

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

**DİNAMİK ATÖLYE ÇİZELGELEMEDE YAPAY SINIR
AĞI İLE TESLİM TARİHİ BELİRLENMESİ**

DOKTORA TEZİ

Mümtaz İPEK

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Tez Danışmanı : Doç. Dr. İ. Hakkı CEDİMOĞLU

Temmuz 2007

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**DİNAMİK ATÖLYE ÇİZELGELEMEDE YAPAY SINIR
AĞI İLE TESLİM TARİHİ BELİRLENMESİ**

DOKTORA TEZİ

Mümtaz İPEK

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 26/07/2007 tarihinde aşağıdaki jüri tarafından Oybirliği / Oyçokluğu ile kabul edilmiştir.

Prof.Dr. Alpaslan FIĞLALI Prof.Dr. Harun TAŞKIN Prof.Dr. Orhan TORKUL

Jüri Başkanı

Jüri Üyesi

Jüri Üyesi

Doç.Dr. İ. Hakkı CEDİMOĞLU Yrd.Doç.Dr. Kasım BAYNAL

Jüri Üyesi

Jüri Üyesi

TEŐEKKÜR

Öncelikle bu tez alıőmasının konu seimini ve her adımında yardımlarını gördüğüm, her an yanımda olan deęerli hocam Do. Dr. İ. Hakkı Cedimoęlu'na teőekkürlerimi bir bor biliyorum.

Tez alıőmam sırasında sıkıntılı anlarımda yardım ve nasihat aldığım kıymetli hocalarım Prof. Dr. Orhan Torkul ve Yrd. Do. Dr. Bayram Topal'a, alıőmanın başlangıcında deęerleri fikirlerini aldığım hocam Yrd. Do. Dr. M. Fatih Taőgetiren'e teőekkür ediyorum.

Özellikle tezin yazım aőamasında manevi olarak güç aldığım, her an yanımda olan, özveri göstererek yardım eden deęerli kardeőim, arkadaőım Dr. Alper Göksu'ya yürekten teőekkür ediyorum.

Baőta, bölümdeki iş yükümü hafifleterek ve teővikleriyle alıőmamda yardımcı olan bölüm başkanımız kıymetli hocam Prof. Dr. Harun Taőkın olmak üzere, bütün bölüm hocalarım ve arkadaőlarıma anlayışları için çok teőekkür ediyorum.

Gösterdikleri sabır, yardım ve anlayışlarından dolayı tüm aileme teőekkürü bir bor biliyorum.

Mümtaz İPEK

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER.....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vii
ŞEKİLLER LİSTESİ.....	ix
TABLolar LİSTESİ.....	x
ÖZET.....	xii
SUMMARY.....	xiii
BÖLÜM 1.	
GİRİŞ.....	1
1.1 Giriş.....	1
1.2. Çizelgeleme Problemleri.....	2
1.3. Genel Atölye Tipi Çizelgeleme.....	4
1.3.1. Statik çizelgeleme.....	5
1.3.2. Dinamik çizelgeleme.....	5
1.4. Teslim Tarihi Belirleme.....	6
1.4.1. Regresyon analizi.....	8
1.4.2. Yapay sinir ağları.....	9
1.5. Öncelik Kuralları.....	10
1.6. Performans Ölçütleri	10
1.7. Varsayımlar.....	11
1.8. Çalışmanın İçeriği.....	12
1.9. Tezin Amacı.....	13
1.10. Tezin Düzenlenmesi.....	13

BÖLÜM 2.

LİTERATÜR TARAMASI.....	15
2.1. Giriş.....	15
2.2. Sayısal Yöntemler.....	15
2.3. Bulgusal Yöntemler.....	18
2.4. Yapay Sinir Ağları.....	21
2.5. Sonuç.....	25

BÖLÜM 3.

TESLİM TARİHİ BELİRLEMEDE YAPAY SİNİR AĞI YAKLAŞIMI.....	26
3.1. Giriş.....	26
3.2. Regresyon Modelleri.....	27
3.2.1. Sadece iş karakteristiklerini kullanan modeller.....	27
3.2.2. İş karakteristikleri ve atölye bilgilerini kullanan modeller....	28
3.2.3. Regresyon analizi.....	28
3.3. Regresyon Modelleri ile Teslim Tarihi Belirleme.....	30
3.3.1. Regresyon analizi için veri elde edilmesi.....	32
3.3.2. Regresyon modellerinin oluşturulması.....	32
3.3.3. Regresyon modelleri ile teslim tarihi belirleme.....	34
3.4. Yapay Sinir Ağları.....	35
3.4.1. Temeller.....	35
3.4.2. Genel özellikler.....	39
3.4.3. Öğrenme.....	40
3.4.4. Bu çalışma için seçilen sinir ağı modeli.....	41
3.4.5. Geriye yayılım ağları.....	41
3.4.6. Bir geriye yayılım ağının yapısı.....	41
3.4.7. Geriye yayılım öğrenme işlemi.....	43
3.4.8. Gerçek çıktuların belirlenmesi (ileri doğru hesaplama).....	44
3.4.9. Hatayı yayma (geriye doğru hesaplama).....	45
3.5. Yapay Sinir Ağı Modeli.....	47
3.5.1. Öğrenme setinin oluşturulması.....	47
3.5.2. Yapay sinir ağının oluşturulması.....	52
3.5.3. Yapay sinir ağının eğitilmesi.....	53

3.5.4. Yapay sinir ađı ile teslim tarihi belirleme.....	53
3.6. Her İki Modelin Performans Ölçütlerine Göre karşılaştırılması....	61
BÖLÜM 4.	
BENZETİM VE YAPAY SİNİR AĐI İLE MODELLEME.....	62
4.1. Giriş.....	62
4.2. Düşünsel Atölye Sistemi.....	63
4.3. Benzetim.....	68
4.3.1. Benzetim tanımı ve yaklaşımı.....	69
4.3.2. Benzetimin faydaları.....	70
4.3.3. Benzetimin sakıncaları.....	71
4.3.4. Benzetim modelleri.....	71
4.3.5. Benzetim elemanları.....	72
4.3.6. Benzetimin üretim sistemlerinde uygulama alanları.....	74
4.3.7. Benzetimin aşamaları.....	75
4.4. Benzetim Yazılımı.....	77
4.4.1. Benzetim yazılımı ana programı.....	77
4.4.2. Varış() fonksiyonu.....	79
4.4.3. Bitiş() fonksiyonu.....	82
4.5. Yapay Sinir Ađı Yazılımı.....	85
4.5.1. YeniAđ() fonksiyonu.....	87
4.5.2. EskiAđ() fonksiyonu.....	90
4.5.3. YenidenÇađır() fonksiyonu.....	92
4.5.4. Test() fonksiyonu.....	93
BÖLÜM 5.	
DENEYSEL SONUÇLAR.....	96
5.1. Giriş	96
5.2. Regresyon Modelleri ile Elde Edilen Benzetim Sonuçları.....	96
5.2.1. EKİS önce öncelik kuralı ile regresyon modelleri benzetim sonuçları.....	97
5.2.2. EETT önce öncelik kuralı ile regresyon modelleri benzetim sonuçları.....	100

5.3. Yapay Sinir Ağı Modeli ile Benzetim Sonuçları	103
5.3.1. EKİS önce öncelik kuralı ile yapay sinir ağı modeli benzetim sonuçları.....	103
5.3.2. EETT önce öncelik kuralı ile yapay sinir ağı modeli benzetim sonuçları.....	104
5.4. Modellerin Karşılaştırılması.....	104
BÖLÜM 6.	
SONUÇLAR VE ÖNERİLER.....	107
6.1. Çalışma Özeti.....	107
6.2. Bulgular.....	108
6.3. Katkılar.....	108
6.4. İleriye Dönük Çalışma Alanları.....	110
KAYNAKLAR.....	111
EKLER.....	118
ÖZGEÇMİŞ.....	152

SİMGELER VE KISALTMALAR LİSTESİ

α	: Regresyon sabiti
β	: Regresyon katsayısı
ε	: Tesadüfi hata terimi
ΔW	: Değişim miktarı
C_i	: Tamamlanma zamanı
d_i	: i. işin teslim tarihi
D_t	: t proses elemanının istenen çıktısı
E_i	: i. işin negatif gecikme ya da erken bitme zamanı
E_t	: t proses elemanının hata miktarı
F_i	: i'inci işin akış zamanı
$I_{.j}$: proses elemanı
I_j	: i proses elemanının çıktısı
L_i	: i. işin gecikme zamanı
MS	: Mutlak sapma
Net_i	: i proses elemanının net girdisi
N_i	: i. işin operasyon sayısı
O_p	: p proses elemanının çıktısı
O_t	: t proses elemanının çıktısı
P_i	: i. işin işlem zamanı
r_i	: i. işin hazır zamanı
SK	: Sapma karesi
T_i	: i. işin pozitif gecikme zamanı
W_{ij}	: i ve j proses elemanları arasındaki bağlantı ağırlıkları
X	: Bağımsız değişken
Y	: Bağımlı değişken

EETT	:	En erken teslim tarihli
EKİS	:	En kısa işlem süreli
JIQ _i	:	i. iş için kuyruklarda bekleyen iş sayıları toplamı
KİS	:	Kuyruktaki iş sayısı
NGİS	:	Negatif geciken iş sayısı
OGST	:	İşlerin ortalama gecikme süreleri toplamı
OMS	:	Ortalama mutlak sapma
ONG	:	Ortalama negatif gecikme
OPG	:	Ortalama pozitif gecikme
OS	:	Operasyon sayısı
OSK	:	Ortalama sapma karesi
OST	:	Operasyon süreleri toplamı
AOST	:	Atölyedeki operasyon süreleri toplamı
PGİS	:	Pozitif geciken iş sayısı
RAS	:	Rassal teslim tarihi belirleme yöntemi
SAB	:	Sabit teslim tarihi belirleme yöntemi
TİS	:	Atölyedeki toplam iş sayısı
TİY	:	Toplam iş yüküne göre teslim tarihi belirleme yöntemi
TOS	:	Kuyruklarda bekleyen işlerin toplam operasyon sayısı
WIQ _i	:	i. iş için atölyedeki toplam iş zamanı
YSA	:	Yapay sinir ağı

ŞEKİLLER LİSTESİ

Şekil 1.1.	Teslim tarihi atama yöntemleri.....	7
Şekil 3.1.	Regresyon modelleri ile teslim tarihleri belirleme	31
Şekil 3.2.	Bir proses elemanının genel yapısı.....	37
Şekil 3.3.	Bir geriye yayılım sinir ağının topolojisi.....	42
Şekil 3.4.	Önerilen yapay sinir ağı modeli.....	48
Şekil 3.5.	Yapay sinir ağı hata grafiği (EKİS).....	55
Şekil 3.6.	Yapay sinir ağı hata grafiği (EETT).....	57
Şekil 4.1.	Kümülatif ortalama iş sayıları.....	65
Şekil 4.2.	Benzetim yazılımı ana program akış şeması.....	78
Şekil 4.3.	Variş() fonksiyonu akış şeması.....	80
Şekil 4.4.	Bitiş() fonksiyonu akış şeması.....	83
Şekil 4.5.	Yapay sinir ağı yazılımı ana program akış şeması.....	86
Şekil 4.6.	YeniAğ() ve EskiAğ() fonksiyonları akış şeması.....	88
Şekil 4.7.	YenidenÇağır() ve Test() fonksiyonları akış şeması.....	94

TABLolar LİSTESİ

Tablo 3.1.	Regresyon modellerinin katsayılarının belirlenmesi için veri seti örneđi (EKİS).....	33
Tablo 3.2.	Regresyon modellerinin katsayılarının belirlenmesi için veri seti örneđi (EETT).....	33
Tablo 3.3.	Regresyon modelleri katsayıları (EKİS).....	34
Tablo 3.4.	Regresyon modelleri katsayıları (EETT).....	34
Tablo 3.5.	Toplama ve transfer fonksiyonu örnekleri.....	38
Tablo 3.6.	Yapay sinir ađının girdileri.....	47
Tablo 3.7.	Yapay sinir ađı eğitim seti (EKİS).....	50
Tablo 3.8.	Yapay sinir ađı eğitim seti - Normalize edilmiş deđerler (EKİS).....	51
Tablo 3.9	Yapay sinir ađı parametreleri (EKİS, EETT).....	54
Tablo 3.10.	Yapay sinir ađı öğrenme başarıları.....	54
Tablo 3.11.	Yapay sinir ađı girdi seti (EKİS).....	59
Tablo 3.12	Yapay sinir ađı girdi seti - Normalize edilmiş deđerler (EKİS).	60
Tablo 4.1.	İşlere ait rotalar.....	63
Tablo 4.2.	Rassal sayı üretimi.....	64
Tablo 4.3.	Oluşturulan işe ait bilgiler.....	81
Tablo 5.1.	TİY ve EKİS ile yapılan benzetim sonuçları.....	97
Tablo 5.2.	OS ve EKİS ile yapılan benzetim sonuçları.....	98
Tablo 5.3.	TİY+OS ve EKİS ile yapılan benzetim sonuçları.....	98
Tablo 5.4.	KİS ve EKİS ile yapılan benzetim sonuçları.....	99
Tablo 5.5.	KİY ve EKİS ile yapılan benzetim sonuçları.....	99
Tablo 5.6.	TİY ve EETT ile yapılan benzetim sonuçları.....	100
Tablo 5.7.	OS ve EETT ile yapılan benzetim sonuçları.....	101
Tablo 5.8.	TİY+OS ve EETT ile yapılan benzetim.....	101
Tablo 5.9.	KİS ve EETT ile yapılan benzetim sonuçları.....	102

Tablo 5.10.	KİY ve EETT ile yapılan benzetim sonuçları.....	102
Tablo 5.11.	YSA ve EKİS ile yapılan benzetim sonuçları.....	103
Tablo 5.12.	YSA ve EETT ile yapılan benzetim sonuçları.....	104
Tablo 5.13.	EKİS için modellerin performans ölçütleri ortalama değerleri..	105
Tablo 5.14.	EETT için modellerin performans ölçütleri ortalama değerleri.	105

ÖZET

Anahtar Kelimeler: Dinamik Çizelgeleme, Teslim Tarihi, Yapay Sinir Ağları

Bu çalışmada dinamik atölye çizelgelemede teslim tarihi belirleme yöntemlerine alternatif olarak daha iyi sonuçlar üretebileceği düşünülen bir yapay sinir ağı yaklaşımı sunulmakta ve olurluluğu gösterilmektedir. Yapay sinir ağı modelinin performansını ölçebilmek amacıyla karşılaştırma yapılabilecek beş farklı regresyon modeli seçilmiştir.

Çalışmayı yapabilmek için öncelikle olay artımlı bir benzetim yazılımı geliştirilmiştir. Yazılımın ürettiği verilerle regresyon model denklemleri oluşturulmuş ve ayrıca yapay sinir ağı modelinin teslim tarihi belirlemek üzere eğitimi yapılmıştır. Yapay sinir ağı modeli olarak geriye yayımlı yapay sinir ağı kullanılmış ve bir yapay sinir ağı yazılımı geliştirilmiştir. Regresyon modelleri oluşturulduktan ve yapay sinir ağı eğitildikten sonra benzetim yazılımı tekrar bu defa karşılaştırma amaçlı olarak çalıştırılmıştır. Regresyon modelleri için her bir modeldeki denklem kullanılarak teslim tarihleri belirlenerek benzetim yapılarak performans ölçütleri hesaplanmış ve saklanmıştır. Yine yapay sinir ağı kullanılarak teslim tarihlerinin belirlendiği benzetim yapılmış ve performans ölçütleri kaydedilmiştir.

Söz konusu bu çalışmalar en kısa işlem süreli ve en erken teslim tarihli öncelik kuralları için ayrı ayrı yapılmıştır. Modelleri karşılaştırmak amacıyla performans ölçütleri olarak teslim tarihinden ortalama mutlak sapma, teslim tarihinden mutlak sapmalarının kareleri ortalaması, ortalama pozitif gecikme, pozitif geciken iş sayısı, ortalama negatif gecikme ve negatif geciken iş sayıları kullanılmıştır. Çalışma sonucunda yapay sinir ağı modelinin teslim tarihi belirlemede olurluluğu görülmüştür. Hem en kısa işlem süreli önce ve hem de en erken teslim tarihli önce öncelik kuralları açısından birkaç performans ölçütü açısından iyi sonuçlar vermiştir. Yapay sinir ağının en kısa işlem süreli performans ölçütünde genel olarak daha iyi sonuç verdiği görülmüştür.

DUE DATE DETERMINATION IN DYNAMIC JOB SHOP SCHEDULING USING ARTIFICIAL NEURAL NETWORKS

SUMMARY

Keywords: Dynamic Scheduling, Due Date, Artificial Neural Networks

In this study, an artificial neural network approach is considered to perform better than conventional due date determination methods in dynamic job shop scheduling is presented and its feasibility is showed. Five different regression models are selected to compare the performance of the artificial neural network model.

An event oriented simulation software is developed to perform this study. Data produced by this software are used to build the regression models and to train the artificial neural network. Building the regression models and training the artificial neural network, the simulation software is run to compare these models. Performance measures are computed and recorded using the regression models. Also simulation software is run using the artificial neural network model to determine due dates and the performance measures are recorded.

Studies mentioned above are done for each dispatching rule of shortest processing time and earliest due date. Performance measures are mean absolute deviation of lateness, mean squared error of lateness, mean tardiness, number of tardy jobs, mean earliness and number of early jobs. At the end of the study, it is observed that the artificial neural network model is feasible to determine due dates. Better results are obtained in some performance criteria for the shortest processing time and the earliest due date dispatching rules. In general, the artificial neural network model gives the better results for the shortest processing time rule.

BÖLÜM 1. GİRİŞ

1.1. Giriş

Bir üretim sisteminde, karar vermeyi gerektiren faaliyetler hiyerarşik olarak üç seviyede oluşur. Bunlar; stratejik, taktik ve kontrol seviyesidir. Stratejik seviyede, üretim planlarının pazar taleplerini karşılaması istenir. Taktik seviyede, planlanan üretim çizelgesi, stok, makina kapasitesi, bakım planı ve işçi verimliliği gibi bazı atölye kısıtları ile koordine edilir. Kontrol seviyesinde ise, planlanan üretim çizelgelerinin uygulanmasını gerçekleştirmek için işlerin akışı sürekli olarak düzenlenir ve beklenmedik olaylarla bozulan çizelgeler derhal güncelleştirilir.

Temel olarak, üretim yönetimi, atölye verimliliğini artırmada iki yönteme sahiptir. Birincisi, ilk iki seviye faaliyetlerinin amacı olan atölyedeki işlemleri etkin bir şekilde planlamaktır. İkincisi ise, üçüncü seviye faaliyetlerinin amacı olan bu planların atölye bazında yürütülmesini sağlamaktır. Bir iş sevkıyat planına dayanarak işler atölyeye sevk edildiği zaman planlanan üretim çizelgesi, atölyede gerçekleştirilmelidir. Başka bir deyişle, işler atölyede işlenmelidir.

Atölyede işlerin makinalarda veya iş merkezlerinde işlenmesi ve akışı ile ilgili problemler atölye bazlı problemlerdir. Eğer üretim faaliyetleri planlandığı gibi gerçekleşseydi atölye bazındaki problemleri çözmek çok kolay olacaktı. Ancak, atölyenin dinamik ortamı yüzünden, atölye bazındaki problemler genellikle çok karmaşıktır. Ayrıca, bu problemlere ait kararlar, zaman kısıtı altında alınır ve çok fazla bilgiyi gerektirir. Üretim çizelgeleme problemleri bunların bir örneğidir.

1.2. Çizelgeleme Problemleri

Üretim çizelgeleme, bir ürünü oluşturan iş parçalarının eldeki tek veya çok sayıda makinalarda hangi sırada ve ne zaman işleneceğinin saptanmasıdır (Taşgetiren, 1999). Üretim çizelgeleme problemleri, üretim tipine göre çok farklı biçimlerde olabilir. Literatürde, üretim çizelgeleme problemleri için pek çok sınıflandırmalar yapıldığı görülür. Herhangi bir sınıflandırmanın amacı, problem sınıflarının anlaşılmasını sağlamak ve her bir sınıfın farklı özelliklerini saptamaktır. Graves (1981), üretim çizelgeleme problemlerinin beş boyutlu bir sınıflandırmasını yapmıştır.

İlk boyut, gereksinim üretmedir. Gereksinimler açık ve kapalı atölyelerde üretilir. Açık atölyede gereksinimler doğrudan doğruya müşteri siparişleriyle üretilir. Kapalı bir atölyede ise gereksinimler stoktan karşılanır ve üretim görevleri sadece mevcut stoktan sipariş kararı vermektir.

Gereksinim üretmeye bağlı olarak üretim çizelgeleme problemi, çok farklı biçimlerde olabilir. Açık bir atölye için en basit biçim, siparişlerin her bir makinada sıralandığı sıralama problemidir. Kapalı bir atölye için üretim çizelgeleme problemi, hem sıralama kararlarını hem de yeniden sipariş verme işlemiyle ilgili parti büyüklüğü kararlarını içerir. Şüphesiz, gerçek yaşamda, tamamıyla açık veya kapalı bir atölye ortamı olmamasına rağmen üretim çizelgeleme problemlerinin çoğu açık veya kapalı olarak düşünülür.

İkinci boyut, işlem karmaşıklığı, her bir üretim görevleriyle ilgili işlem kademelerinin sayısı ile ilgilidir. İşlem karmaşıklığı boyutu, kademe sayısına göre dört farklı kısımda incelenebilir:

Tek-Kademe Tek-Makina problemi, en basit problem biçimidir. Burada bütün işler, tek makinada işlenmek üzere tek bir işlem kademesini gerektirir. Çivi üretimi buna bir örnektir. Bu çok basit biçim, çok karmaşık problemler için çözüm aramada başlangıç noktasıdır.

Tek-Kademe Paralel-Makina problemi, tek makina problemine benzemektedir. İkisi arasındaki fark, her bir iş paralel makinaların birisinde işlenmek üzere yine tek bir işlem kademesini gerektirir. Aynı işi yapan makina sayısı fazladır. Yine, çivi üretimi buna örnek gösterilebilir. Çivi imal edilen makinadan birkaç tanesinin atölyede bulunmasıyla oluşan atölye tipidir.

Çok Kademe problemleri, her bir işin işlem sırasında çok kesin bir öncelik ilişkisinin bulunduğu durumlardır. Her bir iş, makinalar gurubunda öncelik ilişkisine göre işlenmeyi gerektirir. Çok kademeli problemler, akış tipi ve atölye tipi olmak üzere iki şekilde incelenebilir. Akış tipi problemde, bütün işler aynı işlem sırasıyla aynı makina gurubunda işlenir. Diğer bir deyişle, işlerin makinalardaki işlem sırası (teknolojik kısıt) ve öncelik ilişkisi aynıdır. Atölye tipi problem ise, sınıflandırmadaki en genel ve en karmaşık olanıdır. Belli bir işe ait işlem kademeleri sayısı üzerine hiçbir kısıt yoktur. Başka bir deyişle, atölye tipi problemde her bir iş, farklı makinalarda işlenmek üzere kendine özgü bir işlem sırasına sahiptir.

Üçüncü boyut, çizelgeleme ölçütü, çizelgenin nasıl değerlendirileceğini gösterir. Çizelge değerlendirmede genel olarak iki ölçüt kullanılır. İlk ölçüt, çizelge maliyetidir. Belirli bir çizelgeye ait maliyet; üretim hazırlıkları ile ilgili sabit maliyetleri, değişken ve fazla mesai maliyetlerini, stok elde bulundurma maliyetlerini ve de siparişleri karşılayamama maliyetlerini kapsar. İkinci değerlendirme ölçütü, çizelge performansdır. Çizelge performansını değerlendirmede kullanılan etkinlik ölçütleri daha sonra ayrıntılı olarak incelenecektir. Bu ölçütler, ya işlerin atölyede harcandığı zamanı enazlamaya yönelik ölçütler ya da gecikmelerin mümkün olduğunca enazlanmasını sağlayacak olan teslim tarihine dayalı ölçütlerdir. Çizelge değerlendirme, genellikle, hem maliyet hem de etkinlik ölçütlerinin karışımına dayanır.

Sınıflandırmadaki dördüncü boyut, gereksinim tanımlama, problemin parametrelerinin başka bir deyişle bütün sayısal değerlerin önceden bilinip bilinmemesiyle ilgilidir. Eğer bütün parametreler miktar olarak önceden biliniyor ve sabit ise, problem deterministik olarak tanımlanır. Aksi halde problem stokastik olarak tanımlanır. Örneğin, açık atölye için işin her bir kademesine ait işlem zamanı

bilinir. Bu durumda, atölye deterministik olarak tanımlanır. Tam tersine, işlem zamanı bilinmeyebilir ve belli bir olasılık dağılımından rastgele üretilen bir değişken olabilir. Bu durumda atölye, stokastik olarak tanımlanır. Benzer şekilde, kapalı bir atölye için stoktan sipariş verme kararlarına yol açan müşteri talep prosesi önceden tahmin edilebilir ve bu durumda problem, deterministik olarak tanımlanır. Yine, tam tersine, proses bilinmeyebilir ve belli bir olasılık dağılımından üretilen rasgele bir değişken olabilir; bu durumda, proses stokastik olarak tanımlanır.

Beşinci boyut, çizelgeleme ortamı, üretilecek gereksinimler için gerekli girdiler üzerine varsayımlarla ilgilidir. Çizelgeleme periyodu boyunca üretilecek gereksinimlerin miktarı ve buna bağlı olarak atölye ortamına giren işlerin miktarı önceden saptanır ve atölye ortamına sonradan ek iş girişi yapılmaz ve de atölye ortamında hiçbir belirsizlik durumu söz konusu değil ise, çizelgeleme ortamı statik olarak tanımlanır. Diğer yanda, problem, çizelgeleme periyodu boyunca üretilecek gereksinimlerin miktarı ve buna bağlı olarak atölye ortamına giren işlerin miktarına sonradan ek iş girişi yapılabilecek biçimde tanımlanabilir. Başka bir deyişle, çizelgeleme periyodu boyunca atölye ortamına herhangi bir anda yeni iş girdileri olabilir ve bu durumda atölye ortamı dinamik olarak tanımlanır. Gerçekte, çizelgeleme problemleri stokastik ve dinamiktir. Ancak, çoğu problem, statik ve deterministik olarak tasarlanır. Bunun nedeni, çeşitli varsayımlarla basitleştirilen statik ve deterministik modellerin iyice anlaşılması, gerçek-zamanlı modellerin ortaya konmasının ilk adımıdır. Ayrıca, mikroişlemciler ve robotlar üretim hatlarına girmekte ve dolayısıyla, işlem zamanlarında belirlilik sağlanmaktadır.

1.3. Genel Atölye Tipi Çizelgeleme

Genel atölye çizelgeleme problemi bir sabit kaynaklar setinde işlenmesi gerekli işler setiyle karakterize edilir. m tane makinada işlenmek üzere n tane iş mevcuttur. Her bir işin her bir makinada sadece ve sadece bir kez işlem gördüğü varsayılır. İşler, makinalarda belli bir sıra dahilinde işlenir ve bu sıra, teknolojik kısıt, iş seyri veya rota olarak adlandırılır. Genel atölye tipi üretim için teknolojik kısıtların oluşumu ile ilgili hiçbir sınırlama yoktur. Makinada işin işlenmesine operasyon denir. Her operasyonun yürütülmesi belirli bir zaman alır. Bu zaman uzunluğu, operasyon

süresi (işlem zamanı) olarak adlandırılır. Basitleştirmek amacıyla, işi yürütmek üzere gereken makinayı ayarlama veya hazırlama için gerekli olan zamanın, yani, hazırlık zamanının, işlem zamanı içinde bulunduğu varsayılır. İş makinaya taşımak amacıyla geçen zamanın da işlem zamanının içinde bulunduğu varsayılır. Her bir iş kendi operasyon sırasına sahiptir ve diğer işlerin operasyon sıralarından bağımsızdır. Bu işler için operasyon sayısı, operasyon süreleri ve rotaların hepsi birbirinden bağımsızdır. Hiyerarşik bir montaj olmadığı kabul edilir. Operasyon sırasında ekstra hazırlıklara maruz kalınacağı için normal olarak öne almaya izin verilmez. Hedef bazı performans ölçütlerini maksimize eden operasyon çizelgesi ya da çizelgeleme kuralını belirlemektir. Yaygın endüstriyel düzenlemelerde problem dinamiktir. Bu problem incelenirken siparişe göre üretim sistemini ifade eden stokastik iş gelişler arası süreleri olduğu varsayılır.

Bununla birlikte bütün işlerin, aynı işlem sırasına sahip olduğunda özel bir durum ortaya çıkar. Böyle durumlarda problem akış tipi çizelgeleme problemi olarak adlandırılır. Atölye çizelgeleme problemi statik ya da dinamik olarak karakterize edilebilir.

1.3.1. Statik çizelgeleme

Statik problem en uygun çizelgeleri elde etmek için en iyi yardımı verir. Statik problemin amacı genellikle maksimum tamamlanma zamanını minimize eden bir sonlu çizelge geliştirmektir. Komşuluk araştırması, dal-sınır araştırması, dinamik programlama ve tamsayılı programlama teknikleri gibi kombinatoral optimizasyon teknikleri kullanılmıştır. Bunların içinde en çok dal-sınır araştırması kullanılmaktadır (Graves, 1981). Statik çizelgeleme probleminin iki makinadan daha fazla olması durumunda NP-hard (polinomal olmayan zor) olduğu gösterilmiştir (Parker ve Rardin, 1982).

1.3.2. Dinamik çizelgeleme

Dinamik problem sonlu çizelge belirlemede en iyi yardımı veremez. Stoğa üretimdekinin aksine bir atölye gelecek iş yüklerini tahmin etmeye yardımcı olacak

ana üretim çizelgesine sahip değildir. Bilinmeyen gelecek iş dalgalanmaları etkin çizelgeleme algoritmaları geliştirme çabalarını sınırlar. Ayrıca, atölye içinde işlerin gelecekteki durumlarını detaylandırmaya çalışan sonlu çizelge teknikleri, işlem süreleri hakkında önemli belirsizliğin olduğu durumda uygun olmayabilir. Sisson (1959) mekanik yaklaşımlar olarak bilinen sonlu çizelgeler geliştiren tekniklere başvurmuştur.

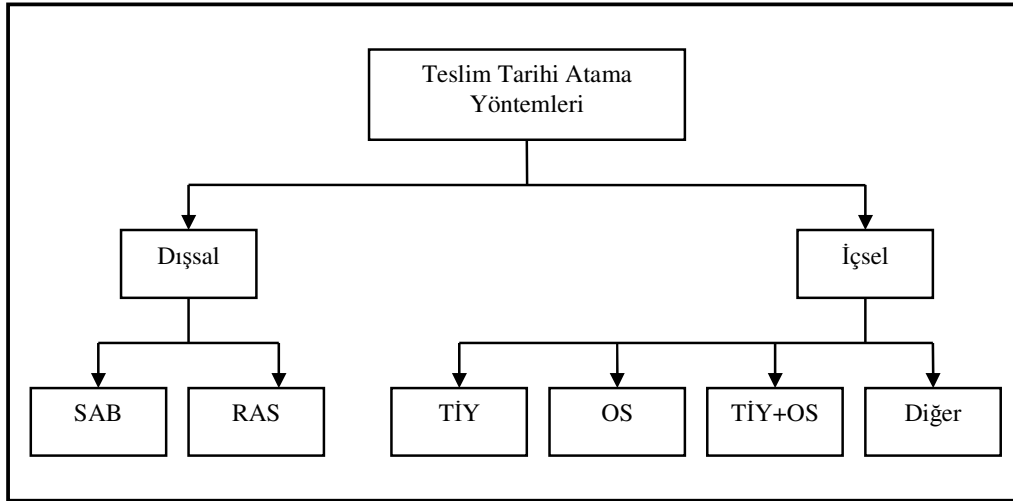
Kuyruk teorisi dinamik problem çalışmasında değerini ispatlamıştır. Önceleri atölyeyi ifade edebilecek bir kuyruk ağı olarak kabul edilmişti. 1957'de çıkarılan kuvvetli bir analitik sonuç, belirli koşullar altında ağıdaki kuyrukların bağımsız olarak incelenmesine izin vermişti (Sisson, 1959). Jackson dekompozisyon prensibi olarak bilinen bu sonuç, atölye içinde çeşitli aşamalarda işlerin kararlı hal olasılıklarını tahmin etmeyi mümkün kılmıştı.

Kesikli olay benzetimi, kullanılan dinamik kontrol ortamı hakkında daha gerçekçi varsayımlara olanak tanımaktadır. Öncelik ve teslim tarihi belirlemek için genellikle kural tabanlı sezgiseller kullanılmıştır. Yerel, tek geçişli öncelik sezgiselleri atölye sıralamasını kontrol eder. İş tamamlanma sürelerinin tahmini iş salıverme zamanında gerekli değildir. Teslim tarihleri çoğunlukla bir akış tahsisat sezgiseli (Teslim tarihi kuralı) kullanılarak belirlenir. Bu teslim tarihleri genellikle tamamlanma süreleri tahminine dayanmamaktadır. Benzetim, farklı sevkıyat ve tahsisat belirleme mekanizmalarının göreceli performansının test edilmesinde kapsamlı bir şekilde kullanılmıştır. Bu en yaygın dinamik atölye çizelgeleme araştırma metodolojisidir.

1.4. Teslim Tarihi Belirleme

Bir üretim sisteminde işlerin teslimi için doğru teslim tarihleri atanmasının önemi akademik araştırmacılar ve pratikteki yöneticiler tarafından iyi anlaşılmıştır. İmalat sistemlerindeki gelişmeler ve envanter sistemlerindeki ideal kavramlardan dolayı, teslim tarihine dayalı araştırma dikkat çekmiş ve bu alanda zengin bir literatür rapor edilmiştir (Cheng, 1989).

Bir üretim sisteminde her bir iş için atölyede işlem görmek üzere serbest bırakılmadan önce teslim tarihi atanır. Literatür analizi teslim tarihi ataması için çeşitli karar kurallarının önerildiğini göstermektedir. Teslim tarihi atama literatürü dinamik çok makinalı durum için teslim tarihi belirlemek için basit, regresyon temelli yaklaşımlar üzerinde durmaktadır (Philipoom, 1994). Eğer teslim tarihleri üretim bölümü tarafından belirlenirse içsel teslim tarihleri olarak, sipariş pazarlama bölümü tarafından müşterilerle yapılan görüşmeler ile belirlenirse dışsal teslim tarihleri olarak anılırlar (Conway, 1965, Ragatz ve Mabert, 1984). Benzer olarak teslim tarihi atama yöntemleri Şekil 1.1'de gösterilen şu kategorilerde incelenebilir (Cheng, 1989, Wagner ve Ragatz, 1994):



Şekil 1.1. Teslim tarihi atama yöntemleri

(1) Dışsal: Bu durumda teslim tarihleri iş geldiğinde bağımsız dış bir kaynak tarafından belirlenir. Sabittirler ve işin verilen bir özelliğidir. Bu kategoride iki tip teslim tarihi atama yöntemi vardır:

- (i) Sabit (SAB): Bütün işlere tamamen aynı akış tahsisatı verilir.
- (ii) Rasgele (RAS): İşin akış tahsisatı rasgele belirlenir.

Bu iki kural şu şekilde formüle edilebilir:

$$\text{SAB: } d_i = r_i + k,$$

$$\text{RAS: } d_i = r_i + e_i,$$

k bir sabit ve e_i bir rassal sayı..

Bu iki metod iş varışları, sistemdeki işler, gelecek olan işler veya atölye karakteristikleri bilgilerini yok sayar.

(2) İçsel: Bu durumda teslim tarihleri, işler atölyeye vardıklarında iş karakteristikleri, atölye bilgisi iş akış zamanı tahminine bağlı olarak çizelgelemeci tarafından belirlenir. Bu kategoride olan bazı teslim tarihi metodları şunlardır:

(i) TİY: Teslim tarihleri toplam iş yüküne bağlıdır.

(ii) OS: Teslim tarihleri işin operasyon sayısına bağlı olarak belirlenir..

(iii) TİY+OS: Teslim tarihleri toplam iş yükü ve işin operasyon sayısına bağlı olarak belirlenir.

Bu metodlar şu şekilde formüle edilebilirler:

$$\text{TİY} \quad : d_i = r_i + kp_i,$$

$$\text{OS} \quad : d_i = r_i + kn_i,$$

$$\text{TİY+OS: } d_i = r_i + k_1p_i + k_2n_i,$$

p_i işin toplam işlem süresi, n_i işin operasyon sayısıdır.

1.4.1. Regresyon analizi

Regresyon analizi, aralarında sebep-sonuç ilişkisi bulunan iki veya daha fazla değişken arasındaki ilişkiyi matematiksel bir fonksiyonla belirlemek, bu ilişkiyi kullanarak bağımlı değişken hakkında tahminler (estimation) ya da kestirimler (prediction) yapabilmek amacıyla sıklıkla başvurulan bir istatistik analiz tekniğidir (Tarı, 1999, Orhunbilge, 2000). Regresyon analizi ile aynı zamanda değişkenler

arasındaki yapısal ilişkiler de ortaya konmuş olur. Ekonomik, sosyal ve doğal olayların çoğunda sebep-sonuç ilişkisine rastlamak mümkündür.

Regresyon modeli uydurulduktan sonra modelin yeterli olup olmadığının kontrolü regresyon analizinin en önemli bölümüdür. Uydurulan modelin doğru modele yeterli derecede yaklaştığını garanti etmek ve en küçük kareler regresyon analizinin tüm varsayımlarını sağlayıp sağlamadığını kontrol etmek gerekir. Regresyon analizinde modelin yeterliliği için genellikle varyans analizi ve çoklu determinasyon katsayısından (R^2) faydalanılır. Modelin yeterliliğinin varyans analizi ile ortaya konması yeterli değildir. Bunun dışında regresyon parametrelerinin de istatistik açıdan anlamlı olması da t testleri ile araştırılmalıdır.

1.4.2. Yapay sinir ağları

Bir yapay sinir ağı (YSA), insan beynine dayalı bir çıkarım mekanizması olarak tanımlanabilir (Negnevitsky, 2002). Diğer bir tanımla, yapay sinir ağları, birbirine paralel bağlı basit elemanlar ile biyolojik sinir sistemlerinin yaptığı gibi gerçek dünya nesneleriyle etkileşimli planlı hiyerarşik yapılardır (Kohonen, 1987). Bir genel sinir ağı modeli proses elemanlarıyla karakterize edilir. Bir proses elemanı beş bileşen içermektedir. Bunlar:

- Girdiler, proses elemanına bilgi getirir. Bu bilgi diğer proses elemanları veya dış kaynaklardan sağlanır. Bazen proses elemanı kendi kendine bilgi verebilir.
- Ağırlıklar, belli bir girdinin proses elemanı üzerindeki etkisini belirler. Ağın doğru çıktuları üretebilmesi için eğitime işlemi sırasında optimize edilmesi gereken ağırlık değerleridir.
- Toplama fonksiyonu, proses elemanının ağırlıklı girdilerini toplar. Çeşitli toplama fonksiyonları vardır (Neuralware Inc, 1990). En yaygın olanı ağırlıklı toplamı bulmaktır. Burada gelen her girdi değeri, kendi ağırlığı ile çarpılarak toplanır (Oztemel, 2003).
- Transfer fonksiyonu, toplama fonksiyonunun sonucunu düzenleyerek proses elemanının çıktısını belirler. Yine çeşitli transfer fonksiyonları vardır

(Neuralware Inc, 1990). Popüler olanlarından bazıları, sigmoid fonksiyon, lineer fonksiyon, step fonksiyonudur.

- Çıktı, transfer fonksiyonunun sonucunu, bağlı olan proses elemanlarına veya dış kaynaklara gönderir.

Ağın topolojisi, ağı karakterize eden ikinci özelliğidir. Bir proses elemanları grubu katman adı verilen yapıyı oluştururlar. Tipik bir sinir ağı bir diğerine bağlı üç katman içerir. Bunlar, dış dünyadan girdi kabul eden girdi katmanı, girdi katmanından gelen bilgileri işleyerek çıktı katmanına gönderen gizli katman (veya gizli katmanlar) ve ağı karar hakkında dış dünyaya bilgi veren çıktı katmanıdır. Bilgi ağı katmanları arasında veya katmanlar içinde akar.

1.5. Öncelik Kuralları

Öncelik kuralları bir makina önündeki kuyrukta, o makina serbest hale geldiğinde iş ataması yapılması gerektiğinde birden fazla iş olduğunda kullanılmaktadır. Yapılan çeşitli çalışmalarda atölyenin yüklü olduğu durumlarda en kısa işlem süreli önce kuralının ortalama pozitif gecikme performans ölçütü açısından daha iyi sonuçlar verdiği görülmüştür (Philipoom, 1994). En erken teslim tarihli önce kuralı teslim tarihi önce olan işe öncelik verdiği için, bu kuralın da iyi sonuçlar verebileceği düşünülmüştür. Bu nedenlerden çalışmada öncelik kuralları olarak en kısa işlem süreli önce ve en erken teslim tarihli önce olmak üzere iki adet öncelik kuralı kullanılmıştır. Her iki öncelik kuralı çizelgeleme açısından farklı akış zamanlı, dolayısıyla farklı tamamlanma zamanlı çizelgeler doğurduğundan her iki kural için incelenen dinamik atölyenin ayrı ayrı benzetimleri yapılmıştır.

1.6. Performans Ölçütleri

Performans ölçütleri (kriterleri) bir çizelgenin başarısı hakkında yorum yapmada yardımcı olmaktadır. Dolayısıyla çizelgenin bir performans ölçütüne göre iyi olup olmadığından bahsedilebilir. Bu çalışmada teslim tarihi belirlemek için bir yapay sinir ağı modeli önerilmiştir. Önerilen model seçilen regresyon modelleri ile performans açısından karşılaştırılmıştır. Teslim tarihleri söz konusu olduğundan

performans ölçütleri olarak, teslim tarihlerinin gerçek tamamlanma zamanlarından olan ortalama mutlak sapmaları, ortalama mutlak sapma kareleri, gerçekleşen tamamlanma zamanlarından pozitif ve negatif gecikmeleri seçilmiştir. Karşılaştırmalar bu ölçütler üzerinden yapılmıştır. Ayrıca her iki tür model için fikir vermesi bakımından pozitif ve negatif geciken iş sayıları da hesaplanmıştır.

1.7. Varsayımlar

Dinamik atölye benzetimi için literatürde hem gerçek hem de düşünsel sistemler bulunmaktadır. Düşünsel sistemler tipik olarak az sayıda makina içermekte (genellikle 10'dan az) ve benzetim çalışmalarında şu varsayımlar yapılmaktadır (Baker, 1974, French, 1982):

- Herbir iş bir bütündür. İş farklı operasyonlardan oluşmasına rağmen, aynı işin iki operasyonu hiçbir şekilde aynı anda işlenemez. Böylece bazı pratik problemleri tartışmamızın dışına çıkarabiliriz. Şöyle ki; son ürün için montaj edilmeden önce komponentlerin aynı anda imal edildiği durumlar.
- İş bölme yoktur. Her bir operasyon, başladığı zaman, diğer operasyon o makinada başlatılmadan önce tamamlanmalıdır.
- Her bir iş, her bir makinada bir tane olmak üzere, m tane farklı operasyona sahiptir. İşin aynı makinada iki defa işlem görmesi olasılığı hesaba katılmaz.
- İş iptali söz konusu değildir. Her bir iş tamamlanıncaya dek işlenmelidir.
- İşlem zamanları çizelgeden bağımsızdır. Burada iki şey var sayılmaktadır: Her bir hazırlık zamanı sıradan bağımsızdır. Yani, işe ait makinayı ayarlamak için gereken zaman en son işlem gören işten bağımsızdır. Makinalar arasında işleri taşımak için gereken zaman ihmal edilmektedir.
- Ara stoğa izin verilir. İşler bir sonraki makinanın boşalması için bekleyebilir.
- Makinanın her bir tipinden sadece bir tane vardır. İşlerin işlenmesi esnasında aynı işi yapan birden fazla makinanın olmadığı varsayılır. Bu varsayım, beklemekten kaçınmak için belli makinaların çoğaltılması durumunu elimine eder.
- Makinalar boş kalabilir.
- Hiçbir makina, bir kerede birden fazla operasyonu işleyemez.

- Makinalar asla bozulmaz ve çizelgeleme periyodu boyunca elverişlidir.
- Teknolojik kısıtlar önceden bilinir ve sabittir.

1.8. Çalışmanın İçeriği

Bu çalışmanın içeriği şu kısımlardan oluşmaktadır:

- Teslim tarihi belirlemede kullanılacak olan regresyon modelleri belirlenebilmesi ve yapay sinir ağı modelinin eğitimi için veri sağlayacak olan dinamik atölye çizelgeleyen olay artımlı benzetim yazılımı geliştir.
- En kısa işlem süreli (EKİS) önce ve en erken teslim tarihli (EETT) önce öncelik kuralı için iki ayrı benzetim çalıştırarak gerekli verileri elde et.
- Yapay sinir ağını modelleyebilmek için geriye yayılım algoritması kullanan bir yapay sinir ağı yazılımı geliştir.
- Benzetimden elde edilen verilerle regresyon modellerini oluşturabilmek için regresyon model denklem katsayılarını bul. İstatistik olarak anlamlı katsayılar olup olmadıklarını araştır.
- Aynı benzetimden elde edilen sonuçlarla bu defa teslim tarihi belirleyecek olan yapay sinir ağını eğit.
- Kurulan regresyon modelleriyle teslim tarihlerini belirleyerek benzetimi çalıştır. Gerekli olan performans ölçütleri değerlerini elde et. Bu işlemleri EKİS ve EETT önce öncelik kuralları için yap.
- Benzetim yazılımının eğitilmiş yapay sinir ağını kullanarak modelleme yapabilmesi için benzetim yazılımında gerekli düzenlemeleri yap.
- Eğitilmiş yapay sinir ağı teslim tarihlerini belirleyerek benzetimi çalıştır. Gerekli performans ölçütleri değerlerini elde et. Bu işlemi EKİS ve EETT önce öncelik kuralları için yap.
- Regresyon modelleri ile yapay sinir ağı modelinin teslim tarihi belirleme performanslarını elde edilmiş olan performans ölçütlerine göre karşılaştır.
- Sonuçların anlamlı olup olmadıklarını istatistik analiz ile test et.

1.9. Tezin Amacı

Bu çalışmanın amacı yapay sinir ağının dinamik atölye çizelgelemede teslim tarihi belirlemede olurluluğunu göstermektir. Bu amaçla olay artımlı benzetim yapan dinamik değişkenlerin kullanıldığı bir yazılım geliştirilmiştir. Yine yapay sinir ağının modellenmesi amacıyla dinamik değişkenlerin kullanıldığı bir geriye yayılım yapay sinir ağı yazılımı sunulmuştur. Regresyon modelleri ve yapay sinir ağı ile teslim tarihleri belirlenerek seçilen performans ölçütlerine göre karşılaştırmaları yapılmıştır. Regresyon modelleri ile teslim tarihi belirlerken gerekli olan verilerdeki eksiklik veya olabilecek olası bir hata teslim tarihi bilgisinin olması gerekenden çok farklı hesaplanmasına yol açabilir. Yapay sinir ağı ise eksik veriyle çalışabilme özelliğinden dolayı böyle bir durumda uygun cevaplar verebilir.

Kullanılan regresyon modellerinde teslim tarihi belirlenirken kullanılan girdi verileriyle aradaki ilişki doğrusal bir denklem ile kurulmaktadır. Oysa yapay sinir ağının girdi verileriyle teslim tarihi arasındaki ilişkiyi kurmadaki yolu doğrusal değil, çok daha farklı ve ilişkiyi ağırlık değerleriyle kurduğundan daha iyi ifade edebilmektedir. Bu nedenlerle yapay sinir ağlarının teslim tarihi belirlemede kullanılabileceği ve daha iyi sonuçlar verebileceği düşünülmüştür.

1.10. Tezin Düzenlenmesi

Yapılan bu çalışma 6 bölümden oluşmaktadır:

Bölüm 1, çizelgeleme problemleri, genel atölye çizelgeleme problemini, dinamik atölye çizelgeleme ve teslim tarihi belirlemedeki çözüm yaklaşımları, tez içeriği ve amacı hakkında bilgi vermektedir.

Bölüm 2'de çizelgeleme problemleri ve teslim tarihi belirleme ile ilgili literatür taramasını içerir. Bu bölümde çözüm yaklaşımları daha ayrıntılı verilmektedir.

Bölüm 3 teslim tarihinde kullanılan regresyon modelleri, regresyon, önerilen yapay sinir ağı modeli ve yapay sinir ağları hakkında detaylı bilgiler vermektedir.

Bölüm 4’de deneysel tasarımın nasıl olduđu açıklanmıştır. Kullanılan dinamik atölyeye ait makina sayıları, iş rota çeşitleri, gelişler arası ve operasyon süreleri hakkında bilgi verilmiştir. Geliştirilen olay artımlı benzetim ve yapay sinir ağı yazılımları akış şemaları ve algoritma yapıları verilerek detaylı olarak anlatılmıştır. Bölüm 5’de deneysel sonuçlar tablolar halinde verilmiştir. Elde edilen sonuçlar istatistik testleri ile birlikte ele alınarak yorumlanmıştır.

Bölüm 6’da sonuçlar yorumlanmış, çalışmanın ve bulguların özeti verilmiştir. Çalışmanın katkısı ve çalışma alanları hakkında bilgi verilmiştir.

BÖLÜM 2. LİTERATÜR TARAMASI

2.1. Giriş

Bir atölye için üretim kontrol sistemi üç sıralı aşamadan oluşan bir yapıda incelenebilir (Philipoom, 1994): Sipariş aşaması, sipariş serbest bırakma ve atölye aşaması. İlk aşamada müşterinin işi ulaşır ve bir teslim tarihi verilir. Teslim tarihi atanması atölye kontrolün ilk önemli görevidir. Teslim tarihi ilişkili performans teslim tarihi atama kurallarının kalitesiyle kendini gösterir. Teslim tarihi ataması ve müşteriye zamanında teslim edilen ürünler müşteri memnuniyeti ve rekabet avantajı sağlayacaktır (Sha ve Hsu, 2004). Teslim tarihi belirleme ile ilgili literatürdeki çalışmalar bu bölümde sayısal, bulgusal ve yapay sinir ağları olmak üzere üç grupta incelenmiştir.

2.2. Sayısal Yöntemler

Bu yöntemlerde problem matematiksel modellerle tanımlanmaktadır. Oluşturulan bu modeller matematik programlama teknikleriyle çözümlenerek en uygun çözüm aranmaktadır.

Fry ve diğerleri (1989) çalışmalarında, iş akış zamanını tahmin etmek için iş ve atölye karakteristiklerinden hangisinin daha etkili olduğunu araştırdılar. Bunu anlamak için hangi faktörün, gerçek akış zamanından planlı tedarik süresinin sapmasına etkili olduğunu bulmaya çalıştılar. İşin akış zamanına etki eden faktörler belirlendikten sonra teslim tarihi belirleme kuralları geliştirilmiştir. Daha sonra benzetim kullanılarak hangi teslim tarihi atama kuralının değişen ürün yapısı ve atölye şartlarında daha iyi performans gösterdiğini belirlemişlerdir.

Ahmed ve Fisher (1992) atölye tipi çizelgelemede, teslim tarihi belirleme, atölyeye sipariş açma ve sıralama arasındaki etkileşimi incelemişlerdir. Araştırmada dinamik 5 makinalı bir atölye tipi üretim seçilmiştir. Benzetim modeli oluşturulurken, performans kriteri olarak yarı mamul maliyeti, nihai mamul elde bulundurma maliyeti ve gecikme cezası ile birlikte maliyet unsuru içermeyen diğer kriterler de dikkate alınmıştır. Sonuç olarak, belirtilen bu 3 faktörün (teslim tarihi, sipariş açma metodu ve sıralama metodu) etkileşiminin önemi, bu faktörlerin ayrı ayrı belirlenmesinin kadar önemli olduğu vurgulanmıştır.

Taşgetiren ve diğerleri (1995), atölye çizelgelemede bir işin tamamlanma zamanı ve işin sistemdeki kuyruk zamanı gibi bir çok faktörden etkilendiğini, aynı zamanda her bir faktörün etki seviyesinin kullanılan öncelik kuralı ve atölye yüküne bağlı olduğunu belirttiler. Çeşitli atölye yükleri altında bu faktörlerin teslim tarihi üzerindeki etkilerini incelemek için birkaç yöntem karşılaştırması yaptılar. Sonuç olarak, teslim tarihi oluşturma yöntemlerinin tahmin yeteneklerinin birbirinden farklı olduğunu ve bu yöntemlerin performanslarının kullanılan öncelik kuralından etkilendiğini gördüler. Ayrıca tahsisata dayalı öncelik verme yaklaşımının akış zamanına dayalı yaklaşımdan daha iyi sonuç verdiğini belirttiler.

Mosheiov (2001) paralel benzer makinalarda teslim tarihi atama problemi ve atölye çizelgeleme üzerine çalışmıştır. Bir çizelgenin maliyetinin maksimum negatif gecikme maliyeti, maksimum pozitif gecikme maliyeti ile teslim tarihi maliyetinin bir fonksiyonu olduğunu ileri sürdü. Çalışmanın amacı bu üç maliyet fonksiyonunu minimize eden bir teslim tarihi belirleme algoritmasının geliştirilmesi idi.

Biskup ve Jahnke (2001), işlere genel bir teslim tarihi ataması ve işlerin tek bir makinada çizelgelenmesi konusu üzerinde durdular. İşlem zamanlarının kontrol edilebilir olduğunu düşündüler. Fakat, önceki yaklaşımların aksine bütün işlem zamanlarının aynı oranda azaltılabileceği durum üzerinde durdular. Teslim tarihi atamanın yanı sıra negatif gecikme, pozitif gecikme geç kalan iş sayısını minimize etmeye konsantre oldular. Polinomal olarak çözülebilen algoritmalar buldular.

Veral (2001) yaptığı çalışmada işlem akış zamanı analizi ile statik teslim tarihi belirlemenin mümkün olduğunu bulmaya çalışmışlardır. Önerilen model ile TİY modeli mukayese edilmiştir. Karşılaştırma atölyenin hafif, orta ve çok yüklü olduğu durumlar göz önüne alınarak gerçekleştirilmiştir. Yazarın modeli TİY'ne kıyasla 3 farklı atölye yükünde de daha iyi sonuç vermiş ve iş akışı zaman tahmin doğruluğunda ise daha iyi performans göstermiştir.

Gupta ve diğerleri (2002) permütasyon akış tipi problemi üzerinde çalıştılar. Bu çalışmada her iş merkezi paralel benzer makinalardan meydana gelmiştir. Her bir işin farklı serbest bırakma tarihleri vardır ve farklı iş merkezlerindeki makinalarda aynı sırada işlem görürler. Çalışmada 20 iş ve 10 iş merkezi dikkate alınmıştır. Ayrıca amaç fonksiyonuna teslim tarihi atama maliyeti ilave edilmiştir.

Wang ve Uzsoy (2002) yaptıkları çalışmada metal işleme ve mikro elektronik endüstrilerinde kullanılan parti işleme makinalarında, işlerin dinamik olarak atölyeye gönderildiği zaman, iş teslim tarihinin olurlu olup olmadığını araştırmışlardır. Dinamik programlama algoritması ile birlikte genetik algoritma tekniği de kullanılmıştır. Çalışmanın mükemmel ortalama performans gösterdiği sonucuna varılmıştır.

Sabuncuoğlu ve Comlekci (2002), detaylı iş, atölye ve makina dengesizlikleri gibi işlerin operasyonları hakkında rota bilgileri kullanan yeni bir akış zamanı tahmin metodu önerdiler. Bu tür bilgilerin şimdilerde bilgisayar bütünleşik imalat sistemlerinde olduğunu belirtmektedirler. Önerdikleri bu metodun performansını çeşitli deneysel koşullar altında benzetim ile ölçtüler. Mevcut akış zamanı tahmin metotlarıyla çeşitli performans ölçütleri açısından karşılaştırma yaptılar. İmalat sistemleri performansının basit metotlardan (TİY) ziyade yoğun bilgi kullanan metotlarla geliştirilebildiğini gördüler. Akış zamanı tahmininde detaylı bilgi kullanmanın, sistem performansını artırma konusunda daha bütünleşik bilgi kullanan metotlara göre önemli gelişmeler sağladığını iddia ettiler. Çalışma sonuçlarının, her bir operasyon için akış zamanı tahmin etmenin geleneksel işe bağlı tahminden daha iyi bir yaklaşım olduğunun görüldüğünü belirttiler.

Song ve diğeri (2002) ürün teslim tarihinin belirlenmesinin üretim planlamasının önemli bir parçası olduğunu belirterek karmaşık çok safhalı montaj işlemlerinde ürün teslim tarihinin belirlenmesi konusunda çalışmışlardır. Ürün teslim süreleri negatif gecikme ve pozitif gecikmeyi minimize etmek için kullanılmıştır.

Sha ve Liu (2005) tam zamanında üretim felsefesinin önem kazanmasına rağmen, siparişleri zamanında teslim edebilmesi müşteri memnuniyetini artıracığı ve bu durumda işletmeye rekabetçi bir avantaj sağlayacağını ileri sürmektedirler. Bu çalışmada teslim tarihi belirleme kurallarından olan TİY'nün performansını arttırmak için dinamik atölye koşullarını EĞER-İSE kuralları ile temsil etmişler ve işin gelişinde atölyedeki durum değerlendirilerek k katsayısı belirlenmiş, bu şekilde TİY metodunun teslim tarihi hatasını azaltmışlardır. Sonuç olarak kurala dayalı TİY yöntemi statik ve dinamik TİY metotlarına göre iyi sonuç vermiştir.

Shabtay ve Steiner (2006) siparişlerin zamanında teslim edilmesinin çizelgeleme ve tedarik zincir yönetiminin en önemli konularından birisi olduğunu ileri sürdüler. Yazarlar tek makina problemi için ağırlıklı negatif gecikme, pozitif gecikme ve teslim tarihi atama cezalarının minimize edilmesi ile ağırlıklı geç kalan işlerin sayısı ve teslim tarihi atama maliyetlerinin minimize edilmesini amaçladılar.

2.3. Bulgusal Yöntemler

Bulgusal yaklaşımlarda deneysel çalışmalardan elde edilen bulgulara dayanarak çözüm, taranacak alanın daraltılmasıyla bulunur. Bulgusal tarama olarak adlandırılır ve gelişmiş tarama algoritmalarına dayanmaktadır (Taşgetiren, 1996).

Bock ve Patterson (1990) çok projeli çizelgeleme probleminde, teslim tarihi oluşturmak için bir sezgisel geliştirmişlerdir. Yazarlar çalışmalarında bir benzetim modeli ile 4 teslim tarihi oluşturma sezgiseli, 5 kaynak atama sezgiseli ve 3 adet kaynak önceden atama sezgiseli kullanmışlardır. Çalışmalarının sonucu, önceden incelenen teslim tarihi belirleme kurallarının etkinliğini destekler nitelikte olmuştur.

Bagchi ve diğeri (1994) tek makina çizelgeleme konusunda çalışmışlardır. Amaçları müşteri siparişlerinin teslim tarihlerinin belirlenmesi ve toplam ceza fonksiyonunu minimize edecek şekilde işleri çizelgelemektir. Toplam ceza fonksiyonu ise iş negatif gecikmesi ve pozitif gecikmesinin toplamından ibarettir. Sonuç olarak bu çalışmada etkili bir prosedür genel problemleri çözmek için ceza fonksiyonu kullanılarak elde edilmiştir. Bu işlem yapılırken problem iki kısma ayrılmıştır. Önce müşteri siparişleri sıralanmış, daha sonra müşteri siparişinin içinde işler sıralanmıştır.

Cheng ve Jiang (1998) literatürde negatif gecikme ve pozitif gecikme ile ilgili olan kaçırılan teslim tarihi performansı ile ilgili yayın eksikliği bulunduğunu ileri sürmüşlerdir. Çalışmalarında iş atama ve teslim tarihi belirleme kurallarının bir atölye tipi üretimde kaçırılan teslim tarihi performansını nasıl etkileyeceğini araştırmışlardır. Yeni bir iş atama ve iki farklı teslim tarihi belirleme modeli geliştirilmiştir. Teslim tarihi ve iş atama kurallarının kombinasyonu halinde iyi sonuçlar alındığı ifade edilmiştir.

Soroush (1999) simültane teslim tarihi problemi üzerinde çalıştı. n işi bir makina üzerinde sıralarken, işlem süreleri tesadüfi olarak üretildi. Sonuç olarak optimal teslim tarihlerini belirlemek için analitik bir yaklaşım getirdiler. Optimal sıralama için iki adet etkin sezgisel metot önerdiler.

İmalat yöneticileri için teslim tarihi belirlemenin zor bir durum olduğunu belirte Philipoom (2000), yaptığı bir çalışmada temin süresi ve pozitif gecikme uzunluğu cezalarına bağlı olarak teslim tarihi belirlenen bir atölyede öncelik kuralı seçimindeki eğilimleri inceledi. Çalışmasının sonucunda en kısa işlem süreli önce kuralının yalın pozitif gecikme cezaları açısından iyi çalıştığını belirtti. Pozitif gecikme için ceza arttıkça ilk giren ilk önce gibi öncelik kurallarının iyi çalıştığını gördü. En erken teslim tarihli önce öncelik kuralı ile teslim tarihi belirleme kuralı parametrelerinin etkileşimli olmasından dolayı en erken teslim tarihli önce kuralının iyi sonuç vermediğini ifade etti.

Yang ve Wang (2001) atölye çizelgelemede yeni bir adaptif yapay sinir ağı ve sezgisel hibrid yaklaşım sundular. Olurlu çözüm sırasında sinir ağı, proses elemanlarının bağlantı ağırlıkları ve bias değerlerini adapte etme özelliğine sahiptir. Yapay sinir ağı ile birleştirilebilen iki sezgisel sundular. Bunlardan biri yapay sinir ağının çözümünü hızlandırmakta ve ağın yaklaşımını garanti etmekte kullanıldı. Diğeri sinir ağı tarafından sağlanan olurlu çözümlerden gecikmesiz çizelgeleri elde etmede kullanıldı. Bilgisayar benzetimleri önerilen hibrid yaklaşımın hızlı ve etkin olduğunu göstermiştir.

Cheng ve diğerleri (2002), yaptıkları çalışmada tek makina problemi üzerine çalışmışlardır. Ele aldıkları problemde bütün işlere ortak bir teslim tarihi atanmıştır. Buradaki amaç, negatif gecikme, pozitif gecikme ve teslim tarihine bağlı olan toplam cezayı minimize edecek teslim tarihini ve çizelgeyi belirlemektir. Yazarlar, iş sırası önceden belli olursa ya da bütün işler aynı işlem süresine sahip olursa, optimal teslim tarihini ve çizelgeyi elde eden algoritmayı geliştirdiklerini ileri sürmektedirler

Xiao ve Li (2002) işlere genel bir teslim tarihi atama ve teslim tarihi, toplam negatif gecikme ve toplam pozitif gecikmenin ağırlıklı toplamalarını minimize ederek işleri paralel makinelerde çizelgeleme problemini düşündüler. Bu problemi çözmek için bir sezgisel ve bu sezgisel için bir mutlak performans oranı geliştirdiler. Sıfır negatif gecikme cezalı durum için daha iyi bir kötü durum sınırlı sezgisel sundular. Ayrıca tamamen polinomal olan bir yaklaşıma şeması geliştirdiler. Sezgisellerinin atölye çizelgeleme ve genel teslim tarihi atama algoritma gelişimine katkı sağladığını iddia ettiler.

Birman ve Mosheiov (2004) iki makinalı bir akış tipi üretimde teslim tarihi ve çizelgeleme problemi üzerinde çalıştılar. Teslim tarihi belirleme problemlerinin son yıllarda yoğun ilgi çektiğini belirttiler. Çalışmalarının amacı maksimum negatif gecikme, pozitif gecikme ve teslim tarihi belirleme maliyetlerini minimize etmektir. Sonuç olarak Johnson algoritması ile daha etkili bir çözüm bulduklarını iddia ettiler.

Min ve Cheng (2006), optimal genel teslim tarihi, her bir makinadaki işlem sırası ve iş sayısını belirlemek için optimal çizelgeleme politikasını belirleyen, teslim tarihi

atama, negatif gecikme, pozitif gecikme maliyetlerini minimize eden bölgesel kodlamaya dayalı bir tür genetik algoritma önerdiler. Genetik algoritma için ayrıca, iyi performans gösteren 3 tip hibrid genetik algoritma yapılandırmak amaçlı tavlama benzetimi mekanizması ve iteratif sezgisel ince ayar operatörü eklediler. Benzer paralel makina çizelgeleme ve genel paralel makina çizelgeleme problemi üzerine odaklanan sayısal hesaplama sonuçları, bu algoritmaların sezgisel algoritmalarından daha iyi sonuç verdiğini ve büyük ölçekli paralel makina negatif gecikme, pozitif gecikme çizelgeleme problemi için uygun olduğunu belirttiler.

Mosheiv ve Oron (2006) çalışmalarında, iş sırasını, genel teslim tarihini ve hızlı bakım aktivitesini yerleşimini belirlemek istediler. Teslim tarihinden sonra veya önce çizelgelenen işler erken veya geç bitiş zamanlarına göre cezalandırılır. Bakım aktivitesinden sonra çizelgelenen bir işin işlem zamanı iş bağımlı bir faktör ile azalır. Amaç minimum toplam negatif gecikme, pozitif gecikme ve teslim tarihi maliyetleridir. Bu problem için polinomal bir çözüm önerdiler. Bakım çizelgeleme ve teslim tarihi atamanın aynı anda yapıldığı bu ilk çalışmalarında, problemin polinomal zamanlı çözülebilir olduğunu ifade ettiler.

2.4. Yapay Sinir Ağları

İnsan beyninin öğrenme, benimseme, ilişkilendirme ve genelleme yeteneği, herhangi bir bilgisayarın yeteneğinden daha üstündür. Bu yetenek, araştırmacıların insan beynini basitleştirilmiş bir formda modellemeye çalışmalarına neden olmuştur (Karna ve Breen, 1989). Bu modellemeler aracılığıyla oluşturulan yapay sinir ağları, çeşitli çizelgeleme problemlerine çözüm bulmada olumlu katkılar sağlamıştır. Bu katkılardan bazıları şunlardır:

Chryssolouris ve diğerleri (1990), bir imalat sistemi tasarımı için bir yapay sinir ağı modeli geliştirdi. Benzetimle vasıtasıyla gerçekleştirilen imalat sistem tasarımı, model parametrelerini yapay sinir ağına öğreterek gerçekleştirir. İstenen performans ölçütü değerleri ağa sunulduğunda ağ, atölyedeki iş merkezlerindeki uygun makina sayılarını belirler. Yani, ağın çıktısı iş merkezlerindeki uygun makina sayısıdır.

Zhang ve diğeri (1991), kendileri tarafından geliştirilen atölye tipi çizelgeleme problemine ait bir yapay sinir ağı modeli kullandıkları çalışmalarında, öncelik ilişkilerini de dikkate alan bir yapay sinir ağı modeli geliştirirler. Geliştirdikleri yöntemle teslim tarihli atölye tipi çizelgeleme problemini öncelik ilişkili atölye tipi çizelgeleme problemine kolaylıkla dönüştürürler. Benzetimden elde ettikleri sonuçlara göre bu yöntemin iyi sonuç verdiğini belirtmektedirler.

Dağlı ve Huggahalli (1991), daha iyi çizelge elde ederken hesaplama zamanını en aza indirebilmek için dinamik çizelgeleme probleminde yapay sinir ağını uyguladı. Eğer bir çizelgeleme problemi, bir vektör gibi şeklinde tanımlanabilirse, bu vektörler yapay sinir ağına girdi olabilir ve çizelgeleme algoritması kullanılarak elde edilecek optimal çözüm saklanabilir. Aynı problem örneğinin tekrarlı sunumu sonucu ağı bu örnekleri öğrenerek optimal çözümü bulabileceği belirtilir. En çok gecikmeyi azaltan Lawlers'in çizelgeleme algoritması ile bu yapay sinir ağının iyi sonuç ürettiği ifade edilir.

Khaw ve Siong (1991), atölye tipi çizelgeleme problemi için üç boyutlu hatayı geriye yayma sinir ağı geliştirdi. Bu yaklaşımın, benzetim ve lineer programlama gibi diğer tekniklerden daha fazla avantajları olduğu ifade edilmektedir. Kolay ve de çizelgeleme problemi ile ilgili hiçbir varsayıma gerek duymayan, sadece girdi ve bu girdilere göre oluşan çıktı değerlerinin geçmişteki verilerinin toplanmasıdır. Bir imalat firmasına ait gerçek verilerle oluşturulan veri seti ile sunulan üç boyutlu model test edilir. Sonuçların ümit verici olduğu belirtilir.

Arizono ve diğeri (1992), toplam akış zamanını azaltmak için yapay sinir ağını kullanan bir çizelgeleme yaklaşımı ortaya koydular. Stokastik yapay sinir ağlarından biri olan Gauss makina modelini kullanarak çizelgeleme problemini çözmeye uğraşır.

Philipoom ve diğeri (1994) yaptıkları çalışmada, bir üretim kontrol sistem sürecinin 3 kısımdan meydana geldiğini belirtmişlerdir. Bu 3 aşama sipariş alma/ana üretim çizelgesi, sipariş açma ve atölyede üretim safhalarıdır. İlk safhada sipariş müşteriden gelmektedir ve siparişe bir teslim tarihi verilmektedir. Sipariş için teslim

tarihi üretim bölümü tarafından verilirse buna içeriden belirlenen teslim tarihi, eğer pazarlama bölümü ile müşteri arasında belirlenirse buna dışarıdan belirlenen teslim tarihi denir. İkinci kısımda sipariş atölyeye gönderilmektedir. Son safhada ise sipariş atölyede üretilir. Yazarlar çalışmalarında üretim kontrol sistem sürecinin birinci safhası üzerinde durmuşlardır. Özellikle içeriden belirlenen teslim tarihini dikkate almışlardır. İçeriden belirlenen teslim tarihi için yapay sinir ağı tahminini kullanmışlardır ve bu tahmini diğer regresyon metotları ile mukayese etmişlerdir. Oluşturulan yapay sinir ağı 23 proses elemanlı giriş katmanına sahiptir. Giriş katmanındaki değerler atölye özelliklerini ve genel iş merkezi özelliklerini göstermektedir. Yapay sinir ağı 9 proses elemanlı bir gizli katman ve bir proses elemanlı bir çıktı katmanına sahiptir. Sonuç olarak, yapay sinir ağının üzerinde çalışmaya değer bir yöntem olduğu yapılan 3 çalışmanın ikisinde sinir ağının daha iyi sonuç verdiği iddia edilmiştir.

Sabuncuoğlu ve Gurgun (1994), tek makina ortalama pozitif gecikme çizelgeleme problemi ve tek makina en az tamamlanma zamanı çizelgeleme problemi için bir yapay sinir ağı geliştirdiler. Önerilen model, yapay sinir ağlarını ve çeşitli algoritmik yaklaşımları birleştirir. Ağın performansı, mevcut çizelgeleme algoritmaları ile karşılaştırılır.

Philipoom ve diğerleri (1997) yaptıkları çalışmada, literatürdeki teslim tarihi araştırmalarının negatif gecikme ve pozitif gecikme maliyetleri üzerine olduğu ve bunun pratikte pek az anlam taşıdığını ileri sürmüşlerdir. Teslim tarihi maliyetleri üzerine regresyon, matematik programlama ve yapay sinir ağı metotları geliştirilmiş ve benzetim tekniği ile üç hipotetik atölyede sonuçlar karşılaştırılmıştır. Çalışmanın sonucunda yapay sinir ağının maliyet fonksiyonu olarak kullanılabilmesi ve daha iyi sonuç alınabileceği iddia edilmiştir.

Geyik ve Cedimoğlu (1999) süreç planlama ve çizelgelemede yapay zeka kullanımı ile ilgili yayınları çalışmalarında incelediler. Çalışmada makaleleri dört grup altında topladılar. Bu gruplar uzaman sistemler, yapay sinir ağları, diğer yapay zeka teknikleri ve hibrid sistemler olarak incelendi.

Raaymakers ve Weijters (2002) parti işleme endüstrisinde regresyon analizi ve yapay sinir ağı arasında karşılaştırma yapmışlardır. Planlama ve çizelgeleme seviyelerinden oluşan hiyerarşik üretim kontrol durumu varsaymışlardır. Çalışmalarında planlama seviyesi üzerine odaklanmışlardır. İşlerin paralel makinalarda tamamlanma zamanlarını regresyon ve yapay sinir ağı kullanarak tahmin etmişlerdir. Bütünleşik bilgi kullanılarak her iki tekniğin de, detaylı bilgi kullanan diğer metotlarla karşılaştırıldığında tamamlanma zamanının tayin edilmesinde daha az zaman harcadığını iddia ettiler. Yeni üretilen iş setleri üzerinde yapılan testler her iki tekniğin de iş sayısı, ortalama işlem süresi değişimleri, daha dengesiz iş yükü ve farklı kaynak konfigürasyonlarına karşı güçlü olduğunu göstermiştir. Sonuç olarak yapay sinir ağlarının tahmin kalitesinin regresyon modellerinden önemli oranda iyi olduğu görülmüştür.

Yildirim ve diğerleri (2006), kaynakların mevcudiyeti, gelen siparişlerin teslim tarihi atamaları ve çizelgede kullanılacak olan öncelik kuralları belirlemede karar vermek için paralel sinir ağlarını kullanan bir sistem önerdiler. Bu sistemi paralel benzer makinaların bulunduğu iş merkezli bir esnek imalat sistemine uyguladılar. Amaçları yapay sinir ağını ve benzetimi, bir imalat sistemi atölyesinde üst yönetim tarafından belirlenen sistem hedeflerine ulaşabilmede işlerin çizelgelenmesinde araç olarak kullanmaktı. Özellikle her bir iş merkezindeki makina sayısı, atölyede kullanılacak olan öncelik kuralı ve siparişlere ait teslim tarihlerinin belirlenmesini hedeflediler. Yapay sinir ağları imalat sistemi dizaynı ile kontrol parametreleri arasında temel oluşturan ilişkiyi ve performans hedeflerini memnun edici bir şekilde yakalayabilmekte olduğunu ifade ettiler.

Chen (2007) çalışmasında, bir bisküvi fabrikasında kritik bir görev olan çıktı zamanının tahmini üzerinde durdu. Çıktı zamanı tahmininde doğruluk konusunda daha başarılı olabilmek için zeki bir hibrid sistem önerdi. Öncelikle, bisküvi partilerini, bulanık geriye yayılım ağı ile çıktı zamanlarını tahmin etmeden önce k-ortalamlar sınıflayıcısı ile ön sınıflamaya tabi tutarak, girdi sınıflama kavramını Chen bulanık geriye yayılım ağı yaklaşımına uyguladı. Farklı kategorilere ait örnekler, farklı fakat aynı topolojiye sahip olan ağlarla öğretildi. İkinci olarak fabrika gelecek sevkiyat planını da zeki hibrid isteme dahil etti. Önerilen metodolojinin

etkinliğini deęerlendirebilmek amacıyla test örneklerini oluşturmak için üretim benzetimi yaptı. Deneysel sonuçlara göre zeki hibrid sistemin tahmin doęruluęunun dięer yaklaşımlardan önemli derecede daha iyi olduęunu belirtti.

Teslim tarihi belirleme hakkında daha fazla bilgi almak için řu yayınlara bakılabilir: Moshiev, 2003, Park ve dięerleri, 2000, Gordon ve dięerleri, 2002, Mosses, 1999, Li ve dięerleri, 1999, Gordon ve Kubiak, 1998, Qi ve dięerleri, 2002.

2.5. Sonuç

Literatürde yapılan çalışmalarda, arařtırmacılar teslim tarihi belirlemenin önemi üzerinde durmuşlardır. Doęru ya da tahmin hatası düşük olarak belirlenen ürün teslim tarihleri, sipariřin zamanında müşteriye teslim edilmesini saęlayacak ve bu şekilde müşteri memnuniyeti artırılmış olacaktır. Bununla beraber zamanında sipariřin teslimi ile birlikte, rekabetçi bir ortamda işletme rakiplere karşı avantaj saęlamış olacaktır.

Geleneksel teslim tarihi belirleme kurallarının yanında, tahmin için yapay sinir aęı kullanılmasına literatürde oldukça fazla karşılaşılmaktadır. Yapay sinir aęının örneklerden öğrenmesi veya atölye ile ilgili deęişik deęerleri dikkate alarak bir sonuca ulaşma özellięi teslim tarihi belirlemek için de kullanılabilir. İşletmelerin teslim tarihlerini doęru olarak belirleyebilmeleri için, klasik teslim tarihi belirleme kuralları yanında, yapay zekanın özelliklerinden faydalanarak, hangi ortamda alınan girdilere göre, hangi kuralı kullanmanın uygun olacaęına kararını yapay zeka teknięinin vermesi daha doęru teslim tarihi belirlenmesini saęlayabilir.

Dinamik atölye çizelgelemede klasik teslim tarihi belirleme kuralları ile yapay sinir aęının performans karşılaştırma çalışması literatürde mevcut olmadığı için ve yapay zekanın örneklerden öğrenme ve uygulama özellięi dikkate alınarak, teslim tarihi belirlemek için böyle bir arařtırmanın yapılması uygun görülmüştür.

BÖLÜM 3. TESLİM TARİHİ BELİRLEMEDE YAPAY SİNİR AĞI YAKLAŞIMI

3.1. Giriş

Bu çalışmanın amacı, dinamik atölye tipi üretimde teslim tarihi belirlenmesinde regresyon modellerinin etkilerini incelemek ve daha iyi teslim tarihi belirleyebilecek olan bir yapay sinir ağı modeli geliştirmektir.

Regresyon modellerinin denklem katsayılarının belirlenmesi ve yapay sinir ağının eğitimi için gerekli olan veri elde etme işlemi benzetim ile yapılmıştır. Benzetim işlemi için bir olay artımlı benzetim programı yazılmıştır. Bu program aynı zamanda regresyon modelleri ve yapay sinir ağının performans karşılaştırmasında da kullanılmıştır.

Gerekli olan veriler elde edildikten sonra, bu veriler kullanılarak regresyon modellerinin denklem katsayıları belirlenmiş ve aynı zamanda yapay sinir ağının eğitimi yapılmıştır. Daha sonra yeniden benzetim yapılarak regresyon modellerinin ve yapay sinir ağı modelinin ürettiği teslim tarihi bilgileri performans kriterleri göz önüne alınarak kıyaslama yapılmıştır.

Yukarıdaki söz konusu işlemler, en kısa işlem zamanlı (EKİS) önce ve en erken teslim tarihli (EETT) önce öncelik kuralları için ayrı ayrı yapılmıştır.

Bu bölüm teslim tarihi belirlenmesinde regresyon modelleri ve yapay sinir ağı modelinin nasıl kullanıldığı hakkında detaylı bilgi vermektedir. Söz konusu aşamalar şu şekilde tanımlanabilir:

1. Regresyon modelleri,
2. Regresyon modelleri ile teslim tarihi belirleme,
3. Önerilen yapay sinir ağı modeli,
4. Yapay sinir ağı modeli ile teslim tarihi belirlenmesi,
5. Regresyon modelleri ve yapay sinir ağı modelinin seçilen performans kriterlerine göre karşılaştırılması.

3.2. Regresyon Modelleri

Bu çalışmada teslim tarihi belirlemek için 5 adet regresyon modeli kullanılmıştır (Philipoom ve diğerleri, 1994). Kullanılan bu 5 model sadece iş karakteristiklerini kullanan ve iş karakteristikleriyle birlikte atölye verilerini de kullanan olmak üzere iki kategoriye ayrılabilir. Bu iki kategorideki modeller şunlardır:

3.2.1. Sadece iş karakteristiklerini kullanan modeller

1. Toplam İş Yüğü (TİY):

$$F_i = kP_i \quad (3.1)$$

i işinin tahmin edilen akış zamanı (F_i) toplam işlem zamanı P_i 'nin bir fonksiyonudur. k bir katsayıdır ve regresyon ile hesaplanmaktadır.

2. Operasyon Sayısı (OS):

$$F_i = kN_i \quad (3.2)$$

Burada işin tahmin edilen akış zamanı, i işinin operasyon sayısının (N_i) bir fonksiyonudur. Yine k bir katsayı olup regresyon ile hesaplanmaktadır.

3. Toplam İş Yüğü ve Operasyon Sayısı (TİY+OS):

$$F_i = k_1 P_i + k_2 N_i \quad (3.3)$$

Bu modelde hem operasyon sayısı hem de toplam işlem zamanı kullanılmaktadır. k_1 ve k_2 katsayıları regresyon ile hesaplanmaktadır.

3.2.2. İş karakteristikleri ve atölye bilgilerini kullanan modeller

1. Kuyruktaki İş Sayısı (KİS):

$$F_i = k_1 P_i + k_2 (JIQ_i) \quad (3.4)$$

i işi atölyeye geldiği anda kuyruklardaki bekleyen iş sayıları toplanır (JIQ_i). Bu atölye verisi iş karakteristiği P_i ile birleştirilir. Buradaki k_1 ve k_2 katsayıları regresyon ile hesaplanmaktadır.

2. Kuyruktaki İş Yüğü (KİY):

$$F_i = k_1 P_i + k_2 (WIQ_i) \quad (3.5)$$

Bu model 2. modelden farklı olarak sadece kuyruklardaki iş sayılarını değil de atölyedeki toplam işlem zamanlarını kullanır. Yine k_1 ve k_2 katsayıları regresyon ile hesaplanmaktadır.

3.2.3. Regresyon analizi

Regresyon analizi, aralarında sebep-sonuç ilişkisi bulunan iki veya daha fazla değişken arasındaki ilişkiyi matematiksel bir fonksiyonla belirlemek, bu ilişkiyi kullanarak bağımlı değişken hakkında tahminler (estimation) ya da kestirimler (prediction) yapabilmek amacıyla sıklıkla başvurulan bir istatistik analiz tekniğidir

(Tarı, 1999, Orhunbilge, 2000). Regresyon analizi ile aynı zamanda değişkenler arasındaki yapısal ilişkiler de ortaya konmuş olur. Ekonomik, sosyal ve doğal olayların çoğunda sebep-sonuç ilişkisine rastlamak mümkündür.

Örnek: Sebep	Sonuç
Gelir	Tüketim
Ağırlık	Uzunluk
Gübre	Verim
Yem miktarı	Süt miktarı
Çalışma süresi	Başarı

Regresyon analizi ile bağımlı ve bağımsız değişkenler arasındaki ilişkiyi açıklamak için matematiksel bir model kullanılır ve bu model regresyon modeli olarak adlandırılır. Bu matematiksel model tek değişkenli (basit) olabileceği gibi çok değişkenli (çoklu) olabilir. Model doğrusal olabileceği gibi eğrisel olarak ta kurulabilir.

Basit doğrusal regresyon modeli

$$Y = \alpha + \beta X + \varepsilon \quad (3.6)$$

şeklinde bir bağımlı ve bir de bağımsız değişken içeren bir modeldir. Burada;

Y: bağımlı (açıklanan) değişken olup belli bir hataya sahip olduğu varsayılır.

X: bağımsız (açıklayan) değişkeni olup hatasız ölçüldüğü varsayılır.

α : regresyon sabiti olup $X = 0$ olduğunda Y'nin aldığı değeri gösterir. Diğer bir ifadeyle regresyon doğrusunun Y eksenini kestiği noktadır.

β : regresyon katsayısı olup, X'de meydana gelen bir birimlik bir değişimin Y de meydana getirdiği değişim miktarını ifade eder. Kısaca eğim katsayısı olarak ta adlandırılır.

ε : tesadüfi hata terimi olup ortalaması sıfır varyansı σ^2 olan normal dağılış gösterdiği varsayılır. Bu varsayım parametre tahminleri için değil katsayıların istatistik olarak anlamlılıklarının sınanması için gereklidir.

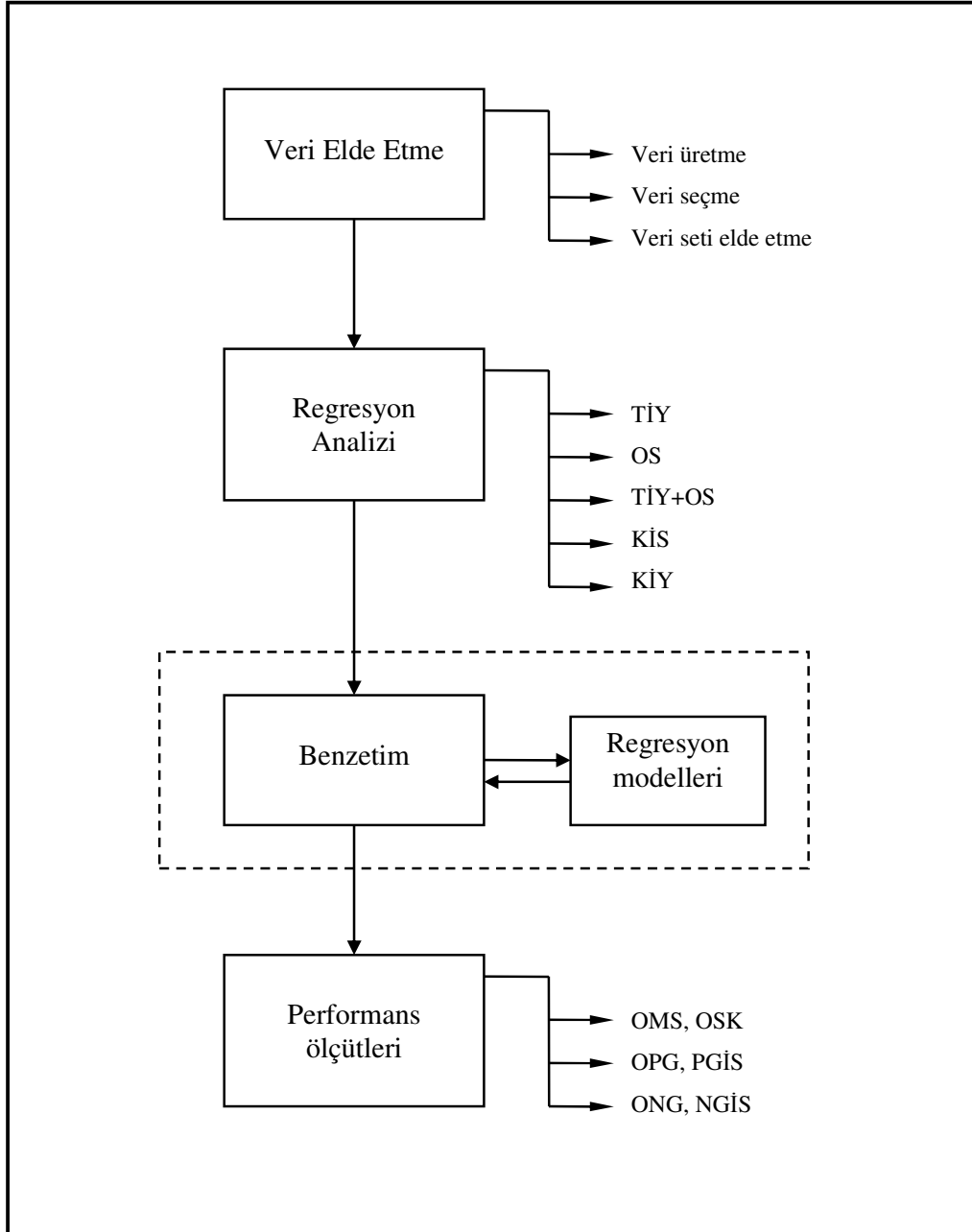
Regresyon modeli uydurulduktan sonra modelin yeterli olup olmadığının kontrolü regresyon analizinin en önemli bölümüdür. Uydurulan modelin doğru modele yeterli derecede yaklaştığını garanti etmek ve en küçük kareler regresyon analizinin tüm varsayımlarını sağlayıp sağlamadığını kontrol etmek gerekir. Regresyon analizinde modelin yeterliliği için genellikle varyans analizi ve çoklu determinasyon katsayısından (R^2) faydalanılır. Modelin yeterliliğinin varyans analizi ile ortaya konması yeterli değildir. Bunun dışında regresyon parametrelerinin de istatistik açıdan anlamlı olması da t testleri ile araştırılmalıdır.

Öte yandan regresyon modelinin çeşitli varsayımları da yerine getirmesi gerekir. Bunlar kısaca ardışık hata terimlerinin otokorelasyonsuz olması, bağımsız değişkenlerin kendi aralarında çoklu doğrusal olmaması. Hata teriminin varyansının sabit olması, hata teriminin dağılımının normal olması gibi varsayımlardır. Bu varsayımlardan sapma modelin yeterliliği ve geçerliliğine olumsuz etki yapmaktadır.

Regresyon modelinin parametrelerinin tahmininde genellikle en küçük kareler ve en çok benzerlik yöntemleri kullanılmaktadır. Uygulamada daha ziyade en küçük kareler yöntemi kullanılmaktadır. En küçük kareler yöntemi hata karelerini minimum yapan yöntemdir. Bu yöntemle serpilme diyagramındaki noktaların regresyon doğrusuna dik uzaklıklarının karelerinin minimum yapılması sağlanmaktadır.

3.3. Regresyon Modelleri ile Teslim Tarihi Belirleme

Söz konusu modellerde akış zamanı tahmin edilmektedir (Şekil 3.1). Tahmin edilen akış zamanlarına işlerin atölyeye geldikleri zaman olan hazır zamanları (r_i) eklendiğinde işin teslim tarihi (d_i) böylece tahmin edilebilmektedir.



Şekil 3.1. Regresyon modelleri ile teslim tarihleri belirleme

3.3.1. Regresyon analizi için veri elde edilmesi

Bahsedilen 5 regresyon modelinde akış zamanının tahmininde kullanılan i işinin toplam işlem zamanı, i işinin operasyon sayısı, i işi atölyeye geldiğinde kuyruklardaki iş sayıları toplamı ve atölyedeki bekleyen işlerin toplam işlem zamanları toplamı (iş yükü) verilerinin katsayılarının belirlenebilmesi için gerekli olan bu veriler benzetim ile elde edilmiştir. EKİS ve EETT öncelik kuralları için ayrı ayrı olmak üzere, her birinde farklı bir rassal sayı ile başlayan 10 benzetim yapılmıştır. Her bir benzetim çalışmasında 5000 işlik ısınma periyodundan sonra 10000 işe ait regresyon modellerinde kullanılacak olan veriler kaydedilmiştir. Böylece sonuçta her birinde 10000 veri olan 10 adet veri dosyası elde edilmiştir. Daha sonra her bir veri dosyasından 10 işten birine ait veriler alınmak üzere 10000 işlik veriye sahip olan tek bir veri seti elde edilmiştir. Tablo 3.1 ve Tablo 3.2’de EKİS ve EETT öncelik kurallarının kullanıldığı bu veri setlerinden 10’ar işe ait birer örnek görülmektedir.

3.3.2. Regresyon modellerinin oluşturulması

Elde edilen 10000 işe ait bu veriler kullanılarak 5 regresyon modeli için gerekli olan k katsayıları doğrusal regresyon ile elde edilmiştir. Tablo 3.1’deki ilk sırada görülen işi örnek alındığında 5 regresyon modelinden ilki olan TİY yöntemi için regresyon denklemi şu şekilde olacaktır:

$$y_1 = kx_1 \quad (3.7)$$

Veya,

$$F_1 = kP_1$$

Buradan,

$$575 = k52$$

Tablo 3.1. Regresyon modellerinin katsayılarının belirlenmesi için veri seti örneği (EKİS)

N	P	JIQ	WIQ	F
...
9	61	41	11930	155
9	99	44	7747	544
9	45	36	9747	132
9	66	32	11038	165
9	58	36	7972	169
9	102	35	10668	449
9	66	39	13630	165
9	97	37	5654	154
9	76	38	8428	254
9	56	41	8114	103
...

Tablo 3.2. Regresyon modellerinin katsayılarının belirlenmesi için veri seti örneği (EETT)

N	P	JIQ	WIQ	F
...
9	52	62	10222	575
9	86	61	13023	670
9	63	57	9544	542
9	69	62	9529	587
9	54	58	10663	472
9	85	56	11082	702
9	27	56	14513	375
9	84	55	13561	689
9	99	54	11936	795
9	71	59	11997	730
...

Bu şekilde veri setindeki tüm işlere ait veriler kullanılarak regresyon ile k katsayıları için bulunan değerler sırasıyla EKİS ve EETT öncelik kuralları için Tablo 3.3 ve Tablo 3.4'de verilmiştir. Bulunan bu katsayılara ait istatistik testler Ek A-J'de verilmiştir.

3.3.3. Regresyon modelleri ile teslim tarihi belirleme

Regresyon model katsayıları elde edildikten sonra, EKİS ve EETT öncelik kurallarının kullanıldığı 10'ar adet benzetim yapılmıştır. Yine bu 10 benzetimde birbirinden ve aynı zamanda regresyon model katsayılarının belirlenmesi için gerekli veri setinin elde edilmesi için yapılan benzetimlerdekenden farklı başlangıç rassal sayıları kullanılmıştır. Kullanılan bu başlangıç rassal sayıları önerilen yapay sinir ağı modeli için de ileride daha sonra ayrıları olarak kullanılacaktır.

Tablo 3.3. Regresyon modelleri katsayıları (EKİS)

Model	k_1	Std.sap.	Anlam	k_2	Std. Sap.	Anlam
TİY	6,963	0,080	0,000	-	-	-
OS	50,062	0,867	0,000	-	-	-
TİY+OS	21,511	0,226	0,000	-143,238	2,124	0,000
KİS	14,294	0,193	0,000	-17,614	0,428	0,000
KİY	12,213	0,175	0,000	-0,048	0,001	0,000

Tablo 3.4. Regresyon modelleri katsayıları (EETT)

Model	k_1	Std.sap.	Anlam	k_2	Std. Sap.	Anlam
TİY	7,336	0,019	0,000	-	-	-
OS	68,550	0,196	0,000	-	-	-
TİY+OS	4,184	0,058	0,000	30,960	0,542	0,000
KİS	3,170	0,025	0,000	6,190	0,038	0,000
KİY	4,254	0,034	0,000	0,027	0,000	0,000

Bu yapılan 10 benzetimde, atölyeye işler geldiği anda teslim tarihleri belirlenirken işlerin geldiği zamana regresyon modeli kullanılarak tahmin edilen akış zamanı eklenerek teslim tarihi belirlenmektedir. Her benzetimde 5000 işlik ısınma periyodundan sonraki 10000 iş için veriler kaydedilmiştir. Yine Tablo 3.1'deki ilk işi örnek alırsak, bu iş için teslim tarihi belirlenirken akış zamanı şöyle tahmin edilmektedir:

$$F_1' = (6,963)(61) \cong 425$$

Bu iş için gerçekleşen akış zamanı $F_1=155$ olduğundan yapılan tahmine göre iş gecikmiştir. Bu gecikme (L_1),

$$L_1 = F_1' - F_1$$

$$L_1 = 425 - 155$$

$$L_1 = 270$$

dir. L_1 pozitif olduğundan iş pozitif gecikmiştir. Dolayısıyla işin pozitif gecikmesi (T_1) 270, negatif gecikmesi veya erken bitmesi (E_1) 0'dır. Aynı zamanda işin tahmin edilen akış zamanının gerçekleşen akış zamanından mutlak sapması (MS) 270 ve sapma karesi (SK) 72900'dür.

Bu hesaplama şekliyle elde edilen 10000 işin tahmin edilen teslim tarihlerinin gerçekleşen tamamlanma zamanlarından (C_i) mutlak farklarının aritmetik ortalaması alınarak ortalama mutlak sapma (OMS) ve yine mutlak fark karelerinin aritmetik ortalaması alınarak ortalama sapma kareleri (OSK) hesaplanmıştır. Ayrıca pozitif geciken iş sayısı (PGİS), ortalama pozitif gecikme (OPG), negatif geciken iş sayısı (NGİS) ve ortalama negatif gecikme (ONG) değerleri hesaplanmıştır. Bu hesaplamalar 10 adet elde edilen veri dosyaları için ayrı ayrı yapılmıştır. Hesaplanan bu 10 değer aritmetik ortalamaları alınarak kullanılan öncelik kuralı ve regresyon modeli için daha sonra önerilen model değerleriyle karşılaştırılmak üzere OMS, OSK, PGİS, OPG, NGİS ve ONG değerleri hesaplanmış olmaktadır.

3.4. Yapay Sinir Ağları

Bu bölümde, çalışmada kullanılan yapay sinir ağı modeli olan geriye yayma metodolojisi incelenmiştir.

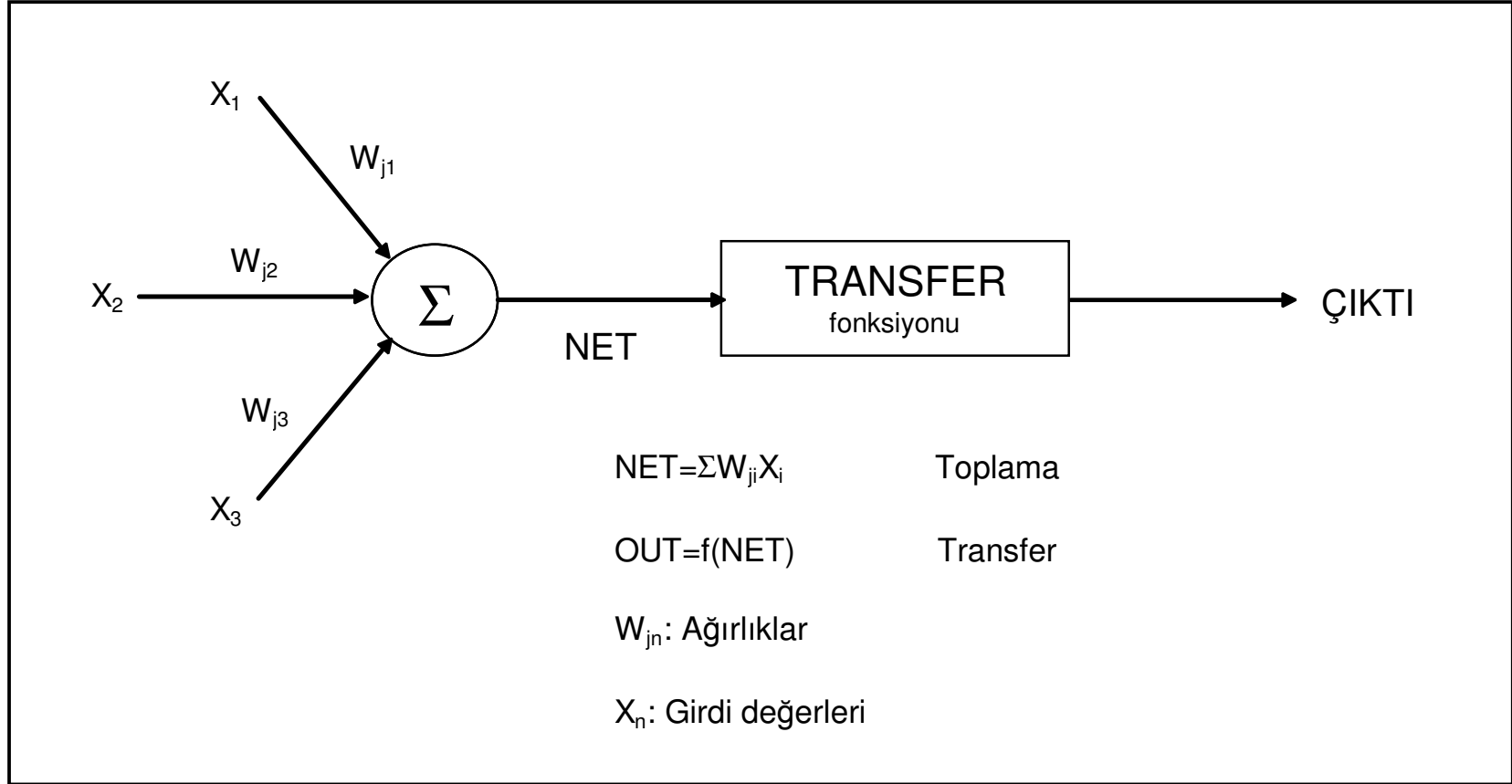
3.4.1. Temeller

Bir sinir ağı, insan beynine dayalı bir çıkarım mekanizması olarak tanımlanabilir (Negnevitsky, 2002). Diğer bir tanımla, yapay sinir ağları, birbirine paralel bağlı

basit elemanlar ile biyolojik sinir sistemlerinin yaptığı gibi gerçek dünya nesneleriyle etkileşimden planlı hiyerarşik yapılardır (Kohonen, 1987). Bir genel sinir ağı modeli Şekil 3.2’de gösterilen proses elemanlarıyla karakterize edilir. Bu şekilde, bir proses elemanının beş bileşen içerdiği görülmektedir. Bunlar:

- Girdiler, proses elemanına bilgi getirir. Bu bilgi diğer proses elemanları veya dış kaynaklardan sağlanır. Bazen proses elemanı kendi kendine bilgi verebilir.
- Ağırlıklar, belli bir girdinin proses elemanı üzerindeki etkisini belirler. Ağın doğru çıktıları üretebilmesi için eğitime işlemi sırasında optimize edilmesi gereken ağırlık değerleridir.
- Toplama fonksiyonu, proses elemanının ağırlıklı girdilerini toplar. Çeşitli toplama fonksiyonları vardır. Bunlardan bazıları Şekil 3.2’de gösterilmiştir (Neuralware Inc, 1990). En yaygın olanı ağırlıklı toplamı bulmaktır. Burada gelen her girdi değeri, kendi ağırlığı ile çarpılarak toplanır (Oztemel, 2003).
- Transfer fonksiyonu, toplama fonksiyonunun sonucunu düzenleyerek proses elemanının çıktısını belirler. Yine çeşitli transfer fonksiyonları vardır (Neuralware Inc, 1990). Popüler olanlarından bazıları, sigmoid fonksiyon, lineer fonksiyon, step fonksiyonudur ve Tablo 3.5’de gösterilmektedir.
- Çıktı, transfer fonksiyonunun sonucunu, bağlı olan proses elemanlarına veya dış kaynaklara gönderir.

Ağın topolojisi, ağı karakterize eden ikinci özelliğidir. Bir proses elemanları grubu katman adı verilen yapıyı oluştururlar. Tipik bir sinir ağı bir diğerine bağlı üç katman içerir. Bunlar, dış dünyadan girdi kabul eden girdi katmanı, girdi katmanından gelen bilgileri işleyerek çıktı katmanına gönderen gizli katman (veya gizli katmanlar) ve ağın kararı hakkında dış dünyaya bilgi veren çıktı katmanıdır. Bilgi ağın katmanları arasında veya katmanlar içinde akar.



Şekil 3.2. Bir proses elemanın genel yapısı

Tablo 3.5. Toplama ve transfer fonksiyonu örnekleri

Toplama fonksiyonları	Transfer fonksiyonları
Maximum: $Net_i = \max(W_{ij} \cdot I_j)$	Sigmoid fonksiyon: $f(x) = \frac{1}{1 + e^{-x}}$
Minimum: $Net_i = \min(W_{ij} \cdot I_j)$	Lineer fonksiyon: $f(x) = x$
Toplam: $Net_i = \sum_j (W_{ij} \cdot I_j)$	Hiperbolik Tanjant fonksiyonu: $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
Kümülatif toplam: $Net_{yeni} = Net_{eski} + \sum_j (W_{ij} \cdot I_j)$	Adım fonksiyonu: $f(x) = \begin{cases} 1, & x > \text{threshold} \\ 0, & x \leq \text{threshold} \end{cases}$
I _j : i proses elemanının çıktısı İ _j : proses elemanı Net _i : i proses elemanının net girdisi W _{ij} : i ve j proses elemanları arasındaki bağlantı ağırlıkları	

3.4.2. Genel özellikler

Sinir ağı modellerini geleneksel sistemlerden ayıran çeşitli karakteristikler vardır. Bu karakteristikler ağın algoritmik olmayan, paralel ve dağıtılmış bilgi işleme yeteneklerine dayanır. Bu yetenekler sinir ağlarının kolaylıkla gerçek zamanlı uygulanabilmesini sağlar ve herhangi bir zorluk olmaksızın karmaşık, lineer olmayan hesaplamaları yapabilmesine, hızlı cevap vermesine izin verir. Sinir ağlarını popüler yapan karakteristiklerden bazıları şunlardır:

- Örneklerden öğrenme. Sinir ağları sadece öğreneceği varsayılan girdi/çıkıtlı ilişkileri örneklerine gerek duyar. Bu örnekleri kullanarak genellemeler yapar.
- Örüntü tanıma veya sınıflama. Sinir ağları örüntüleri girdi olarak alırlar ve örneklerden alınan girdi/çıkıtlı eşleştirmesiyle ilgili bilginin saklandığı dağıtılmış ilişkili belleğini kullanarak uygun çıktıları üretirler.
- Örüntüleri yeniden yapılandırma. Sinir ağları tam olmayan örüntüleri işleyebilir ve girdi örüntüsünü belleğindeki tam olan örüntü ile karşılaştırarak eksik bilgiyi örüntüye yerleştirebilir.
- Kendi kendine organize etme. Bazı sinir ağları kendi kendine organize edebilir ve öğrenebilir. Bir şekilde çevre değiştiğinde bu ağlar kolayca kendilerini yeni bir duruma adapte edebilirler.
- Hata toleransı. Bazı proses elemanları bir takım nedenlerle hasar görür veya kaldırılırsa, sinir ağları çalışmayı kesmez. Bilgi bütün ağ üzerinde dağıtıldığından, bilgideki küçük parçaların kaybı veya hasarı ciddi olarak ağın performansına etki etmez. Bu yetenek bazen “dereceli bozulma” olarak adlandırılır.
- Bulanık veya gürültülü girdilere direnç. Bu önemli bir özelliktir. Sinir ağları bulanık, gürültülü veya eksik veriyle başa çıkabilir. Ağlar eksik veya bulanık veriyle karşılaştığı zaman sunulan girdi örüntüsü için en uygun çıktıyı seçer.

Bu karakteristikler 10 yıldan fazladır araştırmacıları dikkatini çekmekte ve bu nedenle sinir ağlarının kullanımı hızla artmaktadır (Wells, 1992).

3.4.3. Öğrenme

Yapay sinir ağı, zekasını bir öğrenme kuralı kullanarak geliştirir. Bu zeka, bir öğrenme kuralı kullanılarak, ağın, en sonunda uygun değerleri belirlenen bağlantılar veya ağırlık değerlerinde saklanır. Bir ağın bilgisinin veya zekasının onun bütün bağlantılarına dağıtılmış olmasına dikkat edilmelidir. Bu nedenle sinir ağları bazen “connectionist networks” olarak adlandırılırlar.

Bir sinir ağının öğrenme amacı, ağın belirli girdiler için doğru çıktılarını üretmesi için en uygun ağırlık değerlerini bulmaktır. Bilgi bütün bağlantılar üzerinde dağıtıldığından, tek bir bağlantı genellikle herhangi anlamlı bir bilgi içermez. Aslında, mantıklı bilgi oluşturmak için bir grup bağlantı düşünülmesi gereklidir. Bir ağın genellikle, bahis konusu probleme mantıklı ve kabul edilebilir bir cevap verebilmesi için, bütün bağlantılarında uygun ağırlık değerlerine sahip olması gerekmektedir. Bu, öğrenme veya eğitim adı verilen bir proses ile elde edilir. Öğrenme, ağırlık değerlerini değiştirme yolunu belirleyen bir öğrenme kuralına dayanır. Çeşitli öğrenme kuralları bulunmaktadır (Wasserman, 1989), (Simpson, 1990). Bir öğrenme kuralının temel prensipleri kabul edilen öğrenme stratejisiyle belirlenir. Literatürde geniş ölçüde üç tip öğrenme stratejisinden bahsedilmektedir.

Bunlar:

- Öğretmenli öğrenmede ağı denetleyecek bir öğretmene ihtiyaç duyulur. Öğretmen basitçe ağa çıktı katmanında ne üretmesi gerektiğini söyler. Daha detaylı bilgi (Rumelhart, 1986)’da bulunabilir.
- Destekleyici öğrenmede yine bir öğretmene ihtiyaç vardır. Fakat ağa ne üretmesi gerektiği söylenmez. Onun yerine, üretilen çıktının doğru veya yanlış olduğu söylenir (Wells, 1992).
- Öğretmensiz öğrenmede ağa yardımcı olacak herhangi birine gerek yoktur. Bu durumda, ağ girdi/çıktı eşleştirmelerini organize etmek için kendi kriterlerini geliştirir. Bu stratejiyi kullanan ağlar “kendi kendine organize olan ağlar” olarak adlandırılır (Carpenter and Grossberg, 1987), (Nelson and Illingworth, 1991).

3.4.4. Bu çalışma için seçilen sinir ağı modeli

Çoğu sinir ağı modelleri hatları çizilen prensiplere dayanmasına rağmen, çeşitli farklı metotlar, modeller, öğrenme kuralları vb. geliştirilmiştir. Belirli bir ağ, toplama ve transfer fonksiyonu, topolojisi ve kullanılan öğrenme kuralıyla tanımlanır (Karna and Breen, 1989), (Mason and Wang, 1990). Bu bölümde, çeşitli sinir ağı modelleri arasından, bu çalışmada kullanılan geriye yayılım ağları ve onun genelleştirilmiş delta kuralı olarak adlandırılan öğretmenli öğrenme kuralı açıklanacaktır. Diğer sinir ağı modelleri hakkında detaylı bilgi Dayhoff (1990) ve Öztemel (2003)' den alınabilir.

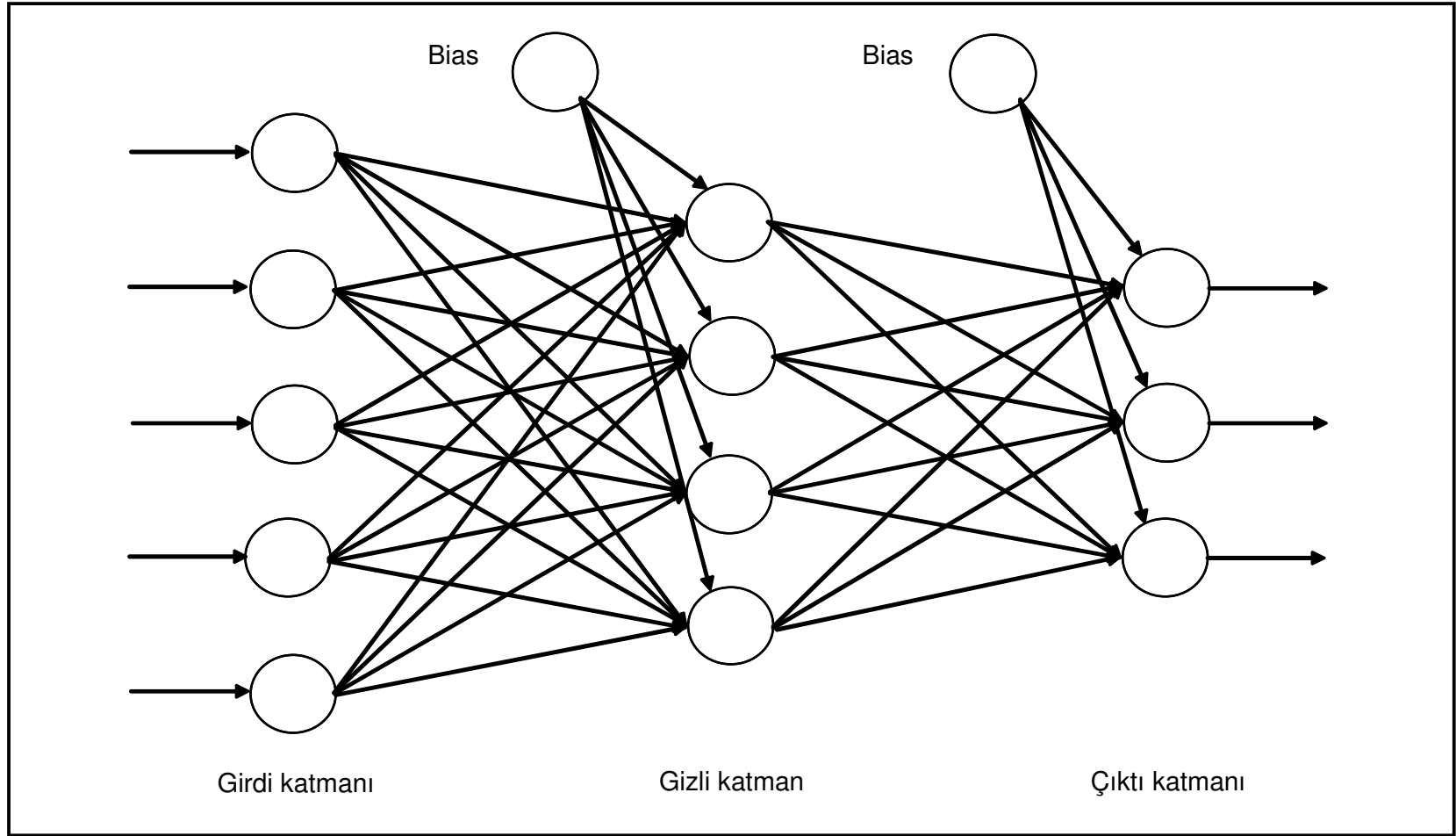
3.4.5. Geriye yayılım ağları

İlk defa Rumelhart ve diğerleri (1986) tarafından sunulan geriye yayılım modeli, en çok kullanılan sinir ağı modellerinden birisidir. Kısıtlamalarına rağmen geriye yayılım, yapay sinir ağlarının uygulanabileceği problem aralığını büyük ölçüde genişletmiş ve özellikle çeşitli girdi/çıkıktı eşleştirme problemlerinde başarılı sonuçlar üretmiştir (Wasserman, 1989, Rabelo ve Alptekin, 1990).

Geriye yayılım ağı, temel olarak girdi örüntüsüyle birlikte ağa sunulan sistemin beklenen çıktılarına gerek duyan öğretmenli öğrenme stratejisini kullanır. Öğrenme mekanizması, ağın gerçek ve beklenen (istenen) çıktıları arasındaki hatayı minimize eden yinelemeli eğitim azalması (iterative gradient descent) metoduna dayanır.

3.4.6. Bir geriye yayılım ağının yapısı

Geriye yayılım ağları çeşitli katmanlar içeren tamamen bağlı ileri besleme ağlardır (Şekil 3.3). Bunun anlamı şudur: Her bir proses elemanı bir sonraki katmandaki her bir proses elemanı ile bağlantılıdır. Fakat bağlantılar geri besleme yoktur. Ayrıca aynı katmandaki proses elemanları arasında herhangi bir bağlantı yoktur. Ağ üç tip katmandan oluşur: Girdi örüntülerini sonraki katmana aktaran bir girdi katmanı, bir veya daha fazla gizli katman ve bir çıktı katmanı. Bahsedilen son iki katman, gerçek bilgi işlemeden sorumludur.



Şekil 3.3. Bir geriye yayılım sinir ağının topolojisi

3.4.7. Geriye yayılım öğrenme işlemi

Geriye yayılım ağıları eğitim esnasında ağı gösterilecek girdi/çıkış çiftlerine gerek duyulan öğretmenli ağılardır. Bir girdi katmanı gösterildiğinde, ağı çıkışları hesaplanır ve istenen çıkışlarla karşılaştırılır. Bu çıkışlar arasındaki hata, azalan eğim algoritması kullanılarak ağı bağlantılarına geriye doğru yayılır. Öğrenme metodu çoğunlukla genelleştirilmiş delta kuralı olarak adlandırılır. Standart bir geriye yayılım ağına proses elemanları toplama fonksiyonu ve transfer fonksiyonu olarak sırasıyla ağırlıklı girdilerin toplamı ve sigmoid fonksiyonuna sahiptir. Öğrenme işleminin detaylı tanımından önce tanımlanması gereken birkaç faktör vardır. Öğrenme katsayısı, momentum ve bias terimleri.

Öğrenme katsayısı: Bu katsayı bağlantılardaki ağırlık değerlerinin değişim miktarını belirler. Bu katsayıyı yüksek almak değişim miktarını artırır, tersi durumda değişim miktarı daha az olur.

Momentum katsayısı: Yaklaşma hızını artırmak için en çok kullanılan sezgisel yaklaşımlardan biri, momentum faktörünün kullanılmasıdır (Rabelo and Alptekin, 1990). Bu ağı hızlı yaklaşmasına yardım eden bir katsayıdır (Lippman, 1987).

Temel olarak bir önceki değişimin bir kısmının şimdiki değişime eklenmesidir. Ana fikir öğrenme sırasında salınımlardan korunmaktır.

Bias: Transfer değeri 1 olan özel bir proses elemanıdır. Bir geriye yayılım ağındaki bias birimlerinin amacı daha hızlı öğrenmeye yardımcı olmak için transfer fonksiyonunun başlangıcını dengelemektir.

Öğrenmeye başlamadan önce, araştırma altındaki problem için uygun bir ağı topolojisi seçilmeli ve öğrenme katsayısı, momentum katsayısı gibi bazı parametrelerin değerleri belirlenmek zorundadır. Ağı başlangıç bağlantı ağırlıkları ise genellikle rasgele belirlenir. Öğrenme işleminde iki aşama vardır (Rumelhart ve diğerleri, 1986):

- Ađın gerek ıktılarının hesaplanması (ileri dođru hesaplama).
- Gerek ıktıyla istenen ıktı arasındaki hatanın geriye yayılması (geriye dođru hesaplama).

3.4.8. Gerek ıktıların belirlenmesi (İleri dođru hesaplama)

Daha nce bahsedildiđi zere, her bir proses elemanın birka girdisi vardır, fakat sonraki katmandaki diđer bir proses elemanın girdisi olabilecek tek bir ıktısı vardır. Bir katmandaki herhangi bir proses elemanın ıktı deđerlerinin hesaplanmasıyla sonraki katmandaki proses elemanlarının girdi deđerleri tanımlanmış olacaktır. ıktı katmanındaki her bir proses elemanın ıktı deđerleri hesaplandıktan sonra, ađın girdi deđerlerine olan gerek cevaplarla karřılařtırılırlar. Girdi katmanında herhangi bir bilgi iřlenmesi olmadıđından, girdi deđerleri otomatik olarak girdi katmanındaki proses elemanlarının ıktı deđerleri olarak atanırlar. Girdi katmanındaki proses elemanlarının sayısının girdi vektrnn boyutuna eřit olması gerekmektedir. Gizli veya ıktı katmanlarındaki herhangi bir proses elemanın ıktılarını hesaplamak iin proses elemanına gelen girdi toplanmalıdır. Eđer toplama fonksiyonu girdilerin ađırlıklı toplamları ise p proses elemanın girdisi (NET_p) řoyledir:

$$NET_p = \sum_{i=1}^k W_{ip} I_i + W_{pBias} \quad (3.8)$$

k proses elemanın girdi sayısı veya basite nceki katmandaki proses elemanı sayısıdır. I girdi deđerleri ve W_{ip} i ve p proses elemanları arasındaki bađlantı ađırlıđıdır. Bias ađırlıklarının da aynı řekilde dřnldđne dikkat edilmelidir. Gizli ve ıktı katmanlarındaki her bir proses elemanı, deđerleri 1 ve onunla ilgili ađırlık deđerleri W_{pBias} olan bir bias girdisine sahiptir.

P proses elemanın ıktısını belirlemek iin NET girdi deđerine bir transfer fonksiyonu uygulanır. Eđer transfer fonksiyonu sigmoid fonksiyon ise p proses elemanın ıktısı (O_p) řu řekildedir:

$$O_p = 1/(1 + e^{-NET_p}) \quad (3.9)$$

Bir gizli katmandaki p proses elemanının çıktı değerleri hesaplandığı zaman, bu değerler eğer varsa bir sonraki gizli katmandaki proses elemanlarının girdi değerleri olurlar. Eğer yoksa çıktı katmanındaki proses elemanları için girdi olurlar. Bunlar aynı yolla transfer ve toplama fonksiyonlarından geçerek ağa sunulan belirli bir girdi örneği için ağın gerçek çıktıları belirlerler.

3.4.9. Hatayı yayma (Geriye doğru hesaplama)

Çıktı katmanındaki bir t proses elemanının gerçek (O_t) ve istenen çıktıları (D_t) arasındaki fark (E_t) şöyle hesaplanır:

$$E_t = (D_t - O_t) \quad (3.10)$$

Minimize edilmesi istenen toplam hata şudur:

$$TE = 1/2 \sum_{t=1}^m (E_t)^2 \quad (3.11)$$

m çıktı katmanındaki toplam proses elemanı sayısıdır.

Geriye yayılım öğrenme algoritmasına göre çıktı katmanı ile son gizli katman arasındaki bağlantı ağırlıklarının değişimlerinin hesabıyla diğer katmanların (gizli katmanlar veya gizli ve girdi katmanı) hesabından farklıdır (Rumelhart ve diğerleri, 1986).

Gizli ve çıktı katmanlarındaki p ve t proses elemanları arasındaki bağlantı değişim miktarı (ΔW) şöyle hesaplanır:

$$\Delta W_{pt}^{yeni} = \lambda \delta_t O_p + \alpha \Delta W_{pt}^{eski} \quad (3.12)$$

$$\delta_t = O_t(1 - O_t)E_t \quad (3.13)$$

λ =öğrenme katsayısı

α =momentum katsayısı

Ağın diğer katmanlarındaki l ve p proses elemanlarının bağlantı ağırlığındaki değişim miktarı da şöyle hesaplanır:

$$\Delta W_{lp}^{\text{yeni}} = \lambda \delta_p O_l + \alpha \Delta W_{lp}^{\text{eski}} \quad (3.14)$$

$$\delta_p = O_p (1 - O_p) \left(\sum_{j=1}^n \delta_j W_{pj} \right) \quad (3.15)$$

δ_j , sonraki katmanın yayılmış hata değerleridir (eğer p son gizli katmanda ise δ_j yukarıdaki hesaplanan δ_t 'ye eşittir).

n, sonraki katmanın proses elemanı sayısıdır.

W_{pj} , sonraki katmanlardaki p ve j proses elemanları arasındaki bağlantı ağırlığıdır.

Bütün ağın yeni bağlantı ağırlıkları şöyle hesaplanır:

$$W_{\text{yeni}} = W_{\text{eski}} + \Delta W_{\text{yeni}} \quad (3.16)$$

Ağın bias bağlantılarının da aynı yolla hesaplandığına dikkat edilmelidir.

Bu ileri ve geriye doğru olan işlemler, çıktı katmanındaki hata kabul edilebilir bir seviyeye düşürülünceye kadar veya önceden belirlenen iterasyon sayısı tamamlanıncaya kadar tekrar edilir.

3.5. Yapay Sinir Ağı Modeli

Bu çalışmada atölye tipi üretimde teslim tarihi belirlemede bir yapay sinir ağı modeli önerilmektedir (Şekil 3.4). Kullanılan yapay sinir ağı, geriye yayılım sinir ağıdır. Daha önce söz konusu edilen regresyon modellerinde, o regresyon modeline ait denklemdeki iş, atölye veya her ikisi hakkında bir veya birkaç bilgi kullanılmaktadır. Önerilen modelde yapay sinir ağının iş akış zamanını tahmin etmede regresyon modellerinde kullanılan bilgilerin yanı sıra daha başka iş ve atölye bilgileri de kullanılmaktadır.

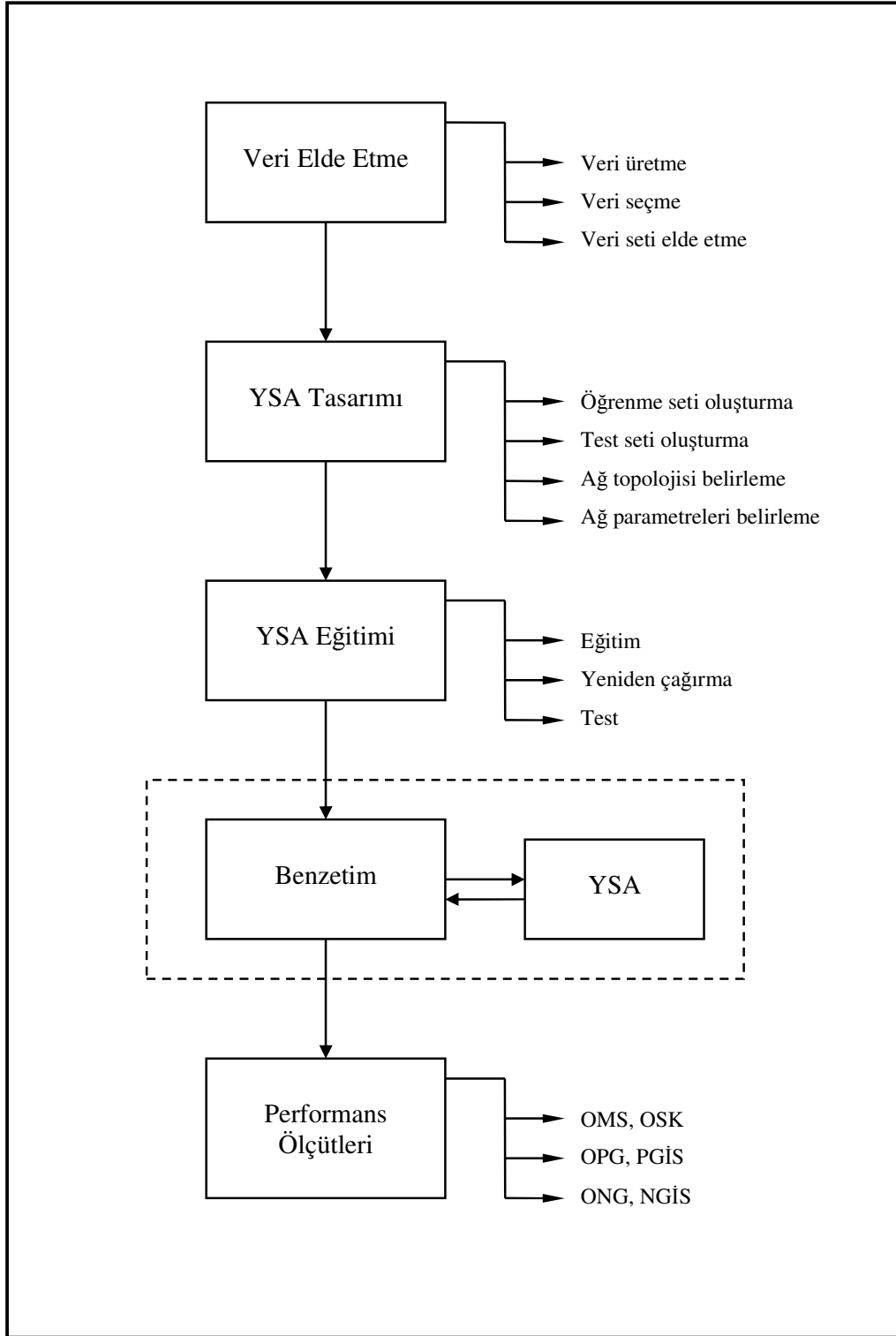
Yapay sinir ağı kullanılarak akış zamanının dolayısıyla teslim tarihinin tahmininde kullanılacak olan bilgiler Tablo 3.6’da gösterilmiştir. Bu bilgiler yapay sinir ağının girdi bilgileri olup sayısı 15 adettir. Dolayısıyla yapay sinir ağının girdi sayısı 15’dir, girdi katmanında 15 proses elemanı bulunmaktadır. Çıktı katmanında tek bir proses elemanı olan akış zamanı bilgisi bulunmaktadır.

Tablo 3.6. Yapay sinir ağının girdileri

Girdi	Bilgi
1	İşin maksimum operasyon süresi (Maks. P)
2	İşin operasyon süreleri toplamı (İş yükü, OST)
3	Atölyedeki toplam iş sayısı (TİS)
4	Kuyruklarda bekleyen işlerin toplam operasyon sayısı (TOS)
5	İşlerin ortalama gecikme süreleri toplamı (OGST)
6	Kuyruklarda bekleyen işlerin operasyon süreleri toplamı (Toplam iş yükü, OST)
7..15	İşin 1, 2, ..., 9. operasyon süreleri (P)

3.5.1. Öğrenme setinin oluşturulması

Yapay sinir ağına öğrenme seti olarak kullanılacak olan örnekler (girdi/çıktı), regresyon modelleri katsayıları belirlemek için üretilen veri setiyle aynıdır. Regresyon modelindeki 5000 işlik ısınma periyodundan sonraki 10000 işlik 10 adet benzetim sonucunda elde edilen 10 veri dosyasından her 10 işten bir tanesi alınmak üzere oluşturulan 10000 işlik tek bir veri seti yapay sinir ağı eğitimi için de



Şekil 3.4. Önerilen yapay sinir ağı modeli

kullanılmıştır. Yine EKİS ve EETT öncelik kurallarının kullanıldığı 2 adet veri seti bulunmaktadır. Her iki öncelik kuralı için farklı yapay sinir ağları eğitilmiştir. Yapay sinir ağı eğitimi için kullanılan veri setindeki bilgilere örnek Tablo 3.7'de verilmiştir.

Veri setindeki bu bilgiler bu halleriyle yapay sinir ağı eğitimi için uygun değildir. Kullanılan yapay sinir ağında aktivasyon fonksiyonu olarak sigmoid fonksiyon kullanılacağından dolayı sonuçlar 0-1 aralığında çıkacaktır. Yani akış zamanı için ağın 1'den büyük veya 0'dan küçük bir değer üretmesi mümkün değildir (Öztemel, 2003). Bu nedenle ağın girdi ve çıktı değerleri normalize edilerek ölçeklendirilmiş veri seti elde edilmiştir. Tablo 3.8'de normalize edilmiş veri seti örneği gösterilmektedir. Ağın eğitildikten sonraki çıktı değerleri de ters normalizasyon yapılarak normal akış zamanı değerleri elde edilmiştir.

Normalizasyon işlemi şu denkleme göre yapılmıştır:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.17)$$

Tablo 3.7'deki son sütunda bulunan F değerlerini normalize etmek istediğimizde, öncelikle bu sütunda bulunan değerlerden minimum ve maksimum olanını bulmamız gerekir. Değerlere baktığımızda minimum değer 103, maksimum değer 544 olduğunu görürüz. Bunun anlamı normalizasyon sonrası 103 için 0, 544 için 1 değeri elde edilecek olmasıdır. Son sütunun ilk satırındaki 155 değerinin normalizasyon sonrası değeri hesaplanması şu şekildedir:

$$x' = \frac{(155 - 103)}{(544 - 103)} = \frac{52}{441} = 0,1179$$

Ters normalizasyon işlemi de, normalizasyon denkleminde şu şekilde elde edilmiştir:

$$x = x' (x_{\max} - x_{\min}) + x_{\min} \quad (3.18)$$

Tablo 3.7. Yapay sinir ağı eğitim seti (EKİS)

Maks. P	OST	TİS	TOS	Girdiler											Çıktı F	
				OGST	AOST	P										
						1	2	3	4	5	6	7	8	9		
...
13	61	41	273	1630	11930	3	5	9	6	9	5	6	13	5	155	
18	99	44	278	1353	7747	7	11	17	8	11	8	18	16	2	544	
6	45	36	235	788	9747	6	5	6	6	3	6	6	6	2	132	
15	66	32	212	662	11038	1	4	11	15	7	10	8	7	4	165	
15	58	36	213	727	7972	2	7	3	5	5	15	4	2	14	169	
50	102	35	222	1058	10668	7	6	7	6	7	50	4	5	10	449	
27	66	39	257	1241	13630	11	3	2	1	10	2	7	3	27	165	
32	97	37	221	537	5654	1	2	1	32	15	5	11	2	26	154	
26	76	38	214	660	8428	6	5	7	2	26	2	5	13	10	254	
21	56	41	244	1267	8114	5	3	4	1	11	21	6	2	3	103	
...	

Tablo 3.8. Yapay sinir ağı eğitim seti – Normalize edilmiş değerler (EKİS)

Maks. P	OST	TİS	TOS	OGST	AOST	Girdiler									Çıktı F	
						P										
						1	2	3	4	5	6	7	8	9		
...
0.1081	0.2083	0.5231	0.5137	0.2062	0.4911	0.0308	0.0580	0.1159	0.0725	0.1231	0.0513	0.0641	0.1538	0.0580	0.0098	
0.0135	0.1250	0.4462	0.4190	0.0942	0.3991	0.0769	0.0580	0.0725	0.0725	0.0308	0.0641	0.0641	0.0641	0.0145	0.0077	
0.1351	0.2344	0.3846	0.3616	0.0775	0.4535	0.0000	0.0435	0.1449	0.2029	0.0923	0.1154	0.0897	0.0769	0.0435	0.0107	
0.6081	0.4219	0.4308	0.3865	0.1301	0.4379	0.0923	0.0725	0.0870	0.0725	0.0923	0.6282	0.0385	0.0513	0.1304	0.0370	
0.2973	0.2344	0.4923	0.4738	0.1545	0.5628	0.1538	0.0290	0.0145	0.0000	0.1385	0.0128	0.0769	0.0256	0.3768	0.0107	
0.3649	0.3958	0.4615	0.3840	0.0609	0.2265	0.0000	0.0145	0.0000	0.4493	0.2154	0.0513	0.1282	0.0128	0.3623	0.0097	
0.2838	0.2865	0.4769	0.3666	0.0772	0.3434	0.0769	0.0580	0.0870	0.0145	0.3846	0.0128	0.0513	0.1538	0.1304	0.0190	
0.2162	0.1823	0.5231	0.4414	0.1579	0.3302	0.0615	0.0290	0.0435	0.0000	0.1538	0.2564	0.0641	0.0128	0.0290	0.0050	
0.3514	0.3073	0.6000	0.4414	0.1364	0.3605	0.4615	0.0435	0.0145	0.1014	0.0462	0.0641	0.0385	0.0128	0.2464	0.0182	
...	

Yapay sinir ağı eğitildikten sonra akış zamanı tahmininde kullanılırken verilen girdi değerlerine örneğin 0,3476 çıktı değerini vermiş olsun. Bu durumda yapılacak olan önceki kayda alınan minimum ve maksimum değerlerini kullanarak ters normalizasyon yaparak gerçek akış zamanı değerini hesaplamaktır. Gerçek akış zamanı değeri şu şekilde hesaplanmaktadır:

$$x = (0,3476)(544 - 103) + 103 = (0,3476)(441) + 103$$

$$x = (153,2916) + 103 \cong 256$$

Normalizasyon ve ters normalizasyon işlemleri her bir girdi ve çıktı için ayrı ayrı yapılmıştır. Yani, her bir girdi ve çıktının minimum ve maksimum değerleri farklıdır. Normalizasyon işlemi sonrasında anlaşılacağı üzere, her bir girdi ve çıktıda en küçük değer 0, en büyük değer ise 1 değerini alacaktır. Yapay sinir ağı eğitimi öncesi yapılması gerekli olan bu normalizasyon işleminde bulunan minimum ve maksimum değerler, daha sonra eğitilmiş yapay sinir ağı kullanılırken ters normalizasyon için gerekli olduğundan bir dosyada bu değerler kayda alınmıştır.

Yapay sinir ağının eğitimi için veri setinin hazır olabilmesi için veri setinin iki parçaya bölünmesi gereklidir. Çünkü bir yapay sinir ağı eğitimi için öğrenme ve test olmak üzere iki set gereklidir. Ağ eğitim sırasında zaten öğrenme setindeki bütün örnekleri görmektedir. İstenilen hata seviyesi yakalanıncaya veya istenen iterasyon veya epok (ağın bütün örnekleri bir defa görmesi) sayısı tamamlanıncaya kadar girdilerden çıktı değerleri hesaplamakta ve istenen çıktı ile arasındaki farka bakarak proses elemanları arasındaki ağırlık değerlerini değiştirmektedir. Dolayısıyla ağ için tek bir setle öğrendi denilemez. Bu nedenle, normalize edilen veri seti öğrenme ve test olmak üzere iki sete ayrılmıştır. İkiye ayırma işlemi, ilk veri setinin %80'i öğrenme %20'si test setinde olacak şekilde rastgele yapılmıştır (Philipoom ve diğerleri, 1994).

3.5.2. Yapay sinir ağının oluşturulması

Yapay sinir ağı ile akış zamanı tahmininde 15 değer girdi olarak kullanılacağından 15 adet girdi ve akış zamanını tahmin eden bir adet çıktı proses elemanı

kullanılmıştır. EKİS ve EETT öncelik kuralı kullanılarak elde edilen veriler için her iki durum akış zamanı tahmininde için yapay sinir ağı sırasıyla 7 ve 9 adet proses elemanı olan tek gizli katmanlı olarak tasarlanmıştır. Her iki öncelik kuralına ait verilerle eğitim işlemi aynı parametrelerle yapılmıştır (Tablo 3.9).

3.5.3. Yapay sinir ağının eğitilmesi

Yapay sinir ağı eğitimi hem EKİS hem de EETT için 2000 epok olarak gerçekleştirilmiştir. Eğitim sırasında 8000 ışık öğrenme seti kullanıldığından, her bir eğitim 16000000 iterasyon olarak gerçekleşmiştir. Eğitim işlemi sırasında her bir epok sonunda çıktı proses elemanı için mutlak sapma ve mutlak sapma kare değerleri hesaplanmış ve kaydedilmiştir. Şekil 3.5 ve Şekil 3.6'da yapay sinir ağlarının ortalama sapma kare değerlerinin grafikleri gösterilmektedir.

Yapay sinir ağı eğitiminde ağın öğrenip öğrenmediğinin belirlenmesi için öğrenme seti ve ağın hiç görmediği test seti ağa girdi olarak verilip alınan çıktılar gerçek çıktı değerleriyle karşılaştırılır. Aradaki fark kabul edilebilir hata payının altında ise o girdi örneğine doğru cevap verdi, değilse yanlış cevap verdi denilir. Tablo 3.10'da her iki ağın öğrenme başarı yüzdeleri verilmiştir.

3.5.4. Yapay sinir ağı ile teslim tarihi belirleme

Yapay sinir ağlarının eğitimi tamamlandıktan sonra, EKİS ve EETT öncelik kurallarının kullanıldığı 10'ar adet benzetim yapılmıştır. Yine bu 10 benzetimde birbirinden ve aynı zamanda yapay sinir ağlarının eğitimleri için ve aynı zamanda regresyon model katsayılarının belirlenmesi için gerekli veri setinin elde edilmesi için yapılan benzetimlerden farklı başlangıç rassal sayıları kullanılmıştır. Kullanılan bu başlangıç rassal sayıları daha önce regresyon modelleri kullanılarak teslim tarihine karar verilen benzetimdekilerle aynı olarak kullanılmıştır.

Tablo 3.9. Yapay sinir ağı parametreleri (EKİS, EETT)

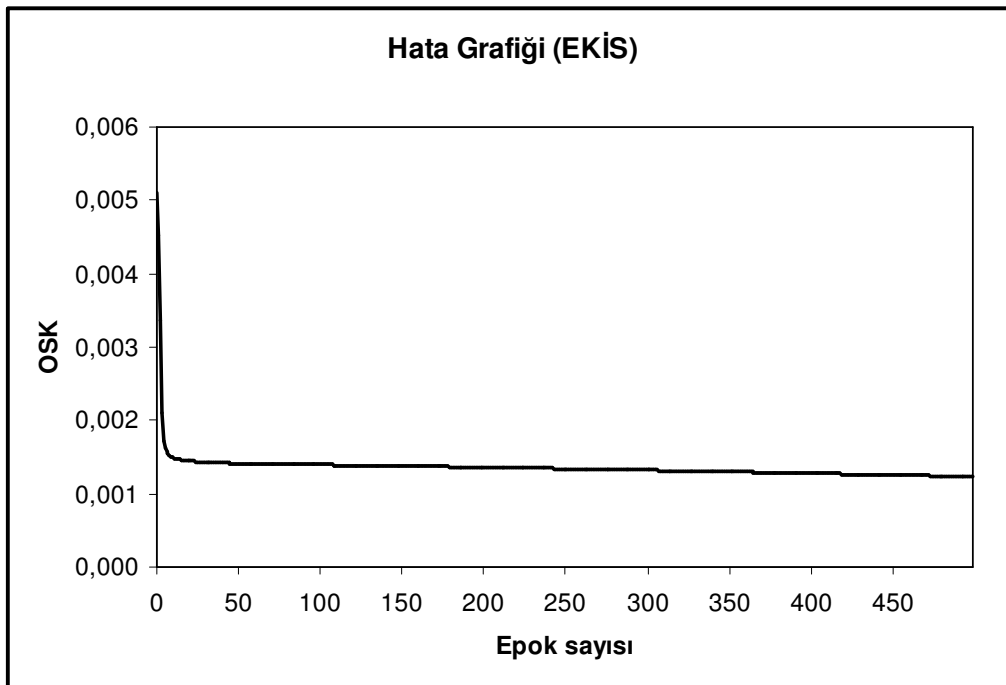
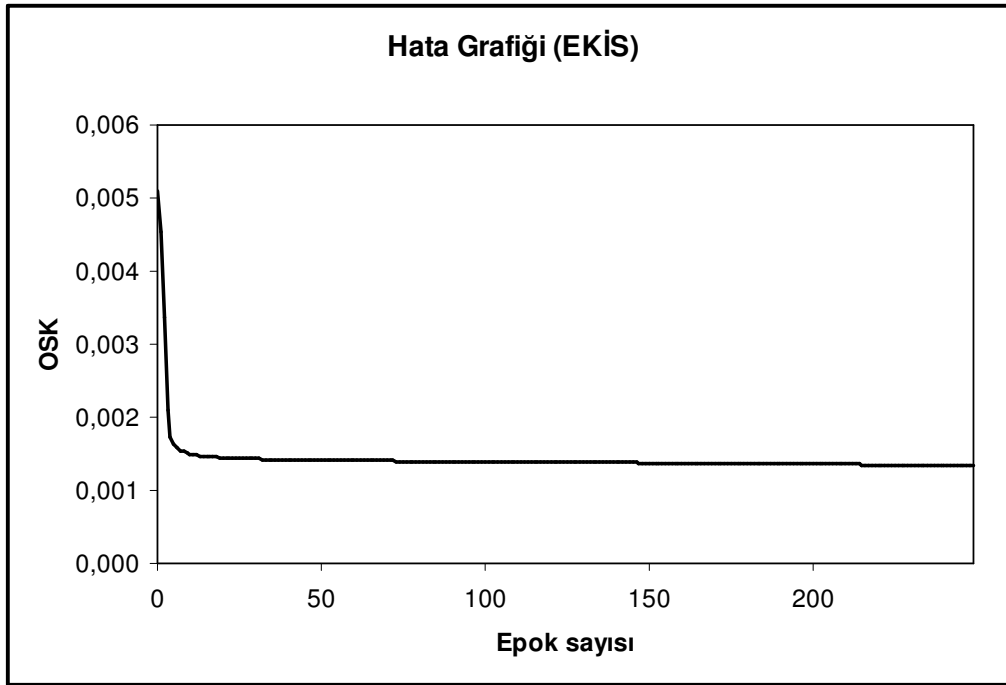
Parametre	Değer
Öğrenme katsayısı	0,2
Momentum katsayısı	0,8
Başlangıç ağırlık değerleri	-0,1 ile 0,1 arasında rastgele
Örnek gösterimi	Sıralı
Epok sayısı	2000

Tablo 3.10. YSA öğrenme başarıları

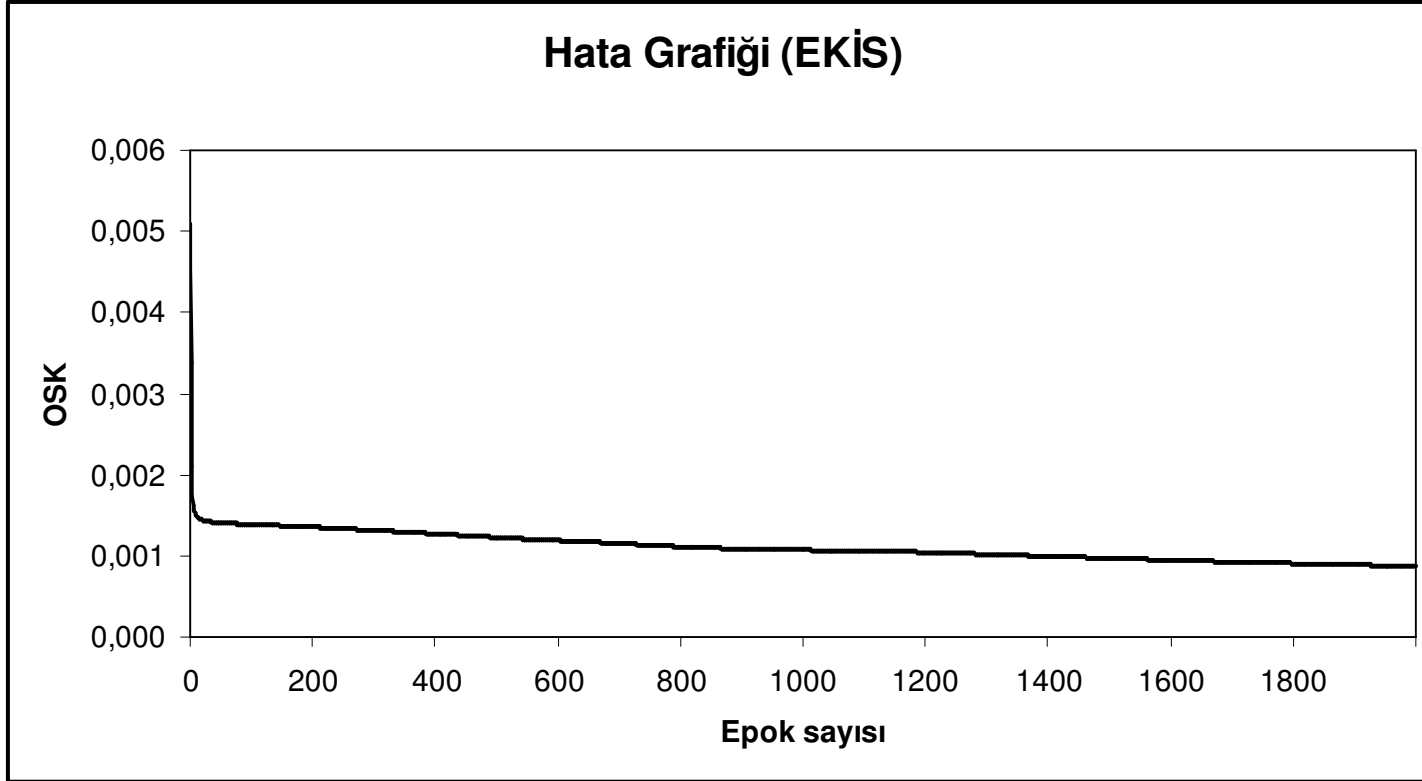
Epok	EKİS		EETT	
	Öğrenme seti	Test seti	Öğrenme seti	Test seti
	(%)	(%)	(%)	(%)
500	96	96	94	94
1000	97	97	95	95
1500	97	97	95	95
2000	97	97	95	95

Bu yapılan 10 benzetimde, atölyeye işler geldiği anda teslim tarihleri belirlenirken işlerin geldiği zamana eğitilmiş yapay sinir ağı kullanılarak tahmin edilen akış zamanı eklenerek teslim tarihi belirlenmektedir. Her benzetimde 5000 işlik ısınma periyodundan sonraki 10000 iş için veriler kaydedilmiştir. Tablo 3.11’de yapay sinir ağına sunulan 10 işe ait örnek girdiler ve bu işe ait gerçekleşen akış zamanı verilmiştir. Bu değerlerin yapay sinir ağına sunulmadan önce normalize edilmesi gerektiğinden, önce değerler normalize edilip daha sonra ağa sunulmuştur. Normalize edilen değerler Tablo 3.12’de verilmiştir. Tablodaki işi örnek alırsak, bu iş için akış zamanı 160 olup, normalize edilmiş akış zamanı 0,0103’tür. Yani yapay sinir ağından beklenen en iyi cevap 0,0103 değeridir. Eğitilen yapay sinir ağının bu işe ait girdilere verdiği akış zamanı tahmini 0,0393’tür. Bu değer ters normalizasyon ile (Denklem 3.18) gerçek değerine çevrilmelidir:

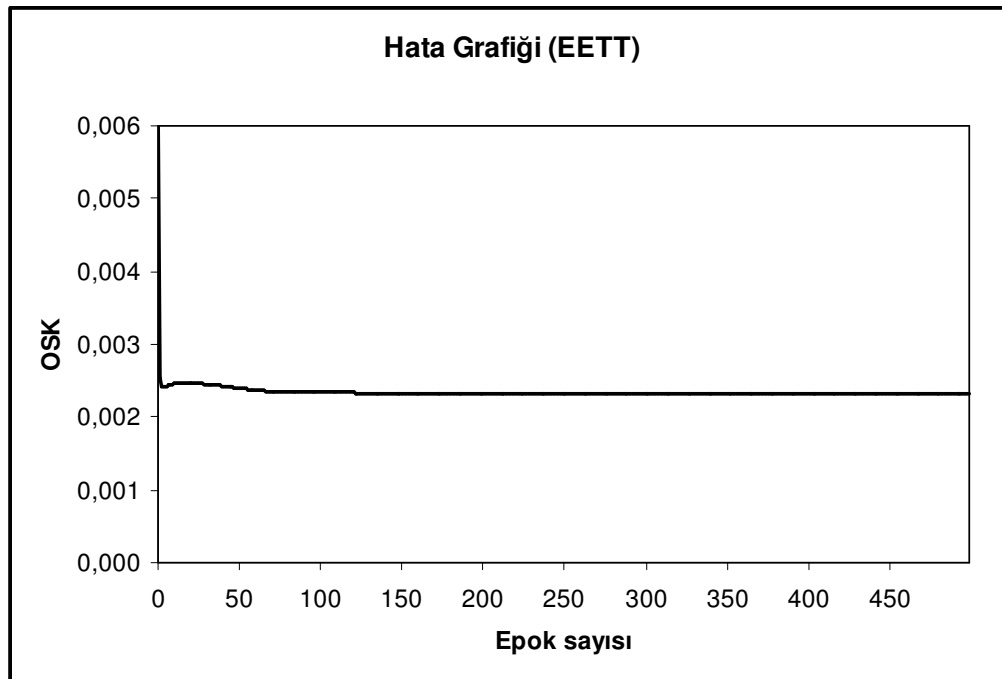
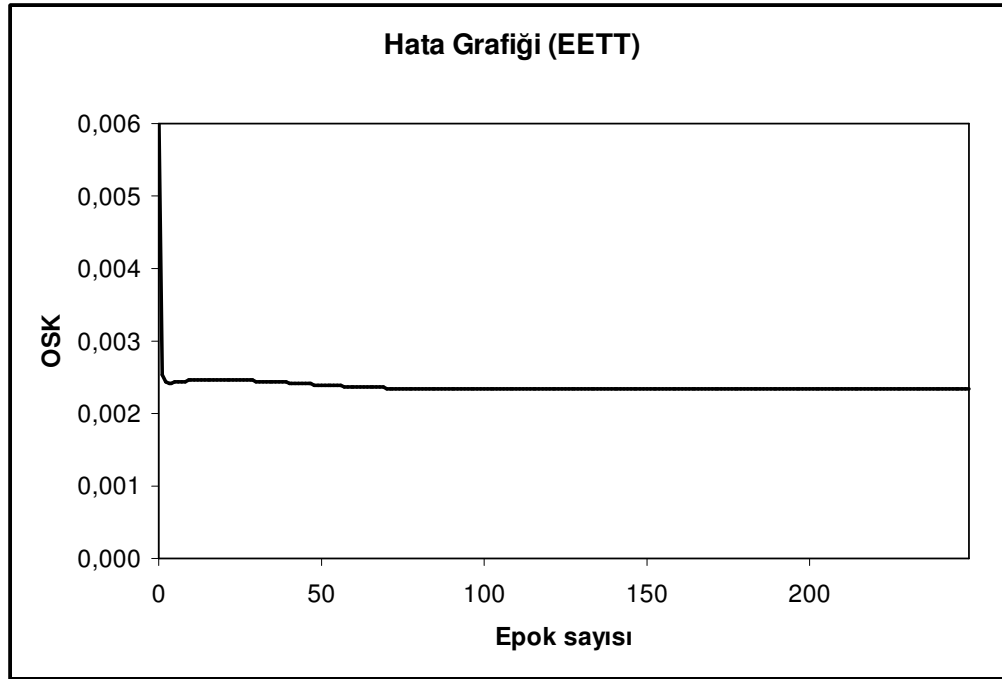
$$F_1' = (0,0393)(10860 - 49) + 49 = 474$$



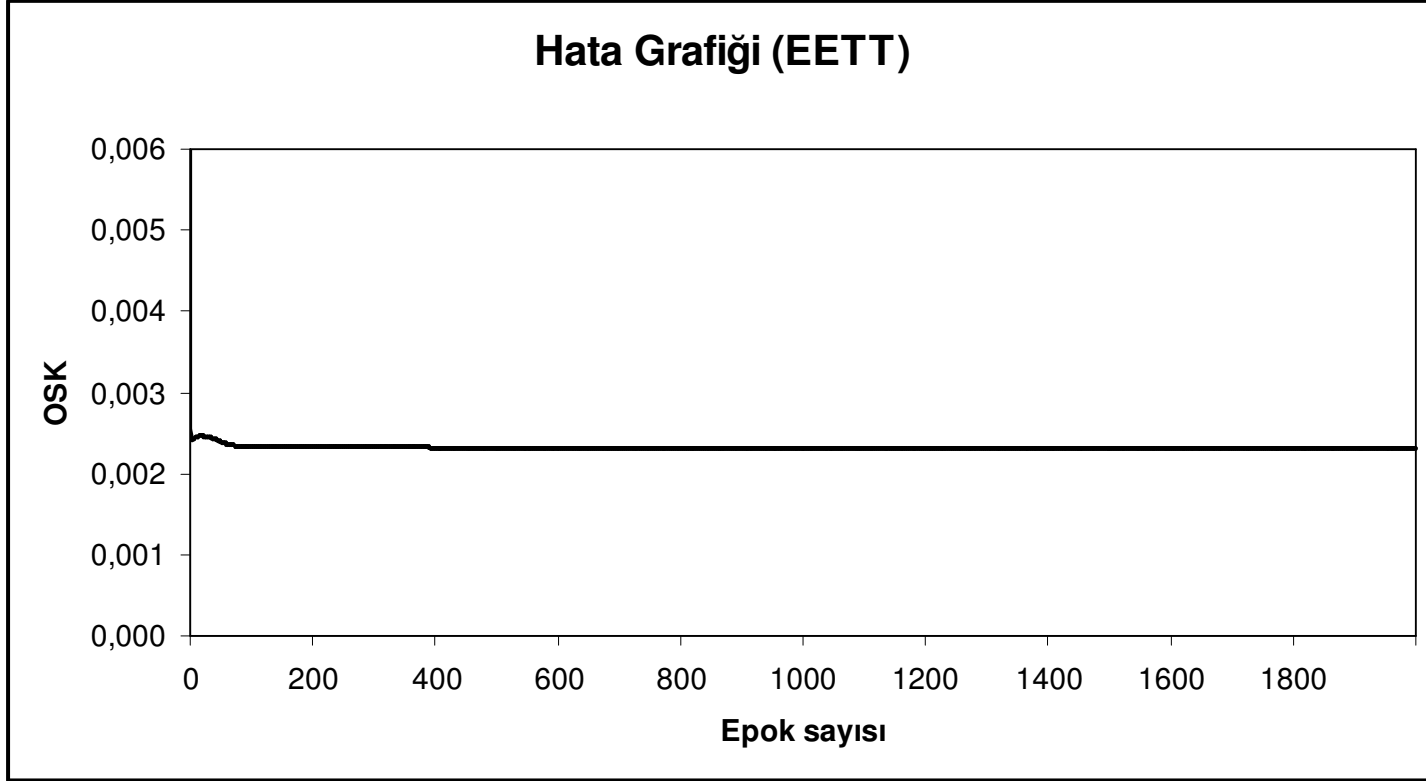
Şekil 3.5. Yapay sinir ađı hata grafiđi (EKİS)



Şekil 3.5. (Devam) Yapay sinir ađı hata grafiđi (EKİS)



Şekil 3.6. Yapay sinir ađı hata grafiđi (EETT)



Şekil 3.6. (Devam) Yapay sinir ađı hata grafiđi (EETT)

Tablo 3.11. Yapay sinir ağı girdi seti (EKİS)

Maks. P	OST	TİS	TOS	Girdiler											Çıktı F
				OGST	AOST	P									
						1	2	3	4	5	6	7	8	9	
...
25	68	43	95	274	499	19	25	3	5	2	2	7	4	1	160
17	75	43	170	275	533	6	4	8	5	6	17	11	5	13	194
10	52	42	186	271	514	5	3	2	9	1	7	10	7	7	132
37	128	42	484	278	542	37	4	21	2	2	11	37	2	11	1445
15	59	42	220	281	516	15	6	13	4	4	4	4	1	9	144
13	64	43	1247	272	540	9	4	12	2	2	4	13	11	7	190
33	89	42	1226	270	532	1	1	10	20	6	33	9	8	1	319
20	69	42	133	261	543	6	4	2	20	7	13	2	13	1	191
38	91	41	595	253	537	27	38	7	1	10	2	2	1	4	412
25	78	41	1157	246	543	6	7	3	12	11	4	5	25	4	372
...

Tablo 3.12. Yapay sinir ağı girdi seti – Normalize edilmiş değerler (EKİS)

Maks. P	OST	TİS	TOS	OGST	AOST	Girdiler									Çıktı F	
						P										
						1	2	3	4	5	6	7	8	9		
...
0.2703	0.2448	0.5538	0.0698	0.0259	0.0091	0.2769	0.3478	0.0290	0.0580	0.0154	0.0128	0.0769	0.0385	0.0000	0.0103	
0.1622	0.2813	0.5538	0.2569	0.0261	0.0105	0.0769	0.0435	0.1014	0.0580	0.0769	0.2051	0.1282	0.0513	0.1739	0.0134	
0.0676	0.1615	0.5385	0.2968	0.0255	0.0097	0.0615	0.0290	0.0145	0.1159	0.0000	0.0769	0.1154	0.0769	0.0870	0.0077	
0.4324	0.5573	0.5385	1.0399	0.0265	0.0109	0.5538	0.0435	0.2899	0.0145	0.0154	0.1282	0.4615	0.0128	0.1449	0.1291	
0.1351	0.1979	0.5385	0.3815	0.0269	0.0098	0.2154	0.0725	0.1739	0.0435	0.0462	0.0385	0.0385	0.0000	0.1159	0.0088	
0.1081	0.2240	0.5538	2.9426	0.0257	0.0108	0.1231	0.0435	0.1594	0.0145	0.0154	0.0385	0.1538	0.1282	0.0870	0.0130	
0.3784	0.3542	0.5385	2.8903	0.0254	0.0105	0.0000	0.0000	0.1304	0.2754	0.0769	0.4103	0.1026	0.0897	0.0000	0.0250	
0.2027	0.2500	0.5385	0.1646	0.0242	0.0109	0.0769	0.0435	0.0145	0.2754	0.0923	0.1538	0.0128	0.1538	0.0000	0.0131	
0.4459	0.3646	0.5231	1.3167	0.0231	0.0107	0.4000	0.5362	0.0870	0.0000	0.1385	0.0128	0.0128	0.0000	0.0435	0.0336	
...

Buradaki 10860 ve 49 değerleri akış zamanı için daha önce bulunup saklanan, sırasıyla x_{\max} (F_{\max}) ve x_{\min} (F_{\min}) değerleridir. Bu iş için gerçekleşen akış zamanı $F_1=160$ olduğundan yapılan tahmine göre iş gecikmiştir. Bu gecikme (L_1),

$$L_1 = F_1' - F_1$$

$$L_1 = 474 - 160$$

$$L_1 = 314$$

dür. L_1 pozitif olduğundan iş pozitif gecikmiştir. Dolayısıyla işin pozitif gecikmesi (T_1) 314, negatif gecikmesi veya erken bitmesi (E_1) 0'dır. Aynı zamanda işin tahmin edilen akış zamanının gerçekleşen akış zamanından mutlak sapması 314, mutlak sapma karesi 98596'dır.

Bu hesaplama şekliyle elde edilen 10000 işin tahmin edilen teslim tarihlerinin gerçekleşen tamamlanma zamanlarından (C_i) mutlak farklarının aritmetik ortalaması alınarak ortalama mutlak sapma ve yine mutlak fark karelerinin aritmetik ortalaması alınarak ortalama mutlak sapma kareleri hesaplanmıştır. Ayrıca pozitif geciken iş sayısı, ortalama pozitif gecikme, erken biten iş sayısı ve ortalama erken bitme değerleri hesaplanmıştır. Bu hesaplamalar 10 adet elde edilen veri dosyaları için ayrı ayrı yapılmıştır. Hesaplanan bu 10 değer aritmetik ortalamaları alınarak kullanılan öncelik kuralı ve yapay sinir ağı modeli için daha önce bahsedilen regresyon modelleri değerleriyle karşılaştırılmak üzere OMS, OSK, PGİS, OPG, NGİS ve ONG değerleri hesaplanmış olmaktadır.

3.6. Her İki Modelin Performans Ölçütlerine Göre Karşılaştırılması

Regresyon modelleri ve yapay sinir ağı modeli ile teslim tarihi belirleme sonucu elde edilen OMS, OSK, PGİS, OPG, NGİS ve ONG performans ölçütleri göz önüne alınarak, regresyon modelleri veya yapay sinir ağı modeli daha iyi veya daha kötü sonuç verip vermedikleri açısından değerlendirilmektedir.

BÖLÜM 4. BENZETİM VE YAPAY SINIR AĞI İLE MODELLEME

4.1. Giriş

Bu bölümde regresyon modelleri ve yapay sinir ağı modeli ile teslim tarihi belirlemede kullanılan dinamik atölye yapısı, benzetim ve benzetim yazılımı, yapay sinir ağı yazılımı hakkında detaylı bilgi verilmektedir.

Benzetim tekniği birçok araştırmacı tarafından çizelgeleme problemlerinin incelenmesinde kullanılmıştır (Conway, 1967, Abdallah, 1973). Regresyon modellerinin katsayılarının belirlenmesinde ve yapay sinir ağının eğitilmesinde gerekli olan verileri üretecek olan ve bu iki modelin performans karşılaştırmasını yapmakta kullanılacak olan iki benzetim modeli geliştirilmiştir. Bu modellerden biri EKİS, diğeri EETT öncelik kuralını kullanandır. Model geliştirilirken benzetimde kullanılan zaman artımlı benzetim ve olay artımlı benzetim tekniklerinden olay artımlı benzetim tekniği kullanılmıştır. Bu teknik benzetim işlemini diğerk tekniğe oranla oldukça hızlandırmaktadır. Bu amaçla bu teknik seçilmiştir.

Bu çalışmada teslim tarihi belirlemek için önerilen yapay sinir ağı modeli için de bir yapay sinir ağı yazılımı geliştirilmiştir. Kullanılan yapay sinir ağı geriye yayılım yapay sinir ağıdır. Bu yazılım regresyon modellerine alternatif olarak teslim tarihi belirlemek üzere tasarlanmıştır. Bir bütün olarak bu yazılım eğitim amaçlı kullanılmıştır. Daha sonra geliştirilen yapay sinir ağı modelinin benzetim sırasında teslim tarihi belirlemesini yapabilmesi için, ilgili yazılım bölümleri benzetim yazılımı içine gömülmüştür. Benzetim modelinde de bu yapıya uyumu sağlayacak olan değişiklikler yapılmıştır.

Hem benzetim hem de yapay sinir ağı yazılımı geliştirilirken geleneksel değişken tanımlamalarından farklı bir yapı olan dinamik değişken tanımlamaları kullanılmıştır. Her iki yazılımda da çalışma sırasında bilginin hafızada saklanması için kısıtlı kullanım sunan dizi değişkenler yerine dinamik değişkenler kullanılmıştır. Bunun sağladığı avantaj, programlama dilinin izin verdiği hafıza alanının dışına çıkarak bilgisayar hafızasının tamamının kullanılabilmesidir.

4.2. Düşünsel Atölye Sistemi

Regresyon modelleri ile yapay sinir ağı modelinin teslim tarihi belirlemedeki performanslarının karşılaştırılabilmesi için 9 makinalı bir atölye düşünülmüştür (Taşgetiren, 1996). Baker ve Dzielinski (1960) 9'dan 30 makina kadar atölyeyi test ederek atölye büyüklüğünün önemli bir etken olmadığını göstermiştir. Her işin 9 makinanın her birinde işlem gördüğü 5 farklı iş tipi vardır. Bu 5 farklı iş tipine (rota) ait makina sırası Tablo 4.1'de verilmiştir. Alternatif rotaya izin verilmemektedir. İşlem zamanları hazırlık ve taşıma zamanlarını da içermektedir. Makinaların önündeki kuyrukların sonsuz kapasiteli olduğu varsayılmıştır. Makina bozulmalarına izin verilmemektedir.

Tablo 4.1. İşlere ait rotalar

Rota	Makinalar
1	3-2-1-5-4-9-6-8-7
2	5-3-4-1-2-9-7-6-8
3	8-1-6-5-2-3-7-9-4
4	3-5-4-1-7-2-9-8-6
5	6-2-2-5-9-1-4-8-7

İşler atölyeye geldiğinde 5 farklı rotadan biri rasgele olarak o iş için belirlenmiştir. İşlerin atölyeye gelişler arası süreleri ve işlem süreleri sırasıyla ortalamaları 10 ve 9 zaman birimi olan üstel dağılımdan elde edilmiştir. Rota belirlenirken ve üstel dağılımdan sayı elde edilirken uniform dağılımdan gelen rassal sayılar kullanılmıştır. Rassal sayı üretmek için şu yöntem kullanılmıştır (Carrie, 1988):

Başlangıç sayısı = 0,645329

ve

asal sayı = 317

olsun. Bu iki sayı birbiri ile çarpılarak bir başka sayı elde edilir. Elde edilen sayının virgülden sonraki kısmı rassal sayı olarak alınır. Alınan bu sayı aynı zamanda bir sonraki rassal sayı üretimi için başlangıç sayısı olmaktadır (Tablo 4.2).

Tablo 4.2. Rassal sayı üretimi

Başlangıç sayısı	Asal sayı	Çarpım sonucu	Rassal sayı
0,645329	317	204,569293	0,569293
0,569293	317	180,465881	0,465881
0,465881	317	147,684277	0,684277
...

Ortalama gelişler arası sürenin 10 ve ortalama işlem süresinin 9 alınmasının amacı %90 atölye kullanım oranını elde etmektir. Beklenen atölye kullanım oranı şu şekilde belirlenebilir (Chang, 1996):

$$\rho = \lambda \mu_p \mu_g / m \quad (4.1)$$

Burada ρ beklenen atölye kullanım oranı, λ iş geliş oranı (=1/ortalama gelişler arası süre), μ_p operasyon başına ortalama işlem zamanı, μ_g işin operasyon sayısı ve m atölyedeki makina sayısını göstermektedir. Bu çalışmada seçilen atölye için $\mu_g = m$ olduğundan beklenen atölye kullanım oranı şu şekilde belirlenebilir:

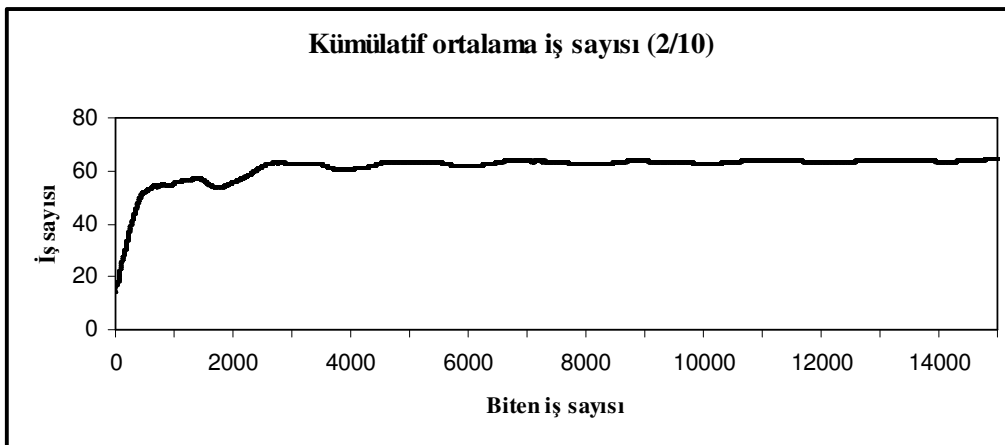
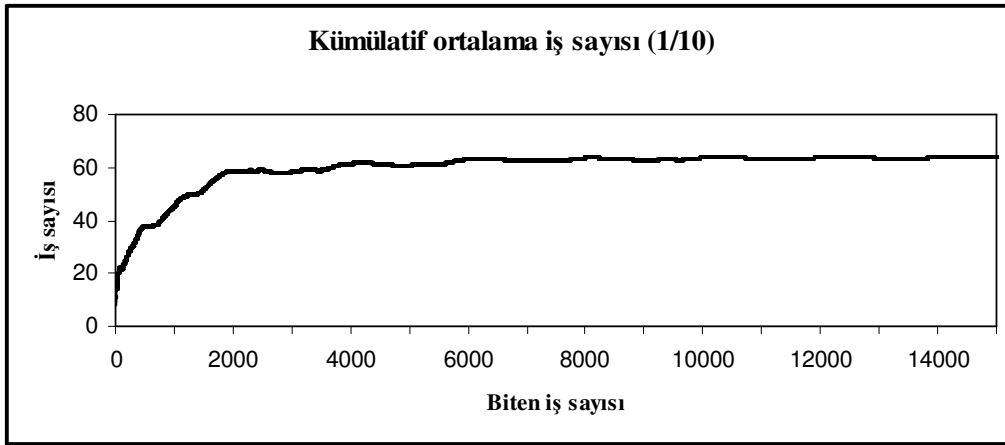
$$\rho = \lambda \mu_p \quad (4.2)$$

Gelişler arası ortalama süre 10, işlem zamanı ortalama 9 seçildiğinden,

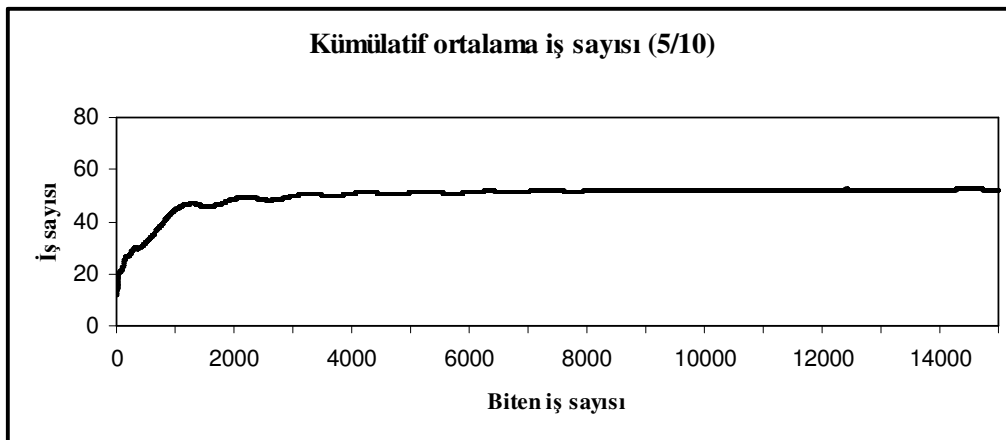
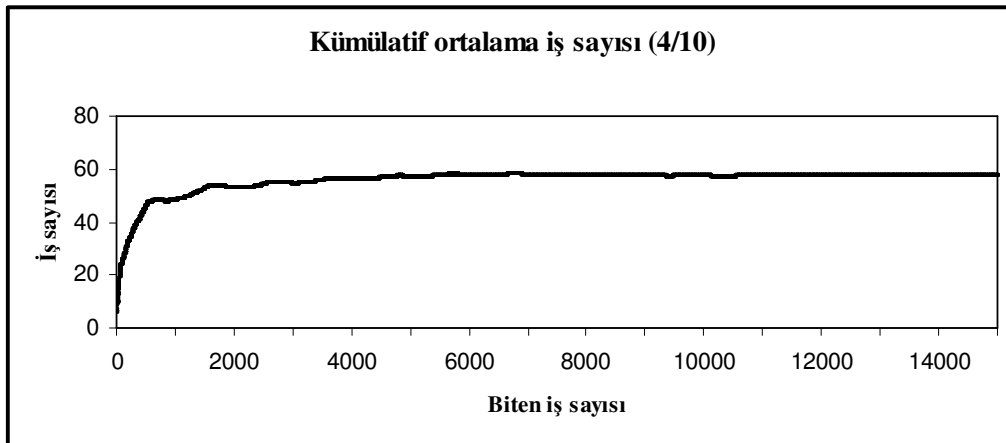
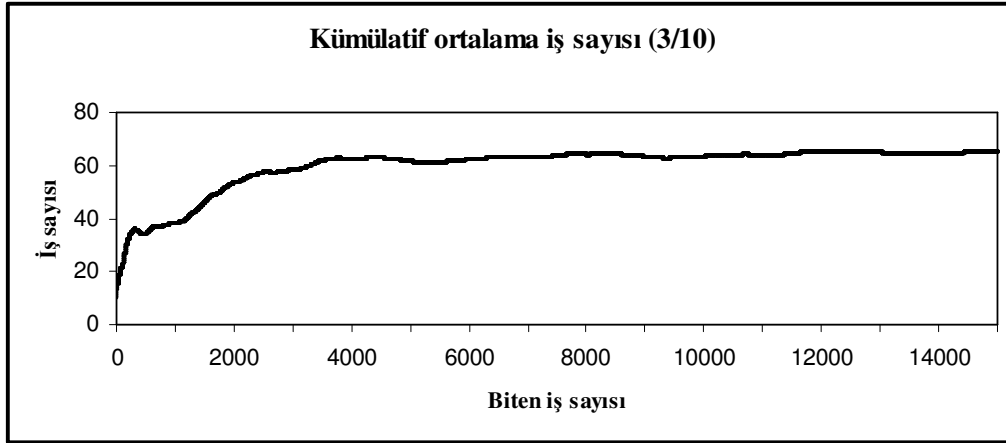
$$\rho = \lambda \mu_p = (1/10)9 = 0,9$$

değeri elde edilir. Yani atölye kullanım oranı yaklaşık olarak %90 olacaktır.

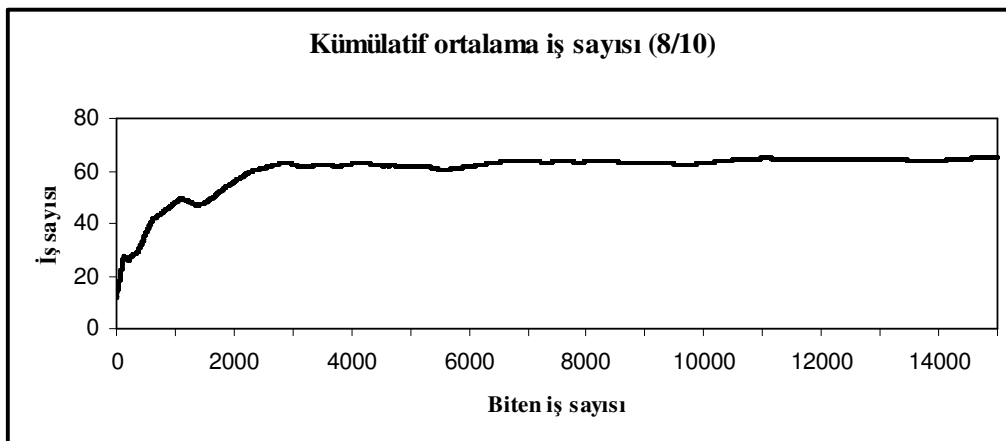
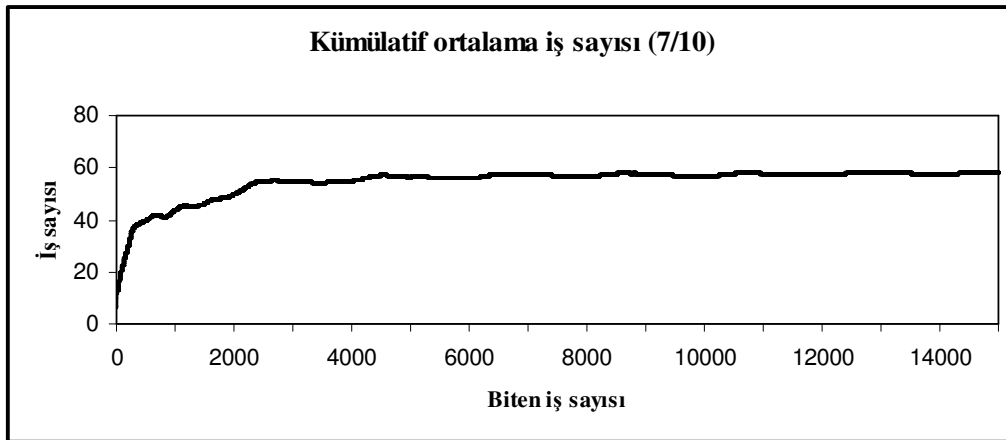
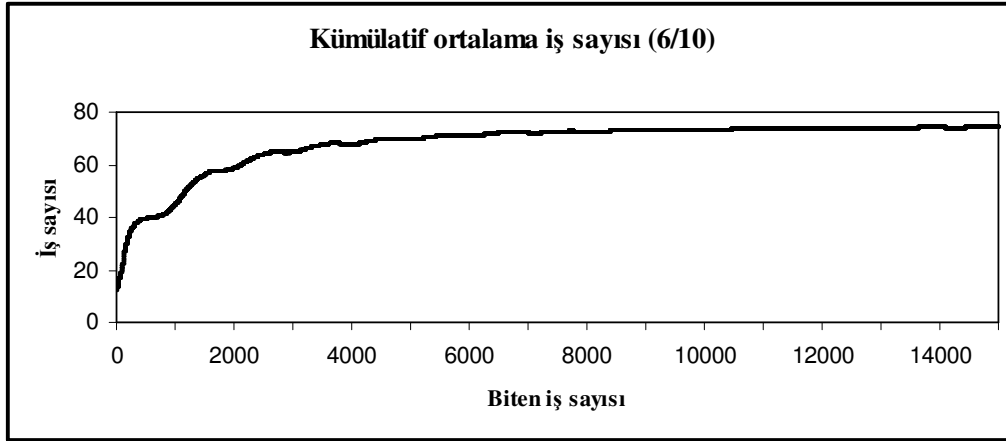
Yapılan her bir benzetimde gerekli veriler 5000 işlik ısınma periyodundan sonraki 10000 iş için kaydedilmiştir. 5000 işlik ısınma periyodun atölyenin durağan hale gelmesi için geçen süredir. Bu sürenin belirlenebilmesi için 10 adet deneme benzetimi yapılmıştır. Her birinde atölyede bulunan kümülatif ortalama iş sayıları kaydedilmiştir. Yapılan bu deneme benzetimlerine ait kümülatif ortalama iş sayılarını gösteren grafik Şekil 4.1’de verilmiştir. Bu grafikler yardımı ile ısınma periyodunun 5000 iş olmasına karar verilmiştir.



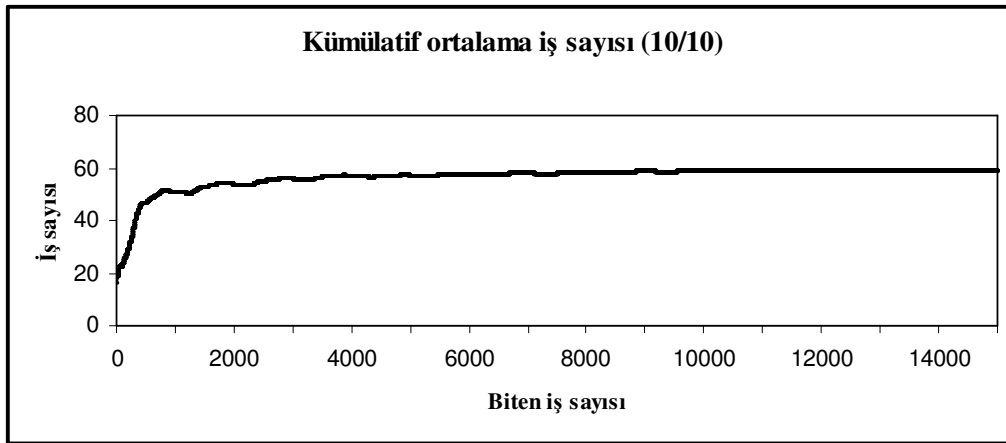
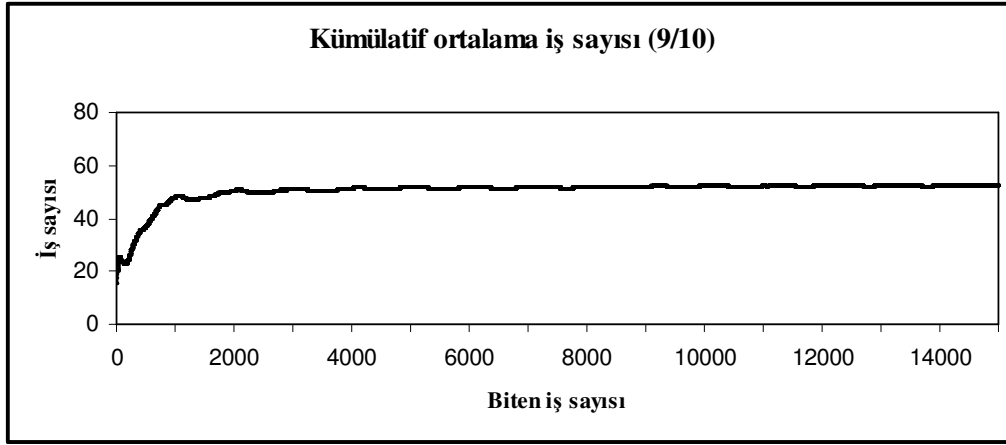
Şekil 4.1. Kümülatif ortalama iş sayıları



Şekil 4.1. (Devam) Küümülatif ortalama iş sayıları



Şekil 4.1. (Devam) Küümülatif ortalama iş sayıları



Şekil 4.1. (Devam) Küümülatif ortalama iş sayıları

4.3. Benzetim

Günümüzde işletmeleri yönetmek gittikçe zorlaşmaktadır. Bunun en önemli sebebi de sistemleri meydana getiren bileşenler arasındaki etkileşimin çok karmaşık olmasıdır. Sistemin bir bileşeni üzerindeki etkinin tüm sistemi etkilemesi ile sistem yaklaşımı ortaya çıkmıştır. Böylece yöneticiler, sistem yaklaşımını kullanırken karmaşık sistemlerin analiz ve tasarımında en önemli araçlardan birisi de benzetim olmaktadır.

4.3.1. Benzetim tanımı ve yaklaşımı

Benzetim, gerçek bir sistemin modelini tasarlama süreci ve sistemin işlemesi için sistemin davranışlarını anlamak veya değişik stratejileri değerlemek amacı ile bu model üzerinde denemeler yapmaktır (Halaç, 1998).

Bir başka tanıma göre ise; benzetim yöntemleri, doğru kararı vermek için geliştirilmiş, incelenen sistemin içindeki aksaklıkları problem olmadan önce ortaya çıkaran, kararları vermeden etkilerini inceleyen bir karar verme mekanizması ve yönetim bakış açısıdır (Güventürk, 1995).

Pek çok modelleme metotları olmasına rağmen, esnek üretim sistemlerinin dinamik yapısının analizinde yaygın olarak kullanılmaktadır. Benzetimle üretim sistemlerinin modellenmesi Carrie (1988), Law (1988), Schorer ve Tseng (1987), Newman (1990), Wang (1989) yanı sıra pek çok diğer araştırmacı tarafından da kullanılmıştır.

Benzetim, emülasyon ve animasyon, üretim sistemlerinin modellenmesinde oldukça faydalı birer araç olmalarına rağmen temel olarak birer optimizasyon tekniği değildirler ve bir üretim sisteminin tasarımında kontrol edilmesi gerekli olan parametre sayısı genellikle çok fazla çıktı nedeniyle kabadır ve analizci bazen optimal veya optimale yakın sonuçları test etmeden bırakabilir ve benzetimle modelleme önemli derecede uzmanlık ve tecrübe gerektirmektedir (Türker, 1997).

Bugün benzetim tekniği yöntemleri üç farklı biçimde uygulanmaktadır (Sarıaslan, 1988):

- Yöneylem oyunları: Bu yöntem benzetilmiş bir ortamda oyuncular ya da karar vericiler arasında menfaat çatışmasının olduğu durumları kapsar.
- Monte Carlo: Olasılıklı ve çözüm için kesin bir formülün bulunmadığı problemleri çözmek için başvurulan bir yöntemdir. Bu yöntem, ana kütleli andıran kuramsal örnekler kullanarak çözüme ulaşmaya çalışır.
- Sistem simulasyonu: Gerçek sistemi temsil eden bir model üzerinde gerçek verileri kullanarak sistemin işleyişi konusunda bilgi edinilmesini sağlar.

Aşağıdaki koşullardan bir veya birkaçı bulunduğu zaman benzetime başvurulmalıdır (Shannon, 1975):

- Problemin tam bir matematik formülasyonu yoktur veya modelin analitik yöntemlerle çözümü henüz bulunamamıştır.
- Analitik yöntemler çözüm için elverişlidir, ancak matematik yöntemler çok karmaşıktır.
- Analitik çözümler vardır ve kullanılabilir, ama problem üzerinde çalışanlarda bu bilgiler yoktur.
- Belirli parametrelerin tahmin edilmesi için benzetime başvurulduğu gözlenmiştir.
- Deneme yapma açısından benzetim tek yol olabilir.
- Sistemlerin veya süreçlerin davranış karakteristiklerini ortaya koymak zaman gerektirebilir.

Vujosevic (1994), yapmış olduğu çalışmada esnek üretim sistemi tasarımında görsel etkileşimli benzetimin önemini belirtmiştir.

Esnek imalat sistemlerinde, sistem arızalarını azaltmak için bakım politikalarını değerlendiren bir çalışma Vineyard ve arkadaşları (2000) tarafından benzetim yöntemiyle yapılmıştır.

4.3.2. Benzetimin faydaları

Benzetimin endüstri uzmanları tarafından belirlenen faydalarından bazıları aşağıdaki gibidir (Banks, 1995):

- Benzetim modelinin geliştirilip kullanılması, sistem üzerinde oynanmasına sağlar. Bu ise, sistemin ve sistem sorunlarının kavranılmasında ve uygun çözümler üretilmesinde son derece önemlidir.
- Sistem analizcilerini daha genel ve daha geniş düşünmeye zorlar.
- Sistem parametrelerinin değişimi karşısında sistem performansının hassasiyetini belirlemek için kullanılabilir.

- Benzetim ile “Eğer böyle olursa ne olur?” türü senaryoları modelleyerek yeni politikalar, üretim teknikleri ve yöntemleri masrafsız ve gerçek sistemle test etmekten doğacak zararlar gerçekleşmeden incelenebilir.
- Benzetimin yardımı ile aksaklıkların sebepleri ortaya çıkarılabilir.
- Benzetim, zamanı daraltarak veya genişleterek olayları hızlandırmaya ya da yavaşlatmaya imkan vermek suretiyle onları etraflıca inceleme fırsatı vermektedir. Eğer istenilirse bütün bir değişim birkaç dakika içerisinde incelenebilir.
- Karar verme konumundaki personel iş başında olmadan hem daha ucuz, hem daha az zarar verici, hem de en etkili şekilde bu yöntemle eğitilebilir.

4.3.3. Benzetimin sakıncaları

- Bir sistemin bilgisayar benzetimini kurmak ve geçerli olduğunu ispatlamanın maliyeti çok yüksektir. Genel olarak her bir sistem için ayrı bir program yazma gereği vardır. Benzetim dilleri bu mahsurları bir dereceye kadar ortadan kaldırmıştır.
- Kurulan bir benzetim programının bilgisayarda çalıştırılması çok zaman alabilir. Bunun ise maliyeti yüksektir.
- Araştırmacılar benzetim tekniğini öğrendikten sonra onu analitik yöntemlerin daha uygun olduğunda kullanma eğilimindedirler (Halaç, 1998).

4.2.4. Benzetim modelleri

Sistemin karakteristiklerine göre modelleme yaklaşımları değişmektedir. Genelde sınıflandırma aşağıdaki gibidir.

- Sürekli/Kesikli modeller
- Statik/Dinamik modeller
- Stokastik/Deterministik modeller

Sürekli modeller: Davranışları zamanla birlikte devamlı değişim gösteren sistemlerle ilgilenmektedir.

Kesikli modeller: Sistemlerin davranışlarındaki değişimleri sadece verilmiş olan bir anda izleyen modellerdir. Örneğin, kuyruk modelleri.

Statik modeller: Sistemin belirli bir zamandaki gösterimidir. Proses zaman içersisinde yayılmaz (Monte Carlo benzetimi gibi).

Dinamik modeller: Zaman içinde yayılan sistemin bir gösterimidir. Buna örnek olarak imalat sistemi benzetimi verilebilir.

3eterministik modeller: Benzetim modeli sistem parametresi olarak hiçbir rasgele değişken içermez. Bir girdi kümesi verildiğinde bir dizi sistem cevabı üretecektir.

Stokastik modeller: Sistem parametrelerinin bir ya da daha fazlası tahmin edilemez, belli bir olasılık dağılımından elde edilebilir. Bir girdi kümesi verildiğinde, çıktuların kendisi rastgele değişkenler olduğundan gerçek sistem cevaplarının bir tahminini verecektir (Taha, 2000 ve Emshoff ve diğerleri, 1970).

4.3.5. Benzetim elemanları

Tüm benzetim modellerinin aşağıdaki elemanlardan oluştuğunu söylemek mümkündür (Erkut, 1992):

- Bileşenler: Bir araya getirildiklerinde sistemi oluşturan parçalardır. Bunlar alt sistem olarak da adlandırılırlar. Bağımsız olarak belirlenirler ve bunların ortak performansı sistemin çıktısını oluşturur. Örneğin bir kent sisteminde bileşenler; eğitim sistemi, sağlık sistemi, ulaşım sistemi gibi.
- Değişkenler: Sistemin özellikleridir. Bunlar dört karşıt kategori içinde sınıflandırılabilirler:
 1. Bağımsız değişkenler: Değiştirmekte serbest olan değişkenlerdir. Dışsal değişken olabilirler, fakat içsel değişken olamazlar; genellikle denetlenebilir değişkenlerdir. Girdi değişkeni olabilirler.

Bağımlı değişkenler: Diğer değişkenlerin etkileri sonucunda değerleri belirlenen değişkenlerdir. Denetlenemez değişken olabilirler; içsel ve çıktı değişken olabilirler.

2. Denetlenebilir değişkenler: Değerleri inceleyici tarafından belirlenen değişkenlerdir. Genellikle bağımsız değişkenlerdir. İçsel ve dışsal olabilirler. Girdi değişkeni olabilirler ancak genelde çıktı değişkeni değildirler.

Denetlenemez değişkenler: Değerleri, sistemin çevresi tarafından veya sistemdeki bileşenlerin etkileşimi sonucunda belirlenen değişkenlerdir. Bağımlı değişken olabilirler. Girdi ve çıktı değişkeni olabilirler.

3. İçsel değişkenler: Değerleri sistem bileşenlerinin performansı tarafından belirlenen değişkenlerdir. Bunlar denetlenemeyen değişkenlerdir. Bağımlı olabilirler ancak genelde bağımsız değişkenlerdir. Bunlar “durum değişkeni” olarak da adlandırılırlar.

Dışsal değişkenler: Değerleri, inceleyici tarafından veya sistem dışındaki etkilerle belirlenen değişkenlerdir.

4. Girdi değişkenleri: Değerleri, sistemin bileşenlerinin performansı veya etkileşimi tarafından belirlenmeyen değişkenlerdir. Dışsal değişken olabilirler, ancak içsel olamazlar. Genellikle bağımsız değişkenlerdir. Denetlenebilir veya denetlenemez olabilir.

Çıktı değişkenleri: Tüm değerleri, sistemin bileşenlerinin performansı veya etkileşimi tarafından belirlenen değişkenlerdir. Genellikle denetlenemeyen değişkenlerdir. İçsel olabilirler; genellikle bağımlı değişkenlerdir.

- Parametreler: Sistem analizcinin keyfi değerler verebildiği miktarlardır. Bunlar analiz boyunca değiştirilemezler, sabit kalırlar.
- İlişkiler: Sistemin bileşenleri, değişkenleri ve parametreleri arasındaki bağlantılardır. Bunlar sistemin durumundaki değişimleri denetlerler. İlişkiler yapısal, işlevsel, sırasal, mekansal, zamansal, neden-sonuç, enerjinin korunumu, mantıksal, matematiksel olabilir.
- Varsayımlar: Modeli gerçek durumdan soyutlayan kabullerdir.
- Kısıtlar: Değişkenlerin değerleri veya kaynakların nasıl tahsis edileceği

üzerindeki sınırlandırmalardır. Bu kısıtlar, tasarımcı tarafından konabilir veya sistemin doğasında vardır.

- Ölçütler: Ölçüt fonksiyonu, sistemin hedeflerinin veya amaçlarının ve bunların nasıl değerlendirileceğinin bir durumudur. Modelin tasarımı ve işletilmesi üzerinde büyük etkisi vardır.

4.3.6. Benzetimin üretim sistemlerinde uygulama alanları

Benzetimin üretim sistemlerinin tasarımı ve analizinde kullanım alanlarından bazıları aşağıdaki gibi sıralanabilir (Simon, 1983):

- Üretim seviyesinin belirlenmesi: Mevcut üretim sistemi bilgisayar üzerinde kurulup, değişik senaryo ve planların etkilerini inceleyip, üretimin ve verimin en yüksek hangi çalışma ve planlama koşullarında oluşacağı bulunabilir.
- Darboğaz belirleme: Fabrika içinde üretimi frenleyen noktaları, yani darboğazları bulup, onları ortadan kaldırmayı sağlayacak senaryoları deneme fırsatı verir.
- Üretim planlaması: Esnek imalat, tam zamanında üretim, minimum stok maliyetleri, yüksek verimlilik gibi kavramlara ulaşacak üretim programlarını yapmak, planların hangi noktalarda problem çıkaracağını bilmek, sistem içindeki belirsizlikleri analiz etmek benzetim yöntemleri ile daha rahat yapılabilir.
- Malzeme taşıma sistemlerinin tasarımı: Malzeme taşıma sistemlerini en aza indirmek, optimum makina yerleşimleri ve problemi kaynağında çözmek benzetimle yapılabilir.
- Yeni sistemlerin tasarımı: Makina parklarını oluşturmak, tasarım aşamasında alternatif projeleri değerlendirmek, sistemin gelecekte alabileceği şekillere uygun olarak bugünden tasarımı yapmak benzetimle mümkündür.
- Kapasite planlaması: Ne yapamayacağını bilme ve bu kısıtlamalara uygun olacak şekilde kapasite çalışmaları yapma imkanı verir.
- İşgücü planlaması

- Montaj hattı dengeleme
- Personel eğitimi

4.3.7. Benzetimin aşamaları

Benzetim sürecinin adımlarını aşağıdaki gibi ele alabiliriz (Erkut, 1992):

- Problemin tanımlanması
 1. Çalışmanın amacı: Yürütülen çalışmanın amacı açık ve eksiksiz olarak belirlenmelidir.
 2. Sistem tanımı: İncelenen sistemin kapsamı dikkatlice çizilmelidir.
 3. Varsayımların belirlenmesi: Sistemin performansını ölçecek kriterlerin belirlenmesi gerekir.
- Benzetim deneylerinin tasarımı
 1. Model kurma: Kağıt üzerinde basit olarak tasarlanarak çıkabilecek aksaklıklar önceden görülebilir.
 2. Benzetim deneyi için veri derleme: Sistem bileşenleri, değişkenleri ve parametreleri hakkında veri toplanmalıdır. Örnekleme: Toplanan verilerin % 90-95 doğru olması gerekir. Ham veriler toplandıktan sonra çeşitli dağılımlara uydurulmalıdır.
- Bilgisayar modelinin kurulması
 1. Kullanılacak benzetim paketini belirleme: Hali hazırda mevcut olan benzetim yazılımları dört ana grup altında toplanabilir (Bayraktar ve diğerleri, 1996):
 - Genel amaçlı benzetim yazılımları
 - Benzetim destekli çizelgeleme yazılımları
 - Robotik benzetim yazılımları
 - İşyeri düzenleme (yerleşim) benzetim yazılımları
 2. Yazılım üzerinde model bileşenlerinin, parametrelerin, olasılık dağılımlarının ve karar kurallarının tanımlanması ve derlenmesi aşaması.
 3. Doğrulama: Modelin doğruluğunun test edilmesidir. Simülasyonlarda, doğrulama amacıyla benzetim saatine göre olayları listelemek ve

lojiklerin içindeki komutları kontrol etmek için kullanılan komutlardan yararlanır.

4. Değerlendirme: Bir bilgisayar modeli eğer modellenen gerçek dünya sisteminin ürettiği sonuçlara yakın sonuçlar ürettiyse, bu bilgisayar modeli geçerlidir denir. Geçersiz olduğu belirlenirse, bunun sebepleri detay seviyesinin belirlenmesinde, kapsamın belirlenmesinde veya veri toplama aşamasında hata olmasındandır.
 5. Modelin ısınma süresinin belirlenmesi: Literatürde, benzetim modelinin denge noktasını bulan ve denge noktasından önceki değerleri atan birçok algoritma tanıtılmaktadır. Conway algoritması, Gafarian-Ancker-Morisaku algoritması, Fishman algoritması ve Gordon algoritması bunlardan bazılarıdır.
 6. Modelin çalışma süresinin belirlenmesi: Bu amaçla aynı şartlar altında sistem ve modelin çalışması karşılaştırılabilir. Çalışma süresi nadir olarak ortaya çıkan olayların meydana gelmesine imkan verecek uzunlukta olmalıdır.
- Benzetim sonuçlarının analizi
1. Benzetim çıktısının gözlem değerlerinin birbirinden bağımsız olması durumunda, denge durumu benzetim hesaplarının doğrudan doğruya yapılması mümkündür. Eğer gözlem değerleri birbirine bağımlı ise, istatistik analiz yöntemlerinin gerektirdiği bağımsız gözlem değerlerine dönüştürmeleri gerekir. Benzetim çıktısındaki gözlem değerleri genellikle birbirine bağımlıdır. Benzetim çıktısındaki gözlem değerlerini bağımsız gözlem değerlerine dönüştüren istatistik yöntemler iki ana grupta toplanabilir (Demirci, 1995):
 - Klasik istatistik yöntemler
 1. Yineleme yöntemi
 2. Parti ortalaması yöntemi
 3. Yenileyici yöntem
 - İleri istatistik yöntemler
 1. Spektral analiz yöntemi
 2. Otoresiv analiz yöntemi
 3. Otokorelasyon analiz yöntemi

4. Zaman serileri yöntemi
2. Sonuçların yorumlanması: İstatistik yöntemlerle elde edilen sonuçların metin, çizelge ve grafik şeklinde kolay anlaşılabilir raporlara dönüştürülmesi, kullanıcıların daha isabetli kararlar vermesi sağlayacaktır.

4.4. Benzetim Yazılımı

Bu çalışmada C++ dili ile bir olay artımlı benzetim yazılımı geliştirilmiştir. Olay artımlı benzetim yapıldığından benzetim sırasında herhangi bir t (benzetim saati) anında kontrol edilmesi gereken, atölyeye yeni bir geliş veya herhangi bir operasyonun bitişi olmak üzere iki olay olma olasılığı vardır. Bu olaylardan hangisi daha erken olacaksa ona öncelik verilir. t benzetim saati o olayın olacağı zamana ilerletilir ve gerekli güncellemeler yapılır. Bu işlemler benzetim sona erinceye kadar tekrarlı bir şekilde yapılır.

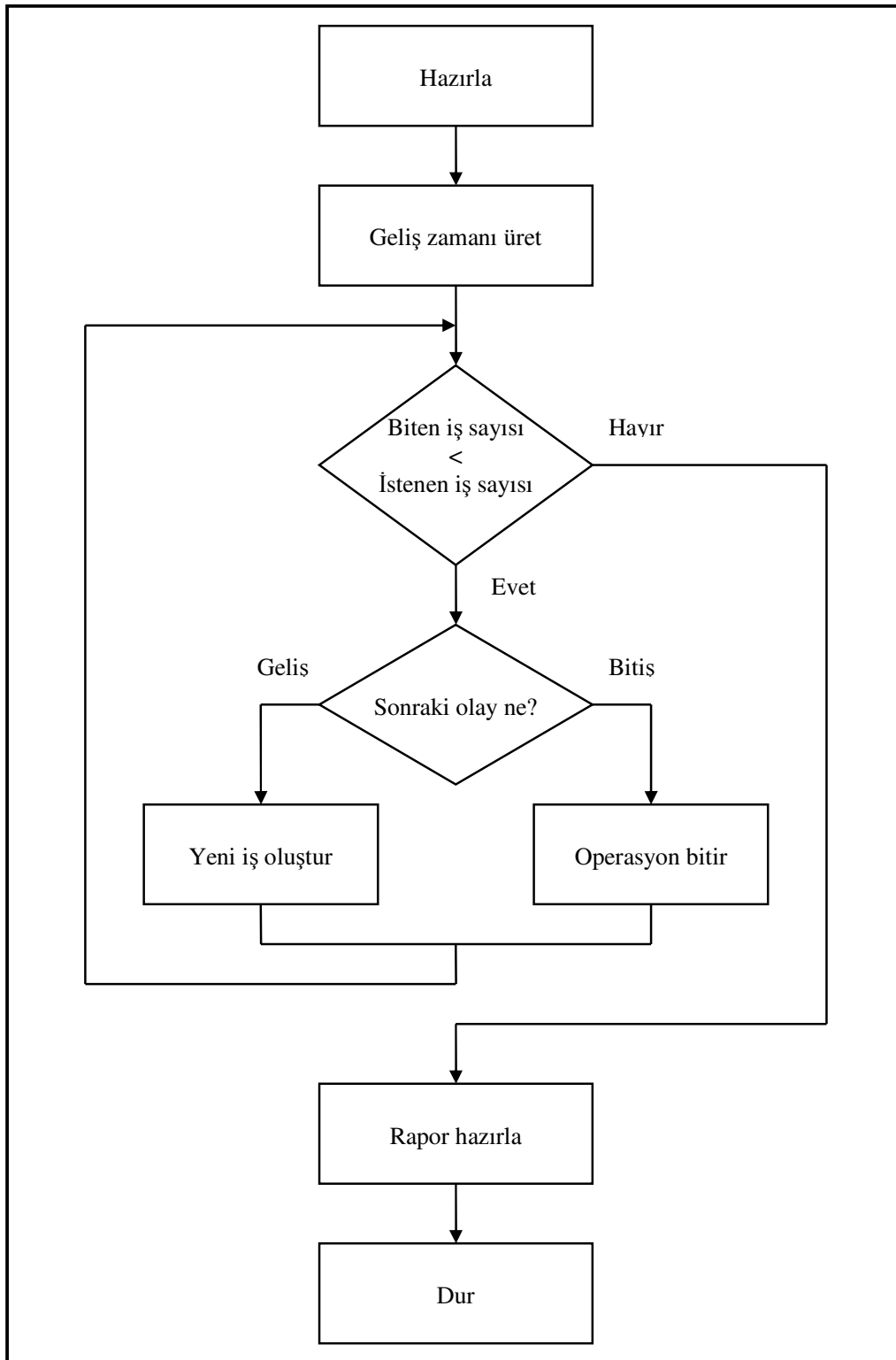
4.4.1. Benzetim yazılımı ana programı

Söz konusu benzetim yazılımına ait program akış şeması Şekil 4.2'de gösterilmektedir. Bu çalışmada kullanılan olay artımlı benzetim tekniğine ait ana program algoritma yapısı da şu şekildedir:

```

main()
{
    Hazırla()
    GelişZamanıÜret()
    SonrakiGelişZamanı = GelişZamanı
    while ( TamamlananİşSayısı < İstenenİşSayısı )
    {
        SonrakiOlayNe()
        switch( SonrakiOlay )
        {
            case VARIŞ: Varış()
            case BİTİŞ: Bitiş()
        }
    }
}

```



Şekil 4.2. Benzetim yazılımı ana program akış şeması

```

RaporHazirla()
}

```

Hazirla(): Bu fonksiyonda benzetimde kullanılan deęişkenlere uygun ilk deęer atamaları yapılmaktadır. Benzetim zamanı $t = 0$ ataması yapılmakta, bitecek iş zamanı büyük bir deęere eşitlenmektedir ($BZ = 999999999$). Bundan amaç ilk anda sadece iş gelişi olacağı, bitecek iş olmayacağından Varış() fonksiyonunun çalıştırılabilmesidir.

GelişZamanıÜret(): Benzetim başladığında ilk anda atölyede herhangi bir iş bulunmamaktadır. Bu nedenle gelecek olan ilk işe ait geliş zamanı deęeri üretilmektedir.

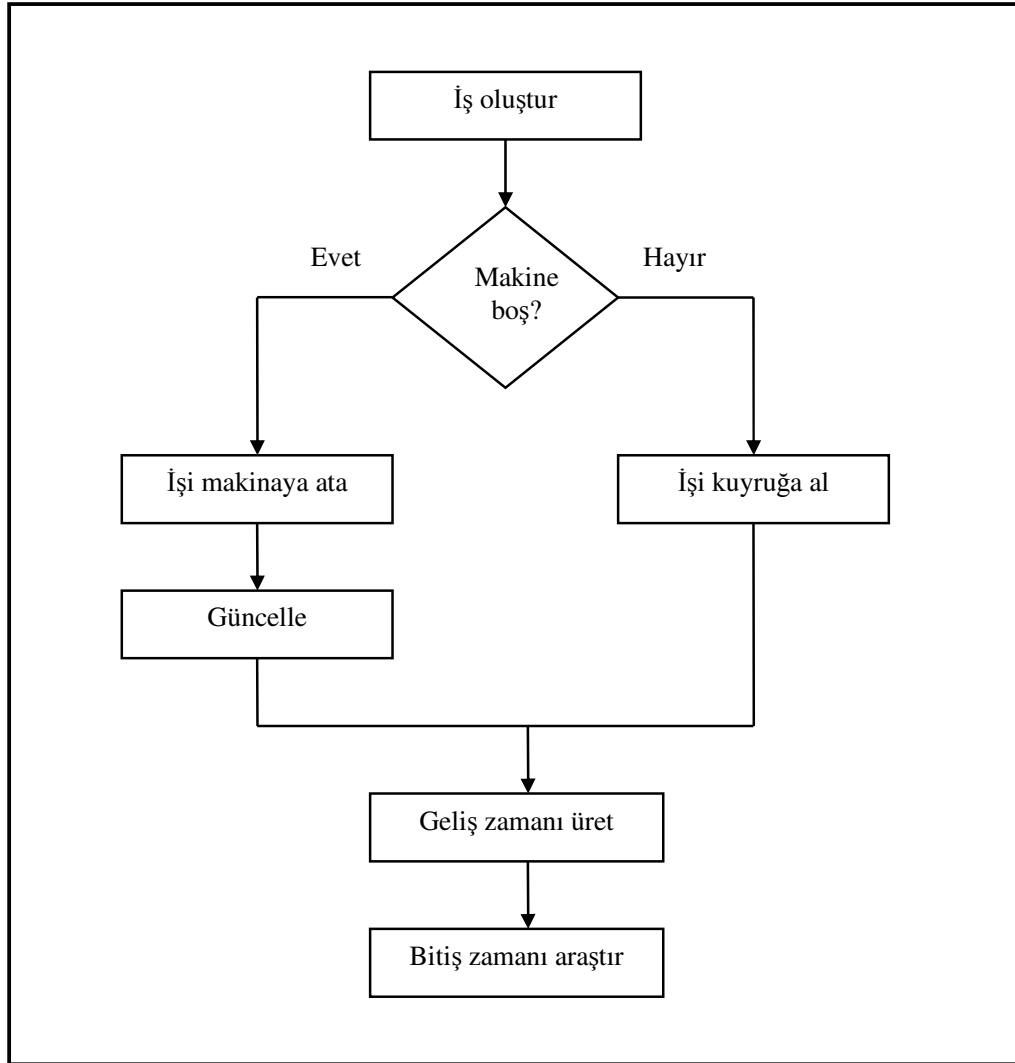
Gelecek olan işin ne zaman biteceęi ve bitecek olan işin ne zaman biteceęi bilgileri belli olduğundan, benzetim döngüsü artık başlatılabilir. Bu amaçla while döngüsü ile benzetimin istenen iş sayısı atölyede işlenene kadar devam edecek olan benzetim başlamaktadır.

SonrakiOlayNe(): Bu fonksiyonda bir sonraki geliş zamanı ile en erken bitecek olan operasyon zamanı karşılaştırması yapılır. Hangisi daha küçükse bir sonraki olay o olacağından switch() yapısı kullanılarak Varış() veya Bitiş() fonksiyonları çalıştırılır.

4.4.2. Varis() fonksiyonu

Bu fonksiyonda yeni bir iş oluşturulur (Şekil 4.3). Atölyedeki bulunan iş sayısı bir arttırılır. Bu işe ait gerekli olan bilgilerin ilgili deęişkenlere atamaları yapılır (Tablo 4.3). İşin işlem göreceęi ilk makina boşsa iş hemen o makinaya atanarak operasyonun biteceęi zaman ve makinanın serbest kalacağı zaman hesaplanır. Bu makinada başka işlerin operasyonu olabileceğinden o işlerin bu makinada operasyona başlama zamanları makinanın serbest kalma zamanına eşitlenmek üzere güncellenmelidir. Bu nedenle Güncelle() fonksiyonu çalıştırılır. Makina boş deęilse, işin ilk operasyonuna başlayabileceęi zaman hesaplanarak iş o makina kuyruğunda beklemeye alınır. İşin o makina için bekleme zamanı, t ile işin o makinada işleme

başlayabileceği zaman arasındaki fark kadar, kuyruk iş yükü, işin operasyon süresi kadar, kuyruktaki iş sayısı ise bir artırılır. Her iki durum için de atölyeye yeni bir iş geldiğinden bir sonraki gelecek olan iş için varış zamanı değeri üretilmelidir. Yeni bir geliş zamanı değeri üretilerek GZ değişkeninde saklanır. Bir sonraki en erken bitiş zamanı için araştırma yapılır.



Şekil 4.3. Varış() fonksiyonu akış şeması

Tablo 4.3. Oluşturulan işe ait bilgiler

Değişken	Açıklama
No	Gelen işlere 1' den başlayarak birer artan bir no verilir
Tip	Rota (İş tüpü) belirlemek için rasgele 1-5 arası bir değer
r	İşin atölyeye geldiği zaman (t) r'ye eşitlenir
N	İşin operasyon sayısı (9)
tN	Tamamlanan operasyon sayısı (0)
kN	Geriye kalan operasyon sayısı (9)
M[1..9]	Rotaya bağlı olarak işlem göreceği makina numaraları atanır
P[1..9]	Rotadaki makinalardaki işlem süreleri belirlenir
P	İşlem süreleri toplanarak atanır
tP	Tamamlanan işlem süresi (0)
kP	Geriye kalan işlem süreleri toplamı (P)
C	Tamamlanma zamanı (0)
F	Akış zamanı (0)
L	Gecikme değeri (0)
T	Pozitif gecikme değeri (0)
E	Negatif gecikme (erken bitme) değeri (0)
ets	Operasyon başlayabilme zamanı (t)
etf	Operasyon bitebilme zamanı (t+P[1])
Pt	Sıradaki işlem zamanı (P[1])
Mt	Sıradaki işlem göreceği makina (M[1])
W[1..9]	İşin makinalardaki bekleme zamanı (0)

Varış() fonksiyonu algoritma yapısı şu şekildedir:

```

Varış()
{
    GelenİşSayısı++
    İş ve iş bilgileri oluştur
    Eğer (İş.Mt == BOŞ) ise
    {
        M[İş.Mt].Durum = MEŞGUL
        İş.etf = t+ İş.Pt
        M.[İş.Mt].SerbestZaman = İş.etf
        Güncelle()
    }
}

```

```

Değilse
{
    İş.ets = M[İş.Mt].SerbestZaman
    İş.etf = 999999999
    İş.W[Mt] += (İş.ets - t)
    M.[İş.Mt].Kuyruk++
    M.[İş.Mt].KuyrukİşYükü += İş.Pt
}
GelisZamaniÜret()
BitisZamaniArastir()
}

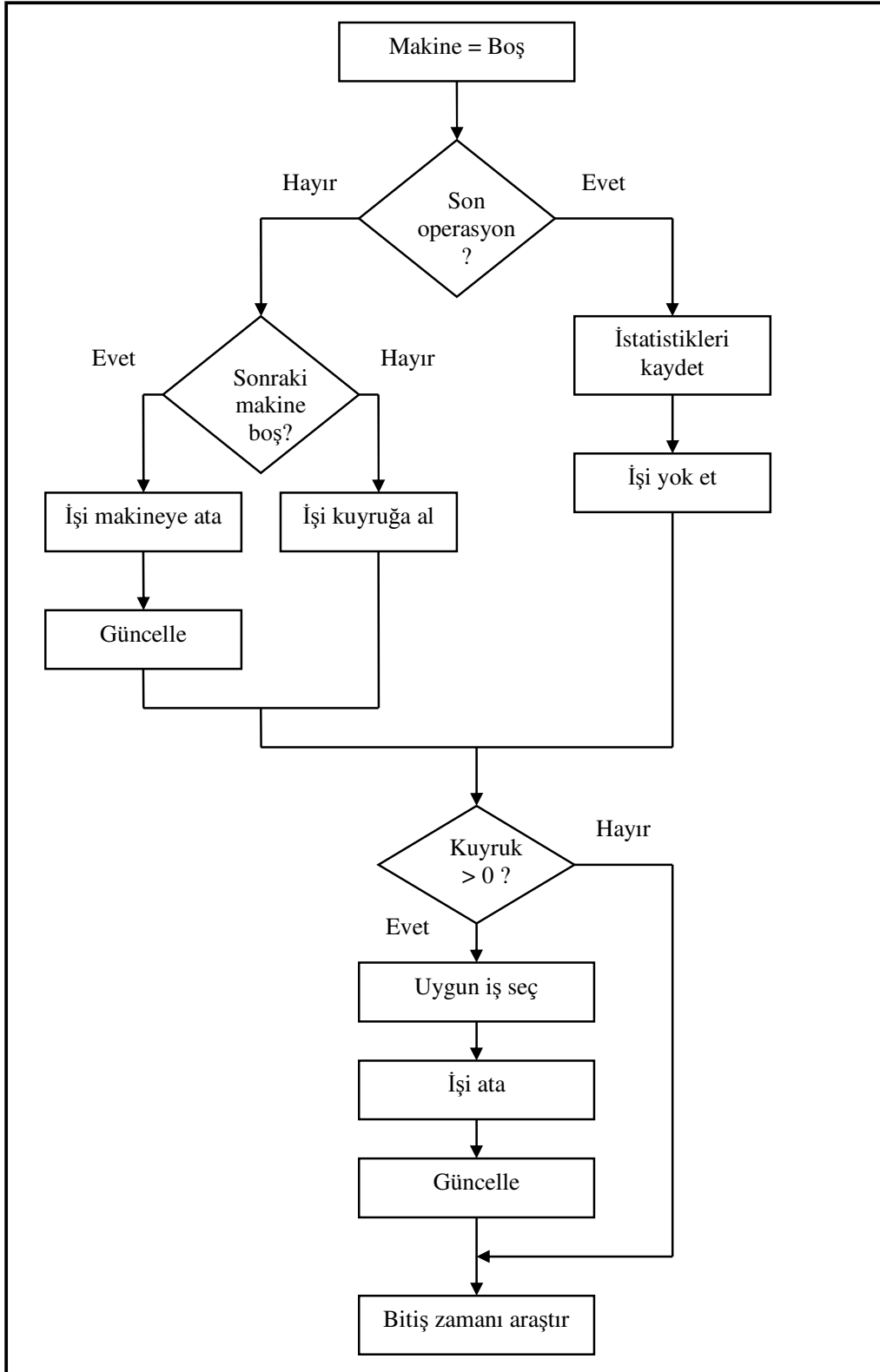
```

4.4.3. Bitiş() fonksiyonu

Bu fonksiyon (Şekil 4.4) herhangi bir işin herhangi bir makinadaki bir operasyonu bittiği zaman çalıştırılır. Bu durumda yapılan ilk şey operasyonun bittiği makinanın “boş” duruma getirilmesidir. Aynı zamanda makina serbest zamanı da t’ye eşitlenir. Arkasından biten iş için tamamlanan operasyon sayısı bir arttırılır, geriye kalan operasyon sayısı bir azaltılır. Tamamlanan işlem zamanı operasyon süresi kadar arttırılırken geriye kalan işlem zamanı tam tersine operasyon süresi kadar azaltılır.

Operasyonu biten iş için biten bu operasyon işin son operasyonu mu değil mi kontrolü yapılır. Değilse, bir sonraki gideceği makinanın boş olup olmadığına bakılır. Makina boşsa makinaya işin ataması yapılır, değilse o makinanın kuyruğuna alınarak makina boşalana kadar bekletilir. Burada yapılan işlemler Varış() fonksiyonundaki iş geldiğinde yapılanlar ile aynı şeylerdir.

Eğer biten bu operasyon işin son operasyonuysa, artık bu iş atölyeden çıkacak anlamına gelir. Benzetim açısından da sistemden atılacağı anlamına gelir. Fakat bundan önce yapılması gereken işlemler vardır. BitenİşSayısı değeri bir arttırılır. Biten iş için kaydedilmek üzere C, F, L, T, E, MS ve SK değerleri hesaplanır. Daha sonra iş sistemden atılır (yok edilir).



Şekil 4.4. Bitiş() fonksiyonu akış şeması

İşin ister son operasyonu bitmiş olsun veya daha önceki bir operasyonu, o makinada işi bittiğinden dolayı makina boşa çıkmıştır ve yeni bir iş için operasyona hazır demektir. Bu nedenle makina kuyruğunda iş olup olmadığı bakılır. Eğer tek bir iş varsa o iş makinaya operasyonu başlamak üzere atanır. Birden fazla iş makina kuyruğunda var ise bu durumda kullanılan öncelik kuralına göre (EKİS veya EETT) makinaya atanacak olan iş seçilir. İş makinaya atanırken daha önce iş atamasında anlatılan işlemler yerine getirilir.

Bitiş() fonksiyonu algoritma yapısı şöyledir:

```

Bitis()
{
    M[İş.Mt].Durum = BOŞ
    M[İş.Mt].SerbestZaman = t
    İş.tN++
    İş.kN--
    İş.kP = İş.P - İş.Pt
    İş.tP += İş.Pt
    İş.Mt = İş.M[tN+1]
    İş.Pt = İş.P[tN+1]
    Eğer (İş.tN < İş.N) ise
    {
        Eğer (M[İş.Mt] = BOŞ ) ise
        {
            M[İş.Mt] = MEŞGUL
            İş.etf = t + İş.Pt
            M[İş.Mt].SerbestZaman = İş.etf
            Guncelle()
        }
    }
    Değilse
    {
        İş.ets = M[İş.Mt].SerbestZaman
        İş.etf = 999999999
        İş.W[Mt] += (İş.ets - t)
        M.[İş.Mt].Kuyruk++
        M.[İş.Mt].KuyrukİşYükü += İş.Pt
    }
}

```

```

Değilse
{
    BitenİŞSayısı++
    İş.C, F, L, T, E, MS, SK hesapla
    İşYokEt()
}
Eğer (M[İş.Mt].Kuyruk > 0 ) ise
{
    Eğer (M[İş.Mt].Kuyruk > 1 ) ise UygunİşSeç()
    M[İş.Mt].Durum = MEŞGUL
    M[İş.Mt].JIQ--
    M[İş.Mt].WIQ -= İş.Pt
    İş.etf = t+ İş.Pt
    M.[İş.Mt].SerbestZaman = İş.etf
    Güncelle()
}
BitisZamaniArastir()
}

```

4.5. Yapay Sinir Ağı Yazılımı

Bu çalışmada kullanılan yapay sinir ağı yazılımı benzetim yazılımında olduğu gibi C++ dili ile yazılmıştır. Şekil 4.5’de akış şeması verilen yazılıma ait ana program algoritma yapısı şu şekildedir:

```

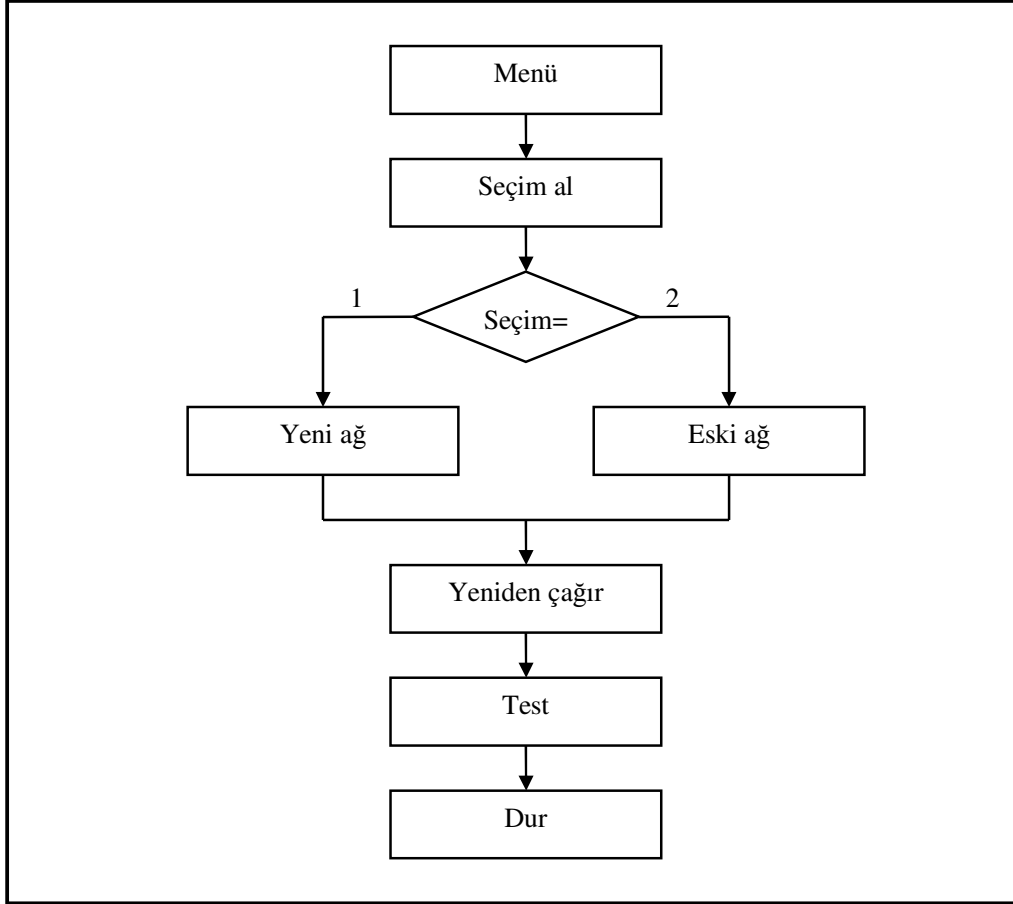
Main()
{
    do
    {
        SecimAl()
        Switch( Seçim )
            1:   YeniAğ()
                YenidenÇağır()
                Test()
            2:   EskiAğ()
                YenidenÇağır()
                Test()
    }
}

```

```

} while ( Seçim <> 3 )
}

```



Şekil 4.5. Yapay sinir ağı yazılımı ana program akış şeması

Program algoritmasından da anlaşılacağı üzere programda yeni ağ eğitimi ve eski ağın eğitime devam etmek üzere iki çeşit öğrenme fonksiyonu bulunmaktadır. YeniAğ() fonksiyonunda ağ parametre bilgileri verilmekte ve başlangıç ağırlık değerleri rasgele olarak alınmaktadır. EskiAğ() fonksiyonunda ise, ağ daha önce belli sayıda epok sayısı kadar eğitilmiştir. Yani daha önceden ağ yapısı hakkında bilgi vardır. Girdi, gizli ve çıktı katmanı proses elemanı sayısı, öğrenme ve momentum katsayıları, daha önceki öğrenme epoklarından dolayı belirli değerlere sahip olan ağırlık değerleri belirlidir. Öğrenme bu yapı üzerinden devam edeceğinden dolayı bilinen bu değerlerin ilgili dosyalardan okunarak öğrenmeye devam edilmesi gerekmektedir.

4.5.1. YeniAğ() fonksiyonu

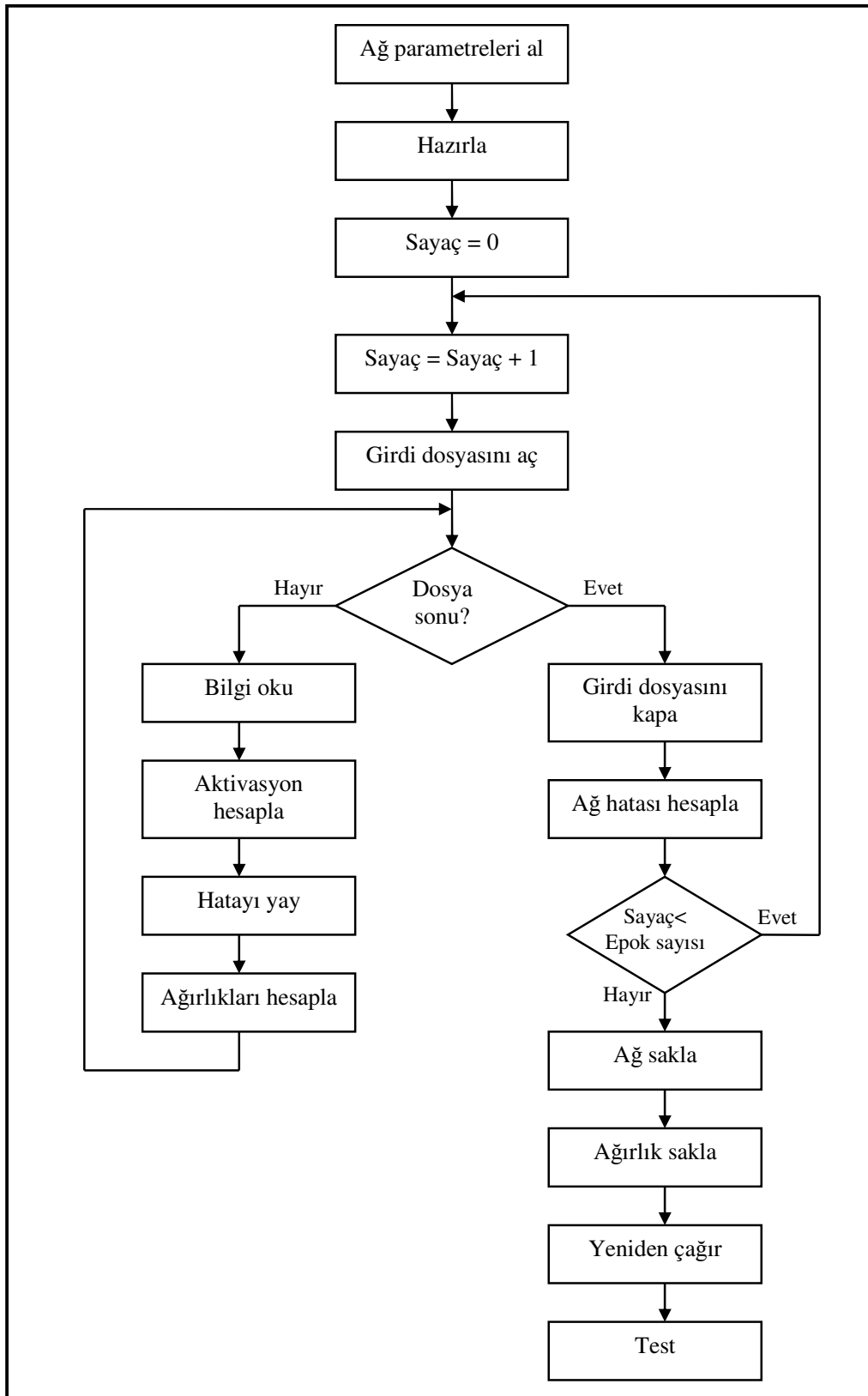
Şekil 4.6'da akış şeması gösterilen YeniAğ() fonksiyonunun algoritma yapısı şu şekildedir:

```

YeniAğ()
{
    ParametreAl()
    Hazırla()
    For( Sayaç=1; Sayaç<=EpokSayısı; Sayaç++)
    {
        GirdiDosyasınıAç()
        While ( DosyaSonu değil )
        {
            BilgiOku()
            AktivasyonHesapla()
            HatayıYay()
            AğırlıkHesapla()
        }

        GirdiDosyasınıKapa()
        AğHatasıHesapla()
    }
    AğSakla()
    AğırlıkSakla()
    YenidenÇağır()
    Test()
}

```



Şekil 4.6. YeniAğ() ve EskiAğ() fonksiyonları akış şeması

Bu fonksiyonda öncelikle yapay sinir ağının girdi, gizli ve çıktı katmanı proses eleman sayıları sorulur. Arkasından öğrenme ve momentum katsayıları istenir. Eğitim işleminin kaç epok sayısı kadar olacağı sorulur. Ayrıca ağın öğrenme işlemini gerçekleştireceği girdi ve çıktı örneklerinden oluşan normalize edilmiş değerleri okuyacağı dosya adı sorulur. Bu işlemlerin arkasından değişkenlere ilk değer atamaları için Hazırla() fonksiyonu çağrılır. Hazırla() fonksiyonunda ağırlıklara başlangıç değerleri -0.1 ile 0.1 arasında rassal olarak belirlenir. Ağın katmanlardaki proses elemanı sayılarına göre dinamik değişken tanımlamaları yapılır.

Daha sonra bir döngü kurularak istenen epok sayısı kadar öğrenme işleminin tekrar etmesi sağlanır. Her bir epok başlangıcında öğrenme dosyası açılır (GirdiDosyasıAç()). Bu dosyadan dosya sonu olmadığı sürece bir girdi seti ve bir çıktı değeri okunur (BilgiOku()). Girdi değerleri kullanılarak gizli katman ve çıktı katmanı için aktivasyon fonksiyonu yarımıyla bu katmanlardaki proses elemanlarının aktivasyon değerleri hesaplanır (AktivasyonHesapla()). Çıktı katmanının aktivasyon değeri yapay sinir ağının çıktı değeridir. Böylece ağın o örnek seti girdilerine karşılık verdiği cevap alınmış olur. Bu cevap ile gerçek çıktı değeri arasındaki fark hesaplanır ve bu hata geriye doğru gizli-çıktı katmanları ve girdi-gizli katman arasındaki ağırlık değerlerine yansıtılır (HatayıYay()). Ağırlık değerlerine yansıtılan hataya göre ağırlık değerleri değişimleri hesaplanır ve bu değişim miktarları kadar ağırlık değerleri arttırılır (AğırlıkHesapla()). Bu işlem dosyadaki bütün girdi-çıktı örnekleri için yapıldığında bir epok tamamlanmış olur ve dosya kapatılır (GirdiDosyasınıKapa()). Ağın bir epok sonundaki ortalama hatası hesaplanır ve kaydedilir (AğHatasıHesapla()). Daha sonra bir sonraki epoka geçilir ve aynı işlemler yapılır. İstenen epok sayısı tamamlanıncaya kadar aynı adımlar tekrarlanır. İstenen epok sayısı kadar öğrenme işlemi yapıldıktan sonra ağ yapısı hakkındaki bilgi ilgili dosyaya kaydedilir (AğSakla()). Hesaplanmış olan ağırlık değerleri, öğrenme işlemine tekrar edilebilir veya edilmeyecekse de ağın öğrenmiş olması bilgisi ağırlık değerlerinde saklandığından eğitilmiş yapay sinir ağı karar vermede kullanılırken bu değerler gerekli olduğundan dolayı ağırlık dosyasına kaydedilirler (AğırlıkSakla()). Arkasından öğrenme için kullanılan öğrenme setindeki girdi-çıktı değerleri eğitilmiş olan ağa sunulur ve cevap vermesi istenir (YenidenÇağır()).

Öğrenme başarı yüzdesi doğru cevap verdiği örnek sayısının öğrenme setindeki toplam örnek sayısına bölünüp 100 ile çarpılmasıyla elde edilir:

$$\text{Başarı (\%)} = \text{Doğru cevap sayısı} / \text{Örnek sayısı} \times 100 \quad (4.3)$$

Daha sonra ağın daha önce hiç görmemiş olduğu test setinde bulundan girdi-çıkıtı örnekleri ağa sunularak cevap vermesi istenir (Test()). Denklem 4.3 ile test setindeki örneklere verilen cevap başarı yüzdesi hesaplanır.

4.5.2. EskiAğ() fonksiyonu

Şekil 4.6'da akış şeması gösterilen EskiAğ() fonksiyonunun algoritma yapısı şu şekildedir:

```

YeniAğ()
{
    ParametreOku()
    Hazırla()
    For( Sayaç=ÖncekiEpokSayısı;
        Sayaç<=(EpokSayısı+ÖncekiEpokSayısı); Sayaç++)
    {
        GirdiDosyasınıAç()
        While ( DosyaSonu değil )
        {
            BilgiOku()
            AktivasyonHesapla()
            HatayıYay()
            AğırlıkHesapla()
        }
        GirdiDosyasınıKapa()
        AğHatasıHesapla()
    }
    AğSakla()
    AğırlıkSakla()
    YenidenÇağır()
    Test()
}

```


}

Bu fonksiyonda öğrenme işlemi bir önceki öğrenmenin devamı şeklinde olacağından öncelikle yapay sinir ağının girdi, gizli ve çıktı katmanları proses eleman sayıları, öğrenme ve momentum katsayıları ilgili dosyadan okunur (AğOku()). Eğitim işleminin kaç epok sayısı kadar devam edeceği sorulur. YeniAğ() fonksiyonunda olduğu gibi ağın öğrenme işlemini gerçekleştireceği girdi ve çıktı örneklerinden oluşan normalize edilmiş değerleri okuyacağı dosya adı sorulur. Bu işlemlerin arkasından değişkenlere ilk değer atamaları için Hazırla() fonksiyonu çağrılır. Ağın katmanlardaki proses elemanı sayılarına göre dinamik değişken tanımlamaları yapılır. Daha sonra önceki eğitimden belirli değerlere sahip olan ağırlık değerleri ilgili dosyadan okunur (AğırlıkOku()).

Daha sonra bir döngü kurularak istenen epok sayısı kadar öğrenme işleminin tekrar etmesi sağlanır. Her bir epok başlangıcında öğrenme dosyası açılır (GirdiDosyasıAç()). Bu dosyadan dosya sonu olmadığı sürece bir girdi seti ve bir çıktı değeri okunur (BilgiOku()). Girdi değerleri kullanılarak gizli katman ve çıktı katmanları için aktivasyon fonksiyonu yarımıyla bu katmanlardaki proses elemanlarının aktivasyon değerleri hesaplanır (AktivasyonHesapla()). Çıktı katmanının aktivasyon değeri yapay sinir ağının çıktı değeridir. Böylece ağın o örnek seti girdilerine karşılık verdiği cevap alınmış olur. Bu cevap ile gerçek çıktı değeri arasındaki fark hesaplanır ve bu hata geriye doğru gizli-çıkı katmanları ve girdi-gizli katman arasındaki ağırlık değerlerine yansıtılır (HatayıYay()). Ağırlık değerlerine yansıtılan hataya göre ağırlık değerleri değişimleri hesaplanır ve bu değişim miktarları kadar ağırlık değerleri artırılır (AğırlıkHesapla()). Bu işlem dosyadaki bütün girdi-çıkı örnekleri için yapıldığında bir epok tamamlanmış olur ve dosya kapatılır (GirdiDosyasınıKapa()). Ağın bir epok sonundaki ortalama hatası hesaplanır ve kaydedilir (AğHatasıHesapla()). Daha sonra bir sonraki epoka geçilir ve aynı işlemler yapılır. İstenen epok sayısı tamamlanıncaya kadar aynı adımlar tekrarlanır. İstenen epok sayısı kadar öğrenme işlemi yapıldıktan sonra ağ yapısı hakkındaki bilgi ilgili dosyaya kaydedilir (AğSakla()). Hesaplanmış olan ağırlık değerleri, öğrenme işlemine tekrar edilebilir veya edilmeyecekse de ağın öğrenmiş olması bilgisi ağırlık değerlerinde saklandığından eğitilmiş yapay sinir ağı karar vermede

kullanılırken bu değerler gerekli olacağından dolayı ağırlık dosyasına kaydedilirler (AğırlıkSakla()). Arkasından öğrenme için kullanılan öğrenme setindeki girdi-çıkıktı değerleri eğitilmiş olan ağa sunulur ve cevap vermesi istenir (YenidenÇağır()). Öğrenme başarı yüzdesi hesaplanır (Denklem 4.3). Daha sonra ağın daha önce hiç görmemiş olduğu test setinde bulundan girdi-çıkıktı örnekleri ağa sunulularak cevap vermesi istenir (Test()). Denklem 4.3 ile test setindeki örneklere verilen cevap başarı yüzdesi hesaplanır.

4.5.3. YenidenÇağır() fonksiyonu

YenidenÇağır() fonksiyonu algoritma yapısı şu şekildedir:

```

YenidenÇağır()
{
    AğOku()
    AğırlıkOku()
    ÖğrenmeSetiDosyasınıAç()
    while ( Dosya sonu değil )
    {
        BilgiOku()
        AktivasyonHesapla()
        HataHesapla()
        Eğer ( Hata < Tolerans ) DoğruSayısı++
        ÖrnekSayısı++
    }
    ÖğrenmeSetiDosyasınıKapa()
    Basari = DoğruSayısı / ÖrnekSayısı x 100
}

```

Bu fonksiyon çağrıldığında yapay sinir ağı öğrenme setiyle en azından belli bir miktar eğitim işlemi yapmış olmalıdır. Çünkü öğrenme setindeki girdi değerlerine karşılık gelen çıkıktı cevabını verebilmesi için ağırlık değerlerine gereksinim duymaktadır. Öncelikle ağın parametrelerinin kayıtlı olduğu dosya açılarak ağ parametreleri okunmalıdır. Daha sonra bu ağa ait eğitim sırasında belirlenmiş olan ağırlık değerleri okunur. Bu ağırlık değerleri yapay sinir ağının öğrenme becerisinin

olduğu yerlerdir. Bu işlemlerden sonra öğrenme seti girdi dosyası olarak açılır. Dosya sonuna ulaşıncaya kadar dosyadan bir örnek seti girdi değerleri okunur, bu değerler ve ağırlık değerleriyle aktivasyon değerleri hesaplanır – çıktı katmanının aktivasyon değerleri çıktı değerleri olup ağın cevabı niteliğindedir – ve gerçek çıktı değerleriyle aralarındaki farklar hesaplanır. Bu farklar belirlenen tolerans değerinden küçük veya eşitse o çıktıya ağın doğru cevap verdiği kabul edilir. Bu şekilde ağın öğrenme setindeki girdilere verdiği cevapların yüzde olarak doğruluk derecesi (başarısı) hesaplanabilir (Denklem 4.1). Şekil 4.7’de YenidenÇağır() fonksiyonunun akış şeması görülmektedir.

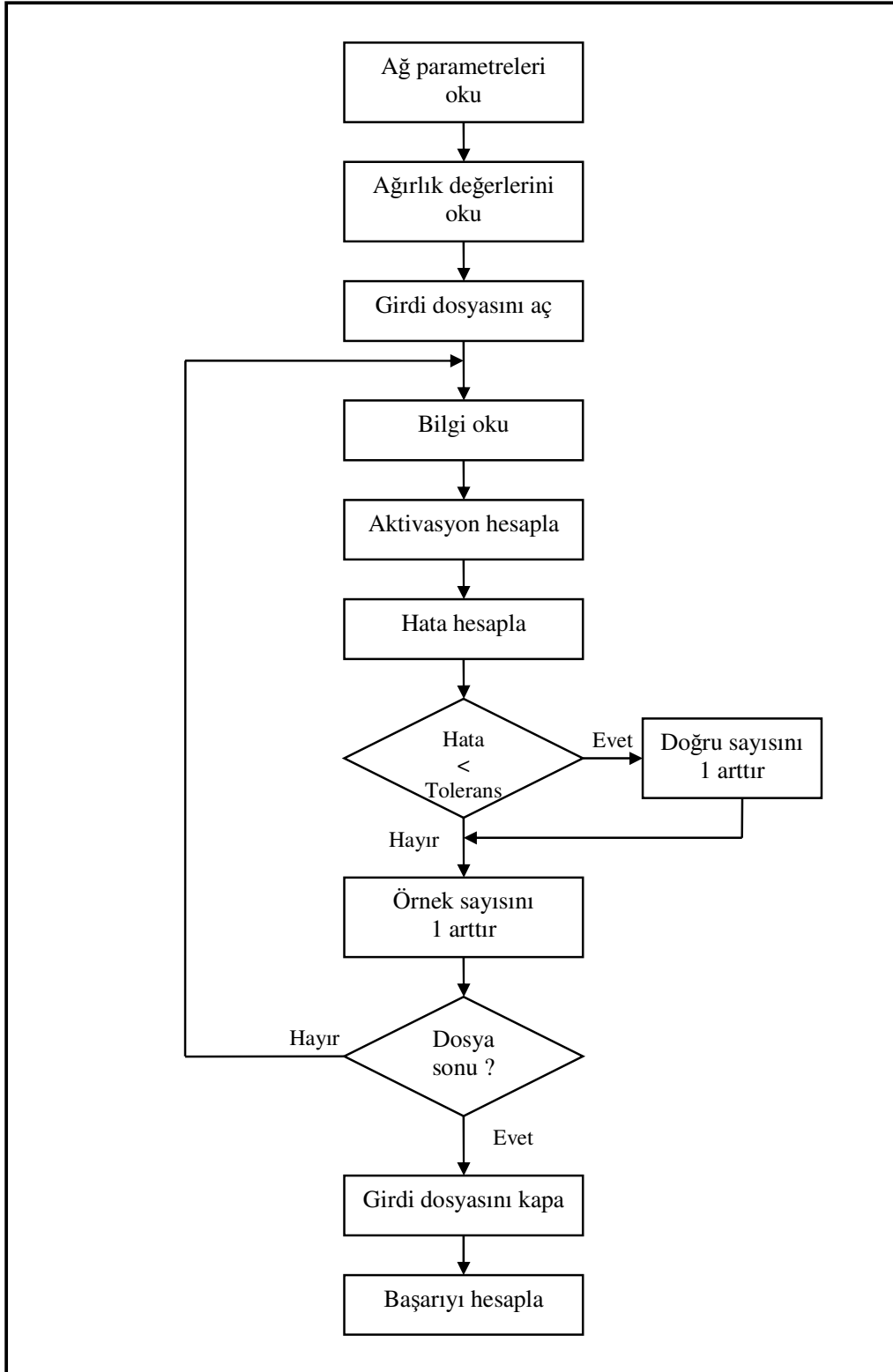
4.5.4. Test() fonksiyonu

Test() fonksiyonu algoritma yapısı şu şekildedir:

```

Test()
{
    AğOku()
    AğırlıkOku()
    TestSetiDosyasınıAç()
    while ( Dosya sonu değil )
    {
        BilgiOku()
        AktivasyonHesapla()
        HataHesapla()
        Eğer ( Hata < Tolerans ) DoğruSayısı++
        ÖrnekSayısı++
    }
    TestSetiDosyasınıKapa()
    Basari = DoğruSayısı / ÖrnekSayısı x 100
}

```



Şekil 4.7. YenidenÇağır() ve Test() fonksiyonları akış şeması

Bu fonksiyon çağrıldığında yapay sinir ağı öğrenme setiyle en azından belli bir miktar eğitim işlemi yapmış olmalıdır. Çünkü öğrenme setindeki girdi değerlerine karşılık gelen çıktı cevabını verebilmesi için ağırlık değerlerine gereksinim duymaktadır. Öncelikle ağın parametrelerinin kayıtlı olduğu dosya açılarak ağ parametreleri okunmalıdır. Daha sonra bu ağa ait eğitim sırasında belirlenmiş olan ağırlık değerleri okunur. Bu ağırlık değerleri yapay sinir ağının öğrenme becerisinin olduğu yerlerdir. Bu işlemlerden sonra öğrenme seti girdi dosyası olarak açılır. Dosya sonuna ulaşıncaya kadar dosyadan bir örnek seti girdi değerleri okunur, bu değerler ve ağırlık değerleriyle aktivasyon değerleri hesaplanır – çıktı katmanının aktivasyon değerleri çıktı değerleri olup ağın cevabı niteliğindedir – ve gerçek çıktı değerleriyle aralarındaki farklar hesaplanır. Bu farklar belirlenen tolerans değerinden küçük veya eşitse o çıktıya ağın doğru cevap verdiği kabul edilir. Bu şekilde ağın öğrenme setindeki girdilere verdiği cevapların yüzde olarak doğruluk derecesi (başarısı) hesaplanabilir (Denklem 4.1). Şekil 4.7’de Test() fonksiyonunun akış şeması görülmektedir.

BÖLÜM 5. DENEYSEL SONUÇLAR

5.1. Giriş

Regresyon modelleri ve yapay sinir ağı modeli ile teslim tarihi belirleyerek dinamik atölye benzetimi yapılmıştır. Bu benzetimler iki farklı öncelik kuralı kullanılarak yapılmıştır (EKİS ve EETT). Bu bölümde yapılan benzetimlerden elde sonuçlar verilmektedir. Her bir teslim tarihi belirleme modeli ve öncelik kuralı kullanılan senaryo için 10 adet benzetim çalışması yapılmıştır. Yapılan bu benzetimlerden elde edilen performans ölçütleri değerlerinin ortalamaları verilmiştir. Yine benzetimlerden elde edilen bu verilerin ortalamaları ve standart sapmaları hesaplanmıştır.

Yapılan benzetim çalışmalarından elde edilen verilerin ait olduğu modellerin istatistik olarak birbirinden farklılıklarının anlamlı olup olmadıkları varyans analizi (Ek K-S) ile %95 güvenle test edilmiştir. Modellerden elde edilen performans değerlerine ait değerlendirmeler yapılırken yapılan bu istatistik analiz göz önüne alınmıştır. Performans değerleri için yapılan istatistik analizler eklerde verilmiştir.

5.2. Regresyon Modelleri ile Elde Edilen Benzetim Sonuçları

Teslim tarihi belirlemede kullanılan regresyon modellerine ait benzetim sonuçları tablolar halinde gösterilmektedir. Tablolarda ilk sütunlarda yapılan benzetim numarası (BN) verilmektedir. İkinci ve diğer sütunlarda hesaplanan performans ölçütleri olarak Ortalama Mutlak Sapma (OMS), Ortalama Sapma Karesi (OSK), Ortalama Pozitif Gecikme (OPG), Pozitif Geciken İş Sayısı (PGİS), Ortalama Negatif Gecikme (ONG) ve Negatif Geciken İş Sayısı (NGİS) değerleri verilmektedir. Her bir sütunun altında da performans ölçütlerine ait aritmetik ortalama değerleri ve standart sapma değerleri verilmektedir.

5.2.1. EKİS önce öncelik kuralı ile regresyon modelleri benzetim sonuçları

Tablo 5.1’de teslim tarihi belirlemede regresyon modellerinden Toplam İş Yüğü (TİY) modeli ve ENKİS önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, PGİS ölçütleri açısından en iyi sonuç 10. , ONG ölçütü açısından 2. ve NGİS ölçütü açısından 3. benzetim çalışmasında alınmıştır.

Tablo 5.2’de teslim tarihi belirlemede regresyon modellerinden Operasyon Sayısı (OS) modeli ve ENKİS önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, PGİS ölçütleri açısından en iyi sonuç 10. , ONG ve NGİS ölçütü açısından 2. benzetim çalışmasında alınmıştır.

Tablo 5.3’de teslim tarihi belirlemede regresyon modellerinden Toplam İş Yüğü + Operasyon Sayısı (TİY+OS) modeli ve ENKİS önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, PGİS ölçütleri açısından en iyi sonuç 8., ONG açısından 9. ve ONGİS ölçütü açısından 5. benzetim çalışmasında alınmıştır.

Tablo 5.1. TİY ve EKİS ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	362	351989	117	1268	246	8732
2	381	482265	140	1378	241	8622
3	394	377541	147	1448	247	8552
4	355	262930	105	1348	250	8652
5	436	777083	193	1329	244	8671
6	378	499501	131	1220	247	8780
7	377	431526	131	1285	247	8715
8	348	274242	98	1189	249	8811
9	418	727611	174	1396	244	8604
10	323	196873	74	1104	249	8896
Ortalama	377.20	438156.10	131.00	1296.50	246.40	8703.50
St. sapma	33.22	191861.31	35.34	104.68	2.76	104.68

Tablo 5.2. OS ve EKİS ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	372	471033	178	2188	193	7812
2	396	627979	204	2259	192	7741
3	403	498186	206	2192	197	7808
4	360	348518	165	2243	195	7757
5	450	982248	256	2192	194	7808
6	389	640109	193	2154	196	7846
7	387	562044	192	2184	194	7816
8	355	370607	158	2129	197	7871
9	430	904806	237	2239	193	7761
10	327	274067	132	2122	195	7878
Ortalama	386.90	567959.70	192.10	2190.20	194.60	7809.80
St. sapma	36.11	231344.35	36.73	46.74	1.71	46.74

Tablo 5.3. TİY+OS ve EKİS ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	281	185315	122	3925	160	6075
2	306	328600	158	4042	148	5958
3	315	252052	167	4102	148	5898
4	300	191683	131	3870	169	6130
5	355	543735	208	4232	147	5768
6	314	360653	155	4175	160	5825
7	311	297870	153	3994	158	6006
8	267	136071	103	3733	163	6267
9	329	519893	185	4048	144	5952
10	299	237836	140	4020	159	5980
Ortalama	307.70	305370.80	152.20	4014.10	155.60	5985.90
St. sapma	24.27	137393.93	30.52	145.75	8.26	145.75

Tablo 5.4’de teslim tarihi belirlemede regresyon modellerinden Kuyruktaki İş Sayısı (KİS) modeli ve ENKİS önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, PGİS ölçütleri açısından en iyi sonuç 4., ONG açısından 9. ve NGİS ölçütü açısından 5. benzetim çalışmasında alınmıştır.

Tablo 5.5’de teslim tarihi belirlemede regresyon modellerinden Kuyruktaki İş Yüğü (KİY) modeli ve ENKİS önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, PGİS ölçütleri açısından en iyi sonuç 2., ONG açısından 5. ve NGİS ölçütü açısından 3. benzetim çalışmasında alınmıştır.

Tablo 5.4. KİS ve EKİS ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	343	319776	122	2836	221	7164
2	332	421954	156	3510	176	6490
3	326	309064	151	3132	175	6868
4	330	188523	49	1397	281	8603
5	424	731155	267	5377	157	4623
6	357	459622	141	3022	216	6978
7	332	369292	133	2929	198	7071
8	349	253198	91	1963	257	8037
9	371	661527	216	4285	155	5715
10	329	232187	90	2036	240	7964
Ortalama	349.30	394629.80	141.60	3048.70	207.60	6951.30
St. sapma	29.96	180089.84	63.08	1162.46	42.96	1162.46

Tablo 5.5. KİY ve EKİS ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	659	556018	63	549	596	9451
2	639	463028	29	380	610	9620
3	667	552805	83	757	584	9243
4	653	502162	50	592	603	9408
5	709	860537	131	720	578	9280
6	675	680665	78	542	597	9458
7	671	616398	76	574	595	9426
8	655	508843	49	484	605	9516
9	694	842372	113	696	581	9304
10	666	645167	82	633	584	9367
Ortalama	668.80	622799.50	75.40	592.70	593.30	9407.30
St. sapma	20.36	137859.16	30.30	114.20	11.02	114.20

5.2.2. EETT önce öncelik kuralı ile regresyon modelleri benzetim sonuçları

Tablo 5.6’de teslim tarihi belirlemede regresyon modellerinden TİY modeli ve EETT önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, PGİS ölçütleri açısından en iyi sonuç 6., ONG açısından 2. ve NGİS ölçütü açısından 3. benzetim çalışmasında alınmıştır.

Tablo 5.7’de teslim tarihi belirlemede regresyon modellerinden OS modeli ve EETT önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, PGİS ölçütleri açısından en iyi sonuç 3., ONG açısından 2. ve NGİS ölçütü açısından 4. benzetim çalışmasında alınmıştır.

Tablo 5.8’de teslim tarihi belirlemede regresyon modellerinden TİY+OS modeli ve EETT önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, PGİS ölçütleri açısından en iyi sonuç 1., ONG açısından 6. ve NGİS ölçütü açısından 8. benzetim çalışmasında alınmıştır.

Tablo 5.6. TİY ve EETT ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	381	355063	111	1186	270	8814
2	400	482524	134	1272	265	8728
3	412	378835	141	1405	271	8595
4	374	268453	99	1255	274	8745
5	455	771815	187	1242	268	8758
6	343	204354	69	1026	274	8974
7	397	500907	125	1122	272	8878
8	396	433301	125	1192	271	8808
9	367	279500	93	1097	274	8903
10	436	724576	168	1299	268	8701
Ortalama	396.10	439932.80	125.20	1209.60	270.70	8790.40
St. sapma	32.85	188066.36	35.06	109.60	3.02	109.60

Tablo 5.7. OS ve EETT ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	477	503677	148	1552	329	8448
2	498	651621	172	1657	326	8343
3	436	322905	103	1419	333	8581
4	507	522843	175	1673	332	8327
5	555	989410	225	1567	330	8433
6	496	668904	163	1510	333	8490
7	492	590382	162	1565	330	8435
8	462	411060	129	1501	334	8499
9	534	917893	206	1616	328	8384
10	464	386083	134	1580	330	8420
Ortalama	492.10	596477.80	161.70	1564.00	330.50	8436.00
St. sapma	35.05	219481.41	36.09	75.69	2.51	75.69

Tablo 5.8. TİY+OS ve EETT ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	391	253461	81	1131	310	8869
2	421	316937	112	1373	309	8627
3	430	415756	124	1310	306	8690
4	506	859756	201	1339	305	8661
5	447	569095	139	1256	309	8744
6	450	551394	148	1398	302	8602
7	446	497468	138	1315	307	8685
8	416	333748	106	1246	310	8754
9	461	437942	153	1474	308	8526
10	487	803427	182	1401	305	8599
Ortalama	445.50	503898.40	138.40	1324.30	307.10	8675.70
St. sapma	33.87	200456.07	35.55	97.05	2.60	97.05

Tablo 5.9’da teslim tarihi belirlemede regresyon modellerinden KİS modeli ve EETT önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG ölçütleri açısından en iyi sonuç 10., PGİS açısından 5, ONG açısından 10. ve NGİS ölçütü açısından 3. benzetim çalışmasında alınmıştır.

Tablo 5.10'da teslim tarihi belirlemede regresyon modellerinden KİY modeli ve EETT önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG ölçütleri açısından en iyi sonuç 8., PGİS açısından 10., ONG açısından 2. ve 8., NGİS ölçütü açısından 2. benzetim çalışmasında alınmıştır.

Tablo 5.9. KİS ve EETT ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	360	404289	147	1747	213	8253
2	395	550968	168	1765	227	8235
3	404	435352	174	1771	230	8229
4	342	297400	137	1769	205	8231
5	468	880364	214	1584	254	8416
6	378	561738	160	1686	218	8314
7	380	493490	159	1700	221	8300
8	331	313588	130	1757	201	8243
9	443	813750	198	1689	245	8311
10	294	222717	108	1762	186	8238
Ortalama	379.50	497365.60	159.50	1723.00	220.00	8277.00
St. sapma	51.81	214869.51	31.48	59.45	20.35	59.45

Tablo 5.10. KİY ve EETT ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	260	387103	169	2472	91	7528
2	285	536803	195	2535	90	7465
3	294	423736	200	2373	94	7627
4	250	280846	157	2410	93	7590
5	339	869742	245	2426	93	7574
6	274	546346	182	2418	92	7582
7	275	475675	183	2459	92	7541
8	214	200994	124	2379	90	7621
9	322	804358	228	2522	93	7478
10	241	293930	149	2370	92	7630
Ortalama	275.40	481953.30	183.20	2436.40	92.00	7563.60
St. sapma	37.26	218235.36	36.23	59.34	1.33	59.34

5.3. Yapay Sinir Ağı Modeli ile Benzetim Sonuçları

Teslim tarihi belirlemede kullanılan yapay sinir ağı modeline ait benzetim sonuçları tablolar halinde gösterilmektedir. Tablolarda ilk sütunlarda yapılan benzetim numarası (BN) verilmektedir. İkinci ve diğer sütunlarda hesaplanan performans ölçütleri olarak Ortalama Mutlak Sapma (OMS), Ortalama Sapma Karesi (OSK), Ortalama Pozitif Gecikme (OPG), Pozitif Geciken İş Sayısı (PGİS), Ortalama Negatif Gecikme (ONG) ve Negatif Geciken İş Sayısı (NGİS) değerleri verilmektedir. Her bir sütunun altında da performans ölçütlerine ait aritmetik ortalama değerleri ve standart sapma değerleri verilmektedir.

5.3.1. EKİS önce öncelik kuralı ile yapay sinir ağı modeli benzetim sonuçları

Tablo 5.11’de teslim tarihi belirlemede yapay sinir ağı (YSA) modeli ve EKİS önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, ölçütleri açısından en iyi sonuç 4., OPG açısından 10., PGİS açısından 5., ONG açısından 4. ve 8., NGİS ölçütü açısından 10. benzetim çalışmasında alınmıştır.

Tablo 5.11. YSA ve EKİS ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	203	259587	77	4008	125	5992
2	215	351766	101	4273	114	5727
3	185	167500	98	4301	87	5699
4	164	137396	63	4089	101	5911
5	276	525751	137	3498	138	6502
6	239	414075	95	4102	144	5898
7	216	293881	86	3864	130	6136
8	193	231892	64	4210	129	5790
9	245	535656	139	4104	106	5896
10	169	202939	55	4315	113	5685
Ortalama	210.50	312044.30	91.50	4076.40	118.70	5923.60
St. sapma	35.34	141604.28	29.08	247.16	17.70	247.16

5.3.2. EETT önce öncelik kuralı ile yapay sinir ağı modeli benzetim sonuçları

Tablo 5.12’de teslim tarihi belirlemede yapay sinir ağı (YSA) modeli ve EETT önce öncelik kuralı kullanılarak yapılan benzetim sonuçları görülmektedir. OMS, OSK, OPG, ONG ve NGİS ölçütleri açısından en iyi sonuç 5., PGİS açısından 10 .benzetim çalışmasında alınmıştır.

Tablo 5.12. YSA ve EETT ile yapılan benzetim sonuçları

BN	OMS	OSK	OPG	PGİS	ONG	NGİS
1	306	381246	152	1972	154	8028
2	339	526469	175	1998	164	8002
3	347	414342	181	1976	166	8024
4	286	278036	143	1977	143	8023
5	241	202242	113	2076	127	7924
6	322	536696	166	1941	156	8059
7	328	472040	165	1899	162	8101
8	276	292562	136	2051	140	7949
9	385	788705	206	1914	179	8086
10	411	845926	219	1800	192	8200
Ortalama	324.10	473826.40	165.60	1960.40	158.30	8039.60
St. sapma	50.58	211185.00	31.88	78.65	18.98	78.65

5.4. Modellerin Karşılaştırılması

Verilen benzetim sonuçları tablolarında, hem regresyon modelleri ve hem de önerilen yapay sinir ağı modeli için EKİS ve EETT önce öncelik kuralları kullanılarak yapılan 10’ar adetlik benzetim sonuçları görülmektedir. Önerilen yapay sinir ağı modeli ve karşılaştırıldığı 5 adet regresyon modeline ait bu benzetim sonuçlarının karşılaştırmasının yapılabilmesi için, 10 benzetimden elde edilen performans ölçütleri değerlerinin aritmetik ortalamaları alınmış ve değerlendirmeler bu ortalama değerler üzerinden yapılmıştır. Regresyon modelleri ve yapay sinir ağı modeli için EKİS ve EETT önce kuralları için elde edilen performans ölçütleri ortalama değerleri Tablo 5.13 ve 5.14’de verilmiştir.

Tablo 5.13. EKİS için modellerin performans ölçütleri ortalama değerleri

Model	OMS	OSK	OPG	PGİS	ONG	NGİS
TİY	377	438156	131	1297	246	8704
OS	387	567960	192	2190	195	7810
TİY+OS	308	305371	152	4014	156	5986
KİS	349	394630	142	3049	208	6951
KİY	669	622800	75	593	593	9407
YSA	211	312044	92	4076	119	5924

Tablo 5.14. EETT için modellerin performans ölçütleri ortalama değerleri

Model	OMS	OSK	OPG	PGİS	ONG	NGİS
TİY	396	439933	125	1210	271	8790
OS	492	596478	162	1564	331	8436
TİY+OS	446	503898	138	1324	307	8676
KİS	380	497366	160	1723	220	8277
KİY	275	481953	183	2436	92	7564
YSA	324	473826	166	1960	158	8040

Tablo 5.13'den hareketle Ek K'da verilen istatistik analizlere dayanarak OMS performans ölçütü göz önüne alındığında en iyi sonucu 211 değeriyle YSA modelinin verdiği görülmektedir. Bu modeli sırasıyla TİY+OS, KİS, TİY ve OS modelleri izlemektedir. Bu sıralama modelleri en iyi performansı verecek şekilde iyiden kötüye doğru sıralamaktadır. Bu performans ölçütü için 6 modelin performansı birbirinden farklıdır.

OSK performans ölçütü açısından Ek L'deki istatistik analizlere dayanarak en iyi sonucu TİY+OS ve YSA modelinin verdiği söylenebilir. İstatistik açıdan bu iki model arasında anlamlı bir fark bulunmamaktadır. Bu modelleri sırasıyla OS, TİY+OS, KİS ve KİY modelleri performans bakımından takip etmektedir. Bu performans ölçütü için 5 grup model birbirinden farklıdır.

Ek M'deki istatistik analizler çerçevesinde OPG performans ölçütü açısından değerlendirildiğinde, en iyi sonucu KİY modeli vermektedir. Ardından ikinci en iyi

sonucu veren model YSA modelidir. Diğerleri başarı sırasına göre bu performans ölçütü açısından TİY, KİS, TİY+OS ve OS modelleridir. Bu defa 6 model bu ölçüt için birbirinden farklı sonuçlar üretmiştir.

ONG performans ölçütü göz önüne alındığına, Ek N'deki istatistik analizlere dayanarak yine her modelin birbirinden farklı sonuçlar ürettiği söylenebilir. Bunlar arasında en başarılı model YSA modelidir. Bu modeli sırasıyla TİY+OS, OS, KİS, TİY ve KİY modelleri takip etmektedir.

Tablo 5.14'de ise EETT öncelik kuralının kullanıldığı modellerin performans değerleri bulunmaktadır. OMS performans ölçütü göz önüne alınarak ve Ek O'daki istatistik analizlerle her bir modelin birbirinden farklı değerler ürettiği söylenebilir. En iyi sonucu KİY modeli üretmiştir. Bu modeli sırasıyla YSA, KİS, TİY, TİY+OS ve OS modelleri izlemektedir.

Yine OSK performans ölçütü için Ek P göz önünde bulunarak birbirinden farklı üç grup model olduğu söylenebilir. Bunlardan birinci grupta olan TİY, YSA ve KİY modelleri en iyi sonuçları vermiştir. İkinci grup olarak YSA, KİY, KİS ve TİY+OS modelleri ve üçüncü olarak OS modeli gelmektedir.

OPG performans ölçütü Ek R ile değerlendirildiğinde en iyi sonucu TİY modeli vermektedir. Sonra TİY+OS modeli ve onu da KİS, OS ve YSA dan oluşan üçüncü grup ve nihayet onları da KİY modeli takip etmektedir.

Benzer şekilde ONG performans ölçütü açısından Ek S'deki istatistik analizlere bakılarak her modelin birbirinden farklı değerler ürettiği ve en iyi sonucu KİY modelinin verdiği görülmektedir. Bu modeli YSA modeli takip etmektedir. Arkasından KİS, TİY, TİY+OS ve OS modelleri gelmektedir.

BÖLÜM 6. SONUÇLAR VE ÖNERİLER

6.1. Çalışma Özeti

Bu çalışmada dinamik atölye çizelgelemede teslim tarihi belirleme yöntemlerine alternatif olarak daha iyi sonuçlar üretebileceği düşünülen bir yapay sinir ağı yaklaşımı sunulmakta ve olurluluğu gösterilmektedir. Yapay sinir ağı modelinin performansını ölçebilmek amacıyla karşılaştırma yapılabilecek beş farklı regresyon modeli seçilmiştir. Bu çalışmayı yapabilmek için öncelikle olay artımlı bir benzetim yazılımı geliştirilmiştir. Bu yazılımın ürettiği verilerle regresyon model denklemleri oluşturulmuş ve ayrıca yapay sinir ağı modelinin teslim tarihi belirlemek üzere eğitimi yapılmıştır. Yapay sinir ağı modeli olarak geriye yayımlı yapay sinir ağı kullanılmış ve bir yapay sinir ağı yazılımı geliştirilmiştir. Regresyon modelleri oluşturulduktan ve yapay sinir ağı eğitildikten sonra benzetim yazılımı tekrar bu defa karşılaştırma amaçlı olarak çalıştırılmıştır. Regresyon modelleri için her bir modeldeki denklem kullanılarak teslim tarihleri belirlenerek benzetim yapılarak performans ölçütleri hesaplanmış ve saklanmıştır. Yine yapay sinir ağı kullanılarak teslim tarihlerinin belirlendiği benzetim yapılmış ve performans ölçütleri kaydedilmiştir. Hem regresyon modelleri oluşturma ve yapay sinir ağı eğitimi için gerekli veriyi sağlayacak olan hem de modellerin karşılaştırmalarını yapmak üzere yapılan benzetim çalışmaları belirli bir ısınma periyodundan sonra gerekli veriler kaydedilmek üzere 10 adet yapılmış, her benzetim çalışmasının sonuçlarından 10 adetten biri alınmak üzere veri setleri düzenlenmiştir. Söz konusu bu çalışmalar en kısa işlem süreli ve en erken teslim tarihli öncelik kuralları için ayrı ayrı yapılmıştır. Modelleri karşılaştırmak amacıyla performans ölçütleri olarak teslim tarihinden ortalama mutlak sapma, teslim tarihinden mutlak sapmalarının kareleri ortalaması, ortalama pozitif gecikme, pozitif geciken iş sayısı, ortalama negatif gecikme ve negatif geciken iş sayıları kullanılmıştır.

6.2. Bulgular

Yapılan çalışmanın sonucunda ortaya çıkan bulgular şu şekilde özetlenebilir:

- Yapay sinir ağı modelinin teslim tarihi belirlemede olurluluğu gösterilmiştir.
- Hem en kısa işlem süreli önce ve hem de en erken teslim tarihli önce öncelik kuralları açısından yapay sinir ağı birkaç performans ölçütü açısından iyi sonuçlar vermiştir.
- Yapay sinir ağının en kısa işlem süreli performans ölçütünde genel olarak daha iyi sonuç verdiği görülmüştür.
- En kısa işlem süreli öncelik kuralı ile birlikte ortalama mutlak sapma, ortalama sapma kare ve ortalama negatif gecikme performans ölçütleri söz konusu olduğunda en iyi sonuçları verdiği gösterilmiştir.
- Yine yapay sinir ağının en erken teslim tarihli önce öncelik kuralı ile birlikte ortalama sapma kare performans ölçütü açısından en iyi sonucu veren modeller arasında olduğu görülmüştür.

6.3. Katkılar

Çalışmanın sağlayabileceği katkılar şöyle özetlenebilir:

- Dinamik atölye çizelgeleme sistemi ve bu sistemin nasıl benzetim modelinin kurulacağı hakkında detaylı bilgi verilmiştir. Model kurulurken bazı varsayımlar yapılmıştır. Burada incelenen atölyede kuyruk kapasiteleri sonsuz, makineler arıza yapmıyor kabul edilmiştir. Fakat bunların benzetim modeline dahil edilmesi çok zor bir durum değildir. Kolaylıkla bu ve benzeri durumlar eklenerek benzetim modeli gerçeğe daha da yakın hale getirilebilir.
- Burada kullanılan benzetim modeli olay artımlıdır. Diğer yöntem ise zaman artımlı olup, sistemdeki en küçük zaman birimi kadar artırımlar yapılarak sistemdeki değişiklikler kontrol edilmektedir. Oysa olay artımlı benzetimde bir sonraki olacak en erken olaya benzetim saati direkt olarak artırıldığından aradaki zaman kayıplarının önüne geçilmekte, benzetim oldukça

hızlanmaktadır. Bu çalışmada olay artımlı benzetim modeli nasıl kurulacağı, nasıl programlanacağı algoritmalar ve akış şemaları yardımıyla açıklanmıştır.

- Yapay sinir ağı teslim tarihi için bir tahmin yöntemi gibi kullanılmıştır. Yapay sinir ağı kullanımının performans ölçütleri dışında avantajlı olduğu durumlar vardır. Bu çalışmada kullanılan regresyon modellerinde bazı modeller tek veri ile bazıları iki veriyle çalışmaktadır. Bunlardan bir veya birkaçı olmadığında sonuçlar oldukça farklı çıkabilmektedir. Oysa yapay sinir ağı modeli girdilerinden bir veya birkaçı eksik olduğunda çok aşırı sapmalara sahip sonuçlar vermemektedir. Bunun nedeni, bir yapay sinir ağının eğitim esnasında verilen girdi ve çıktı aralıkları dışında çalışmamasıdır. Eğitim sırasında verilen çıktı değer aralıkları dışında bir çıktı üretmez. Girdi olarak verilen değerler de her bir girdi için en küçük ve en büyük değerden küçük veya büyük olsa bile, en küçük veya en büyük olarak kabul edileceğinden ağ aşırı sapmalar yapmaz.
- Yine bu çalışmada yapay sinir ağları hakkında açıklayıcı bilgiler verilmiştir. Bir yapay sinir ağının ne olduğu neler yapabileceği, nasıl programlanabileceği açık olarak anlatılmıştır. Ana program ve ana programın kullandığı fonksiyonlar hem algoritma olarak hem de akış şeması olarak açıklayıcı olacak şekilde verilmiştir.
- Yapay sinir ağının teslim tarihi belirlemede ve dolayısıyla genel olarak tahmin amaçlı olarak kullanılabilmesi gösterilmiştir. Bu bakımdan bu ve benzeri konularda çalışacak olanlara yardımcı olacağı düşünülmektedir.
- Çalışmada teslim tarihi belirlerken regresyon modelleri ve yapay sinir ağı modeli kullanılmıştır. Regresyon modellerinde elde edilen sonucun nereden ve nasıl geldiği rahatlıkla belirlenebilir. Fakat yapay sinir ağında bu durum söz konusu değildir. Çünkü yapay sinir ağında öğrenme proses elemanları arasındaki bağlantı ağırlık değerlerinde gizli olduğundan, verilen sonuç için açıklama yapma yeteneği yoktur. Yapay sinir ağının bu yönü bir dezavantaj olarak görülebilir.

6.4. İleriye Dönük Çalışma Alanları

Bu çalışmada teslim tarihi belirlemek için kullanılan yapay sinir ağı modeli geriye yayımlı yapay sinir ağı modelidir. Gizli katman olarak tek katman kullanılmıştır. İki veya daha fazla gizli katman kullanılarak daha iyi sonuçlar elde edilmesi mümkün olabilir.

Ayrıca yine bu çalışmada tam bağlantılı yapay sinir ağı kullanılmıştır. Yani her bir katmandaki her bir proses elemanı bir sonraki katmanın her bir proses elemanı ile bağlantılıdır. Bunun yerine yarı bağlantılı, yani bir proses elemanının bir sonraki katmandaki sadece bir veya birkaç proses elemanı ile bağlantılı olduğu bir ağ modeli kullanılabilir. Bu şekilde daha iyi sonuçlar alınması sağlanabilir.

Yapay sinir ağının teslim tarihi tahmininde iş ve atölyeye ait çeşitli veriler girdi olarak kullanılmıştır. İş ve atölyeye ait daha fazla bilgiler girdi olarak kullanılarak veya kullanılan bazı girdiler kullanılmayarak iyi sonuçlar almak mümkün olabilir. Burada kullanılan yapay sinir ağı geriye yayımlı yapay sinir ağıdır. Bu ağ yerine daha başka yapay sinir ağı modelleri kullanılması daha iyi tahmin sonuçları elde edilmesini sağlayabilir.

KAYNAKLAR

ABDALLAH, M. H., "Computer Simulation Studies for a Scheduling System", PhD Thesis, Cranfield Institute of Technology, 1973.

AHMED, I., FISHER, W. W., "Due Date Assignment, Job Order Release, and Sequencing Interaction in Job Shop Scheduling", Decision Sciences, No. 23-3, s. 633-647, 1992.

ARIZONO, I., YAMAMOTO, A., OHTA, H., "Scheduling for Minimizing Total Actual Flow Time by Neural Networks", International Journal of Production Research, No. 30-3, 1992.

BAGCHI, U., JULIEN, F. M., MAGAZINE, M. J., "Note: Due Date Assignment to Multi Job Customer Orders", Management Science, No. 40-10, s. 1389-1392, 1994.

BAKER, C. T., DZIELINSKI, B. P., "Simulation of a Simplified Job Shop", Management Science, Vol. 6, No. 3, s. 311-323, 1960.

BAKER, K. R., "Introduction to Sequencing and Scheduling", John Wiley & Sons, New York, 1974.

BANKS, J., NORMAN, V. B., "Justifying Simulation in Today's Manufacturing Environment", IIE Solutions, p. 17-19, Nov., 1995.

BAYRAKTAR, E., KADAYIFÇILAR, S., "Gereksinimlerinizi En İyi Karşılacak Benzetim Yazılımı Seçimi", Endüstri Mühendisliği, 7, No 5, s. 19, 1996.

BIRMAN, M., MOSHEIOV, G., "A note on a Due Date Assignment on a Two Machine Flow Shop", Computers and Operations Research, No. 31, s. 473-480, 2004.

BISKUP, D., JAHNKE, H., "Common Due Date Assignment for Scheduling on a Single Machine with Jointly Reducible Processing Times", International Journal of Economics", No. 69, s. 317-322, 2001.

BOCK, D. B., PATTERSON, J. H., "A Comparison of Due Date Setting, Resource Assignment, and Job Preemption Heuristics for the Multiproject Scheduling Problem", Decision Sciences, No. 21-2, s. 387-402, 1990.

CARPENTER, G. A., GROSSBERG, S., "ART2: Self Organisation of Stable Category Recognition Codes for Analog Input Patterns", Applied Optics, No. 26-23, s. 4919-4930, 1987.

CARRIE, A., "Simulation of Manufacturing Systems", John Wiley & Sons, New York, 1988.

CARRIE, A., BITITCI, U. S., "Tool Management in FMS for Integrated Information and Material Flow", Factory 2000 International Conference on Production Systems, Cambridge, s. 301-307, 1998.

CHANG, R., "A Study of Due Ddate Assignment Rules with Constrained Tightness in a Dynamic Job Shop", Computers Ind. Eng., No. 31(1/2), s. 205-208, 1996.

CHEN, T., "An Intelligent Hybrid System for Wafer Lot Output Time Prediction", Advanced Engineering Informatics, No. 21, s. 55-65, 2007.

CHENG, T. C. E., GUPTA, M. C., "Survey of Scheduling Research Involving Ddue Date Determination Decisions", European Journal of Operational Research, No. 38, s. 156-166, 1989.

CHENG, T. C. E., JIANG, J., "Job Shop Scheduling for Missed Due Date Performance", Computers and Industrial Engineering, No. 2, s. 297-307, 1998.

CHENG, T. C. E., CHEN, Z. L., SHAKHLEVICH, N. V., "Common Due Date Assignment and Scheduling with Ready Times", Computers and Operations Research, No. 29, s. 1957-1967, 2002.

CHRYSSOLOURIS, L. M., PIERCE, J., DOMROESE, M., "Use of Neural Networks fort he Design of Manufacturing Systems", Manufacturing Review, No. 3-3, 1990.

CONWAY, R. W., "Priority Ddispatching and Job Lateness in a Job Shop", Journal of Industrial Engineering, No. 16, s. 228-237, 1965.

CONWAY, R. W., MAXWELL, W. L., MILLER, L., "Theory of Scheduling", Addison Welley, New York, 1967.

DAGLI, C., HUGGAHALLI, R., "A Neural Network Architecture for Faster Dynamic Scheduling in Manufacturing Systems", Proceedings of IEEE, International Conference on Robotics and Automation, s. 2408-2413, California, 1991.

DAYHOFF, J. E., "Neural Network Architectures", Van Nostrand Reinhold, New York, 1990.

DEMİRCİ, E., "Üretim Benzetiminde Çıktıların Değerlendirilmesi ve Bir Uygulama", Yöneylem Araştırması Dergisi, No. 7-2, s. 86-87, 1995.

EMSHOFF, J. R., SISSON, R. L., "Design and Use of Computer Simulation Models", The Macmillan Co., Newyork, 1970.

ERKUT, H., "Yönetimde Simülasyon Yaklaşımı", İrfan Yayıncılık, 1992.

FRENCH, S., "Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop", John Wiley & Sons, New York, 1982.

FRY, T. D., PHILIPOOM, P. R., MARKLAND, R. E., "Due Date Assignment in a Multistage Job Shop", IIE Transactions, No. 21-2, s. 153-161, 1989.

GEYİK, F., CEDİMOĞLU, İ. H., "A Review of the Production Scheduling Approaches Based on Artificial Intelligence and the Integration of Process Planning and Scheduling", Proceedings on Swiss Conference of CAD/CAM'99, Neuchatel University, Switzerland, 22-24 February, s.167-174, 1999.

GORDON, V., KUBIAK, W., "Single Machine Scheduling with Release and Due Date Assignment to Minimize the Weighted Number of Late Jobs", Information Processing Letters, No. 68, s. 153-159, 1998.

GORDON, V., PROTH, J. M., CHU, C., "A Survey of the State-of-the-Art Common Due Date Assignment and Scheduling Research", European Journal of Operational Research, No. 139, s. 1-25, 2002.

GRAVES, S. C., 1981, "A Review of Production Scheduling", Operations Research, No. 29-4, s. 646-675, 1981.

GUPTA, J. N. D., KRUGER, K., LAUFF, V., WERNER, F., SOTSKOV, Y. N., "Heuristics for Flow Shops with Controllable Processing Times and Assignable Due Dates", Computers and Operations Research, No. 29, s. 1417-1439, 2002.

GÜVENTÜRK, M., "Simülasyon-Benzetim, Sihir-Teknik-Sanat", Otomasyon, s. 72, Kasım, 1995.

HALAÇ, O., "İşletmelerde Simülasyon Teknikleri", Alfa, İstanbul, 1998.

KARNA, K. N., BREEN, D. M., "An Artificial Neural Networks Tutorial: Part 1 Basics", Neural Networks, No. 1-1, s. 4-23, 1989.

KHAW, J., SIONG, L. B., "Shop Floor Scheduling Using a Three Dimensional Neural Network Model", International Conference on Computer Integrated Manufacturing, ICCIM, s. 563-566, Singapore, Sep 30-Oct 4 1991.

KOHONEN, T., "State of the Art in Neural Computing", IEEE First International Conference on Neural Networks, No. 1, s. 79-90, 1987.

LAW, A., "Simulation of Manufacturing Systems", Proceedings of the 1998 Winter Simulation Conference, s. 40-51, 1998.

LI, C. H., CHENG, T. C. E., "Due Date Determination with Resequencing", IIE Transactions, No. 31-2, s. 183-188, 1999.

LIPPMANN, R. P., "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, No. 4-2, s. 4-22, 1987.

MASSON, E., WANG, Y. J., "Introduction to Computing and Learning in Artificial Neural Networks", European Journal of Operational Research, No. 47, s. 1-28, 1990.

MIN, L., CHENG, W., "Genetic Algorithms for the Optimal Common Due Date Assignment and the Optimal Scheduling Policy in Parallel Machine Earliness/Tardiness Scheduling Problems", Robotics and Computer Integrated Manufacturing, No. 22, s. 279-287, 2006.

MOSHEIOV, G., "A Common Due Date Assignment Problem on Parallel Identical Machines", Computers and Operations Research, No. 28, s. 719-732, 2001.

MOSHEIOV, G., "Due Date Assignment with Asymmetric Earliness Tardiness Cost", Journal of the Operational Research Society, No. 54, s. 1222-1224, 2003.

MOSHEIOV, G., ORON, D., "Due Date Assignment and Maintenance Activity Scheduling Problem", Mathematical and Computer Modelling, No. 44, s. 1053-1057, 2006.

MOSSES, S., "Due Date Assignment Using Feedback Control with Reinforcement Learning", IIE Transactions, No. 31-10, s. 989-999, 1999.

NEGNEVITSKY, M., "Artificial Intelligence: A Guide to Intelligent Systems", Addison-Wesley, 2002.

NELSON, M. N., ILLINGWORTH, W. T., "A Practical Guide to Neural Nets", Addison-Wesley, 1991.

NEURALWARE INC., "NeuralWorks Professional II/Plus: Neural Computing, Pittsburg, 1990.

NEWMAN, S. T., "An Integrated Approach to the Design and Evaluation of Flexible Machining Cells", PhD Thesis, Loughborough University of Technology, 1990.

ORHUNBİLGE, N., "Uygulamalı Regresyon ve Korelasyon Analizi", Avcıol Basım Yayın, İstanbul, 2000.

OZTEMEL, E., "Yapay Sinir Ağları", Papatya Yayıncılık, İstanbul, 2003.

PARK, Y., KIM, S., LEE, Y. H., "Scheduling Jobs on Parallel Machines Applying Neural Networks and Heuristic Rules", Computers and Industrial Engineering, No. 38, s. 189-202, 2000.

PARKER, R. G., RARDIN, R. L., "An Overview of Complexity Theory in Discrete Optimization Part I: Concepts", IIE Transactions, s. 3-10, 1982.

PHILIPOOM, P. R., REES, L. P., WIEGMANN, L., "Using Neural Networks to Determine Internally Set Due Date Assignments for Shop Scheduling", Decision Sciences, No. 25-5/6, s. 825-851, 1994.

PHILIPOOM, P. R., WIEGMANN, L., REES, L. P., "Cost Based Due Date Assignment with the Use of Classical and Neural Network Approaches", Naval Research Logistics, No. 44, s. 21-46, 1997..

PHILIPOOM, P. R., "The Choice of Dispatching Rules in a Shop Using Internally Set Due Dates with Quoted Leadtime and Tardiness Costs", International Journal of Production Research, No. 7, s. 1641-1655, 2000.

QI, X., YU, G., BARD, J. F., "Single Machine Scheduling with Assignable Due Dates", Discrete Applied Mathematics, No. 211-233, 2002.

RAAYMAKERS, W. H. M., WEIJTERS, A. J. M. M., "Makespan Estimation in Batch Process Industries: A Comparison between Regression Analysis and Neural Networks", European Journal of Operational Research, No. 145, s. 14-30, 2003.

RABELO, L. C., ALPTEKİN, S., "Adaptive Scheduling and Control Using Artificial Neural Networks for a Hierarchical/Distributed FMS Architecture, IEEE Second International Conference on Computer Integrated Manufacturing, s. 538-545, 1990.

RAGATZ, G. L., MABERT, V. A., "A Simulation Analysis of Due Date Assignment Rules", Journal of Operations Management, No. 5, s. 27-39, 1984.

RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J., "Learning Representations by Back Propagating Errors", Nature, No. 323, s. 533-536, 1986.

SABUNCUOGLU, I., GURGUN, B., "A Neural Network Model for Scheduling Problems", Working Paper, Bilkent University, 1994.

SABUNCUOGLU, I., COMLEKCI, A., "Operation Based Flow Time Estimation in a Dynamic Job Shop", Omega, No. 30, s. 423-442, 2002.

SARIASLAN, H., "Simülasyon Tekniği", Turhan Kitabevi, Ankara, 1988.

SCHORER, B. J., TSENG, F. T., "Modeling Complex Manufacturing Systems Using Simulation", Proceedings of the 1987 Winter Simulation Conference, 1987.

SHA, D. Y., HSU, S. Y., "Due Date Assignment in Wafer Fabrication Using Artificial Neural Networks", International Journal of Advanced Manufacturing Technology, No. 23, s. 768-775, 2004.

- SHA, D. Y., LIU, C. H., "Using Data Mining for Due Date Assignment in a Dynamic Job Shop Environment", *International Journal of Advanced Manufacturing Technology*, No. 25, s. 1164-1174, 2005.
- SHABTAY, D., STEINER, G., "Two Due Date Assignment Problems in Scheduling a Single Machine", *Operations Research Letters*, No. 43, s. 683-691, 2006.
- SHANNON, R. E., "Systems Simulation: The Art and Science", Prentice-Hall, 1975.
- SIMON, H., "Search and Reasoning in Problem Solving", *Artificial Intelligence*, Vol. 21, No. 2, p. 7-30, 1983.
- SIMPSON, P., "Artificial Neural Networks", Pergamon Pres, New York, 1990.
- SISSON, R. L., "Methods of Sequencing in Job Shops – A Review", *Operations Research*, No. 7-1, s. 10-29, 1959.
- SONG, D. P., HICKS, C., EARL, C. F., "Product Due Date Assignment for Complex Assemblies", *International Journal of Economics*, No. 76, s. 243-256, 2002.
- SOROUSH, S. M., "Sequencing and Due Date Determination in the Stochastic Single Machine Problem with Earliness and Tardiness Costs", *European Journal of Operational Research*, No. 113, s. 450-468, 1999.
- TAHA, H. A., "Yöneylem Araştırması", *Operational Research 6. basımdan çeviri, Literatür Yayıncılık, İstanbul, 2000.*
- TARI, R., "Ekonometri", Alfa Basım Yayım, İstanbul, 1999.
- TAŞGETİREN, M. F., CEDİMOĞLU, İ. H., İNCE., B., "Teslim Tarihi Oluşturma Yöntemleri Üzerine Bir Karşılaştırma", *YA/EM'95 Bildirileri, ODTÜ, Ankara, 10-11 Temmuz 1995.*
- TAŞGETİREN, M. F., "Atölye Tipi Çizelgeleme Problemi için Bir Uzman-Yapay Sinir Ağı Modeli", *Doktora Tezi, İstanbul Üniversitesi, 1996.*
- TAŞGETİREN, M. F., *Ders Notları, 1999.*
- TÜRKER, A. K., "İş Parçaları ve Kesici Takımların EİS İçerisinde Dual Akışlarının Tasarımı ve Analizi", *Doktora Tezi, Sakarya Üniversitesi, 1997.*
- VINEYARD, M., GYAMPAH, K. A., MEREDITH, J. R., "An Evaluation of Maintenance Policies for Flexible Manufacturing Systems", *International Journal of Operations and Production Management*, Vol. 20, No. 4, p. 409-426, 2000.
- VUJOSEVIC, R., "Visual Interactive Simulation and Artificial Intelligence in Design of Flexible Manufacturing Systems", *International Journal of Production Research*, Vol. 32, No. 8, p. 1955-1971, 1994.

WAGNER, B. J., RAGATZ, G. L., "The Impact of Lot Splitting on Due Date Performance", *Journal of Operations Management*, No. 12, s. 13-25, 1994.

WANG, W., "A Knowledge Based Modeling System for the Design and Evaluation of Flexible Manufacturing Facilities", PhD Thesis, Loughborough University of Technology, 1989.

WANG, C. S., UZSOY, R., "A Genetic Algorithm to Minimize Maximum Lateness on a Batch Processing Machine", *Computers and Operations Research*, No. 29, s. 1621-1640, 2002.

WASSERMAN, P. D., "Neural Computing: Theory and Practice, Van Nostrand Reinhold, New York, 1989.

WELLS, G., "An Introduction to Neural Networks", *Applications of AI in Process Control*, 1992.

XIAO, W. Q., LI, C. L., "Approximation Algorithms for Common Due Date Assignment and Job Scheduling on Parallel Machines", *IIE Transactions*, No. 34, s. 467-477, 2002.

YANG, S. WANG, D., "A New Adaptive Neural Network and Heuristics Approach for Job Shop Scheduling", *Computers and Operations Research*, No. 28, s. 955-971, 2001.

VERAL, E. A., "Computer Simulation of Due Date Setting in Multimachine Job Shops", *Computers and Industrial Engineering*, No. 41, s. 77-94, 2001.

YILDIRIM, M. B., CAKAR, T., DOGUC, U., MEZA, J. C., "Machine Number, Priority Rule, and Due Date Determination in Flexible Manufacturing Systems Using Artificial Neural Networks", *Computers and Industrial Engineering*, No. 50, s. 185-194, 2006.

ZHANG, H. C., YAN, P., CHANG, T., "Solving Job Shop Scheduling Problem with Priority Using Neural Network", *IEEE, International Journal Conference Neural Networks*, No. 2, Singapore, 1991.

EKLER

Ek A. EKİS öncelik kuralı ile TİY modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	P	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,654	,428	,428	681,26581

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	3474552446,846	1	3474552446,846	7486,273	,000
	Residual	4640766870,155	9999	464123,099		
	Total	8115319317,001	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	P	6,963	,080	,654	86,523	,000

Ek B. EKİS öncelik kuralı ile OS modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	N	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,500	,250	,250	780,12098

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2030040432,641	1	2030040432,641	3335,652	,000
	Residual	6085278884,360	9999	608588,747		
	Total	8115319317,001	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	N	50,062	,867	,500	57,755	,000

Ek C. EKİS öncelik kuralı ile TİY+OS modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	N, P	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,779	,607	,607	564,81998

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4925741218,310	2	2462870609,155	7720,074	,000
	Residual	3189578098,691	9998	319021,614		
	Total	8115319317,001	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	P	21,511	,226	2,021	95,272	,000
	N	-143,238	2,124	-1,431	-67,445	,000

Ek D. EKİS öncelik kuralı ile KİS modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	JIQ, P	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,715	,511	,511	629,95976

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4147619995,286	2	2073809997,643	5225,686	,000
	Residual	3967699321,715	9998	396849,302		
	Total	8115319317,001	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	P	14,294	,193	1,343	74,086	,000
	JIQ	-17,614	,428	-,747	-41,183	,000

Ek E. EKİS öncelik kuralı ile KIY modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	WIQ, P	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,697	,485	,485	646,34076

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	3938591103,792	2	1969295551,896	4713,981	,000
	Residual	4176728213,209	9998	417756,373		
	Total	8115319317,001	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	P	12,213	,175	1,148	69,766	,000
	WIQ	-,048	,001	-,548	-33,328	,000

Ek F. EETT öncelik kuralı ile TİY modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	P	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,967	,935	,935	164,15017

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	3846877217,623	1	3846877217,623	142766,285	,000
	Residual	269425833,378	9999	26945,278		
	Total	4116303051,000	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	P	7,336	,019	,967	377,844	,000

Ek G. EETT öncelik kuralı ile OS modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	N	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,962	,925	,925	176,07464

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	3806311275,996	1	3806311275,996	122775,214	,000
	Residual	309991775,004	9999	31002,278		
	Total	4116303051,000	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	N	68,550	,196	,962	350,393	,000

Ek H. EETT öncelik kuralı ile TİY+OS modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	N, P	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,975	,951	,951	142,53273

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	3913187881,507	2	1956593940,754	96310,021	,000
	Residual	203115169,493	9998	20315,580		
	Total	4116303051,000	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	P	4,184	,058	,551	72,532	,000
	N	30,960	,542	,434	57,132	,000

Ek I. EETT öncelik kuralı ile KİS modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	JIQ, P	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,992	,985	,985	78,56396

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4054592430,669	2	2027296215,335	328450,880	,000
	Residual	61710620,331	9998	6172,296		
	Total	4116303051,000	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	P	3,170	,025	,418	129,192	,000
	JIQ	6,910	,038	,593	183,447	,000

Ek J. EETT öncelik kuralı ile KIY modeli katsayıları için test çıktıları

Regression

Variables Entered/Removed			
Model	Variables Entered	Variables Removed	Method
1	WIQ, P	.	Enter

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,983	,967	,967	116,16501

ANOVA						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	3981386933,834	2	1990693466,917	147520,946	,000
	Residual	134916117,166	9998	13494,311		
	Total	4116303051,000	10000			

Coefficients						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	P	4,254	,034	,561	125,881	,000
	WIQ	,027	,000	,445	99,839	,000

Ek K. EKİS önce öncelik kuralı ile OMS performans ölçütü için modellerin istatistik test çıktıları

Oneway

Test of Homogeneity of Variances OMS			
Levene Statistic	df1	df2	Sig.
346,785	5	59994	,000

ANOVA OMS					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1182740505,285	5	236548101,057	8792,803	,000
Within Groups	1613986659,963	59994	26902,468		
Total	2796727165,248	59999			

Post Hoc Tests

Multiple Comparisons Dependent Variable: OMS Tukey HSD						
(I) Model	(J) Model	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-9,75590(*)	2,31959	,000	-16,3663	-3,1455
	3	69,38010(*)	2,31959	,000	62,7697	75,9905
	4	27,87910(*)	2,31959	,000	21,2687	34,4895
	5	-291,60910(*)	2,31959	,000	-298,2195	-284,9987
	6	166,68980(*)	2,31959	,000	160,0794	173,3002
2	1	9,75590(*)	2,31959	,000	3,1455	16,3663
	3	79,13600(*)	2,31959	,000	72,5256	85,7464
	4	37,63500(*)	2,31959	,000	31,0246	44,2454
	5	-281,85320(*)	2,31959	,000	-288,4636	-275,2428
	6	176,44570(*)	2,31959	,000	169,8353	183,0561
3	1	-69,38010(*)	2,31959	,000	-75,9905	-62,7697
	2	-79,13600(*)	2,31959	,000	-85,7464	-72,5256
	4	-41,50100(*)	2,31959	,000	-48,1114	-34,8906
	5	-360,98920(*)	2,31959	,000	-367,5996	-354,3788
	6	97,30970(*)	2,31959	,000	90,6993	103,9201
4	1	-27,87910(*)	2,31959	,000	-34,4895	-21,2687
	2	-37,63500(*)	2,31959	,000	-44,2454	-31,0246
	3	41,50100(*)	2,31959	,000	34,8906	48,1114
	5	-319,48820(*)	2,31959	,000	-326,0986	-312,8778
	6	138,81070(*)	2,31959	,000	132,2003	145,4211
5	1	291,60910(*)	2,31959	,000	284,9987	298,2195
	2	281,85320(*)	2,31959	,000	275,2428	288,4636
	3	360,98920(*)	2,31959	,000	354,3788	367,5996
	4	319,48820(*)	2,31959	,000	312,8778	326,0986
	6	458,29890(*)	2,31959	,000	451,6885	464,9093
6	1	-166,68980(*)	2,31959	,000	-173,3002	-160,0794
	2	-176,44570(*)	2,31959	,000	-183,0561	-169,8353
	3	-97,30970(*)	2,31959	,000	-103,9201	-90,6993
	4	-138,81070(*)	2,31959	,000	-145,4211	-132,2003
	5	-458,29890(*)	2,31959	,000	-464,9093	-451,6885

* The mean difference is significant at the .05 level.

Homogeneous Subsets

OMS Tukey HSD							
Model	N	Subset for alpha = .05					
		1	2	3	4	5	6
6	10000	210,4581					
3	10000		307,7678				
4	10000			349,2688			
1	10000				377,1479		
2	10000					386,9038	
5	10000						668,7570
Sig.		1,000	1,000	1,000	1,000	1,000	1,000
Means for groups in homogeneous subsets are displayed.							
a Uses Harmonic Mean Sample Size = 10000,000.							

Ek L. EKİS önce öncelik kuralı ile OSK performans ölçütü için modellerin istatistik test çıktıları

Oneway

Test of Homogeneity of Variances OSK			
Levene Statistic	df1	df2	Sig.
242,235	5	59994	,000

ANOVA OSK					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	863484745088242,000	5	172696949017648,400	218,424	,000
Within Groups	47434275485951200,000	59994	790650323131,501		
Total	48297760231039400,000	59999			

Post Hoc Tests

Multiple Comparisons Dependent Variable: OSK Tukey HSD						
(I) Model	(J) Model	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-129804,52160(*)	12574,97772	,000	-165640,6645	-93968,3787
	3	132784,79400(*)	12574,97772	,000	96948,6511	168620,9369
	4	43525,43280(*)	12574,97772	,007	7689,2899	79361,5757
	5	-184644,03840(*)	12574,97772	,000	-220480,1813	-148807,8955
	6	126109,82950(*)	12574,97772	,000	90273,6866	161945,9724
2	1	129804,52160(*)	12574,97772	,000	93968,3787	165640,6645
	3	262589,31560(*)	12574,97772	,000	226753,1727	298425,4585
	4	173329,95440(*)	12574,97772	,000	137493,8115	209166,0973
	5	-54839,51680(*)	12574,97772	,000	-90675,6597	-19003,3739
	6	255914,35110(*)	12574,97772	,000	220078,2082	291750,4940
3	1	-132784,79400(*)	12574,97772	,000	-168620,9369	-96948,6511
	2	-262589,31560(*)	12574,97772	,000	-298425,4585	-226753,1727
	4	-89259,36120(*)	12574,97772	,000	-125095,5041	-53423,2183
	5	-317428,83240(*)	12574,97772	,000	-353264,9753	-281592,6895
	6	-6674,96450	12574,97772	,995	-42511,1074	29161,1784
4	1	-43525,43280(*)	12574,97772	,007	-79361,5757	-7689,2899
	2	-173329,95440(*)	12574,97772	,000	-209166,0973	-137493,8115
	3	89259,36120(*)	12574,97772	,000	53423,2183	125095,5041
	5	-228169,47120(*)	12574,97772	,000	-264005,6141	-192333,3283
	6	82584,39670(*)	12574,97772	,000	46748,2538	118420,5396
5	1	184644,03840(*)	12574,97772	,000	148807,8955	220480,1813
	2	54839,51680(*)	12574,97772	,000	19003,3739	90675,6597
	3	317428,83240(*)	12574,97772	,000	281592,6895	353264,9753
	4	228169,47120(*)	12574,97772	,000	192333,3283	264005,6141
	6	310753,86790(*)	12574,97772	,000	274917,7250	346590,0108
6	1	-126109,82950(*)	12574,97772	,000	-161945,9724	-90273,6866
	2	-255914,35110(*)	12574,97772	,000	-291750,4940	-220078,2082
	3	6674,96450	12574,97772	,995	-29161,1784	42511,1074
	4	-82584,39670(*)	12574,97772	,000	-118420,5396	-46748,2538
	5	-310753,86790(*)	12574,97772	,000	-346590,0108	-274917,7250

* The mean difference is significant at the .05 level.

Homogeneous Subsets

OSK Tukey HSD						
Model	N	Subset for alpha = .05				
		1	2	3	4	5
3	10000	305370,9581				
6	10000	312045,9226				
4	10000		394630,3193			
1	10000			438155,7521		
2	10000				567960,2737	
5	10000					622799,7905
Sig.		,995	1,000	1,000	1,000	1,000
Means for groups in homogeneous subsets are displayed.						
a Uses Harmonic Mean Sample Size = 10000,000.						

Ek M. EKİS önce öncelik kuralı ile OPG performans ölçütü için modellerin istatistik test çıktıları

Oneway

Test of Homogeneity of Variances OPG			
Levene Statistic	df1	df2	Sig.
1155,166	5	59994	,000

ANOVA OPG					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	91085314,241	5	18217062,848	705,785	,000
Within Groups	1548509073,199	59994	25811,066		
Total	1639594387,440	59999			

Post Hoc Tests

Multiple Comparisons Dependent Variable: OPG Tukey HSD						
(I) Model	(J) Model	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-61,30730(*)	2,27205	,000	-67,7822	-54,8324
	3	-24,66580(*)	2,27205	,000	-31,1407	-18,1909
	4	-10,77020(*)	2,27205	,000	-17,2451	-4,2953
	5	55,54860(*)	2,27205	,000	49,0737	62,0235
	6	39,24960(*)	2,27205	,000	32,7747	45,7245
2	1	61,30730(*)	2,27205	,000	54,8324	67,7822
	3	36,64150(*)	2,27205	,000	30,1666	43,1164
	4	50,53710(*)	2,27205	,000	44,0622	57,0120
	5	116,85590(*)	2,27205	,000	110,3810	123,3308
	6	100,55690(*)	2,27205	,000	94,0820	107,0318
3	1	24,66580(*)	2,27205	,000	18,1909	31,1407
	2	-36,64150(*)	2,27205	,000	-43,1164	-30,1666
	4	13,89560(*)	2,27205	,000	7,4207	20,3705
	5	80,21440(*)	2,27205	,000	73,7395	86,6893
	6	63,91540(*)	2,27205	,000	57,4405	70,3903
4	1	10,77020(*)	2,27205	,000	4,2953	17,2451
	2	-50,53710(*)	2,27205	,000	-57,0120	-44,0622
	3	-13,89560(*)	2,27205	,000	-20,3705	-7,4207
	5	66,31880(*)	2,27205	,000	59,8439	72,7937
	6	50,01980(*)	2,27205	,000	43,5449	56,4947
5	1	-55,54860(*)	2,27205	,000	-62,0235	-49,0737
	2	-116,85590(*)	2,27205	,000	-123,3308	-110,3810
	3	-80,21440(*)	2,27205	,000	-86,6893	-73,7395
	4	-66,31880(*)	2,27205	,000	-72,7937	-59,8439
	6	-16,29900(*)	2,27205	,000	-22,7739	-9,8241
6	1	-39,24960(*)	2,27205	,000	-45,7245	-32,7747
	2	-100,55690(*)	2,27205	,000	-107,0318	-94,0820
	3	-63,91540(*)	2,27205	,000	-70,3903	-57,4405
	4	-50,01980(*)	2,27205	,000	-56,4947	-43,5449
	5	16,29900(*)	2,27205	,000	9,8241	22,7739

* The mean difference is significant at the .05 level.

Homogeneous Subsets

OPG Tukey HSD							
Model	N	Subset for alpha = .05					
		1	2	3	4	5	6
5	10000	75,3472					
6	10000		91,6462				
1	10000			130,8958			
4	10000				141,6660		
3	10000					155,5616	
2	10000						192,2031
Sig.		1,000	1,000	1,000	1,000	1,000	1,000
Means for groups in homogeneous subsets are displayed.							
a Uses Harmonic Mean Sample Size = 10000,000.							

Ek N. EKİS önce öncelik kuralı ile ONG performans ölçütü için modellerin istatistik test çıktıları

Oneway

Test of Homogeneity of Variances ONG			
Levene Statistic	df1	df2	Sig.
1272,549	5	59994	,000

ANOVA ONG					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1488894211,596	5	297778842,319	67968,339	,000
Within Groups	262842141,584	59994	4381,140		
Total	1751736353,180	59999			

Post Hoc Tests

Multiple Comparisons Dependent Variable: ONG Tukey HSD						
(I) Model	(J) Model	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	51,55170(*)	,93607	,000	48,8841	54,2193
	3	90,68640(*)	,93607	,000	88,0188	93,3540
	4	38,64510(*)	,93607	,000	35,9775	41,3127
	5	-347,16790(*)	,93607	,000	-349,8355	-344,5003
	6	127,43760(*)	,93607	,000	124,7700	130,1052
2	1	-51,55170(*)	,93607	,000	-54,2193	-48,8841
	3	39,13470(*)	,93607	,000	36,4671	41,8023
	4	-12,90660(*)	,93607	,000	-15,5742	-10,2390
	5	-398,71960(*)	,93607	,000	-401,3872	-396,0520
	6	75,88590(*)	,93607	,000	73,2183	78,5535
3	1	-90,68640(*)	,93607	,000	-93,3540	-88,0188
	2	-39,13470(*)	,93607	,000	-41,8023	-36,4671
	4	-52,04130(*)	,93607	,000	-54,7089	-49,3737
	5	-437,85430(*)	,93607	,000	-440,5219	-435,1867
	6	36,75120(*)	,93607	,000	34,0836	39,4188
4	1	-38,64510(*)	,93607	,000	-41,3127	-35,9775
	2	12,90660(*)	,93607	,000	10,2390	15,5742
	3	52,04130(*)	,93607	,000	49,3737	54,7089
	5	-385,81300(*)	,93607	,000	-388,4806	-383,1454
	6	88,79250(*)	,93607	,000	86,1249	91,4601
5	1	347,16790(*)	,93607	,000	344,5003	349,8355
	2	398,71960(*)	,93607	,000	396,0520	401,3872
	3	437,85430(*)	,93607	,000	435,1867	440,5219
	4	385,81300(*)	,93607	,000	383,1454	388,4806
	6	474,60550(*)	,93607	,000	471,9379	477,2731
6	1	-127,43760(*)	,93607	,000	-130,1052	-124,7700
	2	-75,88590(*)	,93607	,000	-78,5535	-73,2183
	3	-36,75120(*)	,93607	,000	-39,4188	-34,0836
	4	-88,79250(*)	,93607	,000	-91,4601	-86,1249
	5	-474,60550(*)	,93607	,000	-477,2731	-471,9379

* The mean difference is significant at the .05 level.

Homogeneous Subsets

ONG Tukey HSD							
Model	N	Subset for alpha = .05					
		1	2	3	4	5	6
6	10000	118,8104					
3	10000		155,5616				
2	10000			194,6963			
4	10000				207,6029		
1	10000					246,2480	
5	10000						593,4159
Sig.		1,000	1,000	1,000	1,000	1,000	1,000
Means for groups in homogeneous subsets are displayed.							
a Uses Harmonic Mean Sample Size = 10000,000.							

Ek O. EETT önce öncelik kuralı ile OMS performans ölçütü için modellerin istatistik test çıktıları

Oneway

Test of Homogeneity of Variances OMS			
Levene Statistic	df1	df2	Sig.
76,674	5	59994	,000

ANOVA OMS					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	310136136,924	5	62027227,385	1811,303	,000
Within Groups	2054466925,196	59994	34244,540		
Total	2364603062,120	59999			

Post Hoc Tests

Multiple Comparisons Dependent Variable: OMS Tukey HSD						
(I) Model	(J) Model	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-95,99360(*)	2,61704	,000	-103,4516	-88,5356
	3	-49,44880(*)	2,61704	,000	-56,9068	-41,9908
	4	16,58960(*)	2,61704	,000	9,1316	24,0476
	5	120,73830(*)	2,61704	,000	113,2803	128,1963
	6	71,97500(*)	2,61704	,000	64,5170	79,4330
2	1	95,99360(*)	2,61704	,000	88,5356	103,4516
	3	46,54480(*)	2,61704	,000	39,0868	54,0028
	4	112,58320(*)	2,61704	,000	105,1252	120,0412
	5	216,73190(*)	2,61704	,000	209,2739	224,1899
	6	167,96860(*)	2,61704	,000	160,5106	175,4266
3	1	49,44880(*)	2,61704	,000	41,9908	56,9068
	2	-46,54480(*)	2,61704	,000	-54,0028	-39,0868
	4	66,03840(*)	2,61704	,000	58,5804	73,4964
	5	170,18710(*)	2,61704	,000	162,7291	177,6451
	6	121,42380(*)	2,61704	,000	113,9658	128,8818
4	1	-16,58960(*)	2,61704	,000	-24,0476	-9,1316
	2	-112,58320(*)	2,61704	,000	-120,0412	-105,1252
	3	-66,03840(*)	2,61704	,000	-73,4964	-58,5804
	5	104,14870(*)	2,61704	,000	96,6907	111,6067
	6	55,38540(*)	2,61704	,000	47,9274	62,8434
5	1	-120,73830(*)	2,61704	,000	-128,1963	-113,2803
	2	-216,73190(*)	2,61704	,000	-224,1899	-209,2739
	3	-170,18710(*)	2,61704	,000	-177,6451	-162,7291
	4	-104,14870(*)	2,61704	,000	-111,6067	-96,6907
	6	-48,76330(*)	2,61704	,000	-56,2213	-41,3053
6	1	-71,97500(*)	2,61704	,000	-79,4330	-64,5170
	2	-167,96860(*)	2,61704	,000	-175,4266	-160,5106
	3	-121,42380(*)	2,61704	,000	-128,8818	-113,9658
	4	-55,38540(*)	2,61704	,000	-62,8434	-47,9274
	5	48,76330(*)	2,61704	,000	41,3053	56,2213

* The mean difference is significant at the .05 level.

Homogeneous Subsets

OMS Tukey HSD							
Model	N	Subset for alpha = .05					
		1	2	3	4	5	6
5	10000	275,3243					
6	10000		324,0876				
4	10000			379,4730			
1	10000				396,0626		
3	10000					445,5114	
2	10000						492,0562
Sig.		1,000	1,000	1,000	1,000	1,000	1,000
Means for groups in homogeneous subsets are displayed.							
a Uses Harmonic Mean Sample Size = 10000,000.							

Ek P. EETT önce öncelik kuralı ile OSK performans ölçütü için modellerin istatistik test çıktıları

Oneway

Test of Homogeneity of Variances OSK			
Levene Statistic	df1	df2	Sig.
22,846	5	59994	,000

ANOVA OSK					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	139416727513307,100	5	27883345502661,410	24,765	,000
Within Groups	67548521774645800,000	59994	1125921288372,934		
Total	67687938502159100,000	59999			

Post Hoc Tests

Multiple Comparisons Dependent Variable: OSK Tukey HSD						
(I) Model	(J) Model	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-156544,50540(*)	15006,14067	,000	-199308,9709	-113780,0399
	3	-63965,55080(*)	15006,14067	,000	-106730,0163	-21201,0853
	4	-57432,47230(*)	15006,14067	,002	-100196,9378	-14668,0068
	5	-42020,69500	15006,14067	,057	-84785,1605	743,7705
	6	-33893,52670	15006,14067	,211	-76657,9922	8870,9388
2	1	156544,50540(*)	15006,14067	,000	113780,0399	199308,9709
	3	92578,95460(*)	15006,14067	,000	49814,4891	135343,4201
	4	99112,03310(*)	15006,14067	,000	56347,5676	141876,4986
	5	114523,81040(*)	15006,14067	,000	71759,3449	157288,2759
	6	122650,97870(*)	15006,14067	,000	79886,5132	165415,4442
3	1	63965,55080(*)	15006,14067	,000	21201,0853	106730,0163
	2	-92578,95460(*)	15006,14067	,000	-135343,4201	-49814,4891
	4	6533,07850	15006,14067	,998	-36231,3870	49297,5440
	5	21944,85580	15006,14067	,688	-20819,6097	64709,3213
	6	30072,02410	15006,14067	,340	-12692,4414	72836,4896
4	1	57432,47230(*)	15006,14067	,002	14668,0068	100196,9378
	2	-99112,03310(*)	15006,14067	,000	-141876,4986	-56347,5676
	3	-6533,07850	15006,14067	,998	-49297,5440	36231,3870
	5	15411,77730	15006,14067	,909	-27352,6882	58176,2428
	6	23538,94560	15006,14067	,619	-19225,5199	66303,4111
5	1	42020,69500	15006,14067	,057	-743,7705	84785,1605
	2	-114523,81040(*)	15006,14067	,000	-157288,2759	-71759,3449
	3	-21944,85580	15006,14067	,688	-64709,3213	20819,6097
	4	-15411,77730	15006,14067	,909	-58176,2428	27352,6882
	6	8127,16830	15006,14067	,994	-34637,2972	50891,6338
6	1	33893,52670	15006,14067	,211	-8870,9388	76657,9922
	2	-122650,97870(*)	15006,14067	,000	-165415,4442	-79886,5132
	3	-30072,02410	15006,14067	,340	-72836,4896	12692,4414
	4	-23538,94560	15006,14067	,619	-66303,4111	19225,5199
	5	-8127,16830	15006,14067	,994	-50891,6338	34637,2972

* The mean difference is significant at the .05 level.

Homogeneous Subsets

OSK Tukey HSD				
Model	N	Subset for alpha = .05		
		1	2	3
1	10000	439933,0672		
6	10000	473826,5939	473826,5939	
5	10000	481953,7622	481953,7622	
4	10000		497365,5395	
3	10000		503898,6180	
2	10000			596477,5726
Sig.		,057	,340	1,000
Means for groups in homogeneous subsets are displayed.				
a Uses Harmonic Mean Sample Size = 10000,000.				

Ek R. EETT önce öncelik kuralı ile OPG performans ölçütü için modellerin istatistik test çıktıları

Oneway

Test of Homogeneity of Variances OPG			
Levene Statistic	df1	df2	Sig.
41,467	5	59994	,000

ANOVA OPG					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	21470786,637	5	4294157,327	108,432	,000
Within Groups	2375902238,723	59994	39602,331		
Total	2397373025,360	59999			

Post Hoc Tests

Multiple Comparisons Dependent Variable: OPG Tukey HSD						
(I) Model	(J) Model	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-36,32280(*)	2,81433	,000	-44,3431	-28,3025
	3	-12,93010(*)	2,81433	,000	-20,9504	-4,9098
	4	-34,20090(*)	2,81433	,000	-42,2212	-26,1806
	5	-58,10850(*)	2,81433	,000	-66,1288	-50,0882
	6	-40,42310(*)	2,81433	,000	-48,4434	-32,4028
2	1	36,32280(*)	2,81433	,000	28,3025	44,3431
	3	23,39270(*)	2,81433	,000	15,3724	31,4130
	4	2,12190	2,81433	,975	-5,8984	10,1422
	5	-21,78570(*)	2,81433	,000	-29,8060	-13,7654
	6	-4,10030	2,81433	,692	-12,1206	3,9200
3	1	12,93010(*)	2,81433	,000	4,9098	20,9504
	2	-23,39270(*)	2,81433	,000	-31,4130	-15,3724
	4	-21,27080(*)	2,81433	,000	-29,2911	-13,2505
	5	-45,17840(*)	2,81433	,000	-53,1987	-37,1581
	6	-27,49300(*)	2,81433	,000	-35,5133	-19,4727
4	1	34,20090(*)	2,81433	,000	26,1806	42,2212
	2	-2,12190	2,81433	,975	-10,1422	5,8984
	3	21,27080(*)	2,81433	,000	13,2505	29,2911
	5	-23,90760(*)	2,81433	,000	-31,9279	-15,8873
	6	-6,22220	2,81433	,233	-14,2425	1,7981
5	1	58,10850(*)	2,81433	,000	50,0882	66,1288
	2	21,78570(*)	2,81433	,000	13,7654	29,8060
	3	45,17840(*)	2,81433	,000	37,1581	53,1987
	4	23,90760(*)	2,81433	,000	15,8873	31,9279
	6	17,68540(*)	2,81433	,000	9,6651	25,7057
6	1	40,42310(*)	2,81433	,000	32,4028	48,4434
	2	4,10030	2,81433	,692	-3,9200	12,1206
	3	27,49300(*)	2,81433	,000	19,4727	35,5133
	4	6,22220	2,81433	,233	-1,7981	14,2425
	5	-17,68540(*)	2,81433	,000	-25,7057	-9,6651

* The mean difference is significant at the .05 level.

Homogeneous Subsets

OPG Tukey HSD					
Model	N	Subset for alpha = .05			
		1	2	3	4
1	10000	125,2642			
3	10000		138,1943		
4	10000			159,4651	
2	10000			161,5870	
6	10000			165,6873	
5	10000				183,3727
Sig.		1,000	1,000	,233	1,000
Means for groups in homogeneous subsets are displayed.					
a Uses Harmonic Mean Sample Size = 10000,000.					

Ek S. EETT önce öncelik kuralı ile ONG performans ölçütü için modellerin istatistik test çıktıları

Oneway

Test of Homogeneity of Variances ONG			
Levene Statistic	df1	df2	Sig.
1492,456	5	59994	,000

ANOVA ONG					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	420231370,278	5	84046274,056	46220,401	,000
Within Groups	109091916,872	59994	1818,380		
Total	529323287,150	59999			

Post Hoc Tests

Multiple Comparisons Dependent Variable: ONG Tukey HSD						
(I) Model	(J) Model	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	-59,67310(*)	,60306	,000	-61,3917	-57,9545
	3	-36,51690(*)	,60306	,000	-38,2355	-34,7983
	4	50,79220(*)	,60306	,000	49,0736	52,5108
	5	178,86010(*)	,60306	,000	177,1415	180,5787
	6	112,39260(*)	,60306	,000	110,6740	114,1112
2	1	59,67310(*)	,60306	,000	57,9545	61,3917
	3	23,15620(*)	,60306	,000	21,4376	24,8748
	4	110,46530(*)	,60306	,000	108,7467	112,1839
	5	238,53320(*)	,60306	,000	236,8146	240,2518
	6	172,06570(*)	,60306	,000	170,3471	173,7843
3	1	36,51690(*)	,60306	,000	34,7983	38,2355
	2	-23,15620(*)	,60306	,000	-24,8748	-21,4376
	4	87,30910(*)	,60306	,000	85,5905	89,0277
	5	215,37700(*)	,60306	,000	213,6584	217,0956
	6	148,90950(*)	,60306	,000	147,1909	150,6281
4	1	-50,79220(*)	,60306	,000	-52,5108	-49,0736
	2	-110,46530(*)	,60306	,000	-112,1839	-108,7467
	3	-87,30910(*)	,60306	,000	-89,0277	-85,5905
	5	128,06790(*)	,60306	,000	126,3493	129,7865
	6	61,60040(*)	,60306	,000	59,8818	63,3190
5	1	-178,86010(*)	,60306	,000	-180,5787	-177,1415
	2	-238,53320(*)	,60306	,000	-240,2518	-236,8146
	3	-215,37700(*)	,60306	,000	-217,0956	-213,6584
	4	-128,06790(*)	,60306	,000	-129,7865	-126,3493
	6	-66,46750(*)	,60306	,000	-68,1861	-64,7489
6	1	-112,39260(*)	,60306	,000	-114,1112	-110,6740
	2	-172,06570(*)	,60306	,000	-173,7843	-170,3471
	3	-148,90950(*)	,60306	,000	-150,6281	-147,1909
	4	-61,60040(*)	,60306	,000	-63,3190	-59,8818
	5	66,46750(*)	,60306	,000	64,7489	68,1861

* The mean difference is significant at the .05 level.

Homogeneous Subsets

ONG Tukey HSD							
Model	N	Subset for alpha = .05					
		1	2	3	4	5	6
5	10000	91,9407					
6	10000		158,4082				
4	10000			220,0086			
1	10000				270,8008		
3	10000					307,3177	
2	10000						330,4739
Sig.		1,000	1,000	1,000	1,000	1,000	1,000
Means for groups in homogeneous subsets are displayed.							
a Uses Harmonic Mean Sample Size = 10000,000.							

ÖZGEÇMİŞ

1991 yılında İstanbul Teknik Üniversitesi Sakarya Mühendislik Fakültesi Endüstri Mühendisliği bölümünden mezun oldu. 1992 yılında aynı bölümde araştırma görevlisi olarak çalışmaya başladı. 1995 yılında İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği Programında yüksek lisansını tamamladı. Aynı yıl Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği Bilim Dalında doktora öğrenimine başladı.