacak şekilde ürün kombinasyonlarının dağıtımı ve stokların eşgüdümlü olarak yenilenmesi politikalarının birlikte bulunmasını içermektedir. Bu nedenle literatür çalışması, ürün ikameleri ve eşgüdümlü stok yenileme olarak iki bölümde ele alınmıştır.

1.3.1. Ürün ikamesiyle ilgili çalışmalar

Ürün ikamesi (product substitution), bir ürünün talebinin tamamının ya da belli bir miktarının, bu ürün yerine başka bir ürün tarafından karşılanması anlamına gelir. Genellikle müşteriler taleplerini karşılayabilecek birbirinin yerini tutan birden fazla farklı ürün olması durumunda bile, taleplerinin karşılanması için öncelikli ürün tercihlerine sahiptir. Çok ürünlü stok sistemlerinde ürün ikamelerinin ürün stokları üzerinde büyük etkisi vardır. Ürün ikamesini dikkate alan stok literatüründe ürün ikamesi başlıca iki bölümde incelenmiştir: Müşteri kaynaklı ürün ikamesi

(customer-driven substitution) ve firma kaynaklı ürün ikamesi (firm-driven substitution).

Firma kaynaklı ürün ikamesinin oluştuğu ilk durum, stok sisteminin “1” tane farklı ve aşağı tek yönlü ikame ürünlerden oluştuğu durumdur. Aşağı tek yönlü ürün ikamesi (downward substitution) yapısı stoklarda özelliği birbiriyle eşdeğer, ancak farklı performans özelliklerine sahip ürünler bulunması durumunda açığa çıkar. Aşağı tek yönlü ikame durumunda performansı düşük ürünün talebi, stokların durumuna göre firma tarafından daha iyi performanslı bir ürün ile ikame edilebilir; ancak bunun tersi mümkün değildir. Bu tür ürün ikamesi, ürünlerin performans, kalite vb. özelliklerine göre sınıflandırıldığı durumlarda uygundur. Örneğin düşük hızlı bir bilgisayar işlemcisine gelen talebin bir kısmı yada tamamı, stokların durumuna göre daha yüksek hızdaki işlemciyle ikame edilebilir.

Bir ürünün farklı özellikteki çeşitlerinin (ebat, kalite, performans vb.) tümünün stoklanması, depolama kısıtları, ölçek ekonomisi yada tüm farklı çeşitleri belli miktarlarla stokta bulundurmaya ilgili maliyetler nedeniyle mantıklı değildir. Ürünün tüm çeşitlerini stokta bulundurmak yerine daha az sayıda çeşitten belli miktarlarda bulundurulması, stokta bulundurulmayan ürün çeşitlerinin talebi bir ikame maliyetiyle diğer ürün çeşitlerinin stoklarından karşılanabilir. İlişkili maliyetleri en küçükleyecek şekilde hangi ürün çeşidinden ne kadar stok bulundurulması gerektiğinin belirlenmesi problemi, ürün sınıflandırma (product assortment) problemi olarak adlandırılır [4]. Ürün sınıflandırma problemi için Wagner ve Within [5]’de yer alan örnekte mukavemeti yüksek çelik profiller, mukavemeti daha düşük çelik profillerin talebinin ikamesinde kullanılabilir. Leachman [6]’da yer alan diğer bir örnekte ise düşük kapasiteye sahip hafıza çiplerinin talebi, daha yüksek kapasiteye sahip çiplerle ikame edilmektedir.

Talebin stokastik olarak kabul edildiği durum için tek bir yönde ikamenin yapılabildiği ve hazırlık maliyetinin ihmal edildiği çok ürünlü stok problemi, ilk olarak Ignall ve Veinott [7] tarafından incelenmiştir.

Bassok ve Hsu [8] makalelerinde tek periyotlu, çok ürünlü ve aşağı tek yönlü ürün ikamesi durumunu ele almışlardır. Yazarların modelinde üretim girdisi olarak tek bir hammadde ve çıktı olarak N farklı ürün bulunmakta, çıkan ürün ve talep ise rassaldır. Problem, optimal ürün girdisi miktarının ve çıkan ürünlerin, talebi karşılayacak şekilde farklı ürün sınıflarına dağıtımıdır. Yazarlar etkili algoritmalar geliştirmek için problemin şebeke (ağ) yapısını kullanmışlardır.

Bassok ve arkadaşları [9] ele aldıkları ürün sınıflandırma probleminde, her bir sınıfın talebinin stokastik olduğunu varsayarak, problemi sipariş yenileme ve dağıtım problemi olarak iki aşamada ele almıştır. Yazarlar tek periyot için genel bir kâr enbüyükleme modeli geliştirmiş ve açgözlü (greedy) bir ürün dağıtım politikasının optimal olduğunu göstermiştir.

Rao ve arkadaşlarının [10] ele aldığı problem, Bassok ve arkadaşlarının [9] ele aldığı probleme benzerdir. Bassok ve arkadaşlarından [9] farklı olarak Rao ve arkadaşlarının [10] modelinde birim ürün ikame maliyetinin özdeş olması zorunlu değildir. Rao ve arkadaşları [10] problemi, iki aşamalı tam sayı stokastik programlama ile modellemiştir. Yazarlar, ilk aşamada hangi üründen ne kadar üretileceğini ve ikinci aşamada gerçekleşen talebin hangi ürün dağılımıyla karşılanacağını bulmak için, şebeke akışı, dinamik programlama ve simülasyon tabanlı optimizasyon tekniklerinin bir kombinasyonunu kullanarak, etkili sezgiseller geliştirmiştir.

Liu ve Lee [11], tek yönlü ikameye izin veren çok ürünlü (S,S-1) stok politikaları için Markov modeli sunmuştur. Yazarlar, farklı ürünler için gelen talebin ve geciktirilen talebin (back order) ikamesi durumlarını ele almıştır. Ürün sayısı arttıkça Markov modellerini çözmek için gereken hesaplama güçleştiği için, yazarlar bir ayrıştırma yaklaşımı kullanmıştır.

Firma kaynaklı ürün ikamesinin olduğu diğer durum ise, firmaların stok tutma birimleri arasında tek yönlü transit ürün taşınması (transshipment) durumlarında açığa çıkar. Transit taşıma problemleri, ürünlerin stoklarının yeterli olduğu stok birimlerinden, talebin olduğu ve stokun yetersiz olduğu stok birimlerine ürünlerin

taşınmasını içerir. Burada müşterilerin, ürünlerin taşınma süresini beklemeye gönüllü oldukları kabul edilir. Bir ürünün “I” stok birimine tek yönlü transit nakil problemi, bir stok birimindeki “I” ürünün tek yönlü ikamesi problemiyle benzerdir [9, 12]. Oluşan talebin karşılanması için hangi mağazadan transit taşıma yapılacağı kararı firma tarafından verildiği için, bu problem firma kaynaklı ürün ikamesi literatüründe ele alınır. Transit taşıma problemlerinde, stok birimlerinde ne kadar stok bulundurulması ve karşılanamayan talep için hangi birimden taşıma yapılması gerektiği araştırılmaktadır. Literatürde transit taşıma problemleriyle ilgili birçok çalışma vardır, ancak bunların çoğu taşıma yönüyle ilgili bir kısıta sahip değildir. Bununla birlikte, firma kaynaklı ürün ikamesi literatüründe tek yönlü taşımanın dikkate alındığı çalışmalar da vardır.

Axsäter [13], tek yönlü transit taşıma problemleriyle ilgili olarak politika değerlendirme için basit ve etkili bir yaklaşım tekniği sunmuş ve tekniğin performansını simülasyon çalışmasıyla değerlendirmiştir. Axsäter [13]’de ele alınan sistemde her bir lokasyon (S,S-1) ya da (R,Q) politikası uygular ve tedarik süresince gelen talebin normal dağıldığını kabul eder. Stoklardan ya da diğer depolardan nakille karşılanamayan talep ise, ertelenmektedir.

Olsson [14], Axsäter [13]’deki aynı sistemi düşünerek, talebin daha sonra karşılanmak üzere geciktirilebildiği ve kaybolduğu iki durumu ele almıştır. İki makalenin arasındaki temel fark, talebin modellenmesinden gelir. Olsson [14]’da her bir lokasyondaki müşteri talebinin birbirinden bağımsız olduğu ve poisson dağılımına uyduğu kabul edilir ve talebin geciktirilebildiği durum için geliştirilen model, Axsäter [13]’deki modelle doğrudan karşılaştırılır. Olsson [14]’da, özellikle göreceli olarak müşteri talep oranları düşük olduğunda daha iyi sonuçlar elde etmiştir.

Müşteri kaynaklı ürün ikamesinde ikame kararı firma tarafından değil, müşteri tarafından her bir ikame ürünün kendisine olan faydası dikkate alınarak verilir.

Netessine ve Rudi [15], keyfi sayıda ürünün bulunduğu bir stok sisteminde müşteri kaynaklı ürün ikamesi durumundaki optimal stok kontrol politikalarını incelemiştir.

Burada her bir müşterinin ürünler arasından bir ilk seçimi vardır ve eğer müşteri ilk seçimini stoklarda bulamazsa, bu ürün yerine benzer özellikte farklı bir ürün seçebilmektedir. Müşterinin talebi, seçilen bu ürün tarafından da karşılanamazsa talep kaybolmaktadır. Netessine ve Rudi [15], tüm ürün stoklarının toplam kârı en büyükmek isteyen merkezi bir karar verici tarafından yönetildiği ve her bir ürünün stoklarının bu ürünle ilgili kârı en büyükmek isteyen bağımsız karar vericiler tarafından yönetildiği iki senaryoyu ele almıştır.

Smith ve Agrawal [16] makalelerinde, stokastik talep altında perakende ürün sınıflandırmalarının (retail assortments), stok yönetimi ve müşteri hizmet düzeyi üzerindeki etkisini analiz etmiştir. Yazarlar, belli kaynak kısıtları altında beklenen kârı maksimize edecek şekilde, ürün sınıflarını ve bu sınıfların stok düzeylerini belirlemek için bir yöntem geliştirmiştir. Burada firma kaynaklı ürün ikamesinden farklı olarak müşteri, eğer talep ettiği ürün stoklarda yoksa talebini başka bir ürünle karşılayabilmektedir. Bu makalede müşterinin seçiminin deterministik bir ürün ikame modeline uyduğu kabul edilmiştir.

Anupundi ve arkadaşları [17], stoksuz kalmanın müşteri ürün ikame davranışı üzerindeki etkisini incelemiştir. Yazarlar, müşterilerin poisson sürecine göre geldiğini ve stokların periyodik olarak yenilediği varsayımı altında, stoksuz kalma durumunda iki ikame edilebilir ürün sınıfının talebini tahmin etmekle ilgilenmiş, ancak dinamik ürün ikamesinin stoklar üzerindeki etkisini düşünmemiştir.

Noonan [18], müşterilerin bir birinci ve ikinci seçimi olduğunu ve talebin iki aşamada ortaya çıktığını kabul eder. Birinci aşamada birinci seçim talebi gerçekleşir ve mümkün olduğunca mevcut stoklardan karşılanır. Birinci aşamada karşılanamayan birinci seçim talep, ikinci aşamada deterministik ikame oranlarıyla ikinci seçime dönüştürülür.

Mahajan ve Ryzin [19], ürün ikamelerini Noonan [18]'a benzer şekilde modellemiştir. Farklı olarak bu çalışmada seçim, tüm mevcut ürünler arasından yapılmakta ve müşterinin seçimi kendi faydasının en büyükmesine dayanmaktadır. Yani her bir ikame ürün müşterilere farklı derecelerde fayda

sağlamakta ve müşteri, stok durumuna göre kendi faydasını en büyükleyecek şekilde ürün seçimi yapmaktadır. Bu makalede birbirinden farklı rassal fayda modelleri kullanılmıştır.

Rajaram ve Tang [20], sipariş miktarları ve beklenen kârlar üzerinde ürün ikamesinin etkisini analiz etmiştir. Yazarlar bir hizmet oranı için geliştirdikleri sezgiseli, talebin belirsizlik ve korelasyon düzeyinin ne kadar olduğunu ve ürünler arasındaki ikame derecesinin ikame edilebilen talep altında beklenen kârı ve sipariş miktarlarını ne ölçüde etkilediğini incelemek için kullanmışlardır.

Pentico [21], ürün sınıflandırma (assortment) problemi konusundaki son 50 yılı içeren literatürü incelemiştir. Ürün sınıflandırma problemi özelinde ürün ikamesi konusunda daha fazla çalışma için Pentico [21]'ya bakılabilir.

Tezde ele alınan problemin ürün ikamesi yapısı, literatürde yer alan çalışmalardan birçok yönden farklılık gösterir. Ele alınan problemde ürün ikamesi, temelde firma kaynaklı olarak açığa çıkmaktadır; ancak yine de birçok yönden literatürde yer alan firma kaynaklı ürün ikamesinden farklıdır. Diğer taraftan temeldeki farklılığına rağmen ele alınan problemdeki ürün ikamesi, birçok yönden de müşteri kaynaklı ürün ikamesi ile benzerlikler taşır. Her şeyden önce firma kaynaklı ürün ikamesi literatürde, aşağı tek yönlü ürün ikamesi olarak açığa çıkar. Burada stok sisteminde yer alan ürün çeşitleri arasında bir hiyerarşi söz konusudur ve düşük özelliklere sahip ürün çeşitlerinin talebinin stoklardan karşılanamaması durumunda bu ürün çeşidinin talebinin karşılanamayan bölümü, firma tarafından ürün hiyerarşisinde daha üst bölümde yer alan diğer ürün çeşitleri tarafından ikame edilebilir. Örneğin 2 Ghz'lik mikro işlemci talebi, stokların yeterli olmaması durumunda firma tarafından 2.2 Ghz'lik mikro işlemcilerle ikame edilebilir. Burada firma, müşterinin talep ettiği için, bu tür bir ikame müşteri faydasında bir azalma (memnuniyetsizlik) oluşturmaz. Müşteri kaynaklı ürün ikamesinde ise müşteri, istediği ürünü bulamadığında diğer ürünlerin faydasını dikkate alarak, kendisi için en fazla faydayı veren asıl istediği ürünün yerine başka bir ürünü ikame edebilir. Eğer müşteri, mevcut ürünler arasında düşündüğü eşğin üzerinde fayda sağlayacak bir ürün bulamazsa, kayıp satış durumu (lost sales) oluşur.

Çalışmada ele alınan problemde ise, stok sistemindeki ürünler arasında bir hiyerarşi söz konusu değildir ve bu yönüyle firma kaynaklı ürün ikamesinden farklılık gösterir. Yani firma karşılayamadığı müşteri talebini, daha üstün bir ürünle ikame etmez. Bu nedenle tıpkı müşteri kaynaklı ürün ikamesindeki gibi, ürün talebinin farklı ürün ikameleriyle karşılanmasının müşteriye faydası, birbirinden farklıdır.

1.3.2. Eşgüdümlü stok yenileme politikaları ile ilgili çalışmalar

İlk kez Balintfy [22], çok ürünlü stok problemlerinde (S,c,s) tipi eşgüdümlü stok politikaların kullanılmasını savunmuştur. Ignall [23], birlikte sipariş yenilemeyle ölçek ekonomisinden faydalanmayı sağlayan tüm optimal politikaların çok karmaşık bir yapıya sahip olduklarını göstermiştir. Yazar, (S,c,s) politikalarının çalışmasındaki iki ürünlü basit bir problem için bile her zaman optimal olmadığını ifade etmiştir. Bununla birlikte, uygulanmasındaki kolaylık nedeniyle (S,c,s) benzeri optimale yakın stok politikaları literatürde oldukça ilgi görmüştür.

Curry ve arkadaşları [24] ile Schaack ve Silver [25], (S,c,s) stok politikası altında optimal kontrol parametrelerinin belirlenmesi için prosedürler geliştirmiştir, ancak bu prosedürlerin her ikisi de çok fazla zaman tüketici simülasyon tabanlı prosedürlerdir. Özellikle Silver'in (S,c,s) politikası için kontrol parametrelerinin belirlenmesinde simülasyon içermeyen algoritmalar sunduğu çeşitli çalışmalar mevcuttur. Silver [1], talebin poisson dağılımına uyduğu kabulüyle çok ürünlü stok problemlerinde (S,c,s) kontrol parametrelerinin belirlenmesi için pratik bir prosedür geliştirmiştir. Yazar, çok ürünlü stok sisteminde yer alan bir ürünün, birlikte stok yenileme için diğer ürünlere μ oranlı poisson süreciyle tetiklendiği varsayımıyla, her bir ürünün kontrol parametrelerini ayrı ayrı hesaplamaktadır. Silver [1]'in modelinde, stoklardan karşılanamayan talebin daha sonra karşılanmak üzere geciktirildiği (backorder) varsayılmaktadır. Yazar, öncelikle ürünlerin siparişlerinin birbirinden bağımsız olarak yenilendiği durumu ele almış ($c=s$) ve stokların tam olarak "s" ye düşmesiyle yenilendiğini kabul ederek, verilen bir sabit tedarik süresi ve hizmet düzeyi için kontrol parametrelerini bulmuştur. Yazar, ikinci aşamada eşgüdümlü kontrol durumunda tedarik süresini ihmal ederek, verilen hizmet düzeyi kısıtı altında (S,c,s) kontrol parametrelerini hesaplamıştır. Yazar çalışmasında son olarak eşgüdümlü

durumda tedarik süresinin ihmal edildiği durum için hesapladığı kontrol parametrelerinden hareketle, tedarik süresinin sıfırdan farklı olduğu durum için kontrol parametrelerini hesaplamıştır. Bu makaledeki sayısal denemelerde stokların eşgüdümlü ve birbirinden bağımsız olarak yenilediği durumlardaki maliyetler karşılaştırılmış, eşgüdümlü stok yenileme sonucu önemli maliyet kazançları (ortalama % 18,8) elde edilmiştir.

Thompstone ve Silver [2], Silver [1]'i talebin birleşik poisson (compound poisson) dağılımına uyduğu duruma genişletmiştir. Yazarlar tedarik süresinin ihmal edildiği varsayımı altında bağımsız kontrol ve eşgüdümlü kontrol durumları için verilen hizmet düzeyinde kontrol parametrelerini belirlemişlerdir. Gerçekleştirilen denemelerde bağımsız kontrol durumuna göre eşgüdümlü kontrol ile ortalama % 15,9 daha düşük maliyetler bulunmuştur.

Siver [3], Thompstone ve Silver [2]'i tedarik süresinin ihmal edildiği varsayımından, sabit bir tedarik süresinin varsayıldığı duruma genişletmiştir. Yazarlar burada ardışık bir yaklaşım kullanmış, Thompstone ve Silver [2]'de tedarik süresinin ihmal edildiği varsayımıyla hesapladıkları kontrol parametrelerini girdi olarak kullanarak, sabit bir tedarik süresi için verilen bir hizmet düzeyini sağlayacak şekilde kontrol parametrelerini belirleyen bir algoritma geliştirmişlerdir.

Bu güne kadar (S,c,s) politikasındaki parametreleri daha iyi şekilde belirlemek için birçok çalışma yapılmıştır. Federgruen ve arkadaşları [26], tek ürünlü stok sisteminde (S,c,s) politikasının optimal olduğunu varsayarak, optimal politikayı hesaplamak için bir politika yineleme (policy iteration) algoritması geliştirmiştir. Yazarların ele aldığı çok ürünlü stok sistemi, Silver [3]'de ele alınan sistemin aynısıdır. Burada da talebin birleşik poisson sürecine uyduğu, karşılanamayan talebin geciktirildiği ve sabit bir tedarik süresi bulunduğu varsayılmaktadır. Kontrol parametrelerinin belirlenmesi için geliştirilen bu algoritmalarındaki temel düşünce, çok ürünlü bu sistemlerin uygun bir şekilde ayrıştırılmış tek ürünlü sistemlere dönüştürülerek çözülmesidir. Yazarların çalışmasında algoritma sonlu adımlarda durdurulmakta, ancak bir optimal politikaya yakınsandığı kanıtlanmamaktadır.

Yazarlar, modelin benzer maliyet yapılarıyla çok lokasyonlu stok sistemlerinde eşgüdümlü dağıtımlar için de kullanılabileceğini vurgulamıştır.

Zheng [27], (S,c,s) politikasının poisson talepli ve tek ürünlü stok sistemlerinde Markovian indirim fırsatı için uzun dönemde optimal olduğu varsayımını kanıtlamıştır. Zheng [27]'in ele aldığı stok sistemi, Silver [3] ile Federgruen ve arkadaşları [26]'nın stok sistemleriyle aynıdır ve çok ürünlü sistemini, tek ürünlü sisteme ayrıştırmak için aynı yöntem kullanılmıştır. Burada çok ürünlü stok sistemindeki her bir ürün, merkezi olmayan bir şekilde kontrol edilmektedir. Diğer bir deyişle her bir ürün ayrı bir stok tutma birimidir ve kendi toplam maliyetini minimize etmek için yönetilir. Ignall [23]'da verildiği üzere merkezi bir kontrol altında çok ürünlü stok sistemleri için ise, (S,c,s) politikası optimal değildir.

Melchior [28], ürünlerin taleplerinin poisson dağılımına uyduğu çok ürünlü bir stok sistemi ele almış ve (S,c,s) stok politikası parametrelerinin belirlenmesi için bir siparişte yer alan ürünün, sipariş fırsatından yararlanan diğer ürünlerden bir karşılık (tazminat) aldığı yeni bir hesaplama metodu geliştirmiştir. Yazar, kendi hesaplama yönteminin performansını kıyaslamak için gerçekleştirdiği küçük bir simülasyon çalışmasıyla, bulduğu optimal (S,c,s) politikasını, Federgruen ve arkadaşları [26], Thompstone ve Silver [2]'in (S,c,s) politikalarıyla ve periyodik gözden geçirme P(S,s) politikası performanslarıyla karşılaştırmıştır. Melchior [28], (S,c,s) politikalarının periyodik gözden geçirmeden daha iyi sonuçlar verdiği (birincil sipariş maliyetinin göreceli olarak düşük) durumlarda kendi, hesaplama yönteminin daha iyi performansa sahip olduğunu ifade eder.

Stokların eşgüdümlü olarak yenilenmesinin maliyet kazanımları yarattığı birçok çalışmayla açıkça gösterilmesine karşın, problemin karmaşıklığı nedeniyle en iyi eşgüdümlü kontrol politikasının ne olması gerektiği hala açık değildir. Literatürde yer alan birçok çalışmada periyodik gözden geçirme esasına dayanan eşgüdümlü stok politikasının performansı, (S,c,s) politikasının performansı ile kıyaslanmıştır.

Atkins ve Iyogun [29] ile Pantumsinchai [30], ürün taleplerinin poisson dağılımına uyduğu ve stoklardan karşılanamayan talebin geciktirildiği durum için farklı sipariş

maliyetleri düzeyinde (S,c,s) ve periyodik gözden geçirme stok politikalarının performanslarını incelemişlerdir. Yazarlar, Federgruen ve arkadaşları [26]'nın çalışmasındaki algoritmayı kullanarak elde ettikleri (S,c,s) politikasıyla, QS ve RS periyodik gözden geçirme politikalarının performanslarını karşılaştırdıkları çalışmalarında, birincil sipariş maliyet arttıkça çoğu zaman QS ve RS periyodik gözden geçirme politikalarının daha iyi performans gösterdiği ileri sürmüşlerdir. Pantumsinchai [30], yaptığı deneme sonuçlarına göre (S,c,s) politikasının birincil sipariş maliyetlerinin düşük olduğu durumlarda iyi bir performans gösterdiğini ve oldukça da gürbüz (robust) olduğunu göstermiştir. Yazar, QS politikası için ise, her bir ürün çeşidinin bir mutlak sipariş düzeyi (must order level) bulunmadığından, bu politikanın çok gürbüz (robust) olmadığını ve politikanın performansının problem parametrelerinin etkileşimine duyarlı olduğunu belirtmiştir.

Eijs [31], göreceli olarak yüksek birincil sipariş maliyeti durumlarında (S,c,s) politikalarının kötü performans göstermesinin nedeninin politikanın kendisinden değil, Silver[1, 3], Silver ve Thompstone [2] ile diğer yazarların poisson oranlı özel stok yenileme fırsatı kabulünden kaynaklandığını ifade eder. Yazar, birincil sipariş maliyetlerinin göreceli olarak yüksek olduğu durumlar için $c=S-1$ olan (S,S-1,s) politikasının uygulanması gerektiğini ileri sürer. Bunun anlamı stok tutma ve sipariş maliyetlerini en küçükleme için bir ürünün siparişi tetiklendiğinde, stoklardaki tüm ürünlerin siparişlerinin yenilenmesi gerektiğidir.

Johansen ve Melchior [32], farklı talep değişkenliği durumlarında periyodik gözden geçirme politikalarıyla, (S,c,s) politikasının performansını karşılaştırmıştır. Yazarlar, talep değişkenliğinin küçük olması durumunda periyodik gözden geçirme politikasının performansının, (S,c,s) politikası performansından belirgin bir şekilde iyi olduğunu; talep değişkenliğinin yüksek olması durumunda ise (S,c,s) politikasının periyodik gözden geçirme politikasına göre % 15'e varan düzeylerde daha iyi performansa sahip olduğunu göstermiştir. Yazarlar, talep değişkenliğinin yüksek olduğu durumlarda (S,c,s) politikasının çok daha iyi bir seçim olacağını ileri sürmektedir.

Kısaca özetlenecek olursa, birincil sipariş maliyetlerinin göreceli olarak düşük ve talepteki değişkenliğin yüksek olması durumlarında sürekli gözden geçirme (S,c,s) politikaları iyi performans göstermektedir. Tersine durumda ise, periyodik gözden geçirme politikalarının performansı daha iyidir. Birincil sipariş maliyetlerinin aşırı yüksek olması durumunda siparişlere stoktaki tüm ürünlerin katılması, çok düşük olması durumunda ise birbirinden bağımsız stok yenileme politikasının kullanılması gerekir. (S,c,s) politikaları sürekli gözden geçirmeye dayalı olduğu için periyodik gözden geçirme politikalarına kıyasla yeni bilgiye çok daha hızlı tepki verir. Bu nedenle, (S,c,s) politikalarının periyodik gözden geçirme politikalarına göre daha iyi performans göstermesi beklenir. Ancak, literatürdeki çalışmalar birincil sipariş maliyetinin yüksek olduğu durumlar için bunun aksini göstermektedir. Melchior [28]'a göre sezgilerimizin tersini söyleyen bu durum ya (S,c,s) politikasının kendisinden kaynaklanabilir, yani bu politikanın aşırı basit olmasının bir sonucu olabilir, ya da (S,c,s) politikalarından değil, kontrol parametrelerinin hesaplanma yöntemlerinden kaynaklanabilir.

Tezin sonraki bölümleri şu şekilde düzenlenmiştir: 2. bölümde, dağıtım merkezinin ve ürünlerin talep süreçlerinin ele alınan problemi nasıl etkilediği açıklanmış ve problem ürün kombinasyonu dağıtım problemi ve eşgüdümlü stok yenileme problemi olarak ikiye ayrıştırılmıştır. Ürün kombinasyonu dağıtım problemi yarı Markov karar süreci (semi Markov decision process, SMDP) olarak modellenmiş, eşgüdümlü stok yenileme problemi ise literatürdeki çalışmalardan yararlanılarak çözülmüştür. 3. bölümde birbirileri ile etkileşimli bu iki problemi birlikte, sıralı ve yinelemeli olarak çözen bir algoritma geliştirilmiş ve ürün kombinasyonu dağıtım problemi, çok katmanlı yapay sinir ağı kullanan fonksiyon yaklaşımli ödüllü öğrenme (reinforcement learning) algoritmalarıyla çözülmüştür. 4. bölümde farklı ölçeklerdeki uygulamalı örnekler geliştirilen algoritmayla çözümlenerek, deneysel sonuçlar verilmiştir. Son bölümde ise çalışmayla ilgili genel sonuçlar ve öneriler yer almaktadır.

BÖLÜM 2. MODEL

Bölüm 1.2 problem tanımı bölümünde verildiği üzere çalışmada ele alınan dağıtımıcının stok sistemine gelen talep sürecinin, birleşik poisson süreci olduğu kabul edilmektedir. Ancak, stok yenileme politikasının belirlenebilmesi için stok sisteminde yer alan her bir ürün çeşidinin talep sürecinin bilinmesi gerekir. Stok sistemindeki ürün çeşitlerinin talep süreci ise, uygulanan ürün kombinasyonu dağıtım politikasıyla yakından ilişkilidir. Bu nedenle Bölüm 2.1’de dağıtımıcının stok sisteminin ve bu stok sisteminde yer alan ürün çeşitlerinin talep süreçleri tartışılmıştır. Bölüm 2.2’de ürün kombinasyonu dağıtım problemi yarı Markov karar süreci olarak modellenmiş, Bölüm 2.3’te ise literatürdeki çalışmalardan faydalanarak belirli bir hizmet düzeyi kısıtı altında eşgüdümlü bir stok yenileme politikasının parametrelerinin nasıl belirleneceği araştırılmıştır.

2.1. Ürün Talep Süreci

Tezde ele alınan sistem incelendiğinde iki farklı ürün talep süreciyle karşılaşılmaktadır. Bu talep süreçlerinden ilki çok ürünlü stok sistemine (bankamatige) gelen müşteri talep süreci, diğeri ise stok sisteminde yer alan her bir ürün çeşidinin (banknotların) müşteri talep sürecidir. Bu iki süreç birbiriyle yakından ilişkili ve bağımlıdır. Bölüm 2.1.1 ve Bölüm 2.1.2’de bu iki talep süreci ayrıntılı olarak tartışılmıştır. Bölüm 2.1.3’te stok sistemindeki her bir ürün çeşidinin tedarik süresince talebinin dağılımı ve Bölüm 2.1.4’te ise birleşik poisson talep sürecinin, eşdeğer poisson talep sürecine nasıl dönüştürülebileceği araştırılmıştır.

2.1.1. Çok ürünlü stok sistemlerinin ürün talep süreci

Çalışmanın ana motivasyon kaynağı olan bankamatik probleminde, bankamatige müşteri gelişlerinin λ ortalamalı poisson süreci ve gelen müşterilerin bireysel para

talepleri D 'nin bilinen $f_D(\cdot)$ dağılımına uyan bir kesikli rassal değişken olduğu varsayılmaktadır. Doğal olarak, para çekmek amacıyla bankamatığe gelen her müşterinin mutlaka bankamatikten para talebinde bulunacağı ve bu nedenle müşterilerin bireysel talebinin sıfır (0) olamayacağı ($f_D(0) = 0$) kabul edilir. Bu durumda bankamatığın para talebi süreci, müşteri gelişlerinin λ ortalamalı poisson süreci olduğu ve müşterilerin bireysel para taleplerinin $f_D(0) = 0$ olmak şartıyla $f_D(\cdot)$ kesikli dağılımına uyduğu birleşik poisson sürecidir.

2.1.2. Çok ürünlü stok sistemindeki ürün çeşitlerinin talep süreci

Bankamatik stoklarında yer alan her bir banknot çeşidinin bireysel müşteri talebi, tamamen müşterilerin bankamatikten istedikleri bireysel para talebine ve bankamatığın bu talebi karşılamak için uyguladığı banknot kombinasyonu dağıtım politikasına bağlıdır. Bir anlamda her bir banknotun talep süreci, bankamatığın uyguladığı dağıtım politikası tarafından belirlenir. Bu nedenle bankamatığe para çekmek için gelen müşterilerin talebi karşılanırken, stoklarda yer alan bazı banknotların talebi sıfır (0) olarak gerçekleşebilir. Örneğin stoklarında 10 TL, 20 TL, 50 TL ve 100 TL banknotlarından oluşan 4 ürün çeşidine sahip bir bankamatik ele alınsın. Bankamatığe gelen bir müşterinin talebi 100 TL ise ve bankamatik uyguladığı banknot dağıtım politikasına göre bu talebi tek bir 100 TL ile karşılarsa, bu durumda bu müşterinin 10 TL, 20 TL ve 50 TL talebi sıfır (0) ve 100 TL talebi 1 adet olarak gerçekleşmiş olur. D_i , ($i=1,2,\dots,l$) olmak üzere i 'nci ürünün bireysel

talep miktarını gösteren kesikli rassal değişken olsun. Bu durumda $D = \sum_{i=1}^l D_i$ olur.

$f_{D_i}(\cdot)$, i 'nci ürünün kesikli olasılık fonksiyonu olarak tanımlansın. Bankamatikteki tüm banknot çeşitleri için müşteri talebinin sıfır (0) olması söz konusu olduğu için, $f_{D_i}(0) > 0$ 'dır. Bankamatığe müşteriler λ ortalamalı poisson sürecine göre geldiğine göre, i 'nci banknotun talebi, $f_{D_i}(0) > 0$ olmak üzere λ oranlı "birleşik poisson" sürecine uyar.

Axäter [33]'e göre $f_{D_i}(0) > 0$ olması durumunda, birleşik poisson talep sürecini λ'_i ve $f'_{D_i}(\cdot)$ ile karakterize edebileceğimiz eşdeğer bir süreçle değiştirebiliriz. Bu durumda oluşacak yeni süreçte i 'nci banknot için müşteri geliş oranı $\lambda'_i = \lambda(1 - f_{D_i}(0))$ ve bireysel talebin olasılık fonksiyonu Denklem 2.1'deki gibi olur.

$$f'_{D_i}(d_i) = \begin{cases} \frac{f_{D_i}(d_i)}{1 - f_{D_i}(0)} & , d_i = 1, 2, \dots \\ 0 & , \text{diğer} \end{cases} \quad (2.1)$$

2.1.3. Tedarik süresince gelen birleşik poisson talebinin dağılımı

Bölüm 2.1.2'de i 'nci banknot çeşidi için müşteri gelişinin, ortalaması λ'_i olan poisson sürecine, bireysel ürün talebinin ise bankamatik banknot dağıtım politikasına ve bankamatikten istenen bilindiği varsayılan para talebinin dağılımına göre şekillenen kesikli $f'_{D_i}(\cdot)$ olasılık fonksiyonuna uyduğu gösterilmiştir. Burada λ'_i ve $f'_{D_i}(\cdot)$, müşterinin i 'nci banknot talebinin sıfır olma olasılığı göz önünde bulundurularak dönüştürülen λ_i ve $f_{D_i}(\cdot)$ talep sürecinin eş değer sürecidir.

$d_{i_{\text{enb}}}$, i 'nci banknot çeşidinden bir müşterinin isteyebileceği en büyük bireysel talep miktarını; $X_{i(L)}$ ise, L sabit tedarik süresince i 'nci banknot çeşidi için gelen toplam talebi gösteren rassal değişken olarak tanımlansın. Bu durumda i 'nci banknot çeşidi için sabit L tedarik süresince gelen toplam talebin olasılık dağılımı $f_{X_{i(L)}}(\cdot)$, Adelson [34]'da verilen yaklaşım kullanılarak Denklem 2.2'den hesaplanabilir.

$$f_{X_{i(L)}}(x) = \begin{cases} \exp(-\lambda'_i L) & , x = 0 \\ \frac{1}{x} \sum_{j=0}^{x-1} x-j a_{x-j} f_{X_{i(L)}}(j) & , x = 1, 2, \dots \end{cases} \quad (2.2)$$

$$a_{d_i} = \lambda'_i L f'_{D_i}(d_i) \quad , d_i = 1, 2, \dots, d_{i_{\text{enb}}}$$

2.1.4. Birleşik poisson talep sürecine eşdeğer poisson talep süreci

Friend [35], her bir müşterinin bireysel ürün talep miktarının sabit bir birim yerine kesikli bir rassal değişken olduğu durumlarda bile, talep sürecine poisson dağılımının oldukça iyi uyum gösterdiğini belirtmiş ve bu gibi durumları “Eşdeğer Poisson Dağılımı”na uydurmuştur. Friend [35]’de verilen dağılım uydurma tekniği, geçmiş verilerden t anına kadarki toplam talebin ortalamasını ve varyansını tahmin etmeyi ve rassal değişken olan bireysel ürün miktarının, sabit bir eşdeğer bireysel ürün talep miktarıyla değiştirilmesini içerir.

Bölüm 2.1.2 problem tanımında açıklandığı üzere çalışmada ele alınan problem için banknotların bireysel talepleri sabit bir miktar değil, müşterilerin bankamatikten istedikleri para talebinin dağılımına ve bankamatığın uyguladığı banknot dağıtım politikasına göre belirlenen $f'_{D_i}(\cdot)$ olasılık dağılımına sahip kesikli bir rassal değişkendir.

N_i , i’nci banknot çeşidi için birim zamanda gelen müşteri sayısı ve TD_i , i’nci banknot çeşidi için birim zamanda gelen toplam talep olarak tanımlanırsa, birim zamandaki toplam i’nci banknot çeşidinin talebi $TD_i = \sum_{j=1}^{N_i} D_{ij}$ olarak tanımlanır ve birleşik poisson dağılımına uyar. i’nci banknot çeşidi için toplam talep TD_i ’nin beklenen değeri Denklem 2.3’ten hesaplanabilir [36].

$$E[TD_i] = E[N_i].E[D_i] \quad (2.3)$$

i’nci banknot çeşidi için bankamatığe gelen müşterilerin beklenen değeri $E[N_i] = \lambda'_i$ olduğu için, i’nci banknot çeşidinin toplam talebi TD_i ’nin beklenen değeri Denklem 2.4 ve varyansı Denklem 2.5’ten hesaplanabilir [36].

$$E[TD_i] = \lambda'_i.E[D_i] \quad (2.4)$$

$$V[TD_i] = \lambda'_i \cdot E[D_i^2] \quad (2.5)$$

Friend [35]'in yaklaşımıyla, i 'nci banknot çeşidinin bireysel müşteri talebinin bir sabit olduğu, fakat toplam müşteri talebinin varyansının ve beklenen değerinin aynı kaldığı eşdeğer yeni bir dağılım ele alınsın. U_i , i 'nci ürün çeşidi için eşdeğer sabit bireysel müşteri talebi ve $\lambda'_{i\text{eşdeğer}}$, i 'nci ürün çeşidi için bireysel müşteri talebinin sabit U_i olduğu durumdaki eşdeğer ortalama müşteri sayısı olarak tanımlansın. Bu durumda eşdeğer süreç için toplam talebin beklenen değeri, Denklem 2.6'dan hesaplanır.

$$E[TD_i] = E[N_i] \cdot E[U_i] \quad (2.6)$$

$E[N_i] = \lambda'_{i\text{eşdeğer}}$, U_i ise bir sabit olduğundan $E[U_i] = U_i$ 'dir. Bu nedenle i 'nci ürün toplam talebinin beklenen değeri Denklem 2.7'den, varyansı ise Denklem 2.8'den hesaplanabilir.

$$E[TD_i] = \lambda'_{i\text{eşdeğer}} \cdot U_i \quad (2.7)$$

$$V[TD_i] = \lambda'_{i\text{eşdeğer}} \cdot E[U_i^2] \quad (2.8)$$

U_i , bir sabit olduğu için $E[U_i^2] = U_i^2$ 'dir. Bu durumda toplam talebin varyansı, Denklem 2.9'a dönüşür.

$$V[TD_i] = \lambda'_{i\text{eşdeğer}} \cdot U_i^2 \quad (2.9)$$

Eşdeğer süreçte toplam talebin varyansının ve beklenen değerinin değişmediğini varsaydığımız için;

$$E[TD_i] = \lambda'_i \cdot E[D_i] = \lambda'_{i\text{eşdeğer}} \cdot U_i \quad (2.10)$$

$$V[TD_i] = \lambda \cdot E[D_i^2] = \lambda'_{i\text{eşdeğer}} \cdot U_i^2 \quad (2.11)$$

bulunur. Denklem 2.12 ve 2.13'ten hareketle eşdeğer poisson sürecinin sabit müşteri talep miktarı U_i , Denklem 2.14'ten, ortalaması $\lambda'_{\text{eşdeğer}}$ ise 2.15'ten hesaplanabilir.

$$\lambda'_{\text{eşdeğer}} = \frac{E[TD_i]}{U_i} \quad (2.12)$$

$$V[TD_i] = \frac{E[TD_i]}{U_i} \cdot U_i^2 = E[TD_i] \cdot U_i \quad (2.13)$$

$$U_i = \frac{V[TD_i]}{E[TD_i]} = \frac{\lambda'_i \cdot E[D_i^2]}{\lambda'_i \cdot E[D_i]} = \frac{E[D_i^2]}{E[D_i]} \quad (2.14)$$

$$\lambda'_{\text{eşdeğer}} = \frac{E[TD_i]}{U_i} = \frac{\lambda'_i \cdot E[D_i]}{\frac{E[D_i^2]}{E[D_i]}} = \frac{\lambda \cdot E[D_i]^2}{E[D_i^2]} \quad (2.15)$$

Yeni süreçte i'nci banknot çeşidi için müşteriler bankamatiğe ortalaması $\lambda'_{\text{eşdeğer}}$ olan poisson sürecine göre gelmekte ve her bir müşteri sabit U_i adet i'nci banknottan talep etmektedir. Toplam talep sürecinin poisson sürecine uyması için i'nci banknot çeşidinin fiyatının v_i olması durumunda sabit U_i adet banknot, fiyatı $U_i \cdot v_i$ olan tek bir banknot olarak düşünülebilir. Böylece i'nci banknot çeşidine birim zamanda gelen talep, banknot tutarı $v'_i = U_i \cdot v_i$ ve talep ortalaması $\lambda'_{\text{eşdeğer}}$ olan poisson dağılımına uyar.

2.2. Ürün Kombinasyonu Dağıtım Politikasının Modellenmesi

Bu bölümde dağıtımcının stok sisteminde yer alan ürün çeşitleri için verilen bir eşgüdümlü stok politikası altında ürün kombinasyonu dağıtım problemi, yarı Markov karar süreci olarak modellenmiştir.

Modelde \mathbf{x} ile gösterilen sistem durumu (state), stok sistemindeki ürün (banknot) çeşitlerinin mevcut stok miktarları ve dağıtımçıya (bankamatige) gelen müşteri talebi olarak tanımlanmıştır. Bölüm 1.2 problem tanımında verildiği üzere x_i ($i=1,2,\dots,l$), i 'nci ürün (banknot) çeşidinin mevcut stok miktarını ve d , müşterilerin dağıtımçıdan (bankamatikten) istedikleri bireysel talep miktarı D rassal değişkenin aldığı değeri göstermek üzere, sistemin durumu $\mathbf{x}=(x_1, x_2, \dots, x_l, d)$ şeklinde tanımlanmıştır. Müşterilerin dağıtımçıdan istedikleri bireysel talep miktarını gösteren D rassal değişkeninin olasılık dağılımı $f_D(\cdot)$ 'nin, karar verici tarafından bilindiği varsayılmaktadır.

Verilen bir \mathbf{x} durumu için geçerli kararlar (actions) kümesi $A(\mathbf{x})$, müşteriye dağıtılabilecek tüm olası ürün (banknot) kombinasyonlarını içermektedir. Modelde müşteri talebi D 'yi karşılamak için karar verici tarafından i 'nci üründen dağıtılan ürün miktarı n_i ile tanımlanmıştır. Burada doğal olarak tüm ürün (banknot) çeşitleri ($i=1,2,\dots,l$) için $x_i \geq n_i$ ve $\sum_{i=1}^l n_i = D$ olmalıdır. Diğer bir deyişle dağıtılacak hiçbir ürün (banknot) çeşidinin mevcut stok miktarı x_i , dağıtılacak miktar n_i 'den küçük olamaz ve dağıtılan ürün (banknot) toplamı ($\sum_{i=1}^l n_i$) müşteri talebi D 'yi tam olarak karşılamalıdır. Eğer karar verici tarafından stoktan karşılanamayan bir karar seçilmişse, yani müşteriye dağıtılan ürün (banknot) miktarının toplamı, müşterinin dağıtımçıdan (bankamatikten) istediği talebi karşılamazsa, satış kaybı (lost sales) durumu oluşur.

Müşteri dağıtımçıya (bankamatige) gelip, (para) talebini bildirdiğinde süreç, Denklem 2.16'da verilen geçiş fonksiyonuna göre \mathbf{x} durumundan \mathbf{x}' durumuna geçiş yapar:

$$(x_i - n_i + Q_i)I_{ii} + (x_i - n_i)(1 - I_{ii}), \quad \forall i \text{ için } (i = 1, 2, \dots, l) \quad (2.16)$$

Burada Q_i , $i=1,2,\dots,l$ olmak üzere i 'nci ürünün (banknotun) stoklarının yenilenmesi için verilen sipariş miktarını göstermektedir. I_{li} , ise gösterge değişkenidir ve Denklem 2.17'de tanımlanmıştır:

$$I_{li} = \begin{cases} 0, & \text{Eğer } i\text{'nci ürün (banknot) stok yenileme siparişinde yer almazsa} \\ 1, & \text{aksi halde} \end{cases} \quad (2.17)$$

Ele alınan problemde verilen bir stok politikası altında bankamatik sisteminde gerçekleşen olaylar ve sırası aşağıda verilmiştir:

- 1) Müşteri dağıtımçıya (bankamatiğe) gelir.
 - 2) Müşteri dağıtımçıdan (bankamatikten) talep ettiği D bireysel ürün (para) miktarını bildirir.
 - 3) Dağıtımçı (bankamatik) stoklarındaki mevcut ürünlerin (banknotların) hangi kombinasyonu ile D miktarını karşılayacağına karar verir.
 - 4) Dağıtımçının (bankamatiğin) stoklarında müşteri talebini tam olarak karşılayabilecek ürün kombinasyonu mevcut değilse, talep kaybolur.
 - 5) Her bir ürün (banknot) çeşidinin stok pozisyonu (dağıtımçının elindeki stok miktarı+imalatçıya sipariş ettiği ancak henüz teslim almadığı miktar) güncellenir.
 - 6) Dağıtımçı (bankamatik), stoklarında yer alan her bir ürün (banknot) çeşidinin stok pozisyonuyla, ürünler (banknotlar) için uyguladığı stok politikasını karşılaştırır ve imalatçıdan (para dağıtım merkezi) stoklarını yenileyip yenilemeyeceğine, yenileyecekse hangi ürünlerin (banknotların) stoklarını yenileyeceğine karar verir.
- Çalışmada, müşterilerin dağıtımçı (bankamatik) tarafından sunulan ürün kombinasyonunu kabul ettiği varsayılmaktadır. Bununla birlikte, bir müşterinin talebi karşılanırken müşterinin beklediği ürün kombinasyonu ile, dağıtımçının

(bankamatik) sunduğu kombinasyon arasında farklılık arttıkça, müşterinin kazandığı fayda azalır. Çalışmada müşterilerin birbirileri ile özdeş ve kesikli bir fayda fonksiyonuna sahip oldukları kabul edilmektedir. $U(n_1, n_2, \dots, n_l)$, müşterilerin kesikli özdeş fayda fonksiyonunu gösterebilir. Müşterilerin fayda fonksiyonu, dağıtılan (n_1, n_2, \dots, n_l) miktarlarına bağlıdır.

U_{enb} , bir müşterinin dağıtımçıdan (bankamatikten) talep ettiği ürün (para) miktarının en çok arzu ettiği ürün (banknot) kombinasyonu $(n_1^*, n_2^*, \dots, n_l^*)$ ile karşılanması durumunda alacağı en büyük faydayı gösterir. Diğer bir gösterimle $U_{\text{enb}} = U(n_1^*, n_2^*, \dots, n_l^*)$ olsun. Dağıtımçıya (bankamatik) gelen müşterinin (para) talebinin karşılanmasında en çok arzu ettiği ürün (banknot) kombinasyonunun kendisine sağlanamaması durumunda U_{enb} ile $U(n_1, n_2, \dots, n_l)$ arasındaki fark, müşterinin (n_1, n_2, \dots, n_l) kombinasyonunu kabul ederek katlandığı fayda kaybını göstermektedir. U_a , karar vericinin a kararını (action) seçtiğinde ortaya çıkan ürün kombinasyonunun müşteriye faydası ile müşterinin en çok arzu ettiği ürün kombinasyonunun seçilmesinde elde edeceği fayda arasındaki farkı ifade etsin. Müşterinin dağıtılan ürün kombinasyonu ile uğrayacağı bu kayıp, kurulan modelde bir ceza maliyeti (p) ile cezalandırılmaktadır.

Amaç sonsuz ufukta birim zaman başına ortalama maliyetin en küçüklenmesidir. Sistem \mathbf{x} durumundayken a kararını almanın anlık maliyeti, Denklem 2.18'de verilmiştir:

$$g(\mathbf{x}, a) = \sum_{i=1}^l h_i t_i + (K + m.k).I_{1i} + p.U_a(n_1, n_2, \dots, n_l) + w.I_2 \quad (2.18)$$

Denklem 2.18'de h_i , 1 birim i 'nci ürünü (banknotu) birim zaman stokta bulundurma maliyetini, t_i ise i 'nci ürünün \mathbf{x} durumunda kalma süresini göstermektedir. Bu nedenle denklemdeki ilk terim, süreç \mathbf{x} durumundayken açığa çıkan stokta bulundurma maliyetini temsil eder. Denklemdeki ikinci terim, süreç \mathbf{x} durumundayken a kararı alındığında açığa çıkan sipariş maliyetini göstermektedir.

Burada m , siparişte yer alan ürün çeşidi sayısını göstermektedir. Denklemden yer alan üçüncü terim, müşteriye en çok istediği kombinasyonu dağıtamama maliyetidir. Denklemden son terim ise, bir müşterinin dağıtımcıdan (bankamatikten) (para) talebinin tam olarak karşılanamadığında oluşan kayıp satış maliyetidir. Denklemin son teriminde yer alan I_2 de bir gösterge değişkenidir ve Denklem 2.19'da verildiği şekilde tanımlanmıştır:

$$I_2 = \begin{cases} 0, & \text{müşterinin dağıtımcıdan talebi tam olarak karşılanırsa} \\ 1, & \text{aksi halde} \end{cases} \quad (2.19)$$

i 'nci ürün talebi için müşteriler ortalaması λ_i' olan poisson sürecine göre geldiği için t_i bir rassal değişkendir ve bu nedenle t_i 'nin dağılımı $(1/\lambda_i')$ parametrelili üstel dağılıma uyar [36]. Markov karar süreçlerinde (Markov decision process, MDP), zincirdeki herhangi bir geçişin süresi sabit bir birim olması gerekirken, yarı Markov karar süreçlerinde (semi Markov decision process, SMDP) bu süre genel dağılımlı bir rassal değişkendir. Eğer geçiş süreleri üstel dağılıma uyuyorsa, bu durumda süreç, yarı Markov karar süreçlerinin özel bir durumu olan sürekli zaman Markov süreçleri (continuous time Markov decision process, CTMDP) olarak adlandırılır [37]. Çalışmada durumların geçiş süreleri üstel dağılıma uyan rassal değişken olduğu için problem, sürekli zaman Markov karar süreci olarak modellenmiştir. Sürekli zaman Markov karar süreci modelinin sonsuz zaman ufkunda birim zamandaki ortalama maliyet amaç fonksiyonuyla ilişkili optimallik denklemi, Denklem 2.20'de verilmiştir [38].

$$v(\mathbf{x}) = \min_{a \in A(\mathbf{x})} \left[g(\mathbf{x}, a) - \rho^* y(\mathbf{x}, a) + \sum_{\mathbf{x}' \in \mathbf{X}} p(\mathbf{x}' | \mathbf{x}, a) v(\mathbf{x}') \right] \quad (2.20)$$

Optimallik denkleminde $v(\mathbf{x})$, \mathbf{x} durumunda bulunmanın değerini; $p(\mathbf{x}' | \mathbf{x}, a)$, \mathbf{x} durumundayken a kararı seçildiğinde \mathbf{x}' durumuna geçiş olasılığını; $y(\mathbf{x}, a)$, mevcut karar aşamasında \mathbf{x} durumundayken a kararı seçildiğinde sonraki karar aşamasına kadar geçen beklenen süreyi; ρ^* optimal politikanın ortalama maliyetini göstermektedir.

Sürekli zaman Markov karar süreci problemleri, birleştirme (uniformization) süreciyle tam olarak Markov karar süreci problemine dönüştürülüp, dinamik programlama (dynamic programming, DP) algoritmalarıyla optimal olarak çözüm bulunabilir. Ancak, dinamik programlama algoritmalarının kullanılabilmesi için geçiş olasılıklarının (transition probabilities), geçiş ödülllerinin (transition rewards) ve geçiş sürelerinin (transition times) hesaplanması gerekmektedir. Problemin ölçeği ve karmaşıklığı büyüdükçe bu değerlerin hesaplanması giderek zorlaşmakta ve hatta imkânsız hale gelmektedir. Ödüllü Öğrenme (reinforcement learning), Markov ve yarı Markov karar süreci problemlerinin çözümünde, tüm bu geçiş olasılıkları, geçiş ödülleri ve geçiş süreleri hesaplamalarına gerek kalmaksızın optimale yakın sonuç veren simülasyon tabanlı optimizasyon tekniğidir [37, 39]. Ele alınan problem büyük ölçekli ve karmaşık bir yapıya sahip olduğu için modelin çözümünde, Bölüm 3.2.1’de verilen Ödüllü Öğrenme teknikleri kullanılmıştır.

2.3. Eşgüdümlü Stok Yenileme Politikasının Belirlenmesi

Çok ürünlü stok sistemlerinde stok düzeylerinin belirlenmesi için kullanılan en basit yaklaşım, bireysel ürün taleplerinin birbirinden bağımsız olduğunu kabul ederek bir müşteri hizmet düzeyi kısıtı altında ürünlerin optimal stok politikalarının ve parametrelerinin belirlenmesidir. Ancak, bu tür stok sistemlerinde ürün çeşitlerinin stoklarının aynı merkezden yenilenmesi durumunda eşgüdümlü bir stok yenileme politikasının uygulanması oluşacak ölçek ekonomisi nedeniyle önemli maliyet avantajları kazandırabilir. Tezde ele alınan sistemde eşgüdümlü bir stok yenileme politikası kullanılması sonucunda açığa çıkan ölçek ekonomisi nedeniyle, ulaştırma maliyetlerinde (örneğin tam dolu kamyonlarla taşıma nedeniyle) önemli maliyet kazançları sağlanacaktır.

Ele alınan stok sistemiyle ilgili maliyetler aşağıda verilmiştir:

- h_i : i 'nci ürünün stokta bulundurma maliyeti TL/adet/ birim zaman
- K_i : i 'nci ürünün sabit birincil sipariş maliyeti (TL/sipariş)

- k_1 : i'nci ürünün ikincil sipariş maliyeti (TL/sipariş)
- p : müşterinin kaybettiği faydanın fırsat maliyeti (TL/birim fayda)
- w :müşterilerin talebinin stoktan karşılanamaması maliyeti (TL/talebi karşılanmayan müşteri sayısı)

Burada K ile gösterilen birincil sipariş maliyeti, ürün çeşitlerinden herhangi birinin stoklarının yenilenmesi nedeniyle açığa çıkan ve sipariş edilen ürün miktarından bağımsız, sabit sipariş maliyetidir. k ile gösterilen ikincil sipariş maliyeti ise hâlihazırda bir ürünün siparişine diğer ürünlerin eklenmesi nedeniyle açığa çıkan sipariş maliyetidir. İkincil sipariş maliyeti de yine birincil sipariş maliyeti K gibi verilen sipariş miktarlarından bağımsızdır, ancak siparişe eklenen ürün çeşidi başına toplam sipariş maliyetine eklenir.

Bölüm 1.2 problem tanımı bölümünde de verildiği üzere, ele alınan stok sisteminde stoklardan karşılanamayan talebin kaybolduğu varsayılmaktadır. w ile gösterilen maliyet, talebin stoklardan her karşılanamadığı durumda açığa çıkan satış kaybı (lost sale cost) maliyetidir.

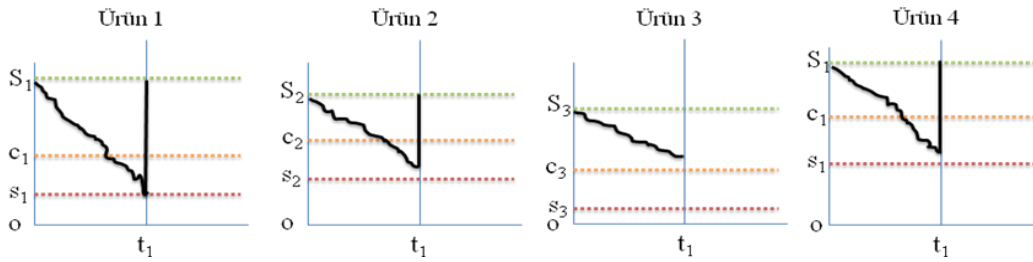
Tezde ele alınan stok sisteminde eşgüdümlü stok yenileme politikası olarak (S,c,s) “can order” politikasının kullanıldığı kabul edilmiştir. Devam eden bölümde (S,c,s) eşgüdümlü stok yenileme politikası ve bu politikanın kontrol parametrelerinin nasıl belirleneceği tartışılmıştır.

2.3.1. (S,c,s) “Can Order” eşgüdümlü stok politikası

(S,c,s) stok politikası stokların sürekli gözden geçirilmesini gerektirir, ancak diğer sürekli gözden geçirme politikalarından farklı olarak burada stok sistemindeki ürün çeşitleri birlikte ele alınmaktadır. Bu stok yenileme politikasında farklı ürün çeşitlerinin stok yenileme kararları birbirileri ile eşgüdümlü olarak belirlenir ve böylece oluşacak ölçek ekonomisiyle maliyetlerde azaltma sağlanır.

(S,c,s) stok politikasında 3 kontrol parametresi mevcuttur. Stok sisteminde ($i=1,2,\dots,1$) farklı ürün (banknot) çeşidi olmak üzere bu parametreler;

- 1) S_i , i'nci ürün (banknot) çeşidinin stoklarının yenileneceği üst seviye (up to level),
- 2) c_i , i'nci ürün (banknot) çeşidinin dışındaki herhangi bir ürün (banknot) çeşidi stoklarının yenilenmesi kararının, i'nci ürün (banknot) çeşidinin siparişini tetiklediği seviye (can order level),
- 3) s_i , diğer ürün (banknot) çeşitlerinden bağımsız olarak i'nci ürün (banknot) çeşidinin stoklarının yenilenmesini tetikleyen seviyedir (must order level).



Şekil 2.1. Dört ürünlü stok sisteminde (S,c,s) stok politikası

Şekil 2.1’de dört ürünlü bir (bankamatik) stok sisteminde, (S_i, c_i, s_i) stok politikası altında ürün (banknot) stoklarının nasıl yenildiği gösterilmiştir. (S_i, c_i, s_i) stok politikası altında ürün (banknot) stoklarının yenilenmesi iki şekilde tetiklenebilir:

- 1) i'nci ürün (banknot) çeşidinin stok pozisyonu “ s_i ” (must order level) seviyesine veya altına düşmüşse, diğer ürün (banknot) çeşitlerinin stoklarının yenilenip yenilmeyeceğine bakılmaksızın, i'nci ürün (banknot) çeşidi için sipariş verilir ve stok pozisyonu “ S_i ” (up to level) seviyesine yükseltilir. Şekil 2.1’de, t_1 anında “ürün 1”in stok pozisyonu s_1 (must order) seviyesine düştüğü için, diğer ürünlerin durumuna bakılmaksızın “ürün 1” için $S_1 - s_1$ kadar sipariş verilir ve “ürün 1”in stok pozisyonu S_1 ’e yükseltilir.
- 2) i'nci ürün (banknot) çeşidi dışındaki bir ürün (banknot) çeşidinin siparişi tetiklendiği anda, i'nci ürünün (banknotun) stok pozisyonu “ c_i ” (can order)

seviyesine veya altına düşmüşse, i 'nci ürün (banknot) çeşidinin siparişi tetiklenir ve stokları yenilenecek diğer ürün (banknot) çeşidiyle birlikte i 'nci ürün (banknot) çeşidinin stokları “ S_i ” (up to level) seviyesine kadar yenilenir. Şekil 2.1’de görüldüğü üzere t_1 anında ürün 2 ve ürün 4’ün stok pozisyonları sırasıyla c_2 ve c_4 seviyelerinin altında olduğu için, bu iki ürünün stok pozisyonları S_2 ve S_4 düzeylerine yükseltilmelidir. Böylece stok pozisyonunu S_2 'ye yükseltecek miktarda ürün 2 ve S_4 'e yükseltecek miktarda ürün 4 siparişleri, ürün 1'in siparişine eklenir.

Bu politikada her zaman için $c_i \geq s_i$ olarak alınır. Ters durumda mantıken c kontrol parametresinin bir anlamı kalmamaktadır.

Bölüm 3.2’de verildiği üzere yaklaşık en iyi ürün dağıtım politikasının belirlenmesi için simülasyon tabanlı bir optimizasyon tekniği olan Ödüllü Öğrenme algoritması kullanılmıştır. Bu tür teknikler, tekrarlı yinelemelere dayanan doğaları gereği hesaplama süresi performansı oldukça düşük olan tekniklerdir. Bu nedenle stok politikasının belirlenmesinde hesaplama süresi kısa olan algoritmaların kullanılması amaçlanmıştır.

(S,c,s) kontrol parametrelerinin belirlenmesi için Silver [1, 3], Thompstone ve Silver [2]’de yer alan algoritmalar simülasyon içermemekte ve kısa sürede sonuç vermektedir. Bu nedenle (bankamatik) stok sisteminde yer alan ürünlerin (S,c,s) politikalarının belirlenmesinde Silver [1, 3] ve Thompstone ve Silver [2]’da yer alan yöntemlerden yararlanılmıştır.

Silver [1,3] ve Thompstone ve Silver [2], çok ürünlü stok sistemindeki ürünlerin stok politikalarını merkezi kontrol olmaksızın ayırtmıştır. Diğer bir deyişle, stok sistemindeki her bir ürünün stok politikası, ilgili ürünün maliyetlerini en küçüklenecek şekilde belirlenmektedir. Burada i 'nci ürünün ölçek ekonomisi nedeniyle var olan bir siparişe eklenmesi fırsatının, μ_i oranlı poisson sürecine uyduğu varsayılmıştır. Özellikle stok sistemindeki ürün sayısı arttıkça, bu olayın gerçekleşmesi daha da rastsallaşacağı için varsayım daha mantıklı hale gelmektedir.

2.3.2. Stok pozisyonu dağılımının belirlenmesi

Stok kontrol politikalarında stokların yenilenip yenilenmeyeceğine karar vermek için genellikle stok düzeyinden (inventory level) ziyade, stok pozisyonu (inventory position) takip edilir. Stok pozisyonu, taleplerin geciktirilebildiği (back order) ve talebin kaybolduğu kayıp satış (lost sales) durumları için farklı şekillerde tanımlanmaktadır.

Stok pozisyonu talebin geciktirilebildiği (back order) durum için, “eldeki stok miktarı+siparişteki stok miktarı-geciktirilen talep miktarı” şeklinde tanımlanırken, kayıp satış (lost sales) durumu için “eldeki stok miktarı+siparişteki stok miktarı” olarak tanımlanır [45].

Örneğin talebin birleşik poisson süreci olduğu ve (S,c,s) stok politikasının uygulandığı bir stok sistemi ele alınsın ve sipariş edilen ürünlerin sabit bir L süresi sonunda sisteme ulaştığı kabul edilsin. Burada ürünlerin stok politikasının belirlenmesi için öncelikle stok düzeyi ya da stok pozisyonuna göre, “ c ” (can order level) ve “ s ” (must order level) parametrelerinin belirlenmesi gerekir. Eğer “ c ” ve “ s ” mevcut stok düzeyine göre belirlenirse, tedarik süresince gelen talebin çok fazla olması durumunda biriken talep çok fazla olacak ve tedarik süresi sonunda sipariş sisteme ulaşmasına karşın stok düzeyi “ s ” nin altında kalmış olabilecektir. Daha da dramatik bir şekilde olasılığı çok az da olsa tedarik süresince talep o kadar büyük olabilir ki, sipariş ulaştığında bile stok düzeyi hala negatifte kalabilir.

Stok pozisyonu kullanılarak “ s ” ve “ c ” parametrelerinin belirlenmesi durumunda ise, henüz ulaşmamış yoldaki siparişler de hesaba katıldığı için stok pozisyonu her zaman $[s+1,S]$ arasında yer alacaktır ve asla “ s ” nin altına düşüp, bozulmuş bir durum oluşturmayacaktır. Bu yüzden stok kontrol parametrelerinin belirlenmesinde stok pozisyonunun kullanılması mantıklıdır. Ancak, karşılanamayan talebin kaybolduğu (lost sales) stok sistemlerinde stok pozisyonu izlenemez ve bunun sonucu olarak bu tür sistemlerde stok pozisyonunun dağılımını belirlemek mümkün olmaz. Bu nedenle, karşılanamayan talebin kaybolduğu (lost sales) sistemlerde stok pozisyonu yerine stok düzeyi takip edilir, ancak bu durumda da yukarıda belirtilen bozulmuş

durumlar açığa çıkar. Özellikle stok sisteminden sağlaması istenen hizmet düzeyi arttıkça, karşılanamayan talebin geciktirildiği durumdaki stok pozisyonunun dağılımıyla, talebin kaybolduğu durumdaki stok pozisyonunun dağılımları birbirine yaklaşıacaktır. Bu nedenle çalışmada, karşılanamayan talebin kaybolduğu durumlar için, talebin geciktirilebildiği durumun stok pozisyonu dağılımı kullanılmıştır.

Stok sistemindeki i 'nci ürün için, (S_i, c_i, s_i) politikasının birleşik poisson talebi altında stok pozisyonu dağılımının belirlenmesinde kullanılan yöntem, Axater [33]'de verilen (S_i, s_i) politikasında stok pozisyonunun dağılımının belirlenmesinde kullanılan yöntemle az farklılık gösterir.

$m_p(j)$, sipariş çevrimi süresince stok pozisyonu I_p 'nin j 'ye ulaşma olasılığı olarak tanımlansın. Stok pozisyonunun alabileceği değerler i 'nci ürün için $[s_i+1, S_i]$ aralığında olduğu için, $(s_i + 1 \leq j \leq S_i)$ 'dir.

Stok pozisyonu I_p 'nin geçişleri bir yenilenme sürecidir. i 'nci ürünün siparişi her yenilendiğinde, yeni bir sipariş çevrimi başlar ve i 'nci ürünün stok pozisyonu S_i 'ye yükselir. Bu nedenle stok pozisyonunun her sipariş çevrimi başlangıcında S_i 'ye yükselmesi bir yenilenmedir. Her sipariş çevriminin başlangıcında stok pozisyonu S_i 'ye yükseleceği için, her sipariş çevriminde stok pozisyonunun S_i 'ye geçiş olasılığı, $m_p(S_i) = 1$ 'dir.

(S_i, c_i, s_i) politikasında i 'nci ürünün siparişini tetikleyen iki kontrol düzeyi vardır:

1) Stok pozisyonu düzeyi, " s_i " düzeyine ya da altına düştüyse kesinlikle yeni bir sipariş tetiklenir ve stok pozisyonu S_i 'ye yükseltilir.

2) Stok pozisyonu, " c_i " düzeyine ya da altına düştüyse;

a) stok sisteminde yer alan diğer ürünler için bir sipariş verilecekse, i 'nci ürün bu siparişe eklenir ve stok pozisyonu S_i 'ye yükseltilir.

b) eğer stok sistemindeki diğer herhangi bir ürün için sipariş verilmeyecekse, i'nci ürün için sipariş verilmez. Stok pozisyonu eldeki stok miktarı – müşteri talebi kadar azalır.

Silver [1]'in yaklaşımında i'nci ürünün diğer ürünlerin siparişine katılma fırsatı, μ_i parametrelili poisson dağılımına uyar. Diğer bir deyişle burada i'nci ürün için müşteri geliş süreci λ_i oranlı ve diğer ürünlerin siparişlerin tetiklenip var olan bir siparişe katılması fırsatı μ_i oranlı poisson sürecidir. Çalışmada Bölüm 2.1.2'de verildiği üzere, stok sistemindeki i'nci ürün çeşidinin müşteri geliş, λ_i' oranlı poisson sürecidir. Bu iki sürecin poisson doğası nedeniyle herhangi bir zamanda sonraki olayın müşteri geliş olma olasılığı, $\gamma_i = \frac{\lambda_i'}{\lambda_i' + \mu_i}$ 'dir. Diğer bir deyişle γ_i , i'nci ürünün stok pozisyonu $[c, s+1]$ aralığındayken diğer ürünler için bir siparişin tetiklenmemesi ve dolayısıyla i'nci ürünün var olan bir siparişe dâhil olmaması olasılığını ifade eder.

(S_i, c_i, s_i) politikasında i'nci ürünün stok pozisyonunun alabileceği değerler kümesi ikiye bölünebilir:

1) Stok pozisyonu, $[S_i, c_i+1]$ aralığında değerler alabilir. Bu durum (S_i, s_i) politikasında, $s_i=c_i$ için gerçekleşen klasik durumdur. Bu aralıkta, ilgilenilen ürünün siparişi tetiklenmez.

2) Stok pozisyonu $[c_i, s_i+1]$ aralığında değerler alabilir. $[c_i, s_i+1]$ aralığında diğer ürünlerin siparişi, i'nci ürünün siparişini $1 - \gamma_i$ olasılıkla tetikler.

Silver [3]'da verilen Denklem 2.21 ile yinelemeli olarak i'nci ürünün stok pozisyonunun geçiş olasılıklarını hesaplamak mümkündür.

$$m_{i_v}(S_i - j) = \begin{cases} \sum_{k=1}^j m_{i_v}(S_i - k + 1) f_{D_i}(j - k + 1) & , j = 1, 2, \dots, S_i - c_i \\ \sum_{k=1}^{S_i - c_i} m_{i_v}(S_i - k + 1) f_{D_i}(j - k + 1) + \gamma_i \sum_{k=S_i - c_i}^{j-1} m_{i_v}(S_i - k) f_{D_i}(j - k) & , j = S_i - c_i + 1, \dots, S_i - 1 \end{cases} \quad (2.21)$$

Stok pozisyonundaki her bir geçiş i'nci ürün için bir müşteri gelişiyle tetiklendiğinden, bir sipariş çevrimi süresince i'nci ürün için gelen ortalama toplam

müşteri sayısı, $\sum_{j=S_i+1}^{S_i} m_j$ dir. i'nci ürün için müşteri gelişi Bölüm 2.1.2'de verildiği

üzere λ_i' ortalamalı poisson sürecine uyduğu için, müşterilerin gelişleri arası süresi $1/\lambda_i'$ ortalamalı üstel dağılıma uyar [36]. Böylece stok pozisyonun olasılık fonksiyonu, değişmeyen durumdan (steady state) hareketle Denklem 2.22'den hesaplanabilir [33].

$$f_{I_p}(k) = \begin{cases} \frac{m_{I_p}(k)}{\sum_{j=S_i+1}^{S_i} m_{I_p}(j)} & , k = S_i + 1, S_i + 2, \dots, S_i \\ 0 & , \text{aksi halde} \end{cases} \quad (2.22)$$

2.3.3. Eksiklik pozisyonu dağılımının belirlenmesi

i'nci ürün için (S_i, c_i, s_i) politikasında tek sipariş tetikleyicisi, stok pozisyonunun “ s_i ” (must order) düzeyi ya da altına düşmesi değildir. Normalde stok pozisyonu “ s_i ” ya da altına düştüğünde sipariş verilecekken, siparişin tetiklenmesi stok pozisyonunun “ c_i ” (can order) seviyesi ya da altına düşmesi ve diğer ürünlerin stok pozisyonlarının kendi “ s ” seviyesi ya da altına düşmesiyle öne alınabilir. Buna bağlı olarak bir sipariş tetiklendiği andaki stok düzeyi $[s_i, c_i]$ arasında ya da bir ıskalama durumunda s_i düzeyinin altında bulunabilir. Burada ıskalama ile kastedilen durum, örneğin i'nci ürün için stok pozisyonu s_i+1 düzeyinde iken, gelen müşterinin 1'den fazla ürün i istediğinde ortaya çıkan durumdur. Bir çevrimde stoktan karşılanamayan ortalama stok miktarını belirlemek için, sipariş verildiği anda stok düzeyinin s_i 'den ne kadar uzakta olduğunun belirlenmesi gerekir. Stoktan karşılanamayan talep miktarının belirlenebilmesi için eksiklik (deficit) pozisyonu olarak isimlendirilen ve Z ile

gösterilen yeni bir kesikli rassal değişken tanımlansın. Bu durumda bu rassal değişken, Z_i : (s_i -“i’nci ürün için sipariş verildiği andaki stok pozisyonu”) olarak tanımlanır. i ’nci ürün için bir sipariş tam olarak stok pozisyonu c_i düzeyindeyken tetiklenirse, bu durumda $z_i = s_i - c_i$ olur. Stok pozisyonu s_{i+1} ’deyken, gelen müşteri $d_{i_{enb}}$ adet ürün i talep ettiğinde $z_i = s_i - (s_{i+1} - d_{i_{enb}}) = d_{i_{enb}} - 1$ olarak elde edilir. Bir ürünün stok pozisyonu, bir sipariş çevriminde sipariş verildikten hemen sonra S ’den başlar.

i ’nci ürün için siparişin stok pozisyonu $[s_{i+1}, c_i]$ aralığındayken tetiklendiği durumda eksiklik pozisyonu Z_i ’nin alabileceği değerler ikiye ayrılabilir:

1) $z = [-1, -2, \dots]$ durumu: i ’nci ürünün stok pozisyonu $[s_{i+1}, s_{i+2}, \dots, c_i]$ iken gelen talep $d_i \geq s_i + 2$ ise, bu durumda Z_i eksiklik pozisyonu kesikli rassal değişkeninin alabileceği değerler $[-1, -2, \dots, s_i - c_i]$ aralığında olur. Eksiklik pozisyonunun $Z_i = z_i$ olabilmesi için “diğer ürünlerin ilgili ürünün siparişini tetiklemesi” ve “stok pozisyonu geçişinin $s_i + z_i$ olması” olaylarının birlikte gerçekleşmesi gerekir. Eksiklik pozisyonunun $Z_i = s_i + z_i$ geçişi ve ilgili ürünün siparişinin diğer ürünlerin siparişi nedeniyle tetiklenmesi olayları birbirinden bağımsız olduğundan, bu durumda Z_i ’nin olasılık fonksiyonu Denklem 2.23’teki gibi hesaplanır.

$$f_{Z_i}(z_i) = m_{IP}(s_i - z_i) \cdot (1 - \gamma_i), \quad z_i = -1, -2, \dots, s_i - c_i \quad (2.23)$$

2) $z = [0, 1, 2, \dots, d_{i_{enb}} - 1]$ durumu: i ’nci ürünün stok düzeyi $[s_i, s_i - 1, s_i - 2, \dots, s_i - d_{i_{enb}} - 1]$ ’e geçtiği anda sipariş tetiklenirse, bu durumda Z_i eksiklik pozisyonu rassal değişkeninin alabileceği değerler $[0, 1, 2, \dots, d_{i_{enb}} - 1]$ olur. Bu durumda eksiklik pozisyonunun $Z_i = z_i$ değerini alabilmesi için stok pozisyonunun alabileceği değerler $[c_i + 1, c_i + 2, \dots, S_i]$ ve $[s_{i+1}, s_{i+2}, \dots, c_i]$ olarak ikiye ayrılabilir. Eksiklik pozisyonunun $Z_i = z_i$ değerini alması olayı, “stok pozisyonunun $[c_i + 1, c_i + 2, \dots, S_i]$ değerlerinden herhangi birine ulaşması ve stok pozisyonu bu değerlere ulaştığında gelen müşterinin talebinin $[c_i - s_i + z_i + 1, c_i - s_i + z_i + 2, \dots, S_i - s_i + z_i]$ olarak gerçekleşmesi” veya “stok pozisyonu $[s_{i+1}, s_{i+2}, \dots, c_i]$ değerlerinden herhangi birine ulaştığında ilgili ürünün siparişinin diğer ürünlerin siparişleri nedeniyle tetiklenmemesi ve gelen müşteri talebinin $[z_i + 1, z_i + 2, \dots, c_i - s_i + z_i]$ ” olmasıdır. i ’nci ürün için Z eksiklik pozisyonu

rassal deęişkeninin $[0,1,2,\dots,d_{i_{\text{enb}}}-1]$ deęerleri almasının olasılık fonksiyonu, Denklem 2.24'ten hesaplanabilir.

$$f_{z_i}(z_i) = \sum_{I=c_i+1}^{s_i} m_{\text{IP}}(I) \cdot f_{D_i}(I-s_i+z_i) + \gamma_i \cdot \sum_{I=s_i+1}^{c_i} m_{\text{IP}}(I) \cdot f_{D_i}(I-s_i+z_i), \quad z_i=0,1,2,\dots,d_{i_{\text{enb}}}-1 \quad (2.24)$$

Bu durumda eksiklik pozisyonu Z_i 'nin olasılık fonksiyonu, Denklem 2.25'teki gibi olur.

$$f_{z_i}(z_i) = \begin{cases} m_{\text{IP}}(s_i - z_i) \cdot (1 - \gamma_i) & , z_i = -1, -2, \dots, s_i - c_i \\ \sum_{I=c_i+1}^{s_i} m_{\text{IP}}(I) \cdot f_{D_i}(I - s_i + z_i) + \gamma_i \cdot \sum_{I=s_i+1}^{c_i} m_{\text{IP}}(I) \cdot f_{D_i}(I - s_i + z_i) & , z_i = 0, 1, 2, \dots, d_{i_{\text{enb}}} - 1 \\ 0 & , \text{aksi halde} \end{cases} \quad (2.25)$$

2.3.4. (S,c,s) kontrol parametrelerinin belirlenmesi

Silver [3] alıřmasında, stok sistemindeki her bir rnn talebinin λ_i ($i=1,2,\dots,l$) poisson daęılımına uyduęu varsayımı altında (S_i, c_i, s_i) parametrelerini hesaplayan algoritmalar geliřtirmiřtir. Blm 1.2.1'de verildięi zere, alıřmada ele alınan problemde stok sistemine mřteri geliři, λ ortalamalı poisson srecidir ve mřterilerin stok sisteminden bireysel talebi, bilindięi varsayılan $f_D(\cdot)$ kesikli bir daęılıma uyar. Dięer bir deyiřle stok sistemine birim zamanda gelen talep, λ ortalamalı birleřik poisson srecidir. Bu durumda i 'nci rnn talep sreci, Blm 1.2.2'de verildięi gibi mřteri geliři ortalaması λ_i' ve rn daęıtım politikasından belirlenen i 'nci rnn bireysel kesikli talep olasılık daęılımının $f_{D_i}(\cdot)$ olduęu birleřik poisson srecidir.

Blm 1.2.4'te ayrıntılı bir řekilde verildięi zere, Friend [35]'in birleřik poisson srecini poisson srecine dnřtrme yaklařımı ile i 'nci rnn λ_i' ortalamalı birleřik poisson sreci, λ_i' eřdeęer ortalamalı eřdeęer poisson srecine dnřtrlebilir. Bylece i 'nci rnn λ_i' eřdeęer ortalamalı poisson talebi iin Silver [3]'teki algoritmalar

kullanılarak, (S_i, c_i, s_i) parametreleri hesaplanabilir. Oluşan yeni eşdeğer talep sürecinde i 'nci ürün için gelen müşterilerin bireysel eşdeğer talep miktarı sabit U_i , i 'nci ürünün eşdeğer birim maliyeti ise $v_i' = U_i \cdot v_i$ olur.

Stok sistemindeki i 'nci ürün için Silver [1, 3] ve Thompstone ve Silver [2]'in çalışmalarından elde edilen çözüm prosedürü aşağıda verilmiştir:

1) Tedarik süresinin ihmal edilerek, $L=0$ için başlangıç (S_i, c_i) parametreleri bulunur. Bu aşamada tedarik süresi ihmal edildiği için elde güvelik stoğu bulundurulmayacak ve doğal olarak $s=0$ olacaktır.

2) Tedarik süresinin ihmal edildiği durum için hesaplanan $S_i(L=0)$, $c_i(L=0)$ ve $s_i=0(L=0)$ parametreleri için, verilen sabit tedarik süresince ($L>0$) gelen talebin dağılımı kullanılarak i 'nci ürün için stok sisteminin elde edilen hizmet düzeyi hesaplanır.

3) Adım 2'de hesaplanan hizmet düzeyi, istenen hizmet düzeyinden küçükse, Tablo 2.1'de verilen ilişkiyle istenen hizmet düzeyi sağlanana dek stok kontrol parametreleri birer birer arttırılır. İstenen hizmet düzeyi ya da fazlası elde edildiğinde (S_i, c_i, s_i) politikasının parametreleri belirlenmiş olur.

Tablo 2.1. ($L=0$) ve ($L>0$) durumlarında (S, c, s) kontrol parametreleri [3]

Parametreler	$L=0$ (Tedarik süresi sıfır)	$L>0$ (Tedarik süresi sabit)
S_i	$S_i(L=0)$	$S_i(L>0)=S_i(L=0)+s_i$
c_i	$c_i(L=0)$	$c_i(L>0)=c_i(L=0)+s_i$
s_i	0	s_i

İzleyen bölümde tedarik süresinin ihmal edildiği ($L=0$) durumu ve sabit tedarik süresi ($L>0$) durumları için (S, c, s) stok politikası parametrelerinin belirlenmesi anlatılmıştır.

2.3.4.1. Tedarik süresinin ihmal edildiği durumda (S,c,s) parametrelerinin belirlenmesi

Stok sistemindeki ürünlerin tedarik süresinin ihmal edilmesi ($L=0$) durumunda, i 'nci ürün için sipariş verildiği anda sipariş stok sistemine ulaşacağından " s_i " doğal olarak sıfır olacaktır. Ancak, i 'nci ürünün diğer ürünlerle birlikte sipariş edilmesinin ortaya çıkaracağı maliyet avantajı nedeniyle, i 'nci ürün için ($s_i=0$) seviyesinin üzerinde bir düzeyde sipariş verilmesi durumu ortaya çıkabilir. Diğer bir deyişle, tedarik süresinin ihmal edildiği ($L=0$) durumunda, i 'nci ürün için birim zamanda beklenen maliyeti minimize eden (S_i, c_i) parametreleri belirlenmek istenmektedir. Daha önce de ifade edildiği üzere, Silver [1,3] ve Thompstone ve Silver [2]'da yapılan kabule göre diğer ürünlerin siparişine eklendiğinde oluşacak maliyet avantajı nedeniyle, i 'nci ürünün sipariş yenileme fırsatı, μ_i oranlı poisson sürecidir.

Birim zamandaki beklenen toplam maliyeti hesaplamak için Silver [1, 46] 'da verilen maliyet yöntemi, Denklem 2.26'da verilmiştir.

$$EC_i = \{S_i - c_i + \gamma_i [1 - \gamma_i^{c_i} / (1 - \gamma_i)]\}^{-1}. \quad (2.26)$$

$$\{ S_i - c_i \quad S_i + c_i + 1 \quad v_i' h_i / 2 + \gamma_i [c_i - \gamma_i [1 - \gamma_i^{c_i} / (1 - \gamma_i)]] v_i' h_i / (1 - \gamma_i) + \lambda_{\text{eşdeğer}}' \gamma_i^{c_i} K + \lambda_{\text{eşdeğer}}' k \}$$

Beklenen toplam maliyet formülünde yer alan maliyetler aşağıda tanımlanmıştır:

- v_i' : i 'nci ürünün eşdeğer birim maliyeti (TL)
- h_i : i 'nci ürünün birim stok maliyeti (adet/TL/zaman)
- K : sabit birincil sipariş maliyeti (TL/sipariş)
- k : ikincil sipariş maliyeti (TL/siparişe eklenen ürün sayısı)
- $\lambda_{\text{eşdeğer}}'$: i 'nci ürünün eşdeğer ortalama talep oranı

- μ_i : i'nci ürünün var olan siparişe eklenme fırsatı oranı
- γ_i : i'nci ürünün diğer ürünlerin siparişine eklenmeme olasılığı

Burada sonraki işlemin talep olması, yani ürün i'nci ürünün var olan bir siparişe eklenmemesi olasılığı, Denklem 2.27'den hesaplanır [1].

$$\gamma_i = \frac{\lambda_i}{\lambda_i + \mu_i} \quad (2.27)$$

Maliyet formülünde geçen γ_i 'nin hesaplanabilmesi için önce, μ_i sipariş yenileme fırsatı oranının hesaplanması gereklidir.

NT_i , birim zaman süresince i'nci ürünün, j'nci ürünü $i \neq j$ tetiklediği ortalama yenileme sayısı olarak tanımlansın. Bu durumda j ürününün sipariş yenileme fırsat oranı $\mu_j = \sum_{\forall i \neq j} NT_i$ ile tahmin edilebilir. NT_i ise, Denklem 2.28' den hesaplanır [1].

$$NT_i = \lambda'_{i\text{ eşdeğer}} \gamma_i^{c_i} / \left[S_i - c_i + \gamma_i \frac{1 - \gamma_i^{c_i}}{1 - \gamma_i} \right] \quad (2.28)$$

Beklenen toplam maliyet formülünde minimum noktanın bulunabilmesi için, toplam maliyet fonksiyonunda ikinci türevinin pozitif olması şartıyla, birinci türev sıfıra eşitlenir. Toplam maliyet formülünün S_i 'ye göre kısmi türevi sıfıra eşitlendiğinde, verilen "c_i" değerine göre en iyi S_i değerini veren Denklem 2.29 elde edilir [1]:

$$\hat{S}_i(c_i) = c_i - \gamma_i(1 - \gamma_i^{c_i}) / (1 - \gamma_i) \quad (2.29)$$

$$+ \sqrt{\frac{2\lambda'_{i\text{ eşdeğer}} (k + K\gamma_i^{c_i}) / v_i' h_i + 2c_i \gamma_i^{c_i+1} / (1 - \gamma_i) - \gamma_i(1 - \gamma_i^{c_i})(1 + \gamma_i^{c_i+1}) / (1 - \gamma_i)^2}{}}$$

Böyle bir durumda $\hat{S}_i(c_i) = c_i$ olarak alınır, verilen bir c_i değeri için minimum maliyetin elde edildiği gösterilebilir [1].

Buna göre ilk adımda i 'nci ürün için S_i , Denklem 2.30'da verilen Ekonomik Sipariş Miktarından (ESM) ve $c_i=0$ olarak belirlenir.

$$ESM_i = \sqrt{\frac{2(K+k)\lambda_i'_{\text{esdeğer}}}{v_i' h_i}} \quad (2.30)$$

$c_i=0$ için en uygun $\widehat{S}_i(0)$ miktarı, Denklem 2.29'dan bulunur ve $\widehat{S}_i(c_i) = c_i$ olarak alınıp, Denklem 2.26'dan maliyeti hesaplanır. Her yinelemede NT_i 'ler ve μ_i 'ler değişecektir, bu nedenle Denklem 2.29'dan hesaplanan $\widehat{S}_i(c_i) = c_i$ de değişir. Maliyetlerin artık azalmadığı yinelemede, en düşük maliyetli (S_i, c_i) parametre değerleri ve μ_i sipariş yenileme oranları elde edilmiş olur.

i 'nci ürün için Silver [1]'da verilen algoritma aşağıda detaylandırılmıştır:

Adım 1: i 'nci ürünün ilk yinelemedeki "c" (can order) düzeyi, $c_i^1 = 0$, i 'nci ürün için ilk yinelemedeki sipariş fırsatı oranı $\mu_i^1 = 0$ olarak atanır. İlk adımda her ürün birbirinden bağımsız ele alınır. Böylece $\gamma_i^1 = \frac{\lambda_i}{\lambda_i + \mu_i} = 1$ hesaplanır. İlk yinelemedeki S_i^1 düzeyi, Denklem 2.30 ekonomik sipariş miktarı denkleminde (ESM) hesaplanır. Yineleme sayısı $m=2$ olarak belirlenir.

Adım 2: i 'nci ürün için $\mu_i^m = \sum_{\forall j \neq i} NT_j$ hesaplanır. İlk adımda $\mu_i = 0$ olarak alınıp Denklem 2.28'den NT_i hesaplanır. Denklem 2.29'dan $\widehat{S}_i(0)$ hesaplanır ve $\widehat{S}_i(c_i) = c_i$ alınarak Denklem 2.26'dan ortalama maliyet bulunur.

Adım 3: i 'nci ürün için önceki adımda hesaplanan μ_i kullanılır ve $\mu_i^{m+1} = \sum_{\forall j \neq i} NT_j$ hesaplanır. Önceki adımda bulunan c_i kullanılarak denklem 2.29'dan $\widehat{S}_i(c_i)$ hesaplanır ve $\widehat{S}_i(c_i) = c_i$ alınarak, Denklem 2.26'dan ortalama maliyet hesaplanır. Yeni hesaplanan ortalama maliyet, önceki adımda hesaplanan ortalama maliyetten

küçük değilse algoritma sonlandırılır. En düşük maliyete sahip S_i ve c_i parametre değeri bulunmuştur. Eğer daha düşük bir maliyet hesaplandıysa, adım 3 tekrar edilir.

2.3.4.2. Sabit tedarik süresi için (S,c,s) parametrelerinin belirlenmesi

$L > 0$ sabit tedarik süresi durumunda, (S,c,s) parametrelerinin belirlenmesi için tedarik süresinin ihmal edildiği ($L=0$) durumunda hesaplanan $S(L=0)$, $c(L=0)$ ve $s=0(L=0)$ parametrelerinden hareket edilir. Bölüm 2.3.4.1’de belirlenen $S(L=0)$, $c(L=0)$ ve $s=0(L=0)$ politikasının mevcut hizmet düzeyi hesaplanır ve Tablo 2.1’deki ilişki kullanılarak istenen hizmet düzeyi sağlanana dek (S,c,s) parametreleri birer birim artırılır [2, 3].

Stok sistemlerinde en uygun politikanın belirlenmesinde iki hesaplama yöntemi söz konusudur. Bu yöntemlerden ilki stoksuz kalma maliyetinin de dikkate alınarak en düşük maliyeti veren politikanın hesaplanmasıdır. Ancak, müşteri talebinin karşılanamamasını maliyetlendirmek oldukça subjektiftir. Bu nedenle genelde, stok politikalarının belirlenmesinde diğer bir yöntem olan hizmet düzeyi kısıtı kullanılır. Stok literatüründe 3 tip hizmet düzeyi söz konudur:

- 1) Birinci tip hizmet düzeyi (type I service level)
- 2) İkinci tip hizmet düzeyi (type II service level)
- 3) Üçüncü tip hizmet düzeyi (type III service level)

Birinci tip hizmet düzeyi, bir sipariş çevriminde stoksuz kalmama olasılığı olarak tanımlanır. Örneğin birinci tip hizmet düzeyinin 0,95 olması, sipariş çevrimlerinin % 95’inde stoksuz kalma durumuyla karşılaşılmayacağını ifade eder [45, 47].

Karşılama oranı (fill rate) olarak da isimlendirilen ikinci tip hizmet düzeyi, gelen müşteri talebinin ne kadarının karşılandığını gösteren bir orandır. Örneğin ikinci tip hizmet düzeyinin 0,95 olması, gelen müşteri talebinin % 95’inin stoklardan karşılandığını gösterir [45, 47].

Hazırlık oranı (ready rate) olarak isimlendirilen üçüncü tip hizmet düzeyi ise, zamanın ne kadarında müşteri talebinin stoklardan karşılandığını gösteren bir orandır. Müşteri talebinin poisson dağılımına uyması halinde hazırlık oranıyla, karşılama oranı birbirine denktir [45].

Çalışmanın ana motivasyonu olan bankamatik probleminde, müşterilerin talebinin ne kadarının karşılandığının belirlenmesi önemli bir performans göstergesidir. Bu nedenle çalışmada, temel performans ölçütü olarak ikinci tip hizmet düzeyi kullanılmıştır.

Sabit tedarik süresi için (S,c,s) politikasının hizmet düzeyinin hesaplanması: Verilen bir (S,c,s) politikasının ikinci tip hizmet düzeyini belirleyebilmek için ilgili değişkenler aşağıda verilmiştir.

- β_{2_i} : i'nci ürünün ikinci tip hizmet düzeyi
- TD_{in_i} : i'nci ürün için bir sipariş çevriminde stoktan karşılanan toplam talep
- TD_{out_i} : i'nci ürün için bir sipariş çevriminde stoktan karşılanamayan toplam talep
- TD_i : i'nci ürün için bir sipariş çevrimindeki toplam talep

Bu durumda ikinci tip hizmet düzeyi Denklem 2.31'den hesaplanabilir [48].

$$\beta_{2_i} = \frac{E[TD_{in_i}]}{E[TD_i]} \quad (2.31)$$

Verilen bir “s_i” için bir çevrimde ortalama stoktan karşılanamayan miktar, eksiklik pozisyonu Z_i'nin dağılımından faydalanarak hesaplanabilir. Bölüm 2.3.3'te Z'nin olasılık fonksiyonu verilmiştir (Bkz. Denklem 2.25).

Sabit bir tedarik süresi ($L>0$) söz konusu olduğunda Tablo 2.1’de verilen ilişki nedeniyle i ’nci ürün için $c_i(L>0)=c_i(L=0)+s_i$ ’dir. Bölüm 2.3.4.1’de belirlenen ihmal edilmiş tedarik süresi ($L=0$) için hesaplanan $(S_i,c_i,0)$ politikasından hareketle, sabit bir tedarik süresi ($L>0$) için (S_i,c_i,s_i) politikası belirlenir. $c_i(L>0)=c_i(L=0)+s_i$ ilişkisi kullanıldığında, eksiklik pozisyonu Z_i ’nin yeni tanım aralığı $z_i = -c_i(L=0), -c_i(L=0)+1, \dots, -2, -1, 0, 1, \dots, d_{i_{enb}} - 1$ olarak belirlenir. Bu durumda eksiklik pozisyonu Z_i ’nin olasılık fonksiyonu, $s_i=0$ için Silver [3]’da verildiği üzere Denklem 2.32’den hesaplanır.

$$f_{Z_i}(z_i) = \begin{cases} m_{ip}(-z_i).(1-\gamma_i) & , z_i = -c_i(L=0), -c_i(L=0)+1, \dots, -2, -1 \\ \sum_{l=c_i+1}^{s_i} m_{ip}(l).f_{D_i}(l+z_i) + \gamma_i \cdot \sum_{l=1}^{c_i} m_{ip}(l).f_{D_i}(l+z_i) & , z_i = 0, 1, 2, \dots, d_{i_{enb}} - 1 \\ 0 & , \text{ aksi halde} \end{cases} \quad (2.32)$$

Bir çevrimde ortalama stoktan karşılanamayan ürün miktarının ne olduğu, “ s_i ” düzeyi ile yakından ilişkilidir. Silver [1] algoritmasına göre önce ihmal edilmiş tedarik süresi ($L=0$) için belirlenen $(S,c,0)$ yani $s=0$ için hizmet düzeyi ölçülür ve sonrasında istenen hizmet düzeyi sağlanana dek $(S+1,c+1,1)$, $(S+2,c+2,2)$,... politikalarının hizmet düzeyleri hesaplanır.

Bir sipariş çevriminde ortalama stoktan karşılanan toplam talep miktarının $E[TD_{in}]$ hesaplanma yöntemleri, stoktan karşılanamayan taleplerin geciktirildiği (back order) ve stoktan karşılanamayan talebin kaybolduğu (lost sales) durumlarda farklıdır.

Stoktan karşılanamayan talebin geciktirilmesi durumu (back order) için ikinci tip hizmet düzeyinin hesaplanması: Talebin geciktirilebildiği durumda bir çevrimde stoktan karşılanamayan talepler eninde sonunda diğer çevrimlerdeki siparişler ile karşılanır. Bu nedenle SM_i , i ’nci ürün için sipariş miktarını göstermek üzere bir çevrimdeki ortalama talep, bir çevrimdeki ortalama sipariş miktarı $E[SM_i]$ ’ye eşittir. Stoktan karşılanamayan talebin geciktirildiği durumda stoklardaki i ’nci ürün için sipariş verildiği anda, eldeki stok düzeyinin sıfırdan büyük (pozitif) ya da küçük (negatif) olmasına bakmaksızın $(S_i - s_i + Z_i)$ kadar sipariş verilir. Bu nedenle stoksuz

kalma durumunda talebin geciktirildiği durum için bir sipariş çevrimindeki ortalama sipariş miktarı, Denklem 2.33'teki gibi hesaplanır.

$$\begin{aligned}
E[TD_i] = E[SM_i] &= \sum_{z_i=-c_i}^{d_{i\text{enb}}-1} E[SM_i | Z_i = z_i] \\
&= \sum_{z_i=-c_i}^{d_{i\text{enb}}-1} (S_i - s_i + z_i) \cdot f_{Z_i}(z_i) \\
&= \sum_{z_i=-c_i}^{d_{i\text{enb}}-1} (S_i - s_i) \cdot f_{Z_i}(z_i) + \sum_{z_i=-c_i}^{d_{i\text{enb}}-1} (z_i) \cdot f_{Z_i}(z_i) \\
&= E[S_i - s_i] + E[Z_i] \\
&= S_i - s_i + E[Z_i]
\end{aligned} \tag{2.33}$$

i'nci ürün için verilen bir “s_i” düzeyi durumunda bir çevrimde ortalama stoktan karşılanamayan ürün miktarlarını hesaplamak için, Z_i'nin olasılık fonksiyonu kullanılır (Bkz. denklem 2.25). Bu durumda i'nci ürünün bir sipariş çevrimindeki stoktan karşılanamayan talep miktarı, Denklem 2.34, 2.35 ve 2.36'dan hesaplanır.

$$\begin{aligned}
E[TD_{\text{out}_i} | s_i, Z_i = z_i] &= \sum_{z_i=-1}^{-c_i} E[TD_{\text{out}_i} | s_i, Z_i] , \quad z_i = -1, -2, \dots, s_i - c_i \\
&= \sum_{z_i=-1}^{-c_i} E[TD_{\text{out}_i} | s_i, Z_i = z_i] \cdot f_{Z_i}(z_i) \\
&= \sum_{z_i=-1}^{-c_i} \left(\sum_{n=0}^{\infty} n \cdot f_{TD_{\text{out}_i}}(n) \right) \cdot f_{Z_i}(z_i) \\
&= \sum_{z_i=-1}^{-c_i} \left(\sum_{n=s_i-z_i+1}^{\infty} (n - s_i + z_i) \cdot f_{X(L)}(n) \right) \cdot (m_{\text{IP}}(-z_i) \cdot (1 - \gamma_i))
\end{aligned} \tag{2.34}$$

$$\begin{aligned}
E[TD_{out_i} | s_i, Z_i = z_i] &= \sum_{z_i=0}^{d_{i_{mb}}-1} E[TD_{out_i} | s_i, Z_i] , & z_i = 0, 1, 2, \dots, d_{i_{mb}} - 1 & \quad (2.35) \\
&= \sum_{z_i=0}^{d_{i_{mb}}-1} E[TD_{out_i} | s_i, Z_i = z_i] \cdot f_{Z_i}(z_i) \\
&= \sum_{z_i=0}^{d_{i_{mb}}-1} \left(\sum_{n=0}^{\infty} n \cdot f_{TD_{out_i}}(n) \right) \cdot f_{Z_i}(z_i) \\
&= \sum_{z_i=0}^{d_{i_{mb}}-1} \left(\sum_{n=s_i-z_i+1}^{\infty} (n - s_i + z_i) \cdot f_{X(L)}(n) \right) \\
&\quad \left(\sum_{I=c_i+1}^{s_i} m_{IP}(I) \cdot f_{D_i}(I + z_i) + \gamma_i \cdot \sum_{I=1}^{c_i} m_{IP}(I) \cdot f_{D_i}(I + z_i) \right)
\end{aligned}$$

$$E[TD_{out_i} | s_i] = E[TD_{out_i} | s_i, Z_i < 0] + E[TD_{out_i} | s_i, Z_i \geq 0] , z_i = s_i - c_i, s_i - c_i + 1, \dots, -1, 0, 1, \dots, d_{i_{mb}} - 1 \quad (2.36)$$

İkinci tip hizmet düzeyi, gelen müşteri talebinin ne kadarının stoklardan karşılandığını gösterdiğine göre, i 'nci ürün için verilen s_i düzeyine göre ikinci tip hizmet düzeyi Denklem 2.37'den hesaplanabilir.

$$\beta_{2_i} = 1 - \frac{E[TD_{out_i} | s_i]}{E[TD_i]} = 1 - \frac{E[TD_{out_i} | s_i]}{S_i - s_i + E[Z_i]} \quad (2.37)$$

Denklemlerde geçen $m_{IP}(\cdot)$, Bölüm 2.3.2'de verildiği üzere stok pozisyonunun geçiş olasılı fonksiyonu, $f_{X(L)}(\cdot)$ ise Bölüm 2.1.3'te verildiği üzere tedarik süresince gelen talebin olasılık fonksiyonudur.

Stoktan karşılanamayan talebin kaybolması (lost sales) durumu için ikinci tip hizmet düzeyinin hesaplanması: Sipariş çevrimlerinde stoktan karşılanamayan talebin kaybolduğu (lost sales) durumda, taleplerin geciktirilebildiği durumdan farklı olarak bir çevrimdeki ortalama talep miktarı, ortalama sipariş miktarına eşit değildir. Çünkü bir çevrimde karşılanamayan talep sonraki çevrimlere yansıtılmamakta ve

kaybolmaktadır. Eğer i 'nci ürün için sipariş verildiği anda stok düzeyi pozitifse, $(S_i - s_i + Z_i)$ miktarı kadar ürün sipariş edilir. Oysa çevrimlerde stoktan karşılanmayan talebin kaybolduğu durumda i 'nci ürün için sipariş verildiği anda stok düzeyi sıfırdan büyük değilse, S_i kadar sipariş verilir. Denklem 2.38'de verildiği gibi bir çevrimdeki ortalama talep miktarı, bir çevrimde stoktan karşılanamayan ortalama talep miktarı ve bir çevrimde stoktan karşılanan ortalama talep miktarının toplamı olarak hesaplanabilir [48].

$$E[TD_i] = E[TD_{in_i}] + E[TD_{out_i}] \quad (2.38)$$

Bu durumda stoktan karşılanamayan talebin kaybolduğu durumda i 'nci ürün için ikinci tip hizmet düzeyi, Denklem 2.39'dan hesaplanır.

$$\beta_{2_i} = \frac{E[TD_{in_i}]}{E[TD_i]} = \frac{E[TD_{in_i}]}{E[TD_{in_i}] + E[TD_{out_i}]} = 1 - \frac{E[TD_{out_i}]}{E[TD_{in_i}] + E[TD_{out_i}]} \quad (2.39)$$

Bir çevrimdeki ortalama stoktan karşılanamayan ürün miktarının hesaplanması, talebin geciktirildiği durumdaki hesaplamalarla aynıdır (Bkz. Denklem 2.34, 2.35 ve 2.36). Daha önce de ifade edildiği gibi farklı olan durum, bir çevrimdeki ortalama talebin hesaplanmasındadır. i 'nci ürün için bir çevrimdeki ortalama stoktan karşılanamayan talep miktarı, Denklem 2.40, 2.41 ve 2.42'den hesaplanabilir.

$$\begin{aligned} E[TD_{in_i} | s_i, Z_i < 0] &= \sum_{z_i=-1}^{-c_i} E[TD_{in_i} | s_i, Z_i] , & z_i = -1, -2, \dots, s_i - c_i & \quad (2.40) \\ &= \sum_{z_i=-1}^{-c_i} E[TD_{in_i} | s_i, Z_i = z_i] \cdot f_{Z_i}(z_i) \\ &= \sum_{z_i=-1}^{-c_i} \left(\sum_{n=0}^{\infty} n \cdot f_{TD_{in_i}}(n) \right) \cdot f_{Z_i}(z_i) \\ &= \sum_{z_i=-1}^{-c_i} \left((S_i + z_i) + \sum_{n=-z_i+s_i}^0 (n) \cdot f_{X(L)}(n) \right) \cdot (m_{IP}(-z_i) \cdot (1 - \gamma_i)) \end{aligned}$$

$$\begin{aligned}
E[TD_{in_i} | s_i, Z_i \geq 0] &= \sum_{z_i=0}^{d_{i_{enb}}-1} E[TD_{in_i} | s_i, Z_i = z_i], \quad z_i = 0, 1, 2, \dots, d_{i_{enb}} - 1 \quad (2.41) \\
&= \sum_{z_i=0}^{d_{i_{enb}}-1} E[TD_{in_i} | s_i, Z_i = z_i] \cdot f_{Z_i}(z_i) \\
&= \sum_{z_i=0}^{d_{i_{enb}}-1} \left(\sum_{n=0}^{\infty} n \cdot f_{TD_{in_i}}(n) \right) \cdot f_{Z_i}(z_i) \\
&= \left(\sum_{z_i=0}^{d_{i_{enb}}-1} \left((S_i - s_i + z_i) + \sum_{n=s_i-z_i}^0 (n) \cdot f_{X(L)}(n) \right) + S_i \cdot (d_{i_{enb}} - s_i - 2) \right) \\
&\quad \left(\sum_{I=c_i+1}^{s_i} m_w(I) \cdot f_{D_i}(I + z_i) + \gamma_i \cdot \sum_{I=1}^{c_i} m_w(I) \cdot f_{D_i}(I + z_i) \right)
\end{aligned}$$

$$E[TD_{in_i} | s_i] = E[TD_{in_i} | s_i, Z_i < 0] + E[TD_{in_i} | s_i, Z_i \geq 0], \quad z_i = s_i - c_i, s_i - c_i + 1, \dots, -1, 0, 1, \dots, d_{i_{enb}} - 1 \quad (2.42)$$

Stoktan karşılanamayan talebin kaybolduğu durumda i'nci ürün için verilen bir "s" düzeyi altında, bir çevrimdeki ortalama talep miktarı Denklem 2.43'ten hesaplanabilir.

$$E[TD_i | s_i] = E[TD_{in_i} | s_i] + E[TD_{out_i} | s_i] \quad (2.43)$$

İkinci tip hizmet düzeyi, müşteri talebinin ne kadarının stoklardan karşılandığını gösterdiği için, stoktan karşılanamayan talebin kaybolduğu durumda Denklem 2.44'ten hesaplanır.

$$\beta_{2_i} = 1 - \frac{E[TD_{out_i} | s_i]}{E[TD_i]} = 1 - \frac{E[TD_{out_i} | s_i]}{E[TD_{in_i} | s_i] + E[TD_{out_i} | s_i]} \quad (2.44)$$

Böylece, öncelikle tedarik süresinin ihmal edildiği (L=0) durumunda i'nci ürün için (S_i, c_i, 0) politikasının hizmet düzeyi bu bölümde verilen denklemlerden hesaplanır. Eğer mevcut politika istenen hizmet düzeyini sağlamıyorsa, istenen hizmet düzeyi sağlanana dek kontrol parametrelerinin değerleri birer birim arttırılır.

BÖLÜM 3. ALGORİTMA VE ÖDÜLLÜ ÖĞRENME

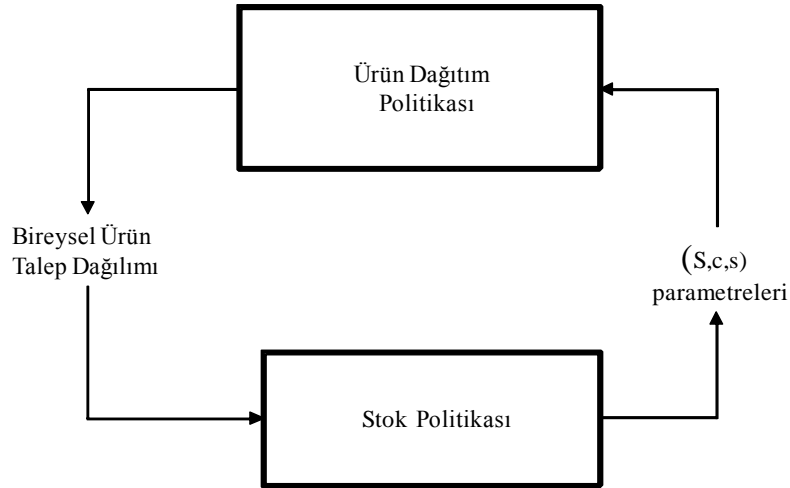
Bölüm 3.1’de, ürün kombinasyonu dağıtım ve stok politikalarının belirlenmesi için bir algoritma geliştirilmiştir. Bölüm 3.2’de ise, yarı Markov karar süreci olarak modellenen ürün kombinasyonu dağıtım probleminin fonksiyon yaklaşımı ödüllü öğrenme algoritmalarıyla çözümü yer almaktadır.

3.1. Ürün Kombinasyonu Dağıtım ve Stok Yenileme Algoritması

Bankamatik stok sisteminde yer alan banknot çeşitlerinin talep süreçleri bilindiğinde, istenen ikinci tip hizmet düzeyini sağlayacak şekilde banknot çeşitlerinin (S,c,s) stok yenileme politikası parametrelerinin nasıl belirlenebileceği Bölüm 2.3’te verilmiştir. Bankamatikteki banknot çeşitlerinin bireysel müşteri taleplerinin dağılımı;

- bankamatiğe gelen müşterilerin bireysel para talebi dağılımına,
- bankamatiğin uyguladığı banknot kombinasyonu dağıtım politikasına bağlıdır.

Tezde ele alınan problemde bankamatiğe gelen müşterilerin bireysel para taleplerinin dağılımı bilindiği varsayıldığından, banknot çeşitlerinin (S,c,s) stok yenileme parametrelerinin belirlenebilmesi için bankamatiğin uyguladığı banknot kombinasyonu dağıtım politikasının bilinmesi yeterlidir. Bölüm 2.2’de yarı Markov karar süreci olarak modellenen banknot kombinasyonu dağıtım politikasının belirlenmesi için ise, banknot çeşitleri için uygulanan (S,c,s) stok yenileme politikasının bilinmesi gerekmektedir.



Şekil 3.1. Ürün kombinasyonu dağıtım ve stok politikalarının etkileşimi

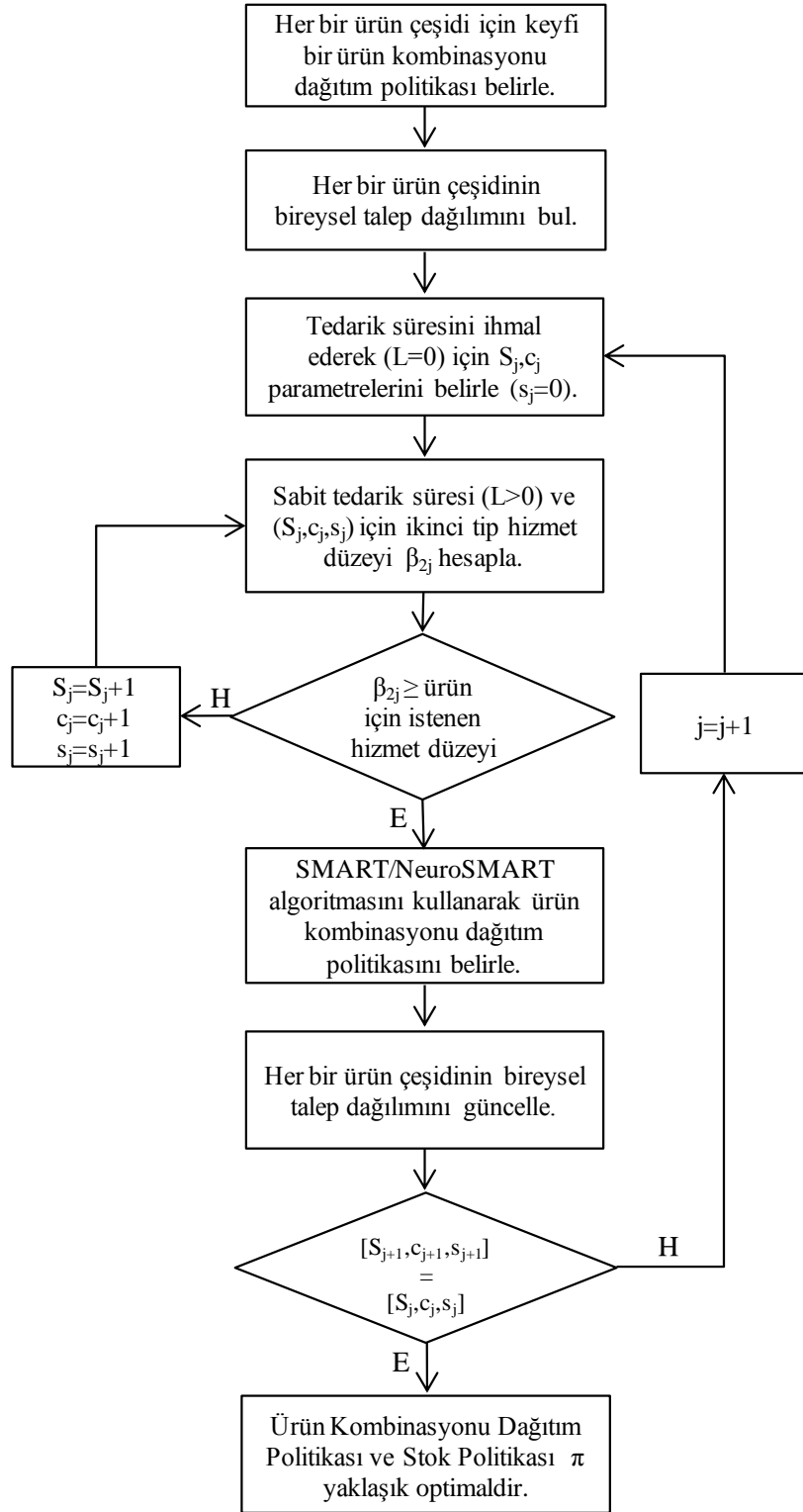
Sonuç olarak Şekil 3.1’de gösterildiği gibi, bankmatik stok sisteminde yer alan banknot çeşitlerinin (S,c,s) stok yenileme politikalarının belirlenebilmesi için bankmatığın banknot kombinasyonu dağıtım politikasının bilinmesi; banknot kombinasyonu dağıtım politikasının belirlenebilmesi için ise banknot çeşitlerinin (S,c,s) stok yenileme politikalarının bilinmesi gerekmektedir. Bu nedenle yaklaşık optimal banknot kombinasyonu dağıtım ve stok politikalarının belirlenmesi için sıralı ve yinelemeli bir algoritma geliştirilmiştir. Geliştirilen algoritmanın akış şeması ve adımları Bölüm 3.1.1’de yer almaktadır.

3.1.1. Algoritmanın adımları

Bu bölümde banknot kombinasyonu dağıtım ve banknot çeşitlerinin (S,c,s) stok yenileme politikalarını optimale yakın bir şekilde belirlemek için bir algoritma geliştirilmiştir. Geliştirilen algoritmada başlangıçta seçilen keyfi bir banknot kombinasyonu dağıtım politikasından hareketle banknotların (S,c,s) stok yenileme ve bankmatığın banknot kombinasyonu dağıtım politikaları, politikalar değişmeyene kadar yinelemeli ve sıralı olarak güncellenmektedir.

Şekil 3.2’de verilen akış şemasında görüldüğü üzere geliştirilen algoritmada, banknot kombinasyonu dağıtım politikasının belirlenmesi için problemin ölçeğine bağlı olarak iki farklı ödüllü öğrenme algoritması kullanılmıştır. Durum-karar sayısı

nispeten az (küçük ölçekli) olan problemlerde, literatürde de yer alan SMART (semi Markov average reward technique) ödüllü öğrenme algoritması kullanılmıştır. Durum-karar sayısı nispeten fazla (orta ve büyük ölçekli) olan problemler için ise SMART ödüllü öğrenme algoritmasından hareketle, durum-karar ikilisi değerlerinin çok katmanlı yapay sinir ağlarıyla tahmin edildiği NeuroSMART fonksiyon yaklaşımı ödüllü öğrenme algoritması geliştirilmiş ve kullanılmıştır. İzleyen ürün kombinasyonu dağıtım probleminin çözümü bölümünde, ödüllü öğrenme ve fonksiyon yaklaşımı detaylı olarak açıklanmıştır.



Şekil 3.2. Yaklaşık optimal ürün kombinasyonu dağıtım ve stok politikası belirleme algoritması

Akış şeması Şekil 3.2’de verilen algoritmanın adımları ise aşağıda verilmiştir:

Adım 1: Geliştirilen algoritmanın ilk adımı, her bir ürün çeşidi ($i=1,2,\dots,l$) için keyfi bir ürün kombinasyonu dağıtım politikasının seçilmesiyle başlar. Bu adımda seçilecek keyfi politikanın miyopik bir politika olması durumunda, ürün çeşitlerinin bireysel talep dağılımları kolaylıkla bulunabilir. Bölüm 4’te verilen uygulamalı örneklerde algoritmanın ilk adımı için, “bankamatige gelen her bir müşterinin para talebinin en az banknot çeşidini veren banknot kombinasyonu ile karşılanması” miyopik politikası kullanılmıştır.

Adım 2: İlk adımda belirlenen ürün kombinasyonu dağıtım politikasını kullanarak, her bir ürün çeşidinin bireysel ürün talep dağılımlarını bul.

Adım 3: 2’nci adımda belirlenen ürün çeşitlerinin talep dağılımlarını kullanarak, Bölüm 2.3.4.1’de verilen yöntemlerle, tedarik süresinin ihmal edildiği ($L=0$) varsayımıyla her bir ürün için (S,c,s) politikası belirle. Bu adımda tedarik süresi ihmal edildiği için tüm ürünlerin “s” kontrol parametresi, sıfır (0)’dır.

Adım 4: Her bir ürün çeşidi için adım 3’te belirlenen $(S,c,0)$ stok politikasından hareketle ve Bölüm 2.3.4.2’de verilen yöntemlerle, verilen sabit bir tedarik süresi ($L>0$) için 2’nci tip hizmet düzeylerini hesapla. Eğer stok sistemindeki ürün çeşitlerinden hizmet düzeyleri istenen düzeyin altında olanlar varsa, bu ürünler için belirlenen (S,c,s) politikasını, $(S+1,c+1,s+1)$ politikasıyla değiştir ve bu ürünler için adım 3’e dön. 2’nci tip hizmet düzeyi, istenen düzeyin üstünde olan ürünler için adım 5’e geç.

Adım 5: İzleyen bölümde verilen SMART/NeuroSMART ödüllü öğrenme algoritmasını kullanarak, adım 4’te belirlenen ürün çeşitlerinin (S,c,s) politikalarının birim zaman başına yaklaşık en düşük beklenen (ortalama) maliyetini veren ürün kombinasyonu dağıtım politikasını belirle. Diğer bir deyişle geçerli ürün kombinasyonu dağıtım politikasını güncelle.

Adım 6: Adım 5'te güncellenen geçerli ürün kombinasyonu dağıtım politikasını kullanarak, stok sistemindeki her bir ürün çeşidinin bireysel talep dağılımlarını yeniden bul ve ürün çeşitlerinin geçerli bireysel talep dağılımlarını güncelle.

Adım 7: Adım 6'da güncellenen ürün çeşitlerinin bireysel talep dağılımlarını kullanarak, $(S_{j+1}, c_{j+1}, s_{j+1})$ stok politikasını belirle. Eğer önceki yinelemede hesaplanan (S_j, c_j, s_j) stok politikası, son yinelemede hesaplanan $(S_{j+1}, c_{j+1}, s_{j+1})$ stok politikasıyla aynı ise, algoritmayı durdur. Belirlenen ürün kombinasyonu dağıtım politikası ve stok politikası yaklaşık optimaldir. Eğer (S_j, c_j, s_j) stok politikasıyla, $(S_{j+1}, c_{j+1}, s_{j+1})$ stok politikası birbirinden farklıysa adım 3'e dön.

3.2. Ürün Kombinasyonu Dağıtım Probleminin Çözümü

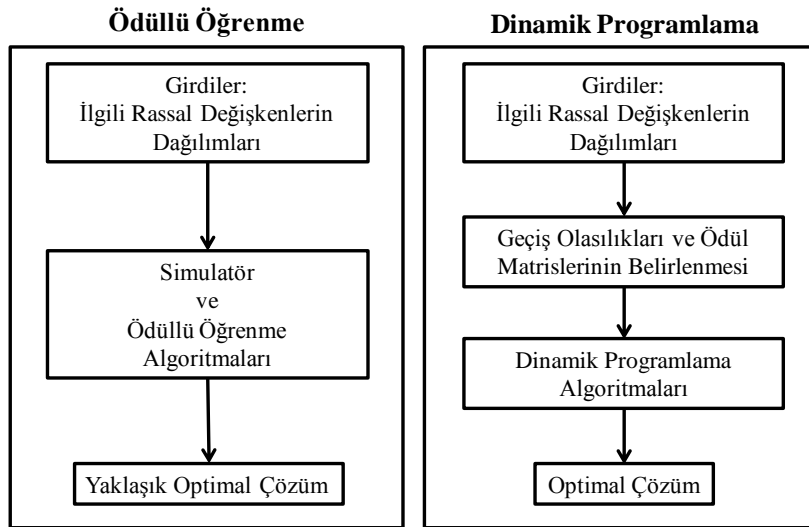
Bölüm 3.2.1'de Markov ve yarı Markov karar süreci problemlerini yaklaşık optimal olarak çözen ve simülasyon tabanlı bir optimizasyon tekniği olan ödüllü öğrenme tartışılmış ve Bölüm 3.2.2'de, durum-karar sayısının nispeten az olduğu (küçük ölçekli) yarı Markov karar süreci problemlerinin çözümü için SMART ödüllü öğrenme algoritması incelenmiştir. Bölüm 3.2.3'te ise durum-karar sayısının fazla olduğu (orta ve büyük ölçekli) yarı Markov karar süreci problemlerinin çözümü için, SMART ödüllü öğrenme algoritmasından hareketle çok katmanlı yapay sinir ağı kullanan fonksiyon yaklaşımı NeuroSMART ödüllü öğrenme algoritması geliştirilmiştir.

3.2.1. Ödüllü öğrenme

Ödüllü öğrenme (reinforcement learning), dinamik programlamaya dayalı tekniklerin geliştirilmesiyle ortaya çıkmıştır. Bu algoritmalar, her bir durum için başlangıçtan sona uzun dönemli olası en küçük maliyeti veren bir değer fonksiyonu hesaplar. Ödüllü Öğrenme, temel olarak simülasyon tabanlı bir optimizasyon tekniğidir ve başlıca Markov/yarı Markov problemlerinde yaklaşık optimal çözüm elde etmek için kullanılır. Bu teknik için literatürde çoğu zaman ödüllü öğrenme yerine, neuro dinamik programlama (neuro-dynamic programming) veya dinamik stokastik

programlama (dynamic stochastic programming) isimlerinin de kullanıldığı görülmektedir [37, 39].

Dinamik programlama algoritmaları, Markov ve yarı Markov karar süreci problemlerinin optimal çözümünü bulmayı garanti eder. Ancak, dinamik programlama algoritmalarının kullanılabilmesi için geçiş olasılıklarının, geçiş ödülleri ve -yarı Markov karar süreci problemleri için- geçiş sürelerinin hesaplanmış olması gerekmektedir. Geçiş olasılıklarının hesaplanması, rassal değişkenlerin olasılık fonksiyonlarını içeren çok katlı integral hesaplamalarını içerir. Büyük ölçekli ve karmaşık problemlerde rassal değişkenlerin sayısı arttıkça, bu hesaplamaların yapılması gittikçe daha zor hale gelir. Dinamik programlamada bu duruma, modellemenin laneti (curse of modeling) ismi verilmektedir. Hatta çoğu zaman küçük ölçekli ancak karmaşık yapıya sahip gerçek hayat problemlerinde bile geçiş olasılıklarının hesaplanması mümkün olmayabilir. Ödüllü öğrenme yöntemi, modellemenin lanetiyle başa çıkabilmek için iyi bir çözümdür. Şekil 3.3'te ödüllü öğrenme ve dinamik programlama algoritmaları karşılaştırılmaktadır. Şekilde görüldüğü üzere dinamik programlamanın aksine, ödüllü öğrenme algoritmalarını çalıştırabilmek için geçiş olasılıklarının, geçiş ödülleri ve geçiş sürelerinin hesaplanması gerekli değildir, çünkü tüm bu hesaplamalar bir simülasyon vasıtasıyla yapılmaktadır. Buna karşılık dinamik programlama algoritmalarıyla optimal çözüm elde edilirken, ödüllü öğrenme de yaklaşık optimal çözüme ulaşılabilmektedir [37].



Şekil 3.3. Yarı Markov karar süreci problemlerinin çözüm yöntemleri [37]

Dinamik programlama algoritmalarıyla ilgili diğeri bir problem, geçiş olasılıklarının depolanmasıyla ilgilidir. Örneğini 1000 adet durum (state) ve her bir durum için 3 kararın (actions) bulunduğu bir problem ele alınsın. Burada her bir faaliyetin geçiş olasılığı matrisi $1000^3 = 10^9$ (bir milyar) elemanlıdır ve dinamik programlama algoritmalarının kullanılabilmesi için bu matrisin bilgisayarda depolanması gerekmektedir. Her bir durumdaki karar sayısının sadece bir artması, yani verilen örnek için 4 olması durumunda, depolanması gereken matrisin eleman sayısı 10^{12} (bir trilyon) olacaktır. Günümüz bilgisayarlarında bile hafıza problemi nedeniyle, bu büyüklükte matrislerin depolanması mümkün değildir. Durum uzayının kesikli değil de sürekli olduğu problemlerde, depolama problemi daha da dramatik bir hal alır. Bu duruma boyutun laneti (curse of dimensionality) adı verilir. Ödüllü öğrenme bir dereceye kadar bu problemle de başa çıkabilmektedir. İlk olarak ödüllü öğrenme metotları geçiş olasılıkları matrisine ihtiyaç duymaz. İkinci olarak ödüllü öğrenme metotları dinamik programlamadaki durumların değer fonksiyonu elemanlarına da ihtiyaç duymaz. Ödüllü öğrenme bunun yerine değer fonksiyonlarını, Q faktör ismi verilen formlarda depolar. Ancak ödüllü öğrenmede bile problemin durum uzayı çok büyüdüğünde ya da durum uzayı sürekli ise, dinamik programlamada olduğu gibi Q faktör değerleri izlenebilir değildir. Bu nedenle Q faktör değerleri tek tek arama (lookup) tablolarına depolanma yerine, yapay sinir ağları (artificial neural network), regresyon ve interpolasyon gibi fonksiyon yaklaşım metotlarıyla tahmin edilir. Böylece az sayıda skaler sayıyla, milyonlarca durum-karar ikilisinin Q faktör değerlerine yaklaşılabilir ve ödüllü öğrenmede fonksiyon yaklaşım metotlarıyla, dinamik programlamadaki boyutun laneti problemiyle baş edilebilir [37, 39, 40].

Çoğu ödüllü öğrenme algoritması, dinamik programlamadaki değer fonksiyonunu hesaplar. Ancak farklı olarak dinamik programlama, verilen bir durumun değer fonksiyonu elemanı ile ilgilenirken; ödüllü öğrenme, verilen bir durum-karar ikilisinin Q faktör değer fonksiyonu elemanı ile ilgilenir. Örneğini iki durum ve her bir durumda iki kararın alınabildiği bir yarı Markov karar süreci problemi ele alınsın. Bu örnek için dinamik programlamada \mathbf{v}^* değer fonksiyonu vektörü Denklem 3.1'deki gibi 2 elemandan oluşurken, ödüllü öğrenmede Q faktör vektörü Denklem 3.2'deki gibi 4 durum-karar ikilisinden oluşur.

$$\mathbf{v}^* = \{\mathbf{v}^*(1), \mathbf{v}^*(2)\} \quad (3.1)$$

$$Q = \{Q(1,1), Q(1,2), Q(2,1), Q(2,2)\} \quad (3.2)$$

Yarı Markov karar süreci problemleri için her bir durum-karar ikilisinin ortalama maliyet Q faktör değer vektörü elemanları, Denklem 3.3'ten hesaplanır.

$$Q(i, a) = \sum_{j=1}^{|\mathcal{X}|} p(i, a, j) \left[r(i, a, j) - \rho^* t(i, a, j) + \mathbf{v}^*(j) \right] \quad (3.3)$$

Bu durumda bir durumun değer fonksiyonu ve bir durumun Q faktör ilişkisi, Denklem 3.4'teki gibidir.

$$\mathbf{v}^*(i) = \min_{a \in A(i)} Q(i, a) \quad (3.4)$$

Eğer Q faktör değerleri bilinirse, verilen bir durumun değer fonksiyonu Denklem 3.4'ten hesaplanabilir. Her bir durum-karar ikilisinin Q faktör değeri, Denklem 3.5'te verildiği gibi bir rassal değişkenin ortalaması olarak ifade edilebilir.

$$\begin{aligned} Q(i, a) &= \sum_{j=1}^{|\mathcal{X}|} p(i, a, j) \left[r(i, a, j) - \rho^* t(i, a, j) + \mathbf{v}^*(j) \right] \quad (3.5) \\ &= \sum_{j=1}^{|\mathcal{X}|} p(i, a, j) \left[r(i, a, j) - \rho^* t(i, a, j) + \min_{b \in A(j)} Q(j, b) \right] \\ &= E \left[r(i, a, j) - \rho^* t(i, a, j) + \min_{b \in A(j)} Q(j, b) \right] \\ &= E \left[\text{Örneklem} \right] \end{aligned}$$

S^i , X rassal değişkeninin i'nci bağımsız örneğini ve n, örneklem büyüklüğünü

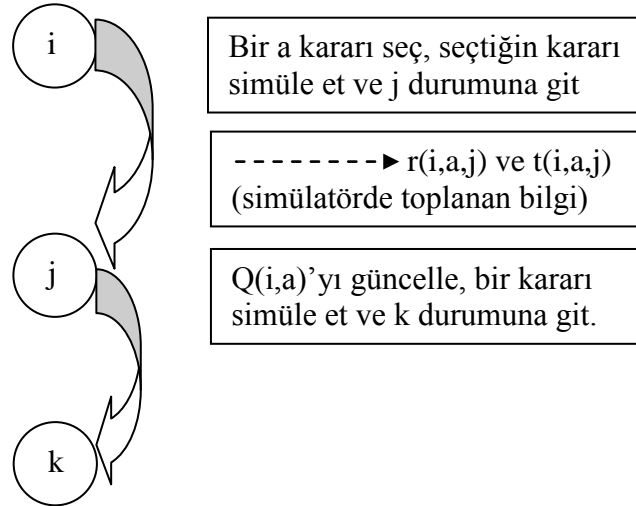
göstermek üzere, güçlü sayılar kanununa göre $P \left(E[X] = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n S^i}{n} \right) = 1$ 'dir [41].

Eğer $\alpha^{n+1} = \frac{1}{n+1}$ olarak ifade edilirse, Robbins-Monro algoritmasına göre $X^{n+1} = (1 - \alpha^{n+1})X^n + \alpha^{n+1}s^{n+1}$ 'dir [37]. Böylece Robbins-Monro algoritması kullanılarak, dinamik programlamada mutlaka bulunması gereken geçiş olasılıklarını hesaplamaksızın Q faktör değerleri, bir simülatörde Denklem 3.6 güncelleme denkleminle tahmin edilebilir.

$$Q^{n+1}(i, a) \leftarrow (1 - \alpha^{n+1})Q^n(i, a) + \alpha^{n+1} \left[r(i, a, j) - \rho^* t(i, a, j) + \min_{b \in A(j)} Q^n(j, b) \right] \quad (3.6)$$

Bu denklemde α , adım sayısı ya da öğrenme oranı; $r(i, a, j)$, i durumundan a kararı altında j durumuna geçmenin anlık maliyeti (immediate cost); ρ^* , sonsuz zaman ufunda birim zaman başına optimal ortalama maliyet ve $t(i, a, j)$, i durumundan a kararı altında j durumuna geçiş süresidir.

Ödüllü öğrenmedeki temel düşünce, iyi kararların ödüllendirilmesi ve kötü kararların cezalandırılmasıdır. Herhangi bir verilen i durumu için iyi kararların Q faktör değerleri iyileştirilir, kötü kararların Q faktör değerleri ise kötüleştirilir. Şekil 3.4'te ödüllü öğrenme algoritmalarının simülatör içindeki genel çalışma prensibi verilmiştir. Şekilde verilen oklar simülatörde durumlar arası geçişleri göstermektedir ve görüldüğü üzere sistem i durumundayken simülatör bir a kararını seçmekte ve bu kararın sonucu olarak sistem j durumuna gitmektedir. Simülatörün i durumundan j durumuna geçişindeki zaman aralığı süresince ödüllü öğrenme algoritması, sistem i durumundayken a kararı altında j durumuna geçmenin anlık maliyeti $r(i, a, j)$ ve i durumundan a kararı altında j durumuna geçiş süresi $t(i, a, j)$ bilgilerini toplar. Simülatör j durumuna ulaştığında $r(i, a, j)$ ve $t(i, a, j)$ kullanılarak, $Q(i, a)$ 'nın yeni bir örnekleme üretilir ve Denklem 3.6 güncelleme denkleminin vasıtasıyla $Q(i, a)$ güncellenir. Böylece, sonraki duruma geçtikten sonra durum-karar ikilisi için güncelleme gerçekleşmiş olur [37].



Şekil 3.4. Simülâtörde Q faktör değerlerinin güncellenmesi [37]

α öğrenme oranı için 1'den küçük sabit bir değer kullanılabileceği gibi, her bir durum-karar ikilisinin kaç defa ziyaret edildiği bilgisi tutularak, bu bilgidir hareketle simülasyon boyunca dinamik olarak da belirlenebilir. Simülâtörde bir durum-karar ikilisinin Q faktör değeri, başka durum-karar ikililerinin Q faktör değerlerinden daha fazla sayıda güncellenebilir. Bu durumun bir sonucu olarak bir hata ortaya çıkar. Eğer α çok küçük seçilirse bu hata ihmal edilebilir derecede azalır, ancak bu durumda algoritma yavaşlar. α , 1'e yakın seçildiğinde ise algoritmanın ilerleme hızı artar, ancak bu durumda hata artık ihmal edilebilir değildir ve bu nedenle algoritma yakınsamayabilir [37, 39].

$V(i,a)$, i durumunda a kararının simülâtörde ziyaret edilme sayısını gösterebilir. α , her bir durum karar ikilisinin ziyaret sayıları $V(i,a)$ 'lar izlenerek dinamik olarak belirleniyorsa, bu durumda ziyaret sayıları büyüdükçe α , sifıra yaklaşmaya başlar. α , sifıra çok yakın bir değere ulaştığında artık Q faktör değerleri belirgin bir şekilde değişmeyeceği için simülasyon durdurulabilir. Simülasyonun diğer bir durdurma kriteri, yeterince büyük seçilmiş yineleme sayısı olabilir. Simülasyonda yeterince büyük seçilmiş yineleme sayısı, α 'nın gittikçe küçülmesini ve Q faktör değerlerinin artık değişmemeye başlamasını garanti eder [37].

Simülâtörde, durumlara uygulanacak kararların seçiminde farklı stratejiler uygulanabilir. Örneğin bir durum için alınabilecek her kararın eşit olasılıkla seçilmesi, gürbüz (robust) bir stratejidir. Diğer bir başarılı strateji ise, keşif

(exploratory) stratejisidir. Bu stratejide en yüksek Q faktör değerine sahip karar (greedy action) yüksek bir olasılıkla, diğer kararlar ise daha düşük, ancak sıfır olmayan olasılıklarla seçilir. Bu durumda simülasyonun başında greedy olmayan kararlar seçilecek, ancak simülasyon ilerledikçe greedy olmayan kararların seçilmesi olasılığı gittikçe azalacaktır. Bu tür bir karar seçimi stratejisi, simülasyonun çalışma uzunluğunu önemli ölçüde kısaltır [37]. Bu nedenle yapılan çalışmada simülatörde karar seçimi için keşif (exploratory) stratejisi uygulanmıştır.

Çalışmada ortalama ödül problemi, maliyet açısından ele alınmıştır. Bu tür problemlerde performans ölçütü olan ortalama ödül skaler bir ölçü birimi olduğu için, ödüllü öğrenme algoritmasıyla öğrenilen politikaların performanslarının ölçülmesi nispeten daha kolaydır. Çalışmada ödüllü öğrenme algoritmasının performansının ölçülmesi için belli yinelemelerde o ana kadar öğrenilen politika simüle edilerek, birim zamandaki ortalama maliyet hesaplanmıştır. Ödüllü öğrenme algoritmasını içinde bulunduran simülasyon ilerledikçe öğrenilen politikanın ortalama maliyeti azalmakta, artık Q faktör değerleri değişmemeye başladığında ise elde edilen politikanın ortalama maliyeti bir değere yakınsamaktadır.

3.2.2. Küçük ölçekli Markov karar süreci problemlerinin çözümü

Çalışmada sonsuz ufukta birim zaman başına ortalama maliyetin en küçüklenmesi amaçlandığı için ödüllü öğrenmede ortalama ödül (average reward) algoritmaları üzerinde durulacaktır. Ortalama ödül yarı Markov karar süreci problemi ya politika yineleme (policy iteration) yaklaşımıyla, ya da değer yineleme (value iteration) yaklaşımıyla çözülebilir. Ödüllü öğrenmede, yarı Markov karar süreci problemlerinin süreci birleştirme (uniformization) işlemiyle Markov karar sürecine dönüştürülüp çözülmesi işleminden kaçınılır. Çünkü hem sürekli zaman Markov süreci problemlerinin dışında birleştirme işlemiyle süreç tam olarak Markov karar sürecine dönüştürülemez ve hem de bu işlem için geçiş olasılıklarının bilinmesi gerekir [37]. Bu nedenle çalışmada durum-karar sayısının nispeten az olduğu problemler, yarı Markov karar süreci problemleri için geliştirilen bir ödüllü öğrenme algoritması olan SMART (semi Markov average reward technique) tekniği ile çözülmüştür.

3.2.2.1. SMART algoritması

Bölüm 2.2’de verildiği üzere ortalama ödül yarı Markov karar süreci modelinin optimallik denkleminde, optimal politikanın maliyeti ρ^* yer almaktadır (Bkz. Denklem 2.20). Bu nedenle değer yineleme (value iteration) için ödüllü öğrenme güncelleme denklemini formüle etmede, Denklem 2.20’de verilen optimallik denklemini doğrudan kullanılamaz. Ortalama ödül yarı Markov karar süreci problemlerinde güncelleme için Denklem 2.20 optimallik denklemini yerine, bunun Q faktör versiyonu olan Denklem 3.7 kullanılır [37].

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

Denklem 3.7’de yer alan α öğrenme oranı, SMART algoritmasında (i,a) durum-karar ikililerinin simülâtördeki ziyaret sayılarından hesaplanmaktadır. $r(i, a, j)$, i durumundan j durumuna geçildiğinde oluşan anlık maliyet ve ρ bilinmeyen ancak çözüm boyunca güncellenerek ρ^* ’a yaklaşan politikanın birim zamandaki ortalama maliyetidir. Sonuç olarak SMART algoritmasında sadece Q-faktör değil, ρ ’da güncellenir. Ancak ρ ’nun algortmada güncellenmesi için, simülâtörde seçilen faaliyetin o ana kadar o durumdaki en iyi Q-faktöre değerine sahip karar (greedy action) olması gerekir. Aksi durumda ρ güncellenmez. Algoritmanın adımları aşağıda verilmiştir [37]:

Adım 1: Tüm Q faktörleri, ziyaret sayılarını, simülasyonda algoritmanın yineleme sayısını, toplam süreyi, toplam maliyeti ve greedy politikanın maliyetini sıfır (0) yap.

- _ $l \in S, u \in A(l)$ için $Q(l, u) \leftarrow 0$ ve $V(l, u) \leftarrow 0$
- _ $k \leftarrow 0$ (k: zincirdeki sıçrama sayısı veya yineleme sayısı)
- _ Toplam süre $\leftarrow 0$
- _ Toplam maliyet $\leftarrow 0$

– $\rho \leftarrow 0$ (ρ : greedy politikanın ortalama maliyeti)

1'den küçük öğrenme oranı sabiti A 'yı belirle ($A < 1$). Yeterince büyük k_{enb} (algoritmanın çalışacağı yineleme sayısı) belirle. Sistemin simülasyonu herhangi keyfi durumdan başlat.

Adım 2: Mevut durum i olsun. Rastgele bir karar a seç. ϕ 'i, Denklem 3.8'e göre belirle.

$$\phi = \begin{cases} 0, & a \in \arg \min_{u \in A(i)} Q(i, u) \text{ ise} \\ 1, & \text{aksi halde} \end{cases} \quad (3.8)$$

Adım 3: Karar a 'yı simüle et. Gidilen durum j olsun. $r(i, a, j)$, a kararı altında i durumundan j durumuna geçildiğinde kazanılan anlık maliyet ve $t(i, a, j)$, aynı geçiş için geçen süreyi gösterebilir. Bu iki değer de simülatör tarafından hesaplanmaktadır. $V(i, a)$, (i, a) ikilisinin ziyaret sayısını ve k 'yi 1 arttır. Öğrenme oranı $\alpha = A / V(i, a)$ 'yı hesapla.

Adım 4a: Denklem 3.7'de verilen güncelleme denklemini kullanarak $Q(i, a)$ 'yı güncelle.

Adım 4b: Eğer seçilen a kararı greedy ise ($\phi = 0$ ise) toplam süreyi Denklem 3.9'dan ve toplam maliyeti Denklem 3.10'dan güncelle.

$$\text{Toplam süre} \leftarrow \text{Toplam süre} + t(i, a, j) \quad (3.9)$$

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

Adım 4c: ρ 'yu Denklem 3.11'i kullanarak güncelle.

$$\rho = \frac{\text{Toplam Maliyet}}{\text{Toplam Süre}} \quad (3.11)$$

Adım 5: Eğer $k < k_{\text{enb}}$ ise adım 2'ye dön. Değilse adım 6'ya geç.

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

3.2.2.2. Politika ortalama maliyetlerinin simülâtörde hesaplanması

Çalışmada ödüllü öğrenme algoritmasıyla öğrenilen politikaların performansını ölçmek için aşağıda verilen algoritma kullanılmıştır. Ödüllü öğrenme algoritmasının simülasyonunda belirli yineleme aralıklarıyla verilen algoritma çalıştırılarak, o ana dek öğrenilen politikanın birim zamandaki ortalama maliyeti hesaplanmaktadır. Ödüllü öğrenme algoritması yinelemeleri ilerledikçe, hesaplanan ortalama maliyetler bir limite yakınsamaktadır.

Adım 1: Toplam süreyi ve toplam maliyeti sıfır (0) yap.

- _ Toplam süre $\leftarrow 0$
- _ Toplam maliyet $\leftarrow 0$

Simülâtörde ortalama maliyetlerin bir değere yakınsamasını sağlayabilecek kadar uzun bir simülasyon süresi belirle.

Adım 2: Mevut durum i olsun. Denklem 3.12'yi kullanarak a kararını seç.

$$a \in \arg \min_{u \in A(i)} Q(i, u) \quad (3.12)$$

Adım 3: Karar a 'yı simüle et. Gidilen durum j olsun. $r(i, a, j)$, a kararı altında i durumundan j durumuna geçildiğinde kazanılan anlık maliyet ve $t(i, a, j)$, aynı geçiş için geçen süreyi gösterebilir. Toplam süreyi Denklem 3.13'ü kullanarak ve toplam maliyeti Denklem 3.14'ü kullanarak güncelle.

$$\text{Toplam süre} \leftarrow \text{Toplam süre} + t(i, a, j) \quad (3.13)$$

$$\text{Toplam maliyet} \leftarrow \text{Toplam maliyet} + r(i, a, j) \quad (3.14)$$

Adım 4: Eğer $\text{Toplam süre} < \text{simülasyon süresi}$ ise, $i \leftarrow j$ yap ve adım 2'ye dön. Değilse adım 5'e git.

Adım 5: Denklem 3.15'i kullanarak öğrenilen politikanın birim zamandaki ortalama maliyetini hesapla.

$$\rho = \frac{\text{Toplam Maliyet}}{\text{Toplam Süre}} \quad (3.15)$$

3.2.3. Orta ve büyük ölçekli Markov karar süreci problemlerinin çözümü

Çalışmada ele alınan problem karmaşık yapılı ve durum-karar sayısı büyük olan bir yarı Markov karar süreci problemidir. Bu nedenle çözümde, modellemenin lanetiyle baş edebilmek için ödüllü öğrenme algoritmaları kullanılmıştır. Ancak, ödüllü öğrenme algoritmaları kullanılmasına rağmen, durum-karar sayısı büyük olduğu için boyutun laneti problemi hala ortadadır. Bu nedenle milyonlarca Q-faktör değerini depolamak yerine, bu değerlere yapay sinir ağları (artificial neural networks) kullanılarak yaklaşmıştır.

3.2.3.1. Q faktör değer fonksiyonuna yaklaşım

Bir Markov ya da yarı Markov karar süreci problemi ödüllü öğrenme algoritmalar ile çözülmek istendiğinde, Q faktör fonksiyonu önceden bilinmediği için bu fonksiyonu

tam olarak temsil etmeyi garanti eden tek yol, simülâtörde elde edilen tüm durum-karar ikililerinin Q faktör değerlerinin arama tablolarında ayrı ayrı depolanmasıdır. Böylece simülasyonun her yinelemesinde, önceki yinelemede depolanan Q faktörler tablolardan alınıp kullanılır. Ancak durum-karar ikilisi sayısı büyüdükçe veya durum-karar ikilileri sürekli değişkenlerse, tüm durum-karar ikililerinin Q faktör değerlerinin depolanması olanaksızlaşır. Hatta çoğu zaman küçük ölçekli problemler için bile tüm durum-karar ikililerinin Q faktör değerlerinin arama tablolarında depolanması, günümüz bilgisayar teknolojisinde bile mümkün değildir. Bu nedenle tüm Q faktör değerlerini depolayamadığımız problemleri çözebilmenin tek yolu, Q faktör fonksiyonunu temsil eden yaklaşımlar kullanmaktır. Q faktör fonksiyonuna yaklaşılabilecek birçok yol vardır. Örneğin Q faktör fonksiyonuna yaklaşmak için durum uzayını kümeleme (state aggregation), fonksiyon enterpolasyonu (function interpolation) ve fonksiyon uydurma (function fitting) gibi yöntemler kullanılabilir [37, 39, 40].

Durum uzayını kümeleme yönteminde fazla sayıdaki durum-karar ikililerinin sayısını azaltmak için bazı durumlar birleştirilmekte ve bu şekilde daha az sayıda durum-karar ikilileri oluşmaktadır. Genel kural olarak benzer karakteristiklere sahip durumlar birleştirilir. Böylece, azalan durum-karar ikililerinin Q faktör değerleri tablolarda depolanabilmekte ve problemin çözümü için ödüllü öğrenme algoritmaları kullanılabilir [37].

Enterpolasyon, matematik literatüründe iyi çalışılmış bir konudur ve literatürdeki enterpolasyon metotları ödüllü öğrenmede Q faktör fonksiyonuna yaklaşmak için kullanılabilir. Temel düşünce Q faktörleri temsilen birkaç Q faktörün depolanması ve tüm diğer Q faktörlerin depolanan bu değerlerden hareketle enterpolasyon teknikleri kullanılarak belirlenmesidir [37].

Çok sayıda durum-karar ikililerinin Q faktör değerlerinin depolanması gerekliliği ile baş edebilmek için kullanılan diğer bir yöntem regresyon ya da yapay sinir ağları gibi fonksiyon uydurma yöntemleriyle Q faktör fonksiyonuna yaklaşılmaktadır [37].

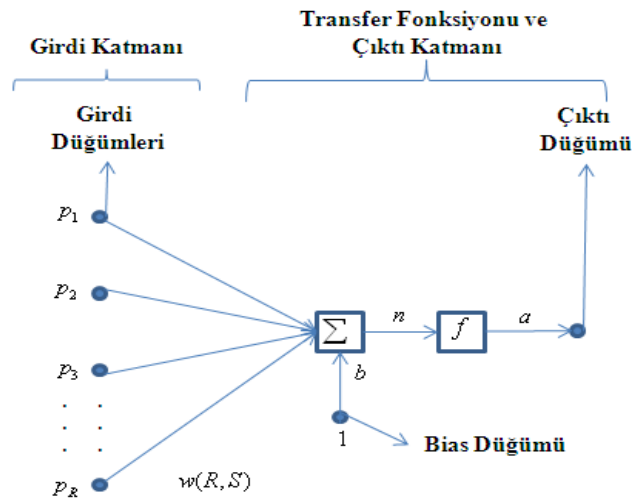
Çalışmada orta ve büyük ölçekli problemlerde tüm durum-karar ikililerinin Q faktör değerlerinin depolanması gerekliliği ile baş edebilmek için, Q faktör fonksiyonuna çok katmanlı yapay sinir ağlarıyla fonksiyon uydurulması yaklaşımı kullanılmıştır.

3.2.3.2. Yapay sinir ağları

Yapay sinir ağlarının (artificial neural networks, ANN) çıkış noktası insan biyolojisinin taklit edilmesi ve ilham kaynağı da insan beynidir. Bilişsel bilim (cognitive science) ve sinir bilimi (neuro science) uzmanları insan beynindeki sınırları modelleyip simüle ederek, beyin fonksiyonları anlamayı amaçlamışlardır. Mühendisliğin insan beyni sınırlarıyla ilgilenmesinin amacı ise onları anlamak değil, kullanışlı makineler için bu tür yapıları kullanmaktır. İnsan beyni mükemmel bir bilgi işleme aracıdır ve görüntü/ses tanıma, öğrenme vb. inanılmaz yeteneklere sahiptir. Eğer insan beyninin bu fonksiyonları nasıl gerçekleştirdiği bulunup mühendislik alanına uygulanabilirse, zor mühendislik problemlerinin çözümünde büyük kolaylıklar sağlanabilir. İnsan beyni bir bilgisayardan çok farklıdır. Bir bilgisayar genellikle birkaç işlemciye sahipken, insan beyni nöron ismi verilen ve birbiriyle paralel çalışan 10^{11} işlemci biriminden oluşur. İnsan beynindeki her bir nöron çevresindeki yaklaşık 10^4 nöronla tamamen paralel çalışan bağlantılara sahiptir. Bu nedenle insan beyninin bir bilgisayarla kıyaslanamayacak ölçüde hesaplama gücüne mevcuttur [42].

Yapay sinir ağları, istenen bir sistem performansını gerçekleştirebilen bir kontrol biçimini öğrenebilmek için uygun ağırlık bağlantılarıyla kendi kendilerini biçimlendirir. Yapay sinir ağları genel olarak verilen giriş-çıkış çiftlerinden ağırlık parametrelerini ayarlama temeline dayanır [43].

Yapay sinir ağları temel olarak doğrusal ve doğrusal olmayan olmak üzere ikiye ayrılır. Doğrusal yapay sinir ağları model bağımlıdır ve yaklaştığı fonksiyonun doğrusal bir modele uyduğunu kabul eder [37]. Şekil 3.5'te doğrusal bir yapay sinir ağının mimarisi verilmiştir. Şekilden de görüldüğü üzere doğrusal yapay sinir ağları, girdi düğümleri ve çıktı düğümü olmak üzere iki katmandan oluşur ve herhangi bir denklem sistemi çözmeksizin doğrusal regresyon gerçekleştirir.



Şekil 3.5. Doğrusal yapay sinir ağı mimarisi

Şekil 3.5'teki p girdi düğümü, R elemandan oluşur. Şekilde verilen tek bias düğümü regresyon denklemindeki sabit katsayıyı (b), diğer girdi düğümleri ise bağımlı değişkenleri ifade eder. Burada $w(\cdot)$ ise ağırlıklardır ve Denklem 3.16'daki gibi $[p_1, p_2, \dots, p_R]$, $[w_{1,1}, w_{1,2}, \dots, w_{1,R}]$ ağırlıkları ile çarpılır ve toplanır. Bu toplama Denklem 3.17'deki gibi bias (b) eklenerek transfer fonksiyonuna net girdi (n) sağlanır [44].

$$Wp = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R \quad (3.16)$$

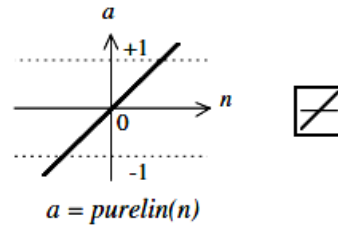
$$n = Wp + b \quad (3.17)$$

Denklem 3.18'de verildiği üzere transfer fonksiyonunda net girdi (n) kullanılarak çıktı düğümü a elde edilir.

$$a = f(n) \quad (3.18)$$

Öğrenme eğrisi olarak da adlandırılan transfer fonksiyonları, bir yapay sinir ağında nöronun çıkış genliğini istenilen değerler arasında sınırlar. Bu değerler çoğunlukla $[0,1]$ veya $[-1,1]$ arasındadır. Yapay sinir ağlarında kullanılacak olan fonksiyonların, türevi alınabilir ve sürekli fonksiyonlar olması gerekmektedir [43].

Dorsal yapay sinir ağlarında Şekil 3.6'te verilen doğrusal transfer fonksiyonu (purelin) kullanılır ve ağıın çıktısı Denklem 3.19'dan hesaplanır.

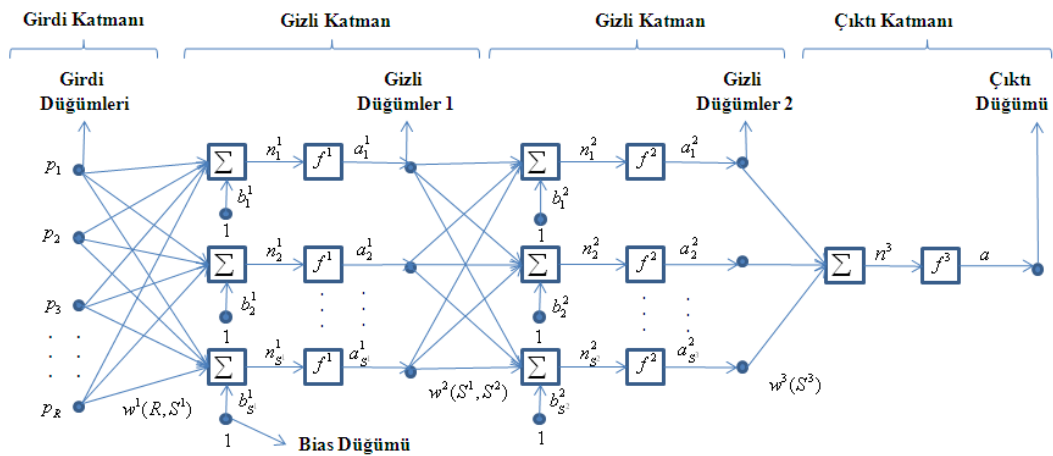


Şekil 3.6. Doğrusal transfer fonksiyonu (purelin) [44]

$$a = f(n) = \text{purelin}(n) = \text{purelin}(Wp + b) = Wp + b \quad (3.19)$$

Doğrusal yapay sinir ağları, doğrusal olmayan bir fonksiyona doğrusal yaklaşım için kullanılabilir ancak doğrusal olmayan bir hesaplama gerçekleştiremez. Doğrusal olmayan ve matematiksel olarak modellenemeyen problemlerin çözümü için çok katmanlı yapay sinir ağları geliştirilmiştir. Çok katmanlı yapay sinir ağlarının en önemli özellikleri matematiksel olarak modellenemeyen fonksiyonlara yaklaşma yetenekleridir [43, 44].

Şekil 3.7'de 1 girdi katmanlı, 2 gizli katmanlı ve 1 çıktı katmanlı yapay sinir ağı mimarisi verilmiştir. Daha fazla gizli katman bulunan yapay sinir ağlarının yapısı da benzerdir.



Şekil 3.7 Dört katmanlı yapay sinir ağı mimarisi

Şekildeki 4 katmanlı yapay sinir ağında girdi katmanında R adet girdi düğümü, 1. gizli katmanda S^1 adet gizli düğüm ve bias düğümü, 2. gizli katmanda S^2 adet gizli düğüm ve bias düğümü, çıktı katmanında ise tek çıktı düğümü vardır. Girdi düğümleri 1. gizli katmandaki her bir gizli düğümle, 1. gizli katmandaki her bir gizli düğüm 2. katmandaki her bir gizli düğümle ve 2. gizli katmandaki her bir gizli düğüm çıktı düğümüyle tek yönlü bağlantılara sahiptir. Şekilde girdi katmanındaki düğümlerle 1. gizli katmandaki düğümler arasındaki ağırlıklar $w^1(R, S^1)$ ile 1. gizli katmandaki düğümler ile 2. gizli katmandaki düğümler arasındaki ağırlıklar $w^2(S^1, S^2)$ ile ve ise 2. gizli katmandaki düğümlerle çıktı düğümü arasındaki ağırlıklar $w^3(S^3)$ ile gösterilmektedir. Çok katmanlı yapay sinir ağı fonksiyon uydurmanın yanı sıra örüntü tanıma, verileri kümeleme vb. birçok alanda kullanılmaktadır ve bu alanlardaki kullanımlarda çıktı katmanı birden fazla düğümde oluşabilmektedir. Ancak yapay sinir ağlarının çalışmadaki kullanımı olan fonksiyon yaklaşımı için, çıktı katmanında tek düğüm yer almaktadır. Girdi katmanındaki R girdi düğümü genellikle her bir bağımsız değişkenle ilişkilidir ve bu nedenle girdi düğüm sayısı, fonksiyon uydurma sürecindeki değişken sayısına eşittir [37].

Gizli katman sayısı, fonksiyon yaklaşımının başarısındaki önemli faktörlerdendir. Uydurulması gereken fonksiyon ne kadar karmaşık ve doğrusallıktan uzaksa, o kadar fazla sayıda gizli katman kullanılması gerekir. Ancak gizli katman sayısı arttıkça, yapay sinir ağının hesaplama yükü de artmakta ve bu nedenle ağın eğitimi ve ağırlıkların hesaplanma süreleri uzamaktadır. Uygun gizli katman sayısını belirlemek için belirli bir kural mevcut değildir, genellikle deneme yanılma yöntemiyle kullanılacak gizli katman sayısı belirlenir [37, 43].

Şekil 3.7’de görüldüğü gibi genellikle çok katmanlı yapay sinir ağlarında, regresyon modelindeki sabit katsayıyı temsil etmek için doğrudan her bir ara çıktı düğümleriyle bağlantılı bias düğümleri kullanılır.

1. gizli katmanın net girdileri $[n_1^1, n_2^1, \dots, n_{s^1}^1]$, Denklem 3.20’den hesaplanır.

$$n_1^1 = p_1 w(1, 1^1) + p_2 w(2, 1^1) + \dots + p_R w(R, 1^1) + b_1^1 \quad (3.20)$$

$$n_2^1 = p_1 w(1, 2^1) + p_2 w(2, 2^1) + \dots + p_R w(R, 2^1) + b_2^1$$

$$\vdots$$

$$n_{S^1}^1 = p_1 w(1, S^1) + p_2 w(2, S^1) + \dots + p_R w(R, S^1) + b_{S^1}^1$$

1. gizli katman net gidileri transfer fonksiyonunda kullanılarak, 1. gizli katmanın çıktıları (2. gizli katmanın girdileri) Denklem 3.21'den hesaplanır.

$$a_1^1 = f(n_1^1) \quad (3.21)$$

$$a_2^1 = f(n_2^1)$$

$$\vdots$$

$$a_{S^1}^1 = f(n_{S^1}^1)$$

2. gizli katmanın net girdileri $[n_1^2, n_2^2, \dots, n_{S^2}^2]$, Denklem 3.22'den hesaplanır.

$$n_1^2 = a_1^1 w(1^1, 1^2) + a_2^1 w(2^1, 1^2) + \dots + a_{S^1}^1 w(S^1, 1^2) + b_1^2 \quad (3.22)$$

$$n_2^2 = a_1^1 w(1^1, 2^2) + a_2^1 w(2^1, 2^2) + \dots + a_{S^1}^1 w(S^1, 2^2) + b_2^2$$

$$\vdots$$

$$n_{S^2}^2 = a_1^1 w(1^1, S^2) + a_2^1 w(2^1, S^2) + \dots + a_{S^1}^1 w(S^1, S^2) + b_{S^2}^2$$

2. gizli katman net gidileri transfer fonksiyonunda kullanılarak, 2. gizli katmanın çıktıları (çıkıtı katmanının girdileri) Denklem 3.23'ten hesaplanır.

$$a_1^2 = f(n_1^2) \quad (3.23)$$

$$a_2^2 = f(n_2^2)$$

$$\vdots$$

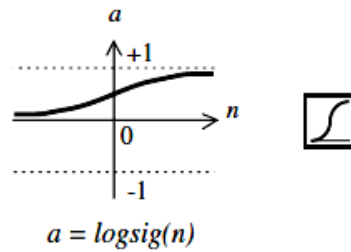
$$a_{S^3}^2 = f(n_{S^2}^2)$$

Çıktı katmanının net girdisi n^3 , Denklem 3.24 ve çıktı değeri a , Denklem 3.25'den hesaplanır.

$$n^3 = a_1^2 w^3(1^3) + a_2^2 w^3(2^3) + \dots + a_{S^3}^2 w^3(S^3) \quad (3.24)$$

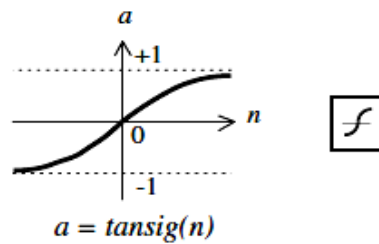
$$a = f(n^3) \quad (3.25)$$

Çok katmanlı yapay sinir ağlarında genellikle sigmoid transfer fonksiyonları kullanılır. Şekil 3.8'de verilen log-sigmoid transfer fonksiyonu, nöronun net girdisi negatiften pozitifte doğru gittikçe 0 ve 1 arasında çıktılar üretir.



Şekil 3.8 Log-sigmoid transfer fonksiyonu [44]

Alternatif olarak çok katmanlı yapay sinir ağlarında tan-sigmoid (tansig) transfer fonksiyonu da kullanılabilir. Şekil 3.9'da verilen tan-sigmoid transfer fonksiyonu ise nöronun net girdisi negatiften pozitifte doğru gittikçe, -1 ve 1 arasında çıktılar üretir.



Şekil 3.9 Tan-sigmoid transfer fonksiyonu [44]

Çok katmanlı ağların eğitimi sırasında hem girdiler ve hem de bu girdilere karşılık üretilmesi gereken çıktılar gösterilir. Ağın görevi, her girdi için o girdiye karşılık gelen çıktıyı üretmektir. Çok katmanlı ağın öğrenme kuralı, en küçük kareler

yöntemine dayanır. Buna göre önce ileri doğru hesaplama ile ağın çıktısı hesaplanır, sonrasında ise ağırlıkların değiştirildiği geriye doğru hesaplama (backpropagation) yapılır [43].

Ağırlıkların güncellendiği geriye doğru hesaplama algoritmaları, temel olarak bir doğrusal olmayan gradyen azaltım (gradient-descent) algoritmalarıdır. Bu algoritmalarda en küçüklenmek istenen amaç fonksiyonu, mevcut çıktı değeri ile ağın ürettiği çıktı değeri farkının kareler toplamı (sum of the square of errors, SSE), karar değişkenleri ise ağdaki ağırlıklardır [37].

Geriye doğru hesaplama algoritması, gradyen azaltım algoritması olduğu için kullanılan transfer fonksiyonlarının türevleri hesaplanabilir fonksiyonlar olması önemlidir. Çalışmada verilen log-sigmoid, tan-sigmoid ve purelin fonksiyonları türevi hesaplanabilen transfer fonksiyonlarıdır [37]. Çok katmanlı yapay sinir ağlarında genellikle gizli katmanlarda sigmoid transfer fonksiyonları, çıktı katmanında ise doğrusal transfer fonksiyonu kullanılır. Böylece doğrusal olmayan transfer fonksiyonlu gizli katmanlar, girdi ve çıktılar arasındaki doğrusal ve doğrusal olmayan ilişkilerin öğrenilmesine, doğrusal çıktı katmanı ise ağın [-1,1] aralığı dışında değer üretmesine olanak tanır. Eğer ağın çıktı değerinin (örneğin 0 ve 1 arasında) kısıtlanması isteniyorsa, bu durumda çıktı katmanında da sigmoid transfer fonksiyonları kullanılabilir [44].

Yapay sinir ağının mimarisi ve başlangıç ağırlıkları keyfi olarak küçük sayılardan, biaslar ise 1 ya da 1'den farklı bir sabit sayı olarak belirlendikten sonra geriye doğru hesaplama algoritmasıyla ağırlıklar ve biaslar eğitilir. Eğitim süresince ağırlıklar ve biaslar, performans fonksiyonunu en küçükleyecek şekilde yinelemeli olarak ayarlanır. Genellikle performans fonksiyonu olarak Denklem 3.26'da verilen ağın çıktısı "a" ile hedeflenen çıktı "t" arasındaki ortalama hata kareleri (mean square of errors, MSE) kullanılır [44].

$$F = MSE = \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2 \quad (3.26)$$

Çok katmanlı ağlarda ağırlıkların eğitimi için kullanılan birçok geriye doğru çalışan eğitim algoritması mevcuttur. Bu algoritmaların tümü performans fonksiyonunu en küçükleyen ağırlıkları belirlemek için performans fonksiyonunun gradyenini (birinci dereceden türevini) kullanır, ancak Levenberg-Marquardt gibi hızlı algoritmalar ikinci türevleri de kullanır [43].

Geriye doğru hesaplama algoritmasının en basit uygulamasında ağırlıklar ve biaslar, performans fonksiyonunun en hızlı şekilde azaldığı yani gradyenin negatif olduğu yönde güncellenir. Bu algoritmanın bir yinelemesi Denklem 3.27'deki gibi yazılabilir [44].

$$x_{k+1} = x_k - \gamma_k g_k \quad (3.27)$$

Denklemden x_k mevcut ağırlık ve bias vektörü, g_k mevcut gradyen ve γ_k öğrenme oranıdır. Gradyen azaltım algoritmasında güncelleme, artımlı güncelleme (incremental updating) ve yığın güncelleme (batch updating) olmak üzere iki farklı şekilde yapılabilir. Artımlı güncellemede gradyen hesaplanır ve her bir girdinin ağa uygulanmasından sonra ağırlıklar güncellenir. Yığın güncelleme de ise tüm girdiler, ağırlıklar güncellenmeden önce ağa uygulanır [44].

Çalışmada yapay sinir ağının eğitilmesinde Levenberg-Marquardt eğitim algoritması kullanılmıştır. Performans fonksiyonu kareler toplamı olduğundan Hessian matrisine Denklem 3.28'deki gibi yaklaşılabilir [44].

$$H = J^T J \quad (3.28)$$

Gradyen ise Denklem 3.29'dan hesaplanır [44].

$$g = J^T e \quad (3.29)$$

Burada J , ağırlıklar ve biaslarla ilgili ağın birinci dereceden türevini içeren Jacobian matrisi, e ise ağ hatalarının bir vektörüdür. Hessian matrisinin tersinin hesaplanamaması durumunun üstesinden gelmek için bu algorithmada, Hessian

matrisine küçük bir μ sabiti eklenir ve Denklem 3.30'da verildiği şekilde ağırlıklar güncellenir [43, 44].

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (3.30)$$

Başlangıçta denklemde küçük bir μ değeriyle başlanır, performans fonksiyonundaki her azalıştan sonra μ 10 kat azaltılır, performans fonksiyonundaki her artıştan sonra da μ 10 kat arttırılır. Bu şekilde performans fonksiyonu algoritmanın her bir yinelenmesinde daima azalır [43, 44].

3.2.3.3. NeuroSMART algoritması

Çalışmada her bir durum-karar ikilisinin Q faktör değer fonksiyonunun tahmini için çok katmanlı yapay sinir ağları kullanılmıştır. Kullanılan yapay sinir ağında girdi katmanındaki düğümler stok sistemindeki ürün çeşitlerinin mevcut stok düzeyleri, stok sistemine gelen müşteri talebi ve bu durum için alınan karardan oluşmaktadır. Örneğin 4 farklı ürün çeşidinden oluşan bir stok sisteminin yapay sinir ağındaki girdi düğümü sayısı, 6'dır. Ağda, deneme yanılma ile en iyi sonucu veren gizli katman sayısı kullanılmıştır. Çıktı katmanı ise tektir ve girdi olarak verilen durum-karar ikilisinin Q faktör değerini vermektedir.

Bu bölümde büyük ölçekli yarı Markov karar süreci problemlerinin çözümü için SMART ödüllü öğrenme algoritmasından hareketle, durum-karar ikililerinin Q faktör fonksiyonuna çok katmanlı yapay sinir ağıyla yaklaşıldığı NeuroSMART algoritması geliştirilmiştir. Geliştirilen algoritmanın adımları aşağıda verilmiştir. Algoritmada yapay sinir ağından tahmin edilen durum-karar ikilisi değerleri q ile gösterilmiştir.

Adım 1: Yapay sinir ağının başlangıç ağırlıklarını $w(.)$ rassal olarak belirle.

Adım 2: Tüm q faktör değerlerini, simülasyonda algoritmanın yinelenme sayısını, toplam süreyi, toplam maliyeti ve greedy politikanın maliyetini sıfır (0) yap.

- _ $l \in S, u \in A(l)$ için $q(l, u) \leftarrow 0$
- _ $k \leftarrow 0$ (k: zincirdeki sıçrama sayısı)
- _ Toplam süre $\leftarrow 0$
- _ Toplam maliyet $\leftarrow 0$
- _ $\rho \leftarrow 0$ (ρ : greedy politikanın ortalama ödülü)

Sabit öğrenme oranı α , ($0 < \alpha < 1$) belirle. Yeterince büyük k_{enb} (yineleme sayısı) belirle. Sistemin simülasyonu herhangi keyfi durumdan başlatılır.

Adım 3: Mevut durum i olsun. Tüm ($u \in A(i)$) kararları için, yapay sinir ağını kullanarak $q(i, u)$ değerlerini tahmin et.

Adım 4: Rastgele a kararı seç. Denklem 3.31'i kullanarak ϕ 'yi belirle. Karar a 'yı simüle et. Geçilen durum j olsun. Simülatörde a kararı altında i durumundan j durumuna geçildiğinde oluşan anlık maliyet $r(i, a, j)$ 'yi ve aynı geçiş için geçen süre $t(i, a, j)$ 'i hesapla.

$$\phi = \begin{cases} 0, & a \in \arg \min_{u \in A(i)} q(i, u) \text{ ise} \\ 1, & \text{değilse} \end{cases} \quad (3.31)$$

Adım 5a: Yapay sinir ağını kullanarak j durumundaki tüm kararlar için $q(j, b)$, ($b \in A(j)$) değerlerini belirle. Denklem 3.32'yi kullanarak j durumunun o anki q faktör değeri q_{next} 'i hesapla. Denklem 3.33'ü kullanarak (i, a) durum-karar ikilisinin q faktör değeri $q(i, a)$ 'yi güncelle.

$$q_{\text{next}} \leftarrow \arg \min_{b \in A(j)} q(j, b) \quad (3.32)$$

$$q(i, a) \leftarrow (1 - \alpha)q(i, a) + \alpha[r(i, a, j) - \rho t(i, a, j) + q_{\text{next}}] \quad (3.33)$$

Adım 5b: Eğer seçilen a kararı greedy ise ($\phi = 0$ ise) Denklem 3.34'ü kullanarak toplam süreyi ve Denklem 3.35'i kullanarak toplam maliyeti güncelle.

$$\text{Toplam süre} \leftarrow \text{Toplam süre} + t(i, a, j) \quad (3.34)$$

$$\text{Toplam maliyet} \leftarrow \text{Toplam maliyet} + r(i, a, j) \quad (3.35)$$

Adım 5c: Denklem 3.36'yı kullanarak ρ 'yu güncelle.

$$\rho = \frac{\text{Toplam Maliyet}}{\text{Toplam Süre}} \quad (3.36)$$

Adım 6: Eğer $k < k_{\text{enb}}$ ise güncellenen $q(i,a)$ 'yı kullanarak yapay sinir ağı güncelle ve adım 3'e dön. Değilse adım 7'ye geç.

Adım 7: Dur. Öğrenilen politika yapay sinir ağının ağırlıklarında depolanır. Bir durumla ilgili kararı belirlemek için, bu durum için izin verilen kararların yapay sinir ağı çıktıları bulunur.

Adım 6'da yapay sinir ağının ağırlıkları artımlı olarak güncellenir. Algoritmanın hızlandırılması için simülatördeki her adımda ağırlıkların güncellenmesi yerine, belirli sayıda geçişten sonra, ağırlıklar yığın şeklinde güncellenebilir.

BÖLÜM 4. UYGULAMALI ÖRNEKLER

Geliştirilen algoritmanın performansını ve sonuçlarını değerlendirmek için bir dizi uygulamalı örnek tasarlanmıştır. Uygulamalı örneklerde başlıca;

- a) ürün kombinasyonu dağıtım probleminde Q faktör fonksiyonuna yapay sinir ağları ile yaklaşılarak elde edilen çözümün kullanılabilirliği,
- b) orta ve büyük ölçekli stok yenileme ve ürün kombinasyonu dağıtım problemlerinin Q faktör fonksiyonuna yapay sinir ağlarıyla yaklaşılarak çözülmesi ve elde edilen politikaların maliyetlerinin miyopik politikalarla kıyaslanması konuları araştırılmıştır.

Oluşturulan uygulamalı örnekler, çalışmanın ana motivasyon kaynağı olan bankamatiklerden para çekme problemi olarak şekillendirilmiştir. Bölüm 3.1’de geliştirilen algoritma, matlab’de kodlanmış ve uygulamalı örnek 1’in;

- verilen bir banknot kombinasyonu dağıtım politikası altında bankamatikteki banknot çeşitlerinin yaklaşık optimal (S,c,s) stok politikası,
- bankamatikteki banknot çeşitleri için verilen (S,c,s) stok politikası altında SMART ve NeuroSMART ödüllü öğrenme algoritmalarıyla yaklaşık optimal banknot kombinasyonu dağıtım politikası belirlenmiştir.

Uygulamalı örnek 2 ve 3 için ise, yaklaşık optimal (S,c,s) stok yenileme ve banknot kombinasyonu dağıtım politikaları belirlenmiş ve miyopik politikalarla kıyaslanmıştır.

4.1. Uygulamalı Örneklerde Kullanılan Ortak Parametreler

Uygulamalı örneklerde bankamatik stoklarında 4 farklı banknot çeşidi (10 TL, 20 TL, 50 TL ve 100 TL) bulunduğu ve bankamatiğe gelen müşterilerin para taleplerinin olasılık fonksiyonu $f_D(\cdot)$ 'nin Tablo 4.1'de verilen kesikli düzgün dağılıma uyduğu varsayılmıştır.

Tablo 4.1. Müşterilerin bireysel para talebi olasılık dağılımı

d (TL)	10	20	30	40	50	60	70	80	90	100
$f_D(d)$	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10

Müşterilerin bankamatikten para taleplerinin tablo 4.1'de verilen kesikli olasılık fonksiyonuna uyması durumunda n_1 , 10 TL'lik banknotlardan; n_2 , 20 TL'lik banknotlardan; n_3 , 50 TL'lik banknotlardan ve n_4 , 100 TL'lik banknotlardan müşteriye dağıtılan miktarları göstermek üzere, her bir talep için oluşturulabilecek (n_1, n_2, n_3, n_4) kararları Tablo 4.2' de verilmiştir.

Tablo 4.2. Müşteri talebine göre oluşan kararlar

d (TL)	10	20	30	40	50	60	70	80	90	100
Karar [1]	1,0,0,0	0,1,0,0	1,1,0,0	0,2,0,0	0,0,1,0	0,3,0,0	0,1,1,0	0,4,0,0	0,2,1,0	0,0,0,1
Karar [2]		2,0,0,0	3,0,0,0	2,1,0,0	1,2,0,0	1,0,1,0	1,3,0,0	1,1,1,0	1,4,0,0	0,0,2,0
Karar [3]				4,0,0,0	3,1,0,0	2,2,0,0	2,0,1,0	2,3,0,0	2,1,1,0	0,5,0,0
Karar [4]					5,0,0,0	4,1,0,0	3,2,0,0	3,0,1,0	3,3,0,0	1,2,1,0
Karar [5]						6,0,0,0	5,1,0,0	4,2,0,0	4,0,1,0	2,4,0,0
Karar [6]							7,0,0,0	6,1,0,0	5,2,0,0	3,1,1,0
Karar [7]								8,0,0,0	7,1,0,0	4,3,0,0
Karar [8]									9,0,0,0	5,0,1,0
Karar [9]										6,2,0,0
Karar [10]										8,1,0,0
Karar [11]										10,0,0,0

Tasarlanan tüm örnekler için Bölüm 3.1'de geliştirilen algoritmanın ilk adımı olarak bankamatiğin, müşteri taleplerini “en az banknot sayısını içeren kombinasyonla karşılama” miyopik politikasını uyguladığı kabul edilmiştir. Bu durumda bankamatiğin başlangıç banknot dağıtım politikası Tablo 4.3'teki gibidir.

Tablo 4.3. Bankamatik'in başlangıç banknot dağıtım politikası

d (TL)	10	20	30	40	50	60	70	80	90	100
(n_1, n_2, n_3, n_4)	1,0,0,0	0,1,0,0	1,1,0,0	0,2,0,0	0,0,1,0	1,0,1,0	0,1,1,0	1,1,1,0	0,2,1,0	0,0,0,1

Bankamatik'in her bir müşteri talebini en az banknot sayısını içeren kombinasyonla sağlama politikası altında, i 'nci banknot çeşidinin bireysel talebi D_i 'nin dağılımı, Toplam Olasılık Yasası kullanılarak Denklem 4.1'deki gibi bulunur.

$$f_{D_i}(n_i) = \sum_{\forall d} f_{D_i|D}(n_i|d) \cdot f_D(d), \quad d=10,20,\dots ; \quad n_i=0,1,2,\dots \quad (4.1)$$

Stoklardaki 10 TL'lik banknotların belirtilen başlangıç banknot dağıtım politikası altında bireysel ürün talebi olasılık fonksiyonunun hesabı, Denklem 4.2, 4.3, 4.4 ve 4.5'te verilmiştir.

$$f_{D_1}(0) = \sum_{\forall d} P(D_1 = 0|D = d) \cdot P(D = d) \quad (4.2)$$

$$= P(D_1 = 0|D = 10) \cdot P(D = 10) + P(D_1 = 0|D = 20) \cdot P(D = 20) + \dots \\ + P(D_1 = 0|D = 100) \cdot P(D = 100)$$

$$f_{D_1}(0) = 0 \cdot (0,10) + 1 \cdot (0,10) + 0 \cdot (0,10) + 1 \cdot (0,10) + 1 \cdot (0,10) \\ + 0 \cdot (0,10) + 1 \cdot (0,10) + 0 \cdot (0,10) + 1 \cdot (0,10) + 1 \cdot (0,10) \\ = 0,60 \quad (4.3)$$

$$f_{D_1}(1) = \sum_{\forall d} P(D_1 = 1|D = d) \cdot P(D = d) \quad (4.4)$$

$$= P(D_1 = 1|D = 10) \cdot P(D = 10) + P(D_1 = 1|D = 20) \cdot P(D = 20) + \dots \\ + P(D_1 = 1|D = 100) \cdot P(D = 100)$$

$$\begin{aligned}
f_{D_1}(1) &= 1.(0,10) + 0.(0,10) + 1.(0,10) + 0.(0,10) + 0.(0,10) \\
&\quad + 1.(0,10) + 0.(0,10) + 1.(0,10) + 0.(0,10) + 0.(0,10) \\
&= 0,40
\end{aligned} \tag{4.5}$$

$d_i=2,3,\dots$ ve $d=10,20,\dots,100$ için $P(D_1 = d_i | D = d) = 0$ olduğundan, $d_i=2,3,\dots$ için $f_{D_1}(d_i) = 0$ 'dır. 10 TL'lik banknotlar için hesaplanan bireysel talep olasılık fonksiyonu Denklem 4.6'da verilmiştir.

$$f_{D_1}(n_1) = \begin{cases} 0,60 & , \quad n_1 = 0 \\ 0,40 & , \quad n_1 = 1 \\ 0 & , \quad \text{aksi halde} \end{cases} \tag{4.6}$$

Benzer şekilde hesaplanan 20 TL'lik, 50 TL'lik ve 100 TL'lik banknotların belirtilen başlangıç ürün kombinasyonu dağıtım politikası altındaki olasılık fonksiyonları, sırasıyla Denklem 4.7, 4.8, 4.9'da verilmiştir.

$$f_{D_2}(n_2) = \begin{cases} 0,40 & , \quad n_2 = 0 \\ 0,40 & , \quad n_2 = 1 \\ 0,20 & , \quad n_2 = 2 \\ 0 & , \quad \text{aksi halde} \end{cases} \tag{4.7}$$

$$f_{D_3}(n_3) = \begin{cases} 0,50 & , \quad n_3 = 0 \\ 0,50 & , \quad n_3 = 1 \\ 0 & , \quad \text{aksi halde} \end{cases} \tag{4.8}$$

$$f_{D_4}(n_4) = \begin{cases} 0,90 & , \quad n_4 = 0 \\ 0,10 & , \quad n_4 = 1 \\ 0 & , \quad \text{aksi halde} \end{cases} \tag{4.9}$$

Çalışmada bankamatiğe gelen müşterilerin, karar verici tarafından bilindiği varsayılan özdeş bir fayda fonksiyonuna sahip olduğu kabul edilmiştir. Müşterilerin kazandıkları faydalar, para taleplerini karşılamak için bankamatik tarafından verilen banknot kombinasyonlarına göre değişmektedir. Örneklerde kullanılan özdeş fayda fonksiyonu Tablo 4.4'te verilmiştir.

Tablo 4.4. Müşterilerin kesikli ve özdeş fayda fonksiyonu

Talep \ Karar	Fayda									
	10	20	30	40	50	60	70	80	90	100
Karar [1]	100	80	100	60	50	50	70	50	90	30
Karar [2]		100	80	100	100	50	80	100	60	50
Karar [3]				80	90	100	100	90	100	60
Karar [4]					60	80	70	60	80	100
Karar [5]						60	60	70	70	60
Karar [6]							50	50	60	90
Karar [7]								40	50	60
Karar [8]									30	40
Karar [9]										30
Karar [10]										20
Karar [11]										10

Müşterilerin bankamatiğe geliş ortalamaları (λ) çalışmadaki uygulamalı örneklerde birbirinden farklı alınmıştır. Müşterilerin geliş ortalamaları (λ) değiştikçe, her bir banknot çeşidinin eşdeğer birleşik poisson süreci parametreleri de değişmektedir.

4.2. (S,c,s) Stok Politikası Altında SMART ve Neuro SMART Algoritmalarının Performansları

Çalışmanın bu bölümünde ele alınan problemin ürün kombinasyonu dağıtımını alt probleminin çözümünde kullanılan ödüllü öğrenme algoritması SMART ile Q faktör fonksiyonunun çok katmanlı yapay sinir ağlarıyla tahmin edildiği NeuroSMART algoritmasının ürettiği politikaların sonsuz ufukta birim zamandaki ortalama maliyetlerinin birbirinden farklı olup olmadığı araştırılmıştır. Bölüm 3.2.3'te verildiği üzere problemlerin durum-karar sayısı arttıkça, tüm durum-karar ikililerinin Q faktör değerlerinin arama tablolarında depolanması ve gerektiğinde bu tablolardan

güncelleme için kullanılması mümkün değildir. Bu nedenle durum-karar sayısının fazla olduğu problemlerde doğrudan SMART algoritması kullanılamaz. Eğer Q faktör fonksiyonunun yapay sinir ağları ile tahmin edildiği NeuroSMART algoritması SMART algoritmasına yakın performans gösteriyorsa, bu durumda durum-karar sayısının fazla olduğu orta ve büyük ölçekli Markov ve yarı Markov karar süreci problemlerinde NeuroSMART algoritması başarıyla kullanılabilir. Bu nedenle SMART algoritmasıyla çözülebilecek küçük ölçekli bir problem tasarlanmış ve aynı (S,c,s) stok politikası için bu iki algoritmanın ürettiği politikaların birim zamandaki ortalama maliyetleri karşılaştırılmıştır.

4.2.1. Uygulamalı örnek 1

Çalışmada ürün kombinasyonu dağıtım politikasının belirlenmesinde SMART ve NeuroSMART algoritmalarının performanslarının kıyaslanması için durum karar sayısının az olduğu küçük ölçekli bir örnek tasarlanmıştır. Bu örnekteki bankamatik talep süreci, müşteri gelişleri ortalaması $\lambda = 5$ müşteri/saat ve para talepleri Tablo 4.1'de verilen kesikli dağılıma uyan birleşik poisson sürecidir. Bankamatik stoklarında yer alan banknot çeşitlerinin (S,c,s) stok parametrelerinin belirlenebilmesi için, bankamatige müşteri geliş ortalaması dikkate alınarak banknot çeşitlerinin talep sürecinin, eşdeğer birleşik poisson ve eşdeğer poisson süreçlerine dönüştürülmesi gerekir.

Örnek 1'de, bankamatik stoklarında yer alan tüm banknot çeşitleri için stokta bulundurma maliyeti $h=0.01$ TL/banknot/saat; birincil sipariş maliyeti $K=10$ TL/sipariş; ikincil sipariş maliyeti $k=5$ TL/siparişe eklenen banknot çeşidi sayısı; talebin stoktan karşılanamaması maliyeti $w=10$ TL/siparişi stoktan karşılanamayan müşteri sayısı ve müşterinin kaçırdığı faydanın fırsat maliyeti $p=0,5$ TL/kayıp müşteri faydası olarak tanımlanmıştır. Belirlenen stok politikasının tüm banknot çeşitleri için en az 0,90 ikinci tip hizmet düzeyini sağlaması gerektiği ve verilen siparişlerin tedarik süresinin sabit 0,5 saat olduğu kabul edilmiştir.

4.2.1.1. Uygulamalı örnek 1'in birleşik poisson ürün talebi parametreleri

Bankamatığın başlangıç banknot dağıtım politikası ve bankamatığe gelen müşterilerin bireysel para talebi dağılımı kullanılarak bulunan bankamatik stoklarında yer alan her bir banknotun bireysel talep dağılımları Bölüm 4.1'de verilmiştir (Bkz. Tablo 4.1, Denklem 4.6, 4.7, 4.8 ve 4.9). Ancak, Bölüm 2.1.2'de verildiği üzere hesaplanan bu olasılık fonksiyonlarında banknot çeşitlerinin bireysel talebinin sıfır olması olasılıkları da söz konusu olduğu için daha önce hesaplanan banknot bireysel talep olasılıklarının, bireysel talebin sıfır olamadığı eşdeğer bireysel talep olasılık fonksiyonu ve eşdeğer banknot müşteri geliş oranı hesaplanmalıdır. Bu durumda belli bir banknot çeşidi için gelen müşteri talebi süreci Bölüm 2.1.2'de verildiği şekilde ortalaması λ' ve bireysel banknot talep olasılık fonksiyonu $f'_{D_i}(\cdot)$ olan eşdeğer birleşik poisson sürecidir. Örnek 1 için bankamatik stoklarında yer alan 10 TL, 20 TL, 50 TL ve 100 TL banknotlarının eşdeğer birleşik poisson talep dağılımları aşağıda bulunmuştur.

10 TL'lik banknotlar için birleşik poisson talep parametreleri ($i=1$): 10 TL'lik banknot çeşidi için eşdeğer müşteri geliş ortalaması ve eşdeğer bireysel banknot talep dağılımı hesaplamaları sırasıyla Denklem 4.10, 4.11 ve 4.12'de verilmiştir.

$$\lambda'_1 = \lambda(1 - f_{D_1}(0)) = 5.(1 - 0,60) = 2 \text{ müşteri/saat} \quad (4.10)$$

$$f'_{D_1}(1) = f_{D_1}(1) / (1 - f_{D_1}(0)) = 0,40 / 0,40 = 1 \quad (4.11)$$

$$f'_{D_1}(n_1) = \begin{cases} 1, & n_1 = 1 \\ 0, & \text{aksi halde} \end{cases} \quad (4.12)$$

20 TL'lik banknotlar için birleşik poisson talep parametreleri ($i=2$): 20 TL'lik banknot çeşidi için eşdeğer müşteri geliş ortalaması ve eşdeğer bireysel banknot talep dağılımı hesaplamaları sırasıyla Denklem 4.13, 4.14, 4.15 ve 4.16'da verilmiştir.

$$\lambda'_2 = \lambda(1 - f_{D_2}(0)) = 5.(1 - 0,40) = 3 \text{ müşteri/saat} \quad (4.13)$$

$$f'_{D_2}(1) = f_{D_2}(1) / (1 - f_{D_2}(0)) = 0,40 / 0,60 = 2/3 \quad (4.14)$$

$$f'_{D_2}(2) = f_{D_2}(2) / (1 - f_{D_2}(0)) = 0,20 / 0,60 = 1/3 \quad (4.15)$$

$$f_{D_2}(n_2) = \begin{cases} 2/3, & n_2 = 1 \\ 1/3, & n_2 = 2 \\ 0, & \text{aksi halde} \end{cases} \quad (4.16)$$

50 TL'lik banknotlar için birleşik poisson talep parametreleri (i=3): 50 TL'lik banknot çeşidi için eşdeğer müşteri geliş ortalaması ve eşdeğer bireysel banknot talep dağılımı hesaplamaları sırasıyla Denklem 4.17, 4.18 ve 4.19'da verilmiştir.

$$\lambda'_3 = \lambda(1 - f_{D_3}(0)) = 5.(1 - 0,50) = 2,5 \text{ müşteri/saat} \quad (4.17)$$

$$f'_{D_3}(1) = f_{D_3}(1) / (1 - f_{D_3}(0)) = 0,50 / 0,50 = 1 \quad (4.18)$$

$$f_{D_3}(n_3) = \begin{cases} 1, & n_3 = 1 \\ 0, & \text{aksi halde} \end{cases} \quad (4.19)$$

100 TL'lik banknotlar için birleşik poisson talep parametreleri (i=4): 100 TL'lik banknot çeşidi için eşdeğer ortalama ve eşdeğer bireysel banknot talep dağılımı hesaplamaları sırasıyla Denklem 4.20, 4.21 ve 4.22'de verilmiştir.

$$\lambda'_4 = \lambda(1 - f_{D_4}(0)) = 5.(1 - 0,90) = 0,5 \text{ müşteri/saat} \quad (4.20)$$

$$f'_{D_4}(1) = f_{D_4}(1) / (1 - f_{D_4}(0)) = 0,10 / 0,10 = 1 \quad (4.21)$$

$$f_{D_4}(n_4) = \begin{cases} 1, & n_4 = 1 \\ 0, & \text{aksi halde} \end{cases} \quad (4.22)$$

4.2.1.2. Uygulamalı örnek 1'in eşdeğer poisson ürün talebi parametreleri

Bölüm 2.3'te verilen yöntemlerle (S,c,s) stok politikasının belirlenebilmesi için birleşik poisson dağılımına uyan banknot çeşidi talep dağılımlarının, Bölüm 2.1.4'te verildiği gibi Friend [35]'in yaklaşımıyla eşdeğer poisson dağılımına dönüştürülmesi gerekir. Bankamatığın stoklarında yer alan 10 TL, 20 TL, 50 TL ve 100 TL banknot çeşitlerinin eşdeğer poisson parametreleri aşağıda tanımlanmıştır.

10 TL'lik banknotlar için eşdeğer poisson talep parametreleri (i=1): 10 TL'lik banknot çeşidinin eşdeğer poisson talebi parametrelerine ait hesaplamalar Denklem 4.23, 4.24, 4.25, 4.26, 4.27'de verilmiştir.

$$E[D_1] = \sum_{\forall n_1} n_1 \cdot f_{D_1}'(n_1) = 1.1 = 1 \quad (\text{adet 10 TL banknot}) \quad (4.23)$$

$$E[D_1^2] = \sum_{\forall n_1} (n_1)^2 \cdot f_{D_1}'(n_1) = 1^2 \cdot 1 = 1 \quad (\text{adet}^2 \text{ 10 TL banknot}) \quad (4.24)$$

$$U_1 = \frac{E[D_1^2]}{E[D_1]} = \frac{1}{1} = 1 \quad (\text{adet 10 TL banknot}) \quad (4.25)$$

$$\lambda'_{\text{eşdeğer}} = \frac{\lambda_1' \cdot E[D_1]^2}{E[D_1^2]} = \frac{2 \cdot 1^2}{1} = 2 \quad (\text{adet 10 TL banknot/saat}) \quad (4.26)$$

$$v_1' = v_1 \cdot U_1 = 10 \cdot 1 = 10 \quad (\text{TL}) \quad (4.27)$$

20 TL'lik banknotlar için eşdeğer poisson talep parametreleri (i=2): 20 TL'lik banknot çeşidinin eşdeğer poisson talebi parametrelerine ait hesaplamalar Denklem 4.28, 4.29, 4.30, 4.31, 4.32'de verilmiştir.

$$E[D_2] = \sum_{\forall n_2} n_2 \cdot f_{D_2}'(n_2) = 1 \cdot \frac{2}{3} + 2 \cdot \frac{1}{3} = \frac{4}{3} \quad (\text{adet 20 TL banknot}) \quad (4.28)$$

$$E[D_2^2] = \sum_{\forall n_2} (n_2)^2 \cdot f_{D_2}'(n_2) = 1^2 \cdot \frac{2}{3} + 2^2 \cdot \frac{1}{3} = 2 \quad (\text{adet}^2 \text{ 20 TL banknot}) \quad (4.29)$$

$$U_2 = \frac{E[D_2^2]}{E[D_2]} = \frac{2}{\frac{4}{3}} = 1,5 \quad (\text{adet 20 TL banknot}) \quad (4.30)$$

$$\lambda_{\text{eşdeğer}}' = \frac{\lambda_2' \cdot E[D_2]^2}{E[D_2^2]} = \frac{3 \cdot \left(\frac{4}{3}\right)^2}{2} = \frac{8}{3} \quad (\text{adet 20 TL banknot/saat}) \quad (4.31)$$

$$v_2' = v_2 \cdot U_2 = 20 \cdot 1,5 = 30 \quad (\text{TL}) \quad (4.32)$$

50 TL'lik banknotlar için eşdeğer poisson talep parametreleri (i=3): 50 TL'lik banknot çeşidinin eşdeğer poisson talebi parametrelerine ait hesaplamalar Denklem 4.33, 4.34, 4.35, 4.36, 4.37'de verilmiştir.

$$E[D_3] = \sum_{\forall n_3} n_3 \cdot f_{D_3}'(n_3) = 1 \cdot 1 = 1 \quad (\text{adet 50 TL banknot}) \quad (4.33)$$

$$E[D_3^2] = \sum_{\forall n_3} (n_3)^2 \cdot f_{D_3}'(n_3) = 1^2 \cdot 1 = 1 \quad (\text{adet}^2 \text{ 50 TL banknot}) \quad (4.34)$$

$$U_3 = \frac{E[D_3^2]}{E[D_3]} = \frac{1}{1} = 1 \quad (\text{adet 50 TL banknot}) \quad (4.35)$$

$$\lambda_{\text{eşdeğer}}' = \frac{\lambda_3' \cdot E[D_3]^2}{E[D_3^2]} = \frac{2,5 \cdot 1^2}{1} = 2,5 \quad (\text{adet 50 TL banknot/saat}) \quad (4.36)$$

$$v_3' = v_3 \cdot U_3 = 50 \cdot 1 = 50 \quad (\text{TL}) \quad (4.37)$$

100 TL'lik banknotlar için eşdeğer poisson talep parametreleri (i=4): 100 TL'lik banknot çeşidinin eşdeğer poisson talebi parametrelerine ait hesaplamalar Denklem 4.38, 4.39, 4.40, 4.41, 4.42'de verilmiştir.

$$E[D_4] = \sum_{\forall n_4} n_4 \cdot f_{D_4}'(n_4) = 1 \cdot 1 = 1 \quad (\text{adet 100 TL banknot}) \quad (4.38)$$

$$E[D_4^2] = \sum_{\forall n_4} (n_4)^2 \cdot f_{D_4}'(n_4) = 1^2 \cdot 1 = 1 \quad (\text{adet}^2 \text{ 100 TL banknot}) \quad (4.39)$$

$$U_4 = \frac{E[D_4^2]}{E[D_4]} = \frac{1}{1} = 1 \quad (\text{adet 100 TL banknot}) \quad (4.40)$$

$$\lambda_{\text{eşdeğer}}' = \frac{\lambda_4' \cdot E[D_4]^2}{E[D_4^2]} = \frac{0,5 \cdot 1^2}{1} = 0,5 \quad (\text{adet 100 TL banknot/saat}) \quad (4.41)$$

$$v_4' = v_4 \cdot U_4 = 100 \cdot 1 = 100 \quad (\text{TL}) \quad (4.42)$$

4.2.1.3. Uygulamalı örnek 1'de SMART ve NeuroSMART algoritmalarının çözümlerinin karşılaştırılması

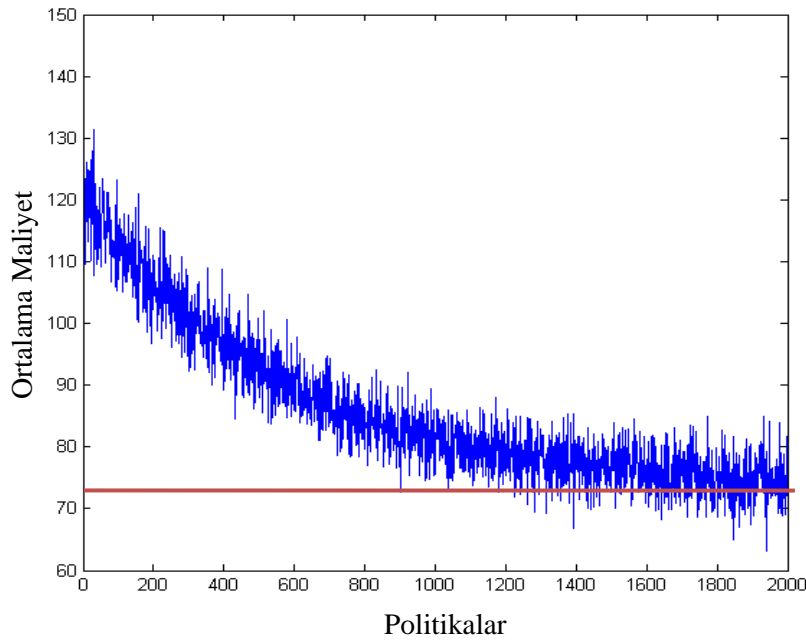
Uygulamalı örnek 1 için bankamatığın “en az banknot sayısını içeren kombinasyonu dağıtma” miyopik politikası altında bulunan (S,c,s) eşgüdümlü stok politikası, Tablo 4.5'te verilmiştir.

Tablo 4.5. Örnek 1'in (S,c,s) stok kontrol politikası parametreleri

Stok parametreleri	S _i	c _i	S _i
Banknot çeşitleri			
10 TL (i=1)	12	4	0
20 TL (i=2)	11	5	2
50 TL (i=3)	8	3	1
100 TL (i=4)	3	2	0

Tablo 4.2’de verilen olası taleplerdeki karar sayıları ve Tablo 4.5’teki stokların tamamlanacağı üst sınır düzeyleri (12, 11, 8, 3) dikkate alındığında, bu çok küçük ölçekli problem için bile durum-karar sayısının 155 232 olduğu görülmektedir.

Tablo 4.5’te verilen (S,c,s) stok yenileme politikası sabit tutularak SMART ödüllü öğrenme algoritmasıyla elde edilen politikaların birim zamandaki ortalama maliyet yakınsama grafiği Şekil 4.1’de verilmiştir.

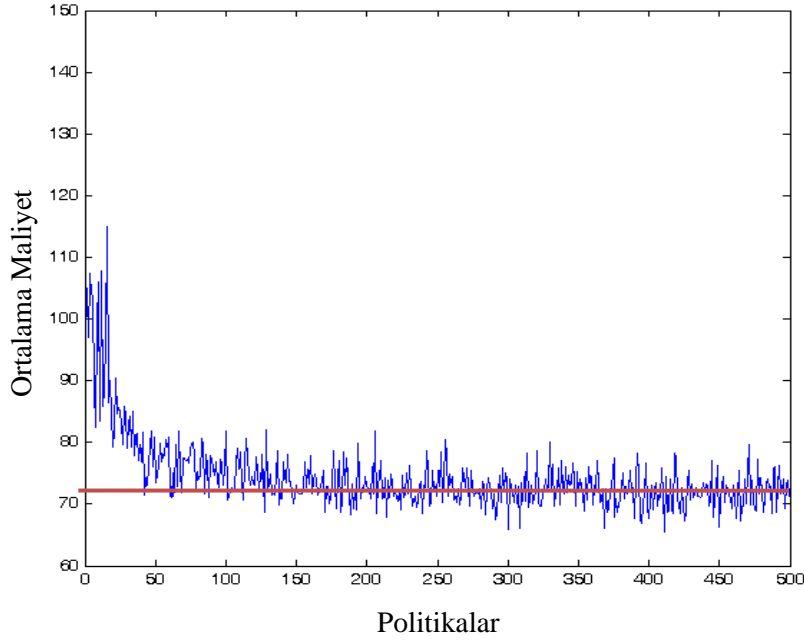


Şekil 4.1. Örnek 1’in SMART algoritması yakınsama grafiği

Örnek 1’in çözümünde SMART algoritmasındaki yineleme sayısı olarak 200.000 müşteri alınmış ve karar seçiminde keşif (exploratory) stratejisi uygulanmıştır. SMART algoritmasında simülatöre gelen her 100 müşteride bir, o zamana kadar üretilen politikanın birim zamandaki ortalama maliyeti hesaplanmıştır. Şekil 4.1’de görüldüğü gibi üretilen politikaların maliyetleri yaklaşık olarak 1200’ncü politika ya da diğer bir deyişle 120.000’nci müşteriden itibaren yaklaşık olarak 72 değerine yakınsamaktadır.

Tablo 4.5’te verilen (S,c,s) stok yenileme politikaları sabit tutularak, Q faktör fonksiyonunun 20 gizli katmanlı yapay sinir ağıyla tahmin edildiği NeuroSMART

ödüllü öğrenme algoritması kullanılmasıyla elde edilen politikaların birim zamandaki ortalama maliyet yakınsama grafiği, Şekil 4.2'deki gibi elde edilmiştir.



Şekil 4.2. Örnek 1'in 20 gizli katmanlı NeuroSMART algoritması yakınsama grafiği

Örnek 1'in çözümünde kullanılan NeuroSMART algoritmasında yineleme sayısı olarak 50.000 müşteri kullanılmış ve karar seçimi için yine keşif (exploratory) stratejisi uygulanmıştır. Yapay sinir ağının başlangıç ağırlıklarının belirlenmesi için girdi olarak başlangıçta rassal olarak üretilen Q faktör değerleri kullanılmış, sonrasında simülörde üretildikçe Q faktörler girdi olarak kullanılıp, yapay sinir ağının ağırlıkları eğitilmiştir. Simülörde her 100 müşteride bir yapay sinir ağı yeniden eğitilmiş ve ağı ağırlıklarında tanımlanmış o ana kadarki politikanın birim zamandaki ortalama maliyeti hesaplanmıştır. Şekil 4.2'de görüldüğü üzere NeuroSMART algoritmasıyla üretilen politikaların maliyetleri çok hızlı bir biçimde yaklaşık olarak 72 değerine yakınsamıştır.

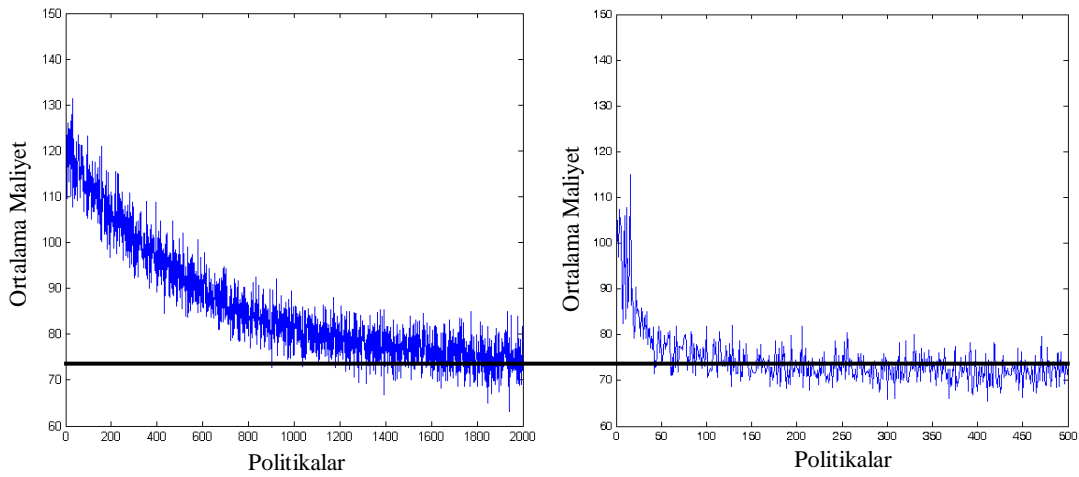
Çalışmada NeuroSMART algoritmasında Q faktör fonksiyonuna yapay sinir ağlarıyla yaklaşılması için MATLAB 2008b Neural Fitting Toolbox kullanılmış ve yapay sinir ağı için yazılan matlab kodu, NeuroSMART algoritması kodunun içine gömülmüştür. Çok katmanlı yapay sinir ağının gizli katmanlarında tan-sigmoid transfer fonksiyonu, çıkış fonksiyonunda ise purelin doğrusal transfer fonksiyonu

kullanılmıştır. Yapay sinir ağının eğitimi için Levenberg-Marquardt backpropagation algoritması seçilmiştir. Yapay sinir ağının her eğitiminde simülatörden elde edilen verilerden rassal olarak % 60'ı eğitimi seti , % 20'si geçerlilik seti ve % 20'si test seti olarak alınmıştır. Eğitim setinde yer alan veriler eğitim süresince ağda kullanılmış ve gradyen hesaplanarak ortalama hata kareleri (MSE) performans ölçütüne göre ağırlıkları ve biaslar ayarlanmıştır.

Eğitim süreci boyunca geçerlilik setinin hataları gözlenir ve geçerlilik setinde yer alan veriler, ağı genelleştirilmesinin ne kadar geçerli olduğunun ölçülmesinde kullanılır. Başlangıçtan itibaren ağı eğitildikçe eğitim setindeki hatalar ve buna bağlı olarak geçerlilik setindeki hatalar gittikçe azalır. Bununla birlikte ağı, eğitim setindeki verileri ezberlemeye başladığında (over fitting) eğitim setindeki hatalar git gide azalmaya devam ettiği halde, geçerlilik setindeki hatalar giderek artmaya başlar. Bu nedenle geçerlilik setindeki hatalar belli bir yinleme sayısında arka arkaya artarsa, bu durumda ağı eğitim seti verilerini ezberlemeye başladığı ve bunun sonucunda fonksiyonu yanlış bir şekilde aşırı uydurduğu düşünülerek eğitim sonlandırılır ve en küçük geçerlilik seti hatasındaki ağırlıklara ve biaslara dönülür. Test verileri, ağı eğitimi tamamlanıp ağırlıklar ayarlanmadan önce ağa gösterilmez ve ağı ürettiği sonuçların ne kadar genelleştirilebileceğinin bağımsız testleri için kullanılır. Eğer test setindeki hatalar geçerlilik setindeki hatalardan belirgin bir şekilde farklı yinlemelerde en küçük değerine ulaşırsa, bu durum verilerin eğitim setine, geçerlilik setine ve test setine rassal olarak ayrılmasında problem olduğunu gösterir.

Yapılan çalışmada yapay sinir ağının girdi verisi olarak durum-karar ikilileri (x_1, x_2, x_3, x_4, d,a), ağı çıktısı olarak ise bu durum-karar ikililerinin Q faktör değerleri kullanılmıştır.

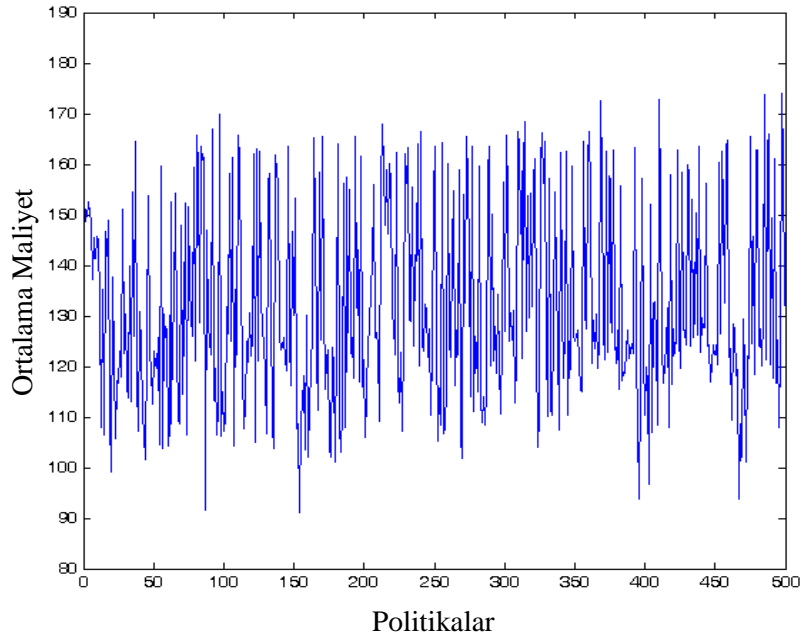
Örnek 1'in SMART ve NeuroSMART algoritmalarıyla elde edilmiş yakınsama grafikleri Şekil 4.3'te kıyaslanmıştır. Şekilden görüldüğü üzere 20 gizli katmanlı NeuroSMART algoritmasının ürettiği politikaların birim zaman başına ortalama maliyeti ile SMART algoritmasının ürettiği politikaların birim zaman başına ortalama maliyeti hemen hemen aynıdır.



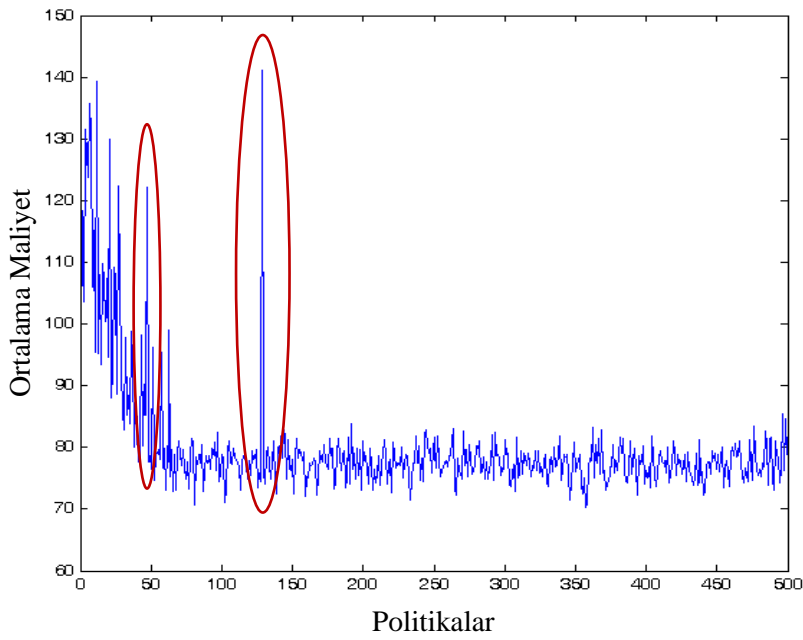
Şekil 4.3 Örnek 1'in SMART ve NeuroSMART algoritmaları yakınsama grafikleri

Yapay sinir ağının yapısındaki gizli katman sayısı, yapay sinir ağının performansını önemli derecede etkileyen bir etmendir. Gizli katman sayısı arttıkça ağın eğitim süresi uzadığı için algoritmanın performansı süre olarak kötüleşmekte, ancak tahmin olarak iyileşmektedir. Bu nedenle tahmin kalitesi açısından yakın sonuçlar verdiği sürece düşük gizli katman sayılarının seçilmesi mantıklıdır. Örnek 1, gizli katman sayısı 2, 5, 10 verilerek yeniden çözülmüştür.

Şekil 4.4'te iki gizli katmana sahip yapay sinir ağıyla Q faktör fonksiyonunun tahmin edildiği NeuroSMART algoritması sonuçları verilmiştir. Şekilden görüldüğü üzere iki gizli katmana sahip yapay sinir ağıyla Q faktör fonksiyonunu tahmin ederek uygulanan NeuroSMART algoritması sonucunda elde edilen politikaların maliyetleri çok geniş bir aralıkta (100-170) salınmakta ve bir limit maliyete yakınsamamaktadır. Bu nedenle elde edilen çözüm kullanılabilir değildir. Farklı ölçekte problemlerde de yapay sinir ağında iki gizli katman kullanıldığı durumlarda benzer sonuçlar gözlenmiştir. Sonuç olarak, NeuroSMART algoritmasının kullanımında gizli katman sayısının iki olarak belirlenmesi uygun değildir.



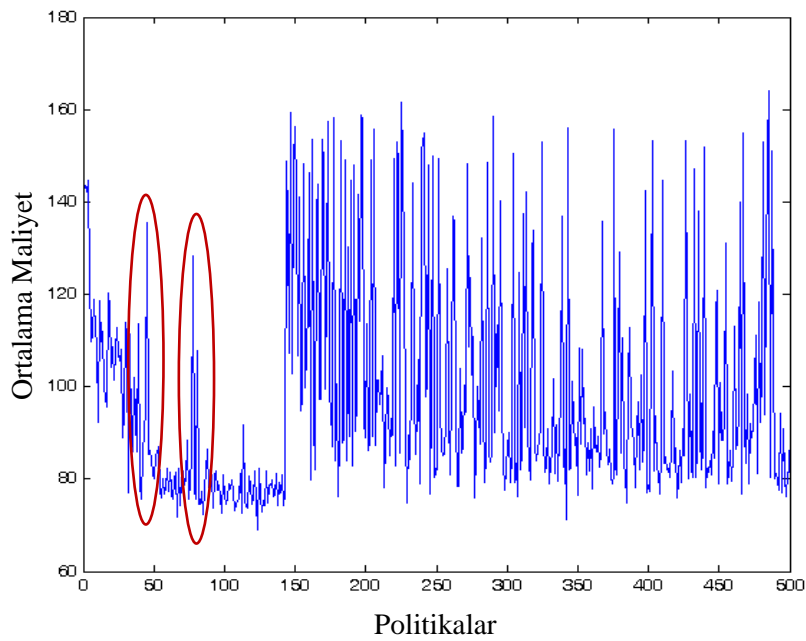
Şekil 4.4. Örnek 1'in 2 gizli katmanlı NeuroSMART algoritması yakınsama grafiği



Şekil 4.5. Örnek 1'in 5 gizli katmanlı NeuroSMART algoritması yakınsama grafiği

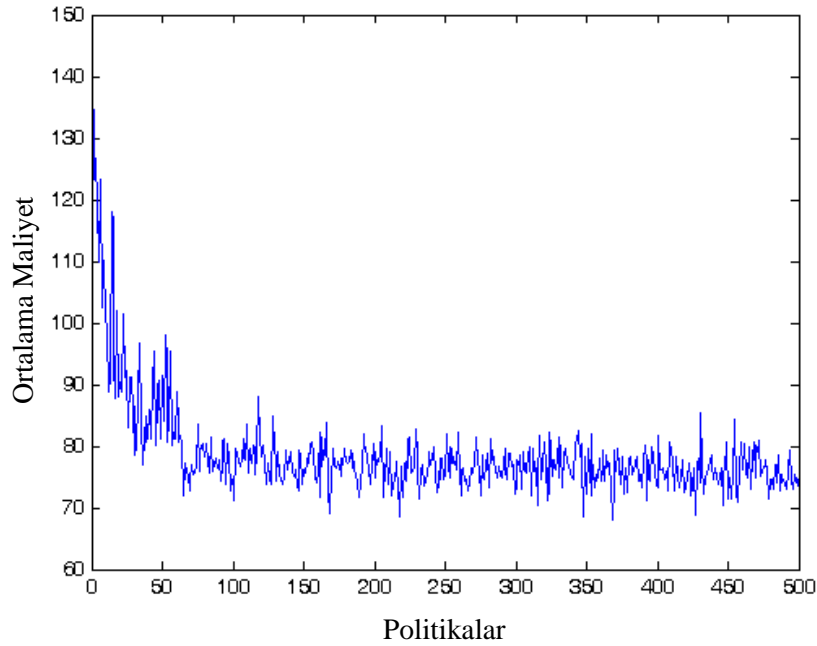
Şekil 4.5'te beş gizli katmana sahip yapay sinir ağıyla Q faktör fonksiyonunun tahmin edildiği NeuroSMART algoritması sonuçları verilmiştir. Şekil 4.5'te görüldüğü üzere 5 gizli katmana sahip yapay sinir ağıyla Q faktör fonksiyonunu

tahmin ederek uygulanan NeuroSMART algoritması sonucunda elde edilen politikaların ortalama maliyetleri, 50'nci yinelemeden sonra 75-80 bandında yakınsamaktadır. Ancak, Şekil 4.5'te elips içine alınan 47'nci ve 129'ncü politikaların ortalama maliyetleri genel salınımdan çok büyük sapma göstermiştir. Her ne kadar simulasyonun ilerleyen yinelemelerindeki politikaların ortalama maliyet hesaplamalarında yakınsama sağlansa da, 47 ve 129 politikalarının ortalama maliyetlerindeki büyük sapmalar, Q faktör fonksiyonun yapay sinir ağıyla tahmininde bu iki noktada yakınsamadan uzaklaşabileceğini göstermektedir.



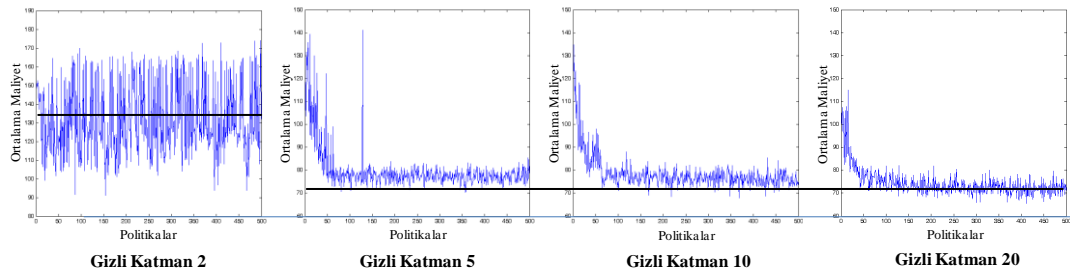
Şekil 4.6. Örnek 1'in 5 gizli katmanlı NeuroSMART algoritması yakınsama grafiği (tekrar çözüm)

Aynı örnek tekrar çözüldüğünde Şekil 4.6'da görüldüğü üzere bu sefer yakınsama sağlanamamıştır. Şekilde elips içine alınmış 45'nci ve 79'ncü politikalarda ortalama maliyetler genel salınımdan aşırı sapma göstermiş ve 145'nci politikadan sonra Şekil 4.5'teki çözümün aksine, yakınsamaktan uzaklaşmıştır. Bu nedenle gizli katman sayısının 5 olarak belirlenmesi durumunda algoritma gürbüz (robust) değildir.



Şekil 4.7. Örnek 1'in 10 gizli katmanlı NeuroSMART algoritması yakınsama grafiği

Şekil 4.7'de 10 gizli katmana sahip yapay sinir ağıyla Q faktör fonksiyonun tahmin edildiği NeuroSMART algoritması sonuçları verilmiştir. Şekilden görüldüğü üzere 50'nci politika hesaplamasından sonra elde edilen politikaların ortalama maliyetleri 75-80 bandında yakınsamakta ve genel salınımdan büyük sapmalar gözlenmemektedir. 10 gizli katmanlı yapay sinir ağı yaklaşımlı NeuroSMART algoritması, verilen örnek için oldukça gürbüzdür.



Şekil 4.8. Örnek 1'in 2, 5, 10, 20 gizli katmanlı NeuroSMART algoritması yakınsama grafikleri

Şekil 4.8'de gizli katman sayısının 2, 5, 10 ve 20 olarak belirlendiği yapay sinir ağlarıyla Q faktör fonksiyonun tahmin edildiği örneklerin ortalama maliyetlerini gösteren grafikler birlikte verilmiştir. Şekilden görüldüğü üzere gizli katman

sayısının 10 ve 20 olarak belirlendiği durumlarda çok iyi sonuçlar alınmış, 2 olarak belirlendiğinde ortalama maliyetlerde yakınsama sağlanamamış ve 5 olarak belirlendiğinde ise yakınsama sağlanmasına rağmen fonksiyon yaklaşımı güvenilir değildir.

4.3. Ürün Kombinasyonu Dağıtım ve Stok Yenileme Probleminin Çözümü

Bu bölümde farklı ölçeklerde iki uygulamalı örnek tasarlanmış ve geliştirilen algoritmayla elde edilen politikaların birim zamandaki ortalama maliyetleri, müşterilere en az banknot sayısını içeren kombinasyonunu dağıtma (miyopik politika 1) ve müşterilere en fazla faydayı sağlayan banknot kombinasyonunu dağıtma (miyopik politika 2) politikalarının birim zamandaki ortalama maliyetleri ile kıyaslanmıştır.

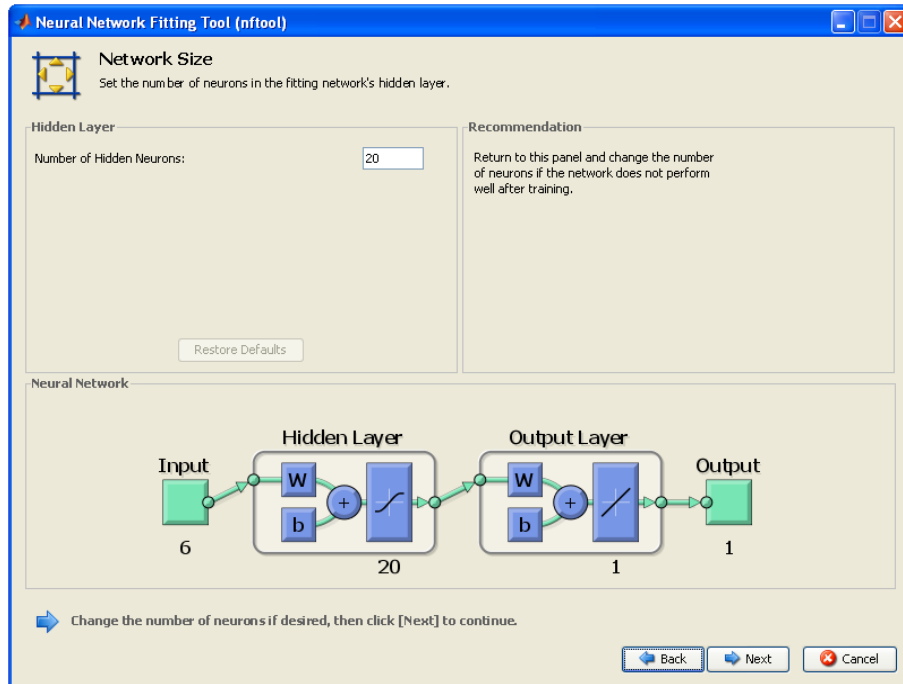
Bu bölümde verilen örnekler için müşterilerin bankamatikten bireysel para taleplerinin kesikli olasılık fonksiyonu, müşterilerin bankamatikten para taleplerine göre oluşan kararlar, algoritmanın ilk adımında kullanılan bankamatiğin başlangıç banknot dağıtım politikası, müşterilerin özdeş kesikli fayda fonksiyonu Bölüm 4.1'dekilerle aynıdır (Bkz. Tablo 4.1, 4.2, 4.3 ve 4.4). Bunun sonucu olarak bankamatikten para taleplerinin dağılımına ve bankamatiğin başlangıç banknot dağıtım politikasına göre hesaplanan banknot çeşitlerinin bireysel talep olasılık fonksiyonları da Bölüm 4.1 ile aynıdır (Bkz. Denklem 4.6, 4.7, 4.8 ve 4.9).

Bölüm 4.2'deki, bir (S,c,s) stok politikası altında ürün kombinasyonu dağıtım politikasının belirlenmesinde SMART ve NeuroSMART algoritmalarının performanslarının kıyaslanmasında elde edilen sonuçlar göz önünde bulundurularak, optimale yakın ürün dağıtım ve stok yenileme politikasını belirlemek için geliştirilen ve adımları Bölüm 3.1 'de verilen algoritmayla ürün kombinasyonu politikasının belirlenmesinde NeuroSMART algoritması kullanılmıştır.

Bu bölümde yer alan her iki örnek için de NeuroSMART algoritmasında yineleme sayısı olarak 100.000 müşteri kullanılmış ve karar seçimi için yine keşif (exploratory) stratejisi uygulanmıştır. Yapay sinir ağının başlangıç ağırlıklarının

belirlenmesi için girdi olarak yine başlangıçta rassal olarak üretilen Q faktör değerleri kullanılmış, sonrasında simülörde üretildikçe Q faktörler girdi olarak kullanılıp, yapay sinir ağı eğitilmiştir. Simülörde her 200 müşteri de bir yapay sinir ağı yeniden eğitilip, ağın ağırlıklarında tanımlanmış ve o ana kadarki politikanın birim zamandaki ortalama maliyeti hesaplanmıştır.

Bölüm 4.2’de olduğu gibi bu bölümde yer alan sayısal örnekler için de NeuroSMART algoritmasının içinde yer alan Q faktör fonksiyonuna yapay sinir ağlarıyla yaklaşılması için MATLAB 2008b Neural Fitting Toolbox kullanılmış ve yapay sinir ağı için yazılan matlab kodu, NeuroSMART algoritma kodunun içine gömülmüştür. Şekil 4.9’da verildiği üzere yapay sinir ağı 1 girdi katmanı, 20 gizli katman, 1 çıktı katmanından oluşturulmuş ve gizli katmanlarda tan-sigmoid transfer fonksiyonu, çıkış fonksiyonunda ise purelin doğrusal transfer fonksiyonu kullanılmıştır. Yapay sinir ağının eğitimi için Levenberg-Marquardt backpropagation algoritması seçilmiştir. Yapay sinir ağının her eğitiminde simülörden elde edilen verilerden rassal olarak % 60’ı eğitimi seti , % 20’si geçerlilik seti ve % 20’si test seti olarak alınmıştır. Yapay sinir ağının girdisi verisi durum-karar ikilileri (x_1, x_2, x_3, x_4, d, a) ve ağıın çıktısı ise bu durum-karar ikilisinin Q faktör değeridir.



Şekil 4.9. Örnek 2 ve 3’te kullanılan yapay sinir ağı mimarisi

4.3.1. Uygulamalı örnek 2 (orta ölçekli problem)

Örnek 2’de müşterilerin bankamatikten para talebi sürecinin, geliş ortalaması $\lambda = 5$ müşteri/saat ve müşterilerin bireysel para talebinin Tablo 4.1’de verilen kesikli dağılıma uyduğu birleşik poisson süreci olduğu kabul edilmiştir. Örnek 2’de bankamatığın stoklarında 4 farklı banknot çeşidi (10 TL, 20 TL, 50 TL ve 100 TL) bulunduğu ve tüm banknot çeşitleri için stokta (h) bulundurma maliyeti 0,001 TL/banknot/saat; (K) birincil sipariş maliyeti 10 TL/sipariş, (k) ikincil sipariş maliyeti 5 TL/siparişe eklenen banknot çeşidi sayısı, (w) talebin stoktan karşılanamaması maliyeti 10 TL/siparişi stoktan karşılanamayan müşteri sayısı ve (p) müşterinin kaçırdığı faydanın fırsat maliyeti 0,01 TL/kayıp müşteri faydası olarak tanımlanmıştır. Belirlenen stok politikasının tüm banknot çeşitleri için en az 0,90 ikinci tip hizmet düzeyini karşılaması gerektiği ve verilen siparişlerin tedarik süresinin sabit 0,5 saat olduğu kabul edilmiştir.

Bankamatik stoklarında yer alan her banknot çeşidinin başlangıç (S,c,s) stok parametrelerinin belirlenebilmesi için bankamatiğe gelen müşteri geliş ortalaması dikkate alınarak banknotların ürün talebi sürecinin, eşdeğer birleşik poisson ve eşdeğer poisson süreçlerine dönüştürülmesi gerekir. 4 banknot çeşidi için hesaplanan birleşik poisson ürün talebi ve eşdeğer poisson ürün talebi parametreleri sırasıyla Tablo 4.6 ve 4.7’de verilmiştir.

Tablo 4.6. Örnek 2’nin başlangıç eşdeğer birleşik poisson banknot talebi parametreleri

Banknotlar (TL) (i=1,2,3,4)	Eşdeğer Müşteri Geliş Ortalaması (λ'_i)	Eşdeğer Bireysel Talep Olasılık Fonksiyonu ($f'_{D_i}(\cdot)$)
10	2	$f_{D_1}(n_1) = \begin{cases} 1 & , n_1 = 1 \\ 0 & , \text{diğer} \end{cases}$
20	3	$f_{D_2}(n_2) = \begin{cases} 2/3 & , n_2 = 1 \\ 1/3 & , n_2 = 2 \\ 0 & , \text{diğer} \end{cases}$
50	2,5	$f_{D_3}(n_3) = \begin{cases} 1 & , n_3 = 1 \\ 0 & , \text{diğer} \end{cases}$
100	0,5	$f_{D_4}(n_4) = \begin{cases} 1 & , n_4 = 1 \\ 0 & , \text{diğer} \end{cases}$

Tablo 4.7. Örnek 2'nin başlangıç eşdeğer poisson banknot talebi parametreleri

Banknotlar (TL) (i=1,2,3,4)	Eşdeğer Banknot Müşteri Geliş Ortalaması ($\lambda'_{\text{eşdeğer}}$)	Eşdeğer Bireysel Sabit Talep (U_i)	Eşdeğer Banknot Fiyatı (v'_i)
10	2	1	10
20	8/3	1,5	30
50	2,5	1	50
100	0,5	1	100

Bölüm 3.1.1'de verilen ürün kombinasyon ve stok politikasını belirleyen algoritmanın ilk adımı için başlangıç banknot dağıtım politikası olarak, müşterilere en az banknot sayısını içeren kombinasyonun dağıtım politikası seçildiğinde elde edilen (S,c,s) stok politikaları Tablo 4.8'de verilmiştir.

Tablo 4.8. Örnek 2'nin başlangıç stok yenileme politikası

Stok parametreleri Banknot çeşitleri	S_i	c_i	s_i
10 TL (i=1)	101	30	0
20 TL (i=2)	78	28	0
50 TL (i=3)	50	18	0
100 TL (i=4)	37	14	0

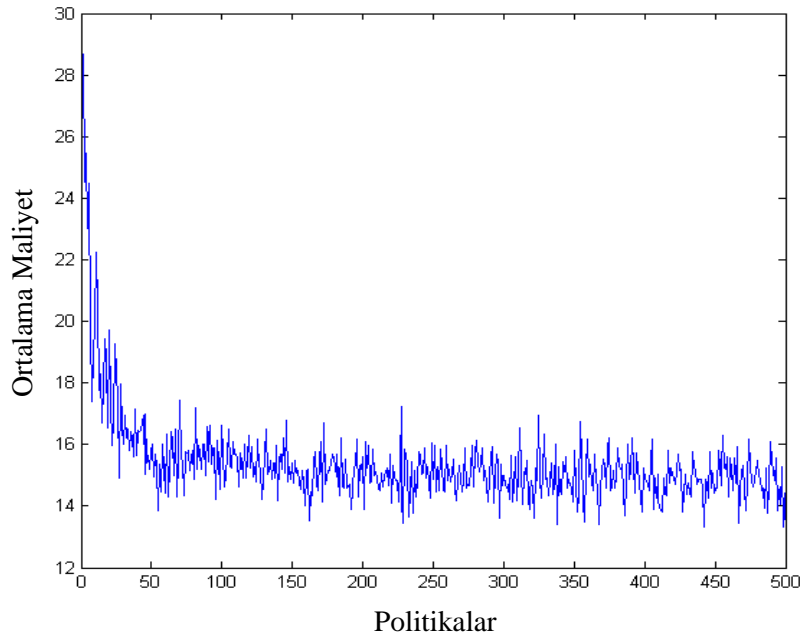
Algoritmanın uygulanmasıyla elde edilen stok yenileme politikası ise Tablo 4.9'da verilmiştir.

Tablo 4.9. Örnek 2'nin yaklaşık optimal stok yenileme politikası

Stok parametreleri Banknot çeşitleri	S_i	c_i	s_i
10 TL (i=1)	108	37	0
20 TL (i=2)	80	30	0
50 TL (i=3)	49	17	0
100 TL (i=4)	36	13	0

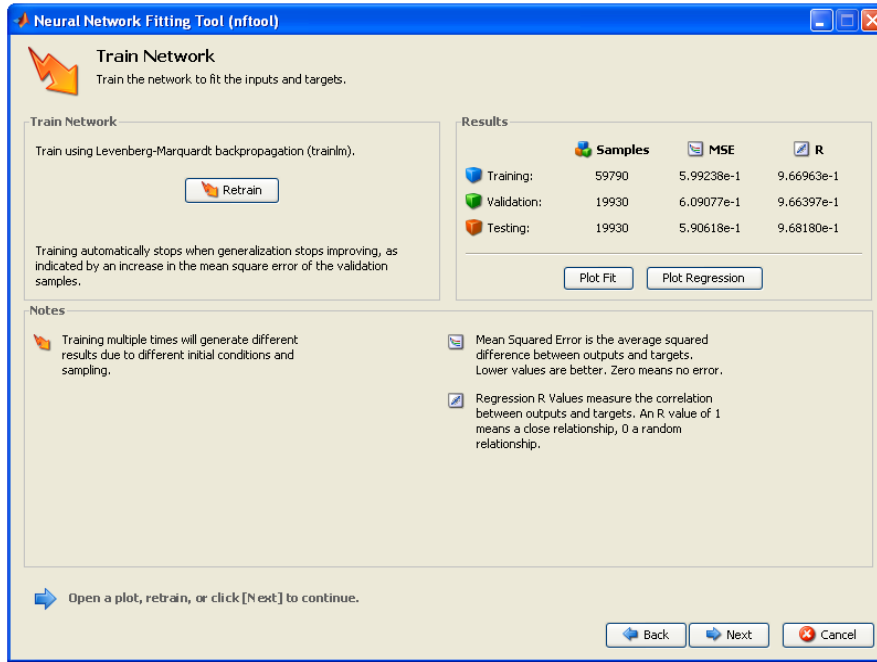
Tablo 4.2’de verilen olası taleplerdeki karar sayıları ve yaklaşık optimal stok politikasındaki stokların tamamlanacağı üst sınır düzeyleri (108, 80, 49, 36) dikkate alındığında ele alınan problemin durum-karar sayısının 746.807.040 olduğu görülmektedir. Durum-karar sayısının bu kadar büyük olduğu bir problemde Q-faktör değerlerinin arama tablolarına depolanıp, güncellenmesi mümkün değildir. Bu nedenle Q faktör fonksiyonu, yapay sinir ağları ile tahmin edilmiştir.

Algoritmanın uygulanmasıyla belirlenen yaklaşık optimal banknot kombinasyonu dağıtım ve (S,c,s) stok politikasının birim zamandaki ortalama maliyet yakınsama grafiği Şekil 4.10’da verilmiştir.



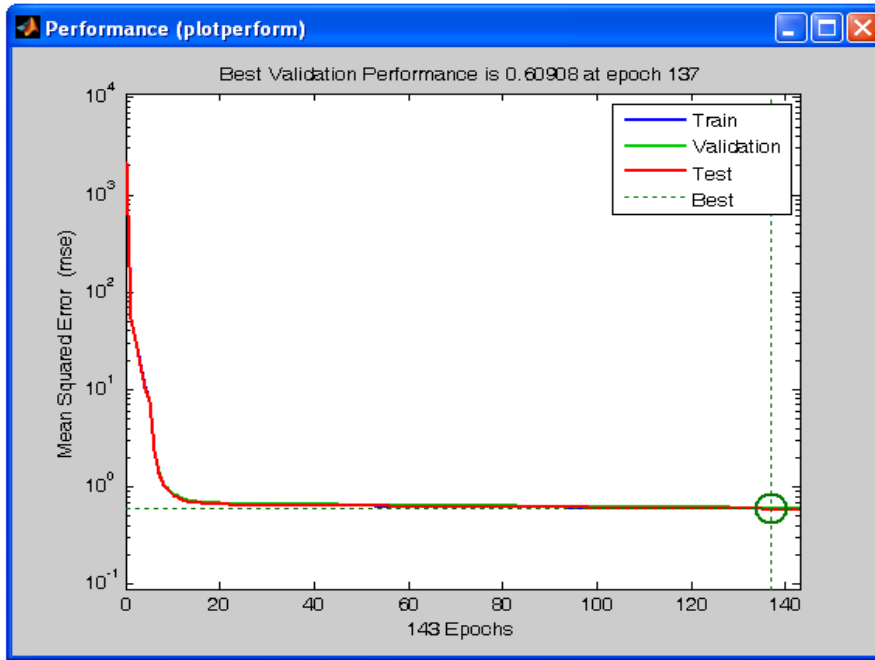
Şekil 4.10. Örnek 2'nin yaklaşık optimal banknot kombinasyonu dağıtım ve stok yenileme politikası yakınsama grafiği

Algoritma, yaklaşık optimal banknot kombinasyonu dağıtım ve stok yenileme politikasını 85 saatte bulmuştur. Bankamatik için elde edilen politikanın 2.tip hizmet düzeyi, simülatörde 0,9878 olarak hesaplanmıştır.



Şekil 4.11. Örnek 2'deki yapay sinir ağının ortalama hata kareleri ve korelasyon katsayıları

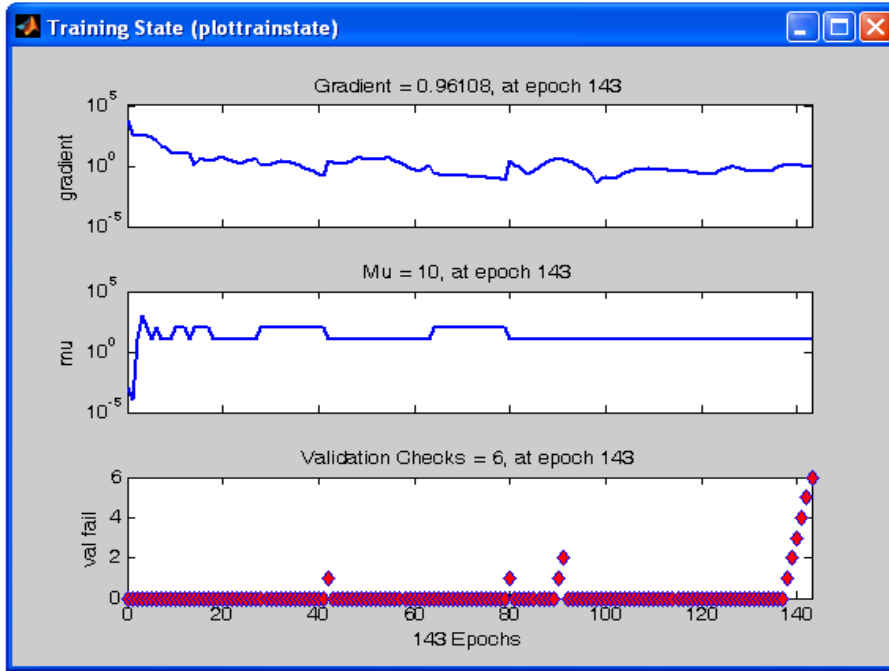
Algoritmanın durduğu andaki son yapay sinir ağının ortalama hata karesi MSE (mean square error) ve oluşan regresyon denkleminin R korelasyon katsayısı (coefficient of correlation) Şekil 4.11'de görülmektedir. Şekilde görüldüğü üzere ortalama hata kareleri oldukça küçüktür ve eğitim seti için 0,529938, geçerlilik seti için 0,609077 ve test seti için 0,590618 olarak gerçekleşmiştir. Hedefler ve çıktılar arasındaki regresyonun korelasyon katsayıları ise eğitim seti için 0,966963, geçerlilik seti için 0,966397 ve test seti için 0,968180 olarak gerçekleşmiştir.



Şekil 4.12. Örnek 2'deki yapay sinir ağının performans grafiği

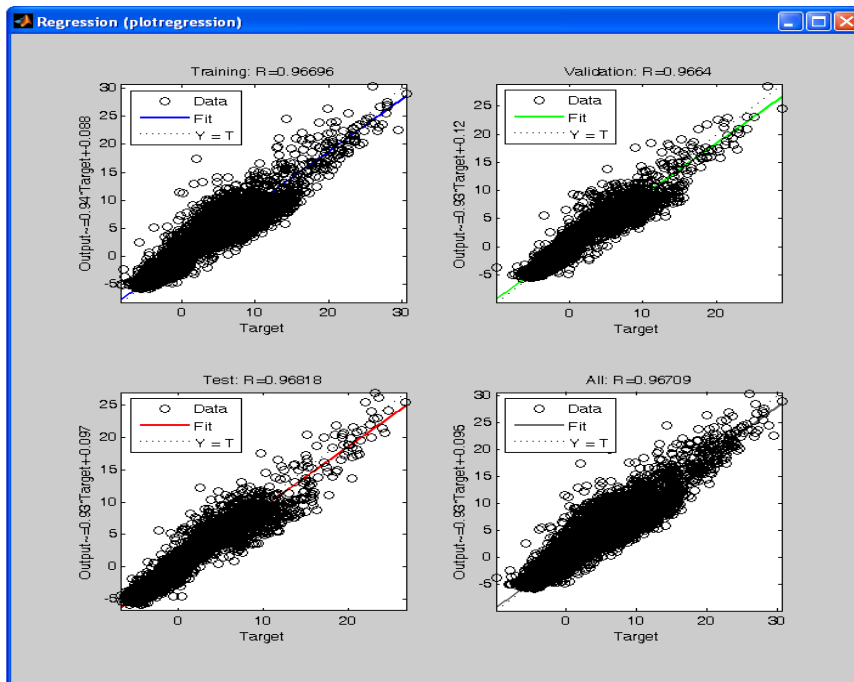
Şekil 4.12'de verilen yapay sinir ağının performans grafiği, yapay sinir ağının performans değerinin gelişimini göstermektedir. Şekildeki performans grafiği incelendiğinde ortalama hata karelerinin (MSE) eğitim algoritmasının yinelemeleri ilerledikçe hızlı bir şekilde 1000'li değerlerden 0,60'lara kadar azaldığı görülmektedir. Diğer bir deyişle performans grafiği, yapay sinir ağının eğitildiğini göstermektedir.

Şekil 4.13'te verilen eğitim durumu grafiğinde, eğitim setinden hesaplanan gradyan değeri, Levenberg-Marquardt eğitim algoritmasında ağırlıkların güncellenmesinde kullanılan μ sabiti ve geçerlilik hatasında artma olup olmadığının yinelemelere göre değişimleri gösterilmiştir. Şekilden görüldüğü üzere 137'nci yinelemede geçerlilik hatası artmaya başlamış ve arka arkaya 6 yinelemede bu artış sürmüştür. Bu nedenle 137'nci yinelemeden itibaren ağın eğitim verilerini ezberlemeye başladığı kararına varılarak, 137'nci yinelemede hesaplanan ağ ağırlıklarına dönülmüştür.



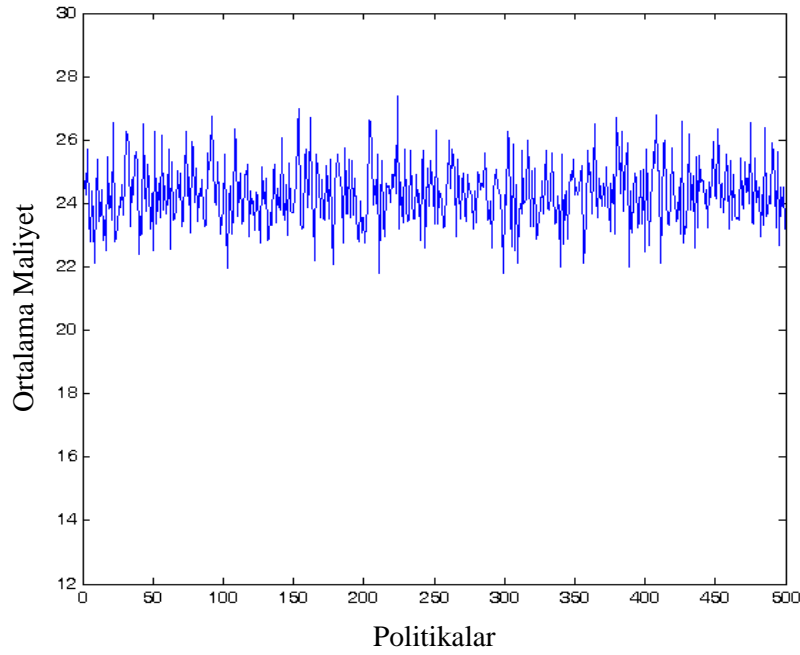
Şekil 4.13.Örnek 2' deki yapay sinir ağının eğitim durumu grafiği

Şekil 4.12 performans grafiği ve Şekil 4.13 eğitim durumu grafiğine göre ağın belirlenen ağırlık ve biasları ile ilgili, fonksiyonun aşırı uydurulması ya da veri setlerin hatalı bölünmesi gibi bir durum gözlenmemektedir.



Şekil 4.14. Örnek 2' deki yapay sinir ağının regresyon grafikleri

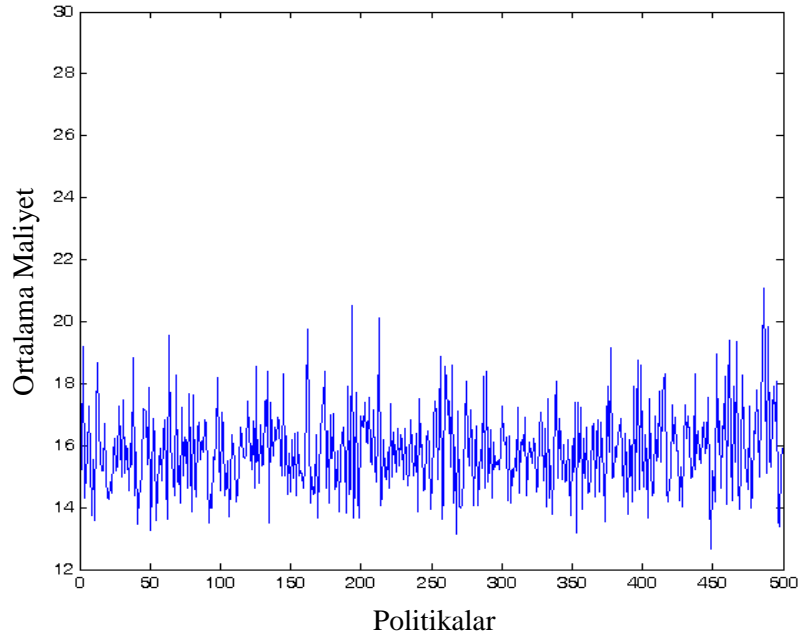
Şekil 4.14'te ağın hedef değerleri ile çıktı değerleri arasındaki ilişkiyi gösteren regresyon grafiği verilmiştir. R korelasyon katsayısının 1'e yaklaşması bu ilişkinin mükemmel olduğunu gösterir. Şekilde görüldüğü üzere eğitim seti, geçerlilik seti, test seti ve tüm veriler için regresyon grafiklerinde ortalama $R=0,96$ 'dır. Buna göre yapay sinir ağıyla Q faktör fonksiyonu çok iyi bir şekilde tahmin edilmekte ve NeuroSMART algoritmasında güvenle kullanılabilir.



Şekil 4.15. Örnek 2'de miyopik politika 1'in yakınsama grafiği

Şekil 4.15'te müşterilere en az sayıda banknot verme politikasının (miyopik politika 1) birim zamandaki ortalama maliyet yakınsama grafiği verilmiştir. Geliştirilen algoritmanın çözümünde ilk adım olarak bu politika (miyopik politika 1) kullanıldığı için stok politikası, Tablo 4.8 başlangıç stok yenileme politikasıyla aynıdır.

Şekil 4.16'da müşterilere en fazla faydayı sağlayan banknot kombinasyonunu verme politikasının (miyopik politika 2) birim zamandaki ortalama maliyet yakınsama grafiği verilmiştir.



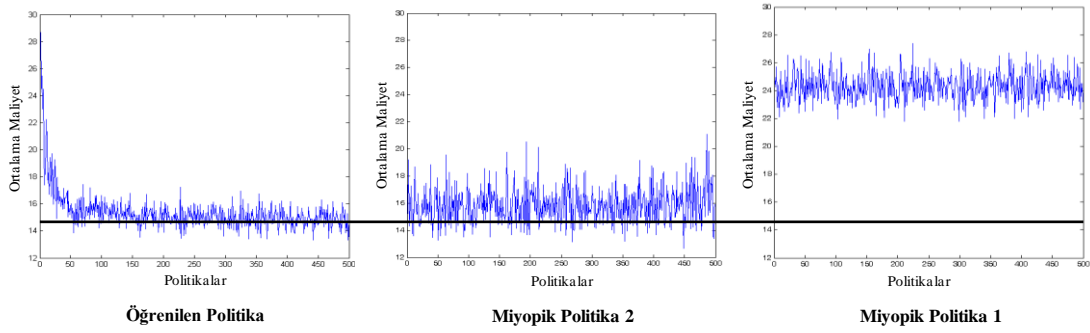
Şekil 4.16. Örnek 2’de miyopik politika 2’nin yakınsama grafiği

Miyopik politika 2 kullanıldığında elde edilen stok politikası tablo 4.10’da yer almaktadır.

Tablo 4.10. Örnek 2’de miyopik politika 2’nin stok yenileme politikası

Stok parametreleri	S_i	c_i	S_i
Banknot çeşitleri			
10 TL (i=1)	107	36	0
20 TL (i=2)	78	28	0
50 TL (i=3)	50	17	0
100 TL (i=4)	36	14	0

Şekil 4.17’de, algoritmayla elde edilen yaklaşık optimal banknot kombinasyonu dağıtım ve stok politikalarının birim zamandaki ortalama maliyetleri, miyopik politika 1 ve miyopik politika 2’nin ürün kombinasyonu dağıtım ve stok yenileme politikalarının birim zamandaki ortalama maliyetleriyle kıyaslanmıştır.



Şekil 4.17. Örnek 2’de öğrenilen politikayla miyopik politika 1 ve 2’nin kıyaslanması

Şekil 4.17’den görüldüğü üzere örnek 2’nin geliştirilen algoritmayla çözümü sonucu elde edilen ürün kombinasyonu dağıtım ve stok yenileme politikasının birim zamandaki ortalama maliyeti, miyopik politika 1 ve miyopik politika 2’den çok daha düşüktür.

4.3.2. Uygulamalı örnek 3 (büyük ölçekli problem)

Uygulamalı örnek 3’te, durum-karar sayısının tezde verilen diğer örneklerden daha büyük olduğu bir problem tasarlanmıştır. Bu örnekte bankamatikten müşterilerin para taleplerinin, müşteri geliş ortalamasının $\lambda = 40$ müşteri/saat ve müşterilerin bireysel para taleplerinin Tablo 4.1’de verilen kesikli olasılık fonksiyonuna uyduğu birleşik poisson süreci olduğu varsayılmıştır (Bkz. Tablo 4.1). Örnekte bankamatığın stoklarında yine 4 farklı banknot çeşidi (10 TL, 20 TL, 50 TL ve 100 TL) bulunduğu ve tüm banknot çeşitleri için stokta bulundurma maliyeti $h=0,001$ TL/banknot/saat; birincil sipariş maliyeti $K=10$ TL/sipariş, ikincil sipariş maliyeti $k=5$ TL/sipariş eklenen banknot çeşidi sayısı, talebin stoktan karşılanamaması maliyeti $w=10$ TL/siparişi stoktan karşılanamayan müşteri sayısı ve müşterinin kaçırdığı faydanın fırsat maliyeti $p=0,01$ TL/kaçan fayda olarak tanımlanmıştır. Belirlenen stok politikasının tüm banknot çeşitleri için en az 0,90 ikinci tip hizmet düzeyini karşılaması gerektiği ve verilen siparişlerin tedarik süresinin sabit 1 saat olduğu kabul edilmiştir.

Bankamatik stoklarında yer alan her banknot çeşidinin (S,c,s) stok parametrelerinin belirlenebilmesi için bankamatige müşteri geliş ortalaması dikkate alınarak

banknotların talep sürecinin, eşdeğer birleşik poisson ve eşdeğer poisson süreçlerine dönüştürülmesi gerekir. 4 banknot çeşidi için hesaplanan birleşik poisson banknot talebi ve eşdeğer poisson banknot talebi parametreleri sırasıyla Tablo 4.11 ve 4.12’de verilmiştir.

Tablo 4.11. Örnek 3’ün başlangıç eşdeğer birleşik poisson banknot talebi parametreleri

Banknotlar (TL) (i=1,2,3,4)	Eşdeğer Müşteri Geliş Ortalaması (λ'_i)	Eşdeğer Bireysel Talep Olasılık Fonksiyonu ($f'_{D_i}(\cdot)$)
10	16	$f_{D_1}(n_1) = \begin{cases} 1 & , n_1 = 1 \\ 0 & , \text{diğer} \end{cases}$
20	24	$f_{D_2}(n_2) = \begin{cases} 2/3 & , n_2 = 1 \\ 1/3 & , n_2 = 2 \\ 0 & , \text{diğer} \end{cases}$
50	20	$f_{D_3}(n_3) = \begin{cases} 1 & , n_3 = 1 \\ 0 & , \text{diğer} \end{cases}$
100	4	$f_{D_4}(n_4) = \begin{cases} 1 & , n_4 = 1 \\ 0 & , \text{diğer} \end{cases}$

Tablo 4.12. Örnek 3’ün başlangıç eşdeğer poisson banknot talebi parametreleri

Banknotlar (TL) (i=1,2,3,4)	Eşdeğer Banknot Müşteri Geliş Ortalaması ($\lambda'_{i\text{ eşdeğer}}$)	Eşdeğer Bireysel Sabit Talep (U_i)	Eşdeğer Banknot Fiyatı (v'_i)
10	16	1	10
20	64/3	1,5	30
50	20	1	50
100	4	1	100

Bölüm 3.1.1’de verilen banknot kombinasyon ve stok politikasını belirleyen algoritmanın ilk adımı için başlangıç banknot dağıtım politikası olarak müşterilere en

az banknot sayısını içeren kombinasyonun dağıtımını miyopik politikası seçildiğinde elde edilen (S_i, c_i, s_i) stok politikaları, Tablo 4.13'te verilmiştir.

Tablo 4.13. Örnek 3'ün başlangıç stok yenileme politikası

Stok parametreleri Banknot çeşitleri	S_i	c_i	s_i
10 TL ($i=1$)	285	84	0
20 TL ($i=2$)	221	80	0
50 TL ($i=3$)	141	51	0
100 TL ($i=4$)	104	40	0

Banknot kombinasyonu dağıtım ve stok yenileme politikalarını belirlemek için geliştirilen algoritmanın uygulanmasıyla elde edilen yaklaşık optimal stok politikası ise Tablo 4.14'te verilmiştir.

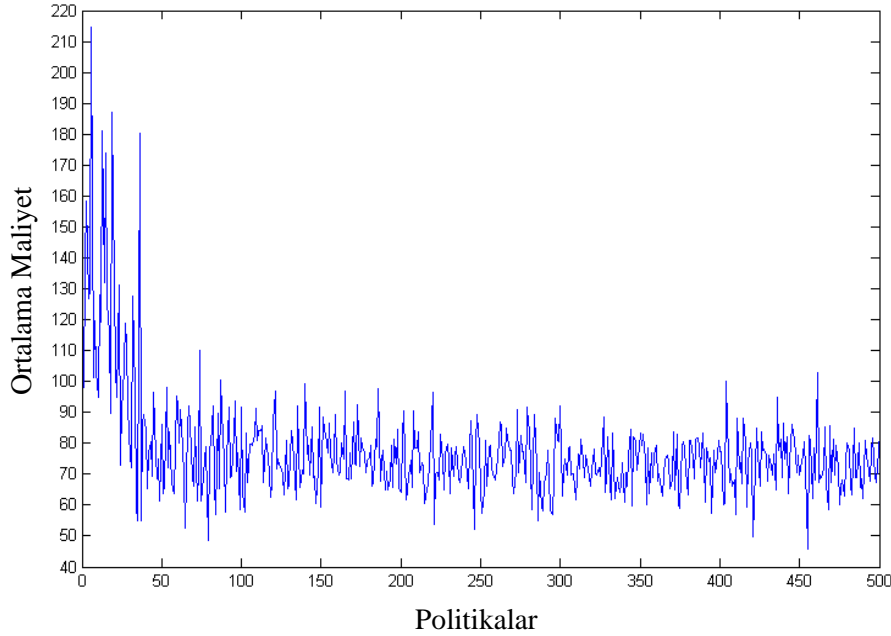
Tablo 4.14. Örnek 3'ün yaklaşık optimal stok yenileme politikası

Stok parametreleri Banknot çeşitleri	S_i	c_i	s_i
10 TL ($i=1$)	334	134	5
20 TL ($i=2$)	221	79	0
50 TL ($i=3$)	132	42	0
100 TL ($i=4$)	97	33	0

Tablo 4.2'de verilen olası taleplerdeki karar sayıları ve yaklaşık optimal stok politikasındaki stokların tamamlanacağı üst sınır düzeyleri (334, 221, 132, 97) dikkate alındığında ele alınan problemin durum-karar sayısının, 46 310 608 344 olduğu görülmektedir. Bu kadar büyük karar-durum sayısına sahip bir problemin Q-faktör değerlerinin arama tablolarına depolanıp, güncellendiği klasik ödüllü öğrenme algoritmalarıyla çözümü mümkün değildir.

Algoritma sonucu belirlenen yaklaşık optimal banknot kombinasyonu dağıtım ve stok yenileme politikasının birim zamandaki ortalama maliyet yakınsama grafiği Şekil 4.18'de verilmiştir. Şekilde görüldüğü üzere geliştirilen algoritmayla elde

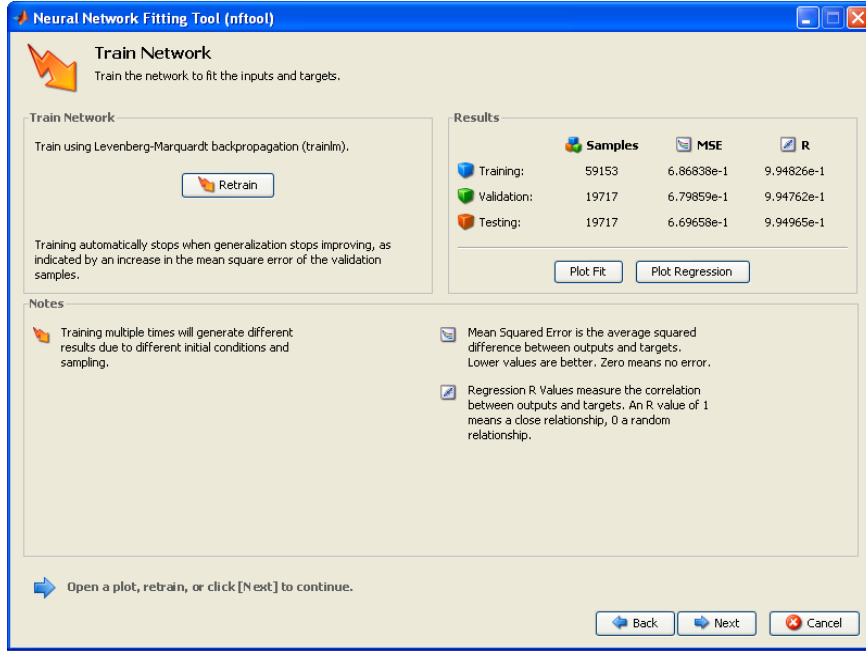
edilen banknot kombinasyonu dağıtım ve stok politikalarının birim zamandaki ortalama maliyeti 70 TL'ye yakınsamaktadır.



Şekil 4.18. Örnek 3'ün yaklaşık optimal banknot dağıtım ve stok yenileme politikası yakınsama grafiği

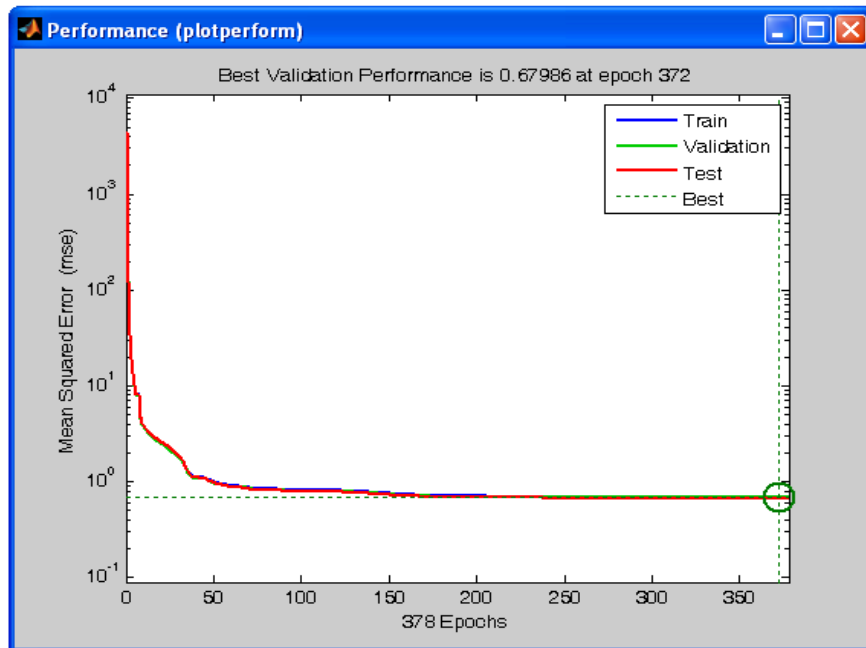
Algoritma yaklaşık optimal banknot kombinasyonu dağıtım ve stok yenileme politikasını 209 saatte bulmuştur. Bankamatik için elde edilen politikanın 2.tip hizmet düzeyi simülâtörde, 0,9395 olarak hesaplanmıştır.

Algoritma durduğu andaki son yapay sinir ağının ortalama hatalarının karesi MSE (mean square error) ve oluşan regresyon denkleminin R korelasyon katsayısı (coefficient of correlation) şekil 4.19'da görülmektedir. Şekilde görüldüğü üzere ortalama hata kareleri oldukça küçüktür ve eğitim seti için 0,686838, geçerlilik seti için 0,679859 ve test seti için 0,669658 olarak gerçekleşmiştir. Hedefler ve çıktılar arasındaki regresyonun korelasyon katsayıları ise eğitim seti için 0,994826, geçerlilik seti için 0,994762 ve test seti için 0,994965 olarak gerçekleşmiştir.



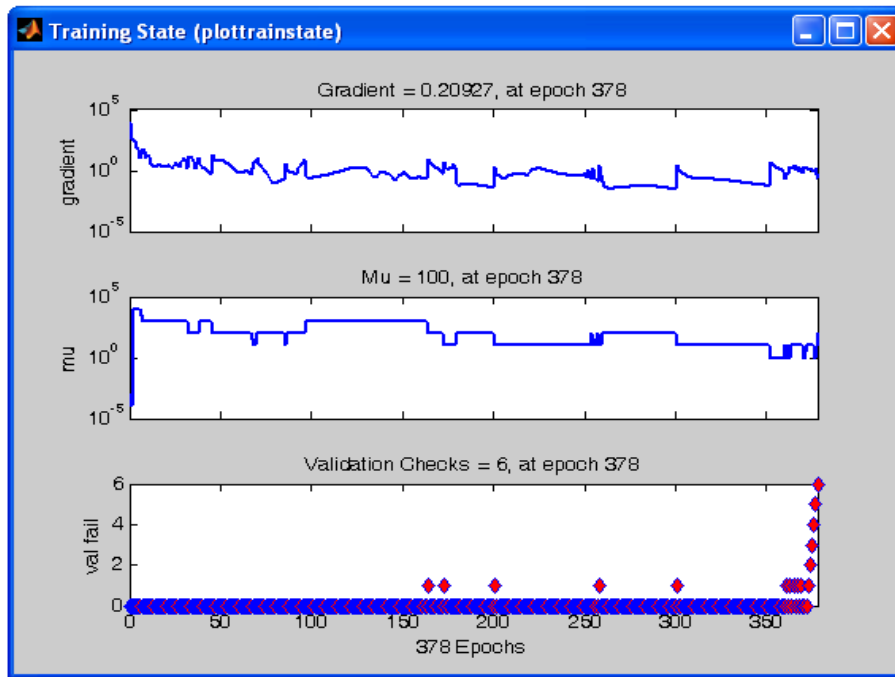
Şekil 4.19. Örnek 3'teki yapay sinir ağının ortalama hata kareleri ve korelasyon katsayıları

Şekil 4.20'de yapay sinir ağının performans grafiği verilmiştir. Şekilde verilen performans grafiği incelendiğinde ortalama hata karelerinin (MSE) eğitim algoritmasının yinelenmeleri ilerledikçe, hızlı bir şekilde 1000'li değerlerden 0,68'lere kadar düştüğü görülmektedir.



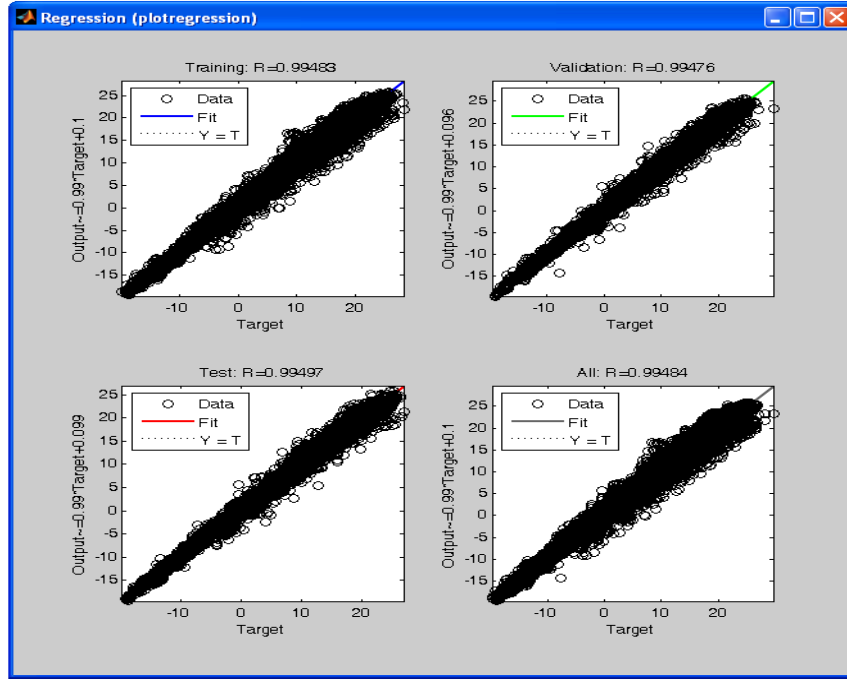
Şekil 4.20. Sayısal örnek 3'teki yapay sinir ağının performans grafiği

Şekil 4.21’de verilen eğitim durumu grafiğinde, eğitim setinden hesaplanan gradyen değerini, Levenberg-Marquardt eğitim algoritmasında ağırlıkların güncellenmesinde kullanılan μ sabitini ve geçerlilik hatasında artma olup olmadığının yinelemelere göre değişimleri gösterilmiştir. Şekilden görüldüğü üzere 372’nci yinelemede geçerlilik hatası artmaya başlamış ve arka arkaya 6 yinelemede bu artış sürmüştür. Bu nedenle 372’nci yinelemeden itibaren ağın eğitim verilerini ezberlemeye başladığı kararına varılarak, 372’nci yinelemede hesaplanan ağ ağırlıklarına dönülmüştür.

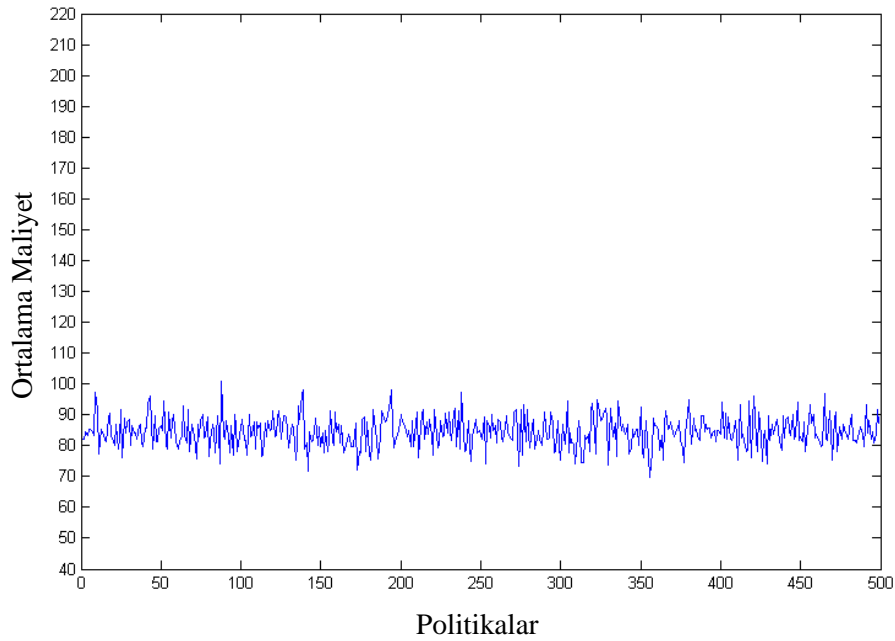


Şekil 4.21. Örnek 3’teki yapay sinir ağının eğitim durumu grafiği

Şekil 4.22’de ağın hedef değerleri ile çıktı değerleri arasındaki ilişkiyi gösteren regresyon grafiği verilmiştir. Şekilde görüldüğü üzere eğitim seti, geçerlilik seti, test seti ve tüm veriler için regresyon grafiklerinin hepsinde de $R=0,99$ ’un üzerindedir. Buna göre yapay sinir ağıyla Q faktör fonksiyonu çok iyi bir şekilde tahmin edilmekte ve NeuroSMART algoritmasında güvenle kullanılabilir.



Şekil 4.22. Örnek 3'teki yapay sinir ağının regresyon grafikleri



Şekil 4.23. Örnek 3'te miyopik politika 1'in yakınsama grafiği

Şekil 4.23'te müşterilere en az sayıda banknot verme politikasının (miyopik politika 1) birim zamandaki ortalama maliyet yakınsama grafiği verilmiştir. Geliştirilen algoritmanın çözümünde ilk adım olarak bu politika (miyopik politika 1)

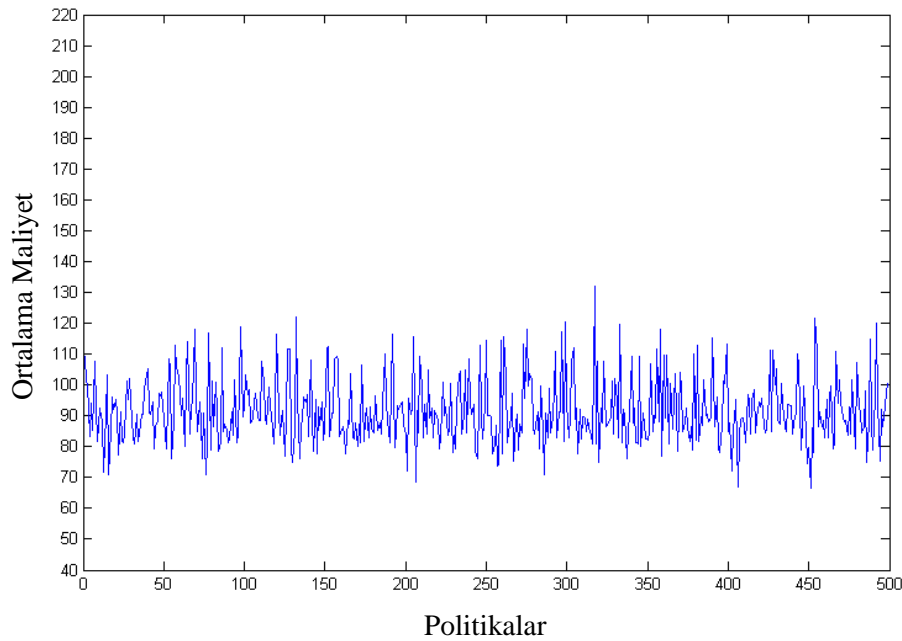
kullanıldığı için stok politikası, Tablo 4.13 başlangıç stok yenileme politikasıyla aynıdır. Şekilde görüldüğü üzere miyopik politika 1'in birim zamandaki ortalama maliyeti 85 TL'ye yakınsamaktadır.

Tablo 4.15'te müşterilere en fazla faydayı sağlayan banknot kombinasyonunun dağıtılması politikasının (miyopik politika 2) hesaplanan stok yenileme politikası verilmiştir.

Tablo 4.15. Örnek 3'te miyopik politika 2'nin stok yenileme politikası

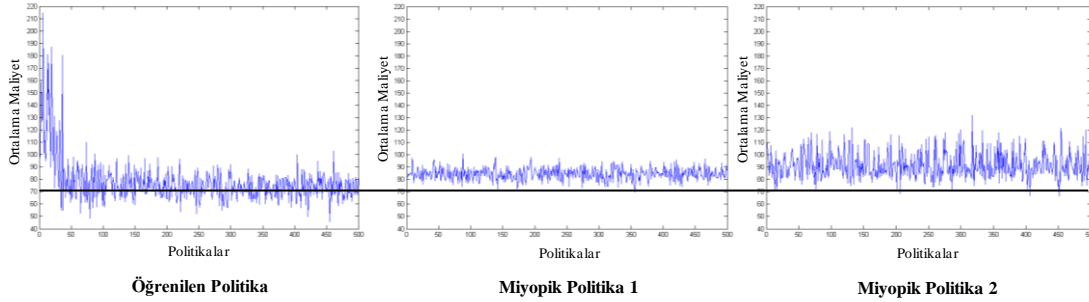
Stok parametreleri Banknot çeşitleri	S_i	c_i	s_i
10 TL (i=1)	306	106	5
20 TL (i=2)	221	79	0
50 TL (i=3)	140	50	0
100 TL (i=4)	103	39	0

Şekil 4.24'te miyopik politika 2'nin ve bu politika altındaki stok yenileme politikasının birim zamandaki ortalama maliyet yakınsama grafiği verilmiştir. Şekilde görüldüğü üzere miyopik politika 2'nin birim zamanda ortalama maliyeti 90 TL'ye yakınsamaktadır.



Şekil 4.24. Örnek 3'te miyopik politika 2'nin yakınsama grafiği

Şekil 4.25'te, geliştirilen algoritmayla elde edilen yaklaşık optimal banknot kombinasyonu dağıtım ve stok yenileme politikasının birim zamandaki ortalama maliyet yakınsama grafikleri ile miyopik politika 1 ve miyopik politika 2'nin birim zamandaki ortalama maliyet yakınsama grafikleri verilmiştir.



Şekil 4.25. Örnek 3'te öğrenilen politikayla miyopik politika 1 ve 2'nin kıyaslanması

Şekil 4.25'ten görüldüğü üzere geliştirilen algoritmayla elde edilen politikaların birim zamandaki ortalama maliyeti, miyopik politika 1 ve miyopik politika 2'nin ortalama maliyetlerinden daha düşüktür.

Literatürde yer alan çalışmalara göre ödüllü öğrenme algoritmalarında fonksiyon yaklaşımı, her problem için iyi sonuçlar üretmemektedir [37, 49]. Ödüllü öğrenme algoritmalarında yapay sinir ağlarının kullanıldığı ve etkileyici sonuçlar alındığı en bilinen örnek Tesauro [50]'nun yapay sinir ağlarını kullandığı tavla oyunu (backgammon) uygulamasıdır.

Boyan ve Moore [49], dinamik programlamadaki arama tablolarının yerine fonksiyon yaklaşımı kullanımının tavla oyunundaki gibi başarılı örneklerinin az olduğunu ve gürbüz sonuçlar vermediğini, hatta bazen tamamıyla yanlış kontrol politikalarıyla sonuçlandığını ifade etmektedir. Yazarlara göre fonksiyon yaklaşımı sonucu 4 olası davranış gözlemlenebilir [49]:

1) İyi Yakınsama: Fonksiyon yaklaşımıyla her bir yinelemedeki Q faktör değer fonksiyonu doğru bir şekilde temsil edilir (diğer bir deyişle yeterince yinelemeden sonra değer fonksiyonu her bir durumun değerini doğru bir şekilde temsil eder) ve yaklaşık optimal V^* değer fonksiyonuna yakınsar.

2) Şanslı Yakınsama: Fonksiyon yaklaşımıyla her bir yinelemedeki Q faktör değer fonksiyonu doğru bir şekilde temsil edilmez, ancak yine de algoritma, politikası optimal olan bir değer fonksiyonuna yakınsamayı başarır.

3) Kötü Yakınsama: Algoritma yakınsar; örneğin belli bir yinelemeden sonra durumların hedef değerlerinin değişimi durur, ancak elde edilen değer fonksiyonu ve politika kötüdür.

4) Iraksama: Verilen diğer 3 durumdan en kötüsüdür. Küçük tahmin hataları, yinelemeler ilerledikçe katlanabilir ve durumların değer fonksiyonu değişimleri asla azalmayabilir.

Öncelikle gerçekleştirilen ve çalışmada verilen/verilmeyen gizli katman sayısının 10 ve üzeri olarak belirlendiği tüm örnekler için, sonsuz zaman ufkunda birim zaman başına ortalama maliyet fonksiyonu değişiminde yakınsama sağlanmıştır. Sadece gizli katman sayısının 5'ten az olarak belirlendiği örneklerde ve bazen 5-10 arasında gizli katman kullanıldığı durumlarda kullanılabilir olmayan sonuçlar görülmüştür. Ayrıca uygulamalı örnek 1'deki küçük ölçekli (durum-karar sayısı nispeten az) problem, hem fonksiyon yaklaşımsız ödüllü öğrenme algoritması SMART ile ve hem de yapay sinir ağı yaklaşımı NeuroSMART algoritmasıyla çözülmüş ve her iki çözümün ürettiği politikaların ortalama maliyetlerinin birbirine çok yakın olduğu görülmüştür. Bu nedenle NeuroSMART algoritmasıyla elde edilen politikaların sonsuz zaman ufkunda birim zaman başına ortalama maliyetlerinin yakınsaması, şans olamaz.

Çalışmada verilen örneklerin yakınsama grafikleri, durumların Q faktör değer değişimlerinin değil, değer yineleme algoritması çalıştıkça belli yineleme sayılarında üretilen politikaların sonsuz zaman ufkunda birim zaman başına ortalama maliyet grafikleridir. Verilen uygulamalı örneklerin grafiklerinden görüldüğü üzere SMART/NeuroSMART ödüllü algoritmalarının yinelemeleri devam ettikçe politika iyileşmekte, artık durumların değer fonksiyonu değişmemeye başladığında ise üretilen politika sabitlenmekte ve elde edilen maliyetler değişmemeye başlamaktadır. Ayrıca çalışmada verilen uygulamalı örnek 2 (orta ölçekli) ve uygulamalı örnek 3

(büyük ölçekli) problemlerinin ürün kombinasyonu dağıtım ve stok yenileme politikalarını birlikte belirleyen algoritmayla çözülmesiyle elde edilen politikaların sonsuz zaman ufkunda birim zaman başına ortalama maliyetlerinin, en az banknot sayısını içeren kombinasyonun dağıtımını miyopik politikasının ve en fazla faydayı veren kombinasyonların dağıtımını miyopik politikasının birim zamandaki ortalama maliyetlerinden çok daha düşük olduğu gözlenmiştir.

Uygulamalı örneklerde elde edilen sonuçlara göre, çalışmada ele alınan problem için geliştirilen ve Q faktör değer fonksiyonuna yapay sinir ağlarıyla yaklaşılan algoritma, Boyan ve Moore [49]'un genellemesinin aksine Tesauro [50]'nun tavla oyunu uygulamasındaki gibi çok iyi sonuçlar üretmiştir. Çalışma, yapay sinir ağlarıyla Q faktör değer fonksiyonuna yaklaşımlı ödüllü öğrenme algoritmaları için iyi bir örnek teşkil etmektedir.

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

Çalışmada, çok ürünli bir stok sistemine gelen talebin karşılanmasında kullanılacak ürün kombinasyonu dağıtım ve stok yenileme politikalarının belirlenmesi problemi ele alınmıştır. Literatürdeki firma kaynaklı müşteri ikamesinin konu edildiği problemlerde sadece aşağı tek yönlü ürün ikamesi söz konusu olmakta, farklı yönlü ikamelerin müşteriye faydası dikkate alınmamaktadır. Müşteri kaynaklı ürün ikamesinin söz konusu olduğunda ise firmalar ürün talebini yönlendirememektedir. Yapılan çalışmada ürün ikamesi firma kaynaklı olarak gerçekleşmekte, ancak ikame literatürden farklı olarak sadece aşağı tek yönlü olarak gerçekleşmemektedir. Ürün ikamelerine, farklı ürün ikamelerinin müşteri faydaları da göz önüne alınarak, firma tarafından karar verilmektedir. Bu yönüyle literatürde benzer bir çalışma yer almamaktadır.

Çalışmada, müşteri talebini karşılayacak şekilde ürün kombinasyonu dağıtım ve ürün stoklarının yenilenmesi politikalarını sıralı ve yinelemeli bir şekilde birlikte belirleyen bir algoritma geliştirilmiştir. Stok yenileme politikası olarak, sürekli gözden geçirmeli ve eşgüdümlü (S,c,s) politikası kullanılmış ve stok sistemindeki her bir ürün çeşidinin (S,c,s) stok kontrol parametrelerinin belirlenmesinde literatürdeki çalışmalardan faydalanılmıştır. Ürün kombinasyonu dağıtım problemi ise yarı Markov karar süreci olarak modellenmiştir. Ele alınan problem endüstride yeri olan karmaşık ve büyük ölçekli gerçek bir yaşam problemidir. Bu nedenle modelin dinamik programlamayla çözümü mümkün olmamış ve çözüm için simülasyon tabanlı bir optimizasyon tekniği olan Ödüllü Öğrenme kullanılmıştır. Çalışmada, yarı Markov karar süreci problemlerinin çözümü için geliştirilmiş SMART ödüllü öğrenme algoritmasından hareketle, Q faktör fonksiyonunu çok katmanlı yapay sinir ağları ile tahmin eden, fonksiyon yaklaşımını NeuroSMART algoritması geliştirilmiştir. Çalışma, durum-karar sayısının büyük olduğu ve bu nedenle ödüllü öğrenme algoritmalarının doğrudan kullanılmadığı durumlarda, Q faktör

fonksiyonunun çok katmanlı yapay sinir ağılarıyla tahmin edildiği ödüllü öğrenme algoritmalarına iyi bir örnek teşkil etmektedir.

Bölüm 4'te gerçekleştirilen uygulamalı örneklerin ilk bölümündeki Q faktör fonksiyonunu tahmin etmek için kullanılan yapay sinir ağında gizli katman sayısının uygun bir şekilde belirlenmesiyle elde edilen sonuçlarda, NeuroSMART algoritmasının SMART algoritması kadar iyi sonuçlar verdiği gözlenmiştir. Uygulamalı örneklerde gizli katman sayısı küçük belirlendiğinde ise algoritma sonuçlarının bir değere yakınsamadığı veya yakınsasa bile güvenilir olmadığı görülmüştür. Bu durum Q faktör fonksiyonun beklendiği gibi doğrusal olmayan, karmaşık bir fonksiyon olduğunu göstermektedir. Uygulamalı örneklerde yapay sinir ağlarında 20 gizli katman kullanımıyla gürbüz (robust) sonuçlar üretmiştir.

Bölüm 4'te gerçekleştirilen uygulamalı örneklerin ikinci bölümünde yaklaşık optimal ürün dağıtım ve stok yenileme politikalarının belirlenmesi için geliştirilen algoritmanın, NeuroSMART algoritmasıyla performansı incelenmiştir. Uygulamalı örneklerde, geliştirilen algoritmayla elde edilen birim zamandaki ortalama maliyetler, en az banknot kombinasyonunun dağıtımı ve en fazla faydayı sağlayan banknot kombinasyonunun dağıtım politikalarının ortalama maliyetleri ile kıyaslanmış ve geliştirilen algoritmayla elde edilen sonuçların çok daha iyi olduğu görülmüştür.

Bu bölümde ayrıca Q faktör fonksiyonunu tahmin etmek için kullanılan yapay sinir ağının performansı ve kullanılabilirliği de araştırılmıştır. Elde edilen sonuçlara göre, geliştirilen algoritmayı kullanan simülasyon sona erdiğinde elde edilen yapay sinir ağı, genelleştirilebilir ve ağın ağırlıklarından istenen durum-karar ikilisi simüle edilerek uygulanacak politika kolaylıkla belirlenebilir.

Elde edilen sonuçlar kısaca aşağıda özetlenmiştir:

- Çalışmada, literatürde benzeri olmayan ancak endüstride yeri olan bir problem ele alınmış ve çözümüyle literatüre önemli katkı sağlanmıştır.

- Uygulamalı örneklerden elde edilen sonuçlara göre, uygun gizli katman sayısına sahip yapay sinir ağıyla Q faktör fonksiyonunun tahmin edildiği NeuroSMART ödüllü öğrenme algoritması, çok iyi sonuçlar vermektedir.
- Dinamik programlama ve ödüllü öğrenmeyle çözülemeyecek büyüklükte durum-karar sayısına sahip Markov ve yarı Markov problemlerinde, çok katmanlı yapay sinir ağı yaklaşımı NeuroSMART ödüllü öğrenme algoritması yaklaşık optimal çözüm veren iyi bir alternatiftir.
- NeuroSMART algoritmasının başarısında gizli katman sayısı önemlidir. Ele alınan problemde gizli katman sayısı 10 ve 10'dan büyük seçildiğinde algoritma başarıyla çalışmaktadır. Gizli katman sayısı 5 olarak belirlendiğinde algoritma gürbüz değildir, 2 olarak belirlendiğinde ise algoritmayla elde edilen çözüm kullanılamamaktadır.
- Geliştirilen algoritmayla elde edilen ürün kombinasyonu dağıtım ve stok yenileme politikalarının sonsuz zaman ufkundaki ortalama maliyetleri, incelenen miyopik politikaların birim zamandaki ortalama maliyetlerinden çok daha düşüktür.

Çalışmada ele alınan stok sistemi, bir imalatçı ve bir dağıtıcıdan oluşmaktadır. İleriki çalışmalarda problem, dağıtıcı sayısı artırılarak, hem stok sistemindeki ürünlerin ve hem de farklı dağıtıcıların aynı imalatçıdan eşgüdüm halinde stoklarını yeniledikleri duruma genişletilebilir. Örneğin ele alınan problem çalışmanın motivasyon kaynağı bankamatik problemi için, bir bankanın bir şehirdeki tüm bankamatiklerindeki banknotların eş güdüm halinde stoklarının yenilenmesi problemine genişletilebilir. Bu durumda problem, ele alınan tüm bankamatiklerin yaklaşık optimal eşgüdümlü stok yenileme politikalarının, her bir bankamatığın banknot kombinasyonu dağıtım politikasının ve stokları yenilenecek bankamatiklere banknotların gönderilmesinde dinamik rotaların belirlenmesi problemine dönüşür.

KAYNAKLAR

- [1] SILVER, E.A., A control system for coordinated inventory replenishment. *Int.J.Prod.Res.*, 1974; 12(6):647-671.
- [2] THOMPSTONE, R.M., SILVER, E.A., A coordinated inventory control system for compound poisson demand and zero lead time. *Int.J.Prod.Res.*, 1975; 13(6):581-602.
- [3] SILVER, E.A., Establishing reorder points in the (S,c,s) coordinated control system under compound poisson demand. *Int. J. Prod. Res.*, 1981; 19(6):743-750.
- [4] PENTICO, D.W., The assortment problem with probabilistic demands. *Management Science*, 1974; 21(3):286-290.
- [5] WAGNER, H., WHITIN, T.W., Dynamic version of the economic lot size model. *Management Science*, 1958; 5: 89-96.
- [6] LEACHMAN, R., Preliminary design and development of a corporate-level production system for the semiconductor industry. Working Paper ORC 86-11, Operations Research Center, University of California, Berkeley, CA, 1986.
- [7] IGNALL, E., VEINOTT, A.F., Optimality of myopic inventory policies for several substitute products. *Management Science*, 1969; 15(5): 284-304.
- [8] HSU, A., BASSOK, Y., Random yield and random demand in a production system with downward substitution. *Operations Research*, 1999; 47(2): 277-290.
- [9] BASSOK, Y., ANUPINDI, R., AKELLA, R., Single-period multiproduct inventory models with substitution. *Operations Research*, 1999; 47(4): 632-642.
- [10] RAO, U.S., SWAMINATHAN, J.M., ZHANG, J., Multi-product inventory planning with downward substitution. *Stochastic demand and setup costs*, *IEEE Transactions*, 2004; 36: 59-71.
- [11] LIU, J., LEE, C.G., Evaluation of inventory policies with unidirectional substitutions. *European Journal of Operational Research*, 2007; 182: 145-163.
- [12] AXÄTER, S., A new decision rule for lateral transshipments in inventory systems. *Management Science*, 2003; 49(9): 1168-1179.
- [13] AXÄTER, S., Evaluation of unidirectional lateral transshipments and substitutions in inventory systems. *European Journal of Operational Research*, 2003; 149: 438-447.

- [14] OLSSON, F., An inventory model with unidirectional lateral transshipments. *European Journal of Operational Research*, 2010; 200: 725-732.
- [15] NETESSINE, S., RUDI, N., Centralized and competitive inventory models with demand substitution. *Operations Research*, 2003; 51(2): 329-335.
- [16] SMTITH A.S., AGRAWAL, N., Management of multi-item retail inventory systems with demand substitution. *Operations Research*, 2000; 48(1): 050-064.
- [17] ANUPUNDI, R., DADA, M., GUPTA, S., Estimation of consumer demand with stock-out based substitution: an application to vending machine products. *Marketing Science*, 1998; 17(4): 406-423.
- [18] NOONAN, P.S., When customers choose: a multi-product, multi-location newsboy model with substitution. Working Paper, Emory University, Atlanta, GA, 1995.
- [19] MAHAJAN, S., RYZIN, G., Inventory competition under dynamic consumer choice. *Operations Research*, 2001; 49(5): 646-657.
- [20] RAJARAM, K., TANG, C.S., The impact of product substitution on retail merchandising. *European Journal of Operational Research*, 2001; 135: 582-601.
- [21] PENTICO, D.W., The assortment problem: a survey. *European Journal of Operational Research*, 2008; 190: 295-309.
- [22] BALINTFY, J.L., On a basic class of multi-item inventory problems. *Management Science*, 1964; 10:287.
- [23] IGNALL, E., Optimal continuous review policies for two product inventory systems with joint setup costs. *Management Science*, 1969; 15(5):278-283
- [24] CURRY, G.L., SKEITH, R.W., HARPER, R.G., A multi-product dependent inventory model. *AIIE Transactions*, 1970; 3(2): 263-270
- [25] SCHAACK, J.P., SILVER, E.A., A procedure, involving simulation, for selecting the control variables of an (S,c,s) joint ordering strategy. *INFOR.*, 1972; 10: 154
- [26] FEDERGRUEN, A., GROENEVELT, H., TIJMS, H.C., Coordinated replenishments in a multi-item inventory system with compound poisson. *Management Science*, 1984; 30(3):344-355.
- [27] ZHENG, Y.S., Optimal control policy for stochastic inventory systems with Markovian discount opportunities. *Operations Research*, 1994; 42(4): 721-737.
- [28] MELCHORS, P., Calculating can-order policies for the joint replenishment problem by the compensation approach. *European Journal of Operational Research*, 2002; 141: 587-595.
- [29] ATKINS, D.R., IYOGUN, P.O., Periodic versus 'can order' policies for coordinated multi-item inventory systems. *Management Sciences*, 1988; 34(6), 791-796.

- [30] PANTUMSINCHAI, P., A comparison of three joint ordering inventory policies. *Decision Sciences*, 1992; 23(1): 111-127.
- [31] EIJS, M.J.G., On the determination of the control parameters of the optimal can-order policy. *Mathematical Methods of Operations Research*, 1994; 39: 289-304.
- [32] JOHANSEN, SG., MELCHORS, P., Can order policy for the periodic-review joint replenishment problem. *Journal of Operational Research Society*, 2003; 54: 283-290.
- [33] AXÄTER, S., *Inventory control*. Kluwer Academic Publishers, 2000.
- [34] ADELSON, R.M., Compound poisson distributions. *Operational Research Quarterly*, 1966; 17:73-75.
- [35] FRIEND, J.K., Stock control with random opportunities for replenishment. *Operational Research Quarterly*, 1961; 11(3):130-136.
- [36] ROSS, S.M., *Introduction to probability models*. Academic Press, 7 ed., 2000.
- [37] GOSAVI, G., *Simulation-based optimization*. Kluwer Academic Publishers, 2003.
- [38] PUTERMAN, M.L., *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.
- [39] DIMITRI P.B., TSITSIKLIS, J., *Neuro-dynamic programming*, Athena Scientific, 1996.
- [40] BUŞONIU, L., BABUŞKA, R., SCHUTTER, B.D., ERNST, D., *Reinforcement learning and dynamic programming using function approximators*. CRC Press, 2010.
- [41] MITRANI, I., *Probabilistic modelling*. Cambridge University Press. 1998.
- [42] APAYDIN, E., *Introduction to machine learning*. MIT Press, 2004.
- [43] GÜLBAĞ, A., *Yapay sinir ağı ve bulanık mantık tabanlı algoritmalar ile uçucu organik bileşiklerin miktarsal tayini*. Doktora Tezi, Sakarya Üniversitesi, Elektrik-Elektronik Mühendisliği, 2006.
- [44] DEMUTH, H., BEALE, M., HAGAN, M., *Neural network toolbox user guide*. The MathWorks Inc, 2008.
- [45] SILVER, A.E., PYKE, D.F., *Inventory management and production planning and scheduling*. John Wiley & Sons, 1998.
- [46] SILVER, E.A., Three ways of obtaining the average cost expression in a problem related to joint replenishment inventory control. *Nav. Res. Logist.Q.*, 1973; 20, 241.
- [47] HOPP, J.W., SPEARMAN, M.L., *Factory physics*. McGraw-Hill Education, 2000.

- [48] TIJMS, H.C., Stochastic models an algorithmic approach. John Wiley & Sons, 1995.
- [49] BOYAN, J.A., MOORE, A.W., Generalization in reinforcement learning: safely approximation the value function. Advanced in Neural Information Processing Systems, 1995; 369-376
- [50] TESAURO, G., Programming backgammon using self-teaching neural nets. Artificial Intelligence, 2002; 134:181-199

ÖZGEÇMİŞ

Mustafa Ahmet Beyazıt Ocaktan, 04.01.1976'da Balıkesir'de doğmuş, ilk, orta ve lise eğitimini Balıkesir'de tamamlamıştır. 1994 yılında Balıkesir Üniversitesi Endüstri Mühendisliği bölümünü kazanarak üniversite eğitimine başlamış ve bu bölümden 1998 yılında mezun olarak endüstri mühendisi unvanı almıştır. 1999-2001 yılları arasında yedek subay olarak askerlik görevini tamamlamıştır. 2001 yılında Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim dalında başladığı yüksek lisans eğitimini 2004 yılında tamamlamış ve 2005 yılında Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim dalında doktora eğitimine başlamıştır. 2002-2006 yılları arasında Balıkesir Üniversitesi Endüstri Mühendisliği Bölümünde Araştırma Görevlisi olarak görev yapmış, 2006 yılından itibaren halen Sakarya Üniversitesi Endüstri Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır.