

154328

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPAY SİNİR AĞLARI İLE ATÖLYE ÇİZELGELEMEDE
YENİ BİR YÖNTEM GELİŞTİRİLMESİ VE BU
YÖNTEMİN UYGULANMASI**

Endüstri Yük. Müh. Gökay AKKAYA

**FBE Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği Programında
Hazırlanan**

DOKTORA TEZİ

Tez Savunma Tarihi : 21 Nisan 2004

Tez Danışmanı : Prof. Dr. Turay GÖKÇEN (YTÜ)

Jüri Üyeleri : Prof. Dr. Hüseyin BAŞLIGİL (YTÜ)

: Prof. Dr. Ataç SOYSAL (İTÜ)

: Prof. Dr. Haluk ERKUT (İTÜ)

: Yrd. Doç. Dr. Tufan DEMİREL (YTÜ)

İSTANBUL, 2004

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ.....	v
KISALTMA LİSTESİ.....	vi
ŞEKİL LİSTESİ.....	vii
ÇİZELGE LİSTESİ.....	viii
ÖNSÖZ	x
ÖZET.....	xi
ABSTRACT.....	xii
1. GİRİŞ	1
2. YAPAY ZEKA NEDİR?	3
2.1 Bilim Olarak Yapay Zeka	7
2.2 Benzetim ve Gerçeklik	10
2.3 Felsefe Mühendisliği	11
2.4 İnsan Zihninin Bilgisayara Yansıtılması	12
3. ÖĞRENME	17
3.1 Makine Öğrenmesi	17
3.1.1 Makine Öğrenmesinin Tarihi Gelişimi	18
3.2 Değişik Seviyelerdeki Öğrenme Metotları.....	22
3.2.1 Bilgi Seviyeli Öğrenme Metotları	22
3.2.1.1 Benzerlik Tabanlı Öğrenme	22
3.2.1.2 Açıklama Tabanlı Öğrenme	23
3.2.1.3 Kavramsal Sınıflama	27
3.2.2 Sembol Seviyeli Öğrenme Metotları.....	28
3.2.2.1 Sınırlandırıcılarda Öğrenme.....	28
3.2.2.2 Genetik Algoritmalarda Öğrenme.....	28
3.2.3 Makine Seviyeli (Device Level) Öğrenme.....	29
3.3 Bilgi Gösterimi ve Öğrenme Arasındaki İlişki	29
3.4 Bilgi Sistemleri ve Makine Öğrenmesinin Sınırları.....	30
4. YAPAY SİNİR AĞLARI.....	33
4.1 Neden Yapay Sinir Ağları?	33
4.1.1 Yapay Sinir Ağlarının Özellikleri	33
4.1.1.1 Öğrenme.....	33
4.1.1.2 Genelleme.....	34

4.1.1.3	Özetleme.....	34
4.1.2	Yapay Sinir Ağlarının Tarihi Gelişimi.....	34
4.1.3	Günümüzde Yapay Sinir Ağları.....	36
4.1.4	Gelecekte Beklentiler.....	38
4.1.5	Yapay Sinir Ağları ve Uzman Sistemler.....	38
4.2	Yapay Sinir Ağlarının Esasları.....	39
4.2.1	Biyolojik Prototip.....	39
4.2.2	Yapay Sinir Hücresi (Nöron).....	40
4.2.3	Aktivasyon Fonksiyonları.....	41
4.2.4	Tek Katmanlı Yapay Sinir Ağları.....	43
4.2.5	Çok Katmanlı Yapay Sinir Ağları.....	44
4.2.6	Doğrusal Olmayan Aktivasyon Fonksiyonu.....	45
4.2.7	Tekrarlanan Ağlar.....	45
4.2.8	Yapay Sinir Ağlarının Eğitilmesi.....	46
4.3	Yapay Sinir Ağlarında Bilgi Gösterimi.....	47
4.4	Algılayıcı ile (Perceptron) Öğrenme.....	48
4.4.1	Algılayıcı ile Gösterim.....	49
4.4.2	Doğrusal Ayrılabilirlik.....	50
4.4.3	Algılayıcı (Perceptron) Öğrenme Algoritması.....	51
4.4.4	Genelleştirilmiş Delta Kuralı.....	52
4.5	Geriye Yayma Algoritması (GYA - Backpropagation).....	52
4.5.1	Yapay Sinir Ağlarının Teknik Yapısı.....	53
4.5.2	Çok Katmanlı Yapay Sinir Ağları.....	55
4.5.3	Yapay Sinir Ağının Çalıştırılması.....	56
4.5.3.1	Yapay Sinir Ağı Üzerinde İleri Doğru Geçiş (Forward Pass).....	57
4.5.3.2	Yapay Sinir Ağı Üzerindeki Geriye Doğru Geçiş (Reverse Pass).....	58
4.5.4	İleri Algoritmalar.....	62
4.5.5	Uygulama.....	62
5.	ÇİZELGELEME.....	65
5.1	Çizelgeleme Seviyeleri.....	68
5.2	Çizelgeleme Problemlerinde Kabuller.....	70
5.3	Çizelgeleme Ortamı Çeşitleri.....	71
5.4	Performans Ölçütleri ve Çizelgeleme Problemlerinin Sınıflandırılması.....	71
5.5	Atölye Tipi Üretim Ortamında İşlerin Çizelgelenmesi.....	73
5.6	Çözüm Yaklaşımları.....	78
5.7	Yapay Sinir Ağları ile Atölye Çizelgeleme.....	80
6.	ATÖLYE ÇİZELGELEME VE BENZETİM.....	84
6.1	Benzetimin Tanımı ve Temel Amaçları.....	84
6.2	Benzetimin Uygulama Alanları.....	85
6.2.1	İleri İmalat Sistemlerinin (İİS) Tasarımı.....	87
6.3	Bilgisayar Benzetiminin Karakteristikleri.....	88
6.3.1	Bilgisayar Benzetiminin Kazanımları.....	88
6.3.2	Bilgisayar Benzetiminin Kısıtlamaları.....	89
6.4	Benzetim Modellerinin Kurulması.....	90
6.4.1	Benzetim Modeli.....	95
6.4.2	Benzetimin Geçerliliği.....	96
6.5	Benzetim Tabanlı Atölye Çizelgeleme.....	100

7.	ATÖLYE ÇİZELGELEME TASARIMINDA YAPAY SİNİR AĞLARI İLE DENEYSEL ÇALIŞMA	103
7.1	Modelin Kurulması ve Tanıtımı	103
7.1.1	Atölye Planı	103
7.1.1.1	İş Merkezleri Tezgah Sayıları	103
7.1.1.2	Her İş Merkezinde Bulunan Girdi/Çıktı (G/Ç) Alanı Büyüklüğü	104
7.1.1.3	İş Merkezleri Arası Mesafe	104
7.1.2	Parça Bilgileri	104
7.1.2.1	Parça Tipi	104
7.1.2.2	Her Bir Parça Tipi İçin Operasyon Sayısı	104
7.1.2.3	Her Bir Parça Tipi İçin İş Merkezi Sayısı	105
7.1.3	İşler	105
7.1.3.1	Parça Tipi	105
7.1.3.2	Sipariş Büyüklüğü	105
7.1.3.3	Teslim Tarihi	105
7.1.3.4	Hazırlık ve İşlem Zamanları	106
7.1.4	Öncelik Kuralı	107
7.1.5	Malzeme Taşıma Sistemi	107
7.1.6	Örnek Konfigürasyonlar	107
7.1.7	Yapay Sinir Ağı Yapısı	108
7.1.8	Eğitme ve Test Setleri	110
7.1.9	Performans Kriterleri	110
7.2	Uygulama ve Deneysel Çalışma	112
7.2.1	Benzetim Deneyleri	115
7.2.2	Yapay Sinir Ağı Eğitme ve Test Sonuçları	116
7.2.2.1	Öncelik Kurallı Temel YSA Yaklaşımı Eğitme ve Test Sonuçları	117
7.2.2.2	İş Merkezine Dayalı Yaklaşım İçin YSA Eğitme ve Test Sonuçları	124
7.2.3	Deneysel Sonuçlar ve Karar Çizelgeleri	127
7.2.3.1	Öncelik Kuralına Dayalı Yaklaşım İçin Deneysel Sonuç ve Karar Çizelgeleri ..	128
7.2.3.2	İş Merkezine Dayalı Yaklaşım İçin Deneysel Sonuçlar ve Karar Çizelgeleri	137
7.2.4	Etki Analizi	142
7.2.5	Deneysel Çalışmanın Sonuçları	164
8.	SONUÇ VE DEĞERLENDİRME	165
	KAYNAKLAR	167
	EKLER	170
	Ek 1 Siman model programı	171
	Ek 2 Siman deneysel programı	177
	ÖZGEÇMİŞ	179

SİMGE LİSTESİ

$a_1, \dots, a_n$	Bireyler
P	Bireylerin sahip olduğu özellik
P_c	Ortak özellik seti
C	Genelleştirilmiş kavram
E	Çalıştırma örneği
O	Operasyonel kriter
T	Alan teorisi
S	Sistemin bulduğu açıklama
X_i	Girdi vektörü
W_i	Ağırlık vektörü
NET	Toplama fonksiyonu
D	Hedef çıktı değeri
A	Gerçek çıktı değeri
δ	Hata değeri
Δ_i	Ağırlıkları düzeltme katsayısı
α	Üstel düzeltme katsayısı
θ	Momentum katsayısı
λ	Öğme oranı
t	Zaman
$N(\mu, \sigma)$	Ortalaması μ ve standart sapması σ olan normal dağılım
S_{ik}	i . işin k . tezgahta başlama zamanı
t_{ijk}	i . işin j . işleminin k . tezgahda yapılması için gerekli olan süre
n_T	Gecikmeli parça yüzdesi

KISALTMA LİSTESİ

YSA	Yapay Sinir Ağları
GYA	Geriye yayma algoritması (Backpropagation)
ÇIKTI	Çıktı değeri
F(NET)	Aktivasyon fonksiyonu
ÇKA	Çok katmanlı algılayıcı
KEP	Kısıtlı eniyileme problemleri (Combinatorial Optimization Problems – COP)
K	Bilgi temsilinin bilişimi
P(K)	Bilginin olasılık yoğunluğu
EKİZ	En kısa işlem zamanı
EETT	En erken teslim tarihi
İGİG	İlk gelen ilk işlem görür
F	Ortalama akış zamanı
T	Ortalama gecikme
L	Gecikme süresi
N	İş sayısı
E	Tezgah (işlem) sayısı
H	Tüm işlerin gerçekleştirilmesi için gerekli toplam süre olmak üzere
İİS	İleri imalat sistemleri
OGA	Otomatik güdümlü araç
C_{max}	Maksimum tamamlanma zamanı
mu1	1. iş merkezindeki ortalama makine kullanım oranı
mu2	2. iş merkezindeki ortalama makine kullanım oranı
mu3	3. iş merkezindeki ortalama makine kullanım oranı
mu4	4. iş merkezindeki ortalama makine kullanım oranı
M1	1 numaralı iş merkezi
M2	2 numaralı iş merkezi
M3	3 numaralı iş merkezi
M4	4 numaralı iş merkezi

ŞEKİL LİSTESİ

Şekil 3.1 Öğrenmenin temel şekilleri.....	18
Şekil 3.2 Simon'ın modelini esas alan öğrenme elemanları	20
Şekil 3.3 LEX'in sezgisel algoritma yapısı.....	25
Şekil 3.4 Episodik Öğrenmede PET'in Kontrol Yapısı.....	26
Şekil 3.5 PET'in problem yapılarını episodlarda operatörlerle birleştirmesi	27
Şekil 4.1 Biyolojik sinir hücresi.....	40
Şekil 4.2 Yapay sinir hücresi	41
Şekil 4.3 Aktivasyon fonksiyonlu yapay sinir hücresi.....	42
Şekil 4.4a Sigmoidal aktivasyon fonksiyonu	42
Şekil 4.4b Hiperbolik tanjant fonksiyonu	44
Şekil 4.5 Tek katmanlı sinir ağları	44
Şekil 4.6 Çift katmanlı sinir ağı	45
Şekil 4.7 Algılayıcı (perceptron) sinir hücresi	48
Şekil 4.8 Tekli sinir hücresi sistemi	50
Şekil 4.9 Aktivasyon fonksiyonu ile bir yapay sinir hücresi	53
Şekil 4.10 Sigmoidal aktivasyon fonksiyonu.....	55
Şekil 4.11 İki katmanlı geriye yayma algoritması	56
Şekil 4.12 Çıktı katmanındaki bir ağırlığın çalıştırılması	59
Şekil 4.13 Gizli katmandaki bir ağırlığın çalıştırılması	60
Şekil 5.1 Örnek bir Gantt şeması	67
Şekil 5.2 İşlere ait uygun olmayan Gantt şeması	75
Şekil 5.3 İşlere ait uygun Gantt şeması	75
Şekil 5.4 Tezgahlara ait uygun olmayan Gantt şeması	76
Şekil 5.5 Tezgahlara ait uygun Gantt şeması	76
Şekil 5.6 Farklı yaklaşımlarla gerekli olan yapay zeka derecesi.....	80
Şekil 6.1 Benzetim modeli gelişim süreci.....	97
Şekil 7.1 Öncelik kuralına dayalı yaklaşım için yapay sinir ağı yapısı	109
Şekil 7.2 İş merkezine dayalı yaklaşım için yapay sinir ağı yapısı.....	109
Şekil 7.3 Yapay sinir ağının oluşturulması ve test edilmesi	113
Şekil 7.4 Karar verme ve etki alanı analizi	114

ÇİZELGE LİSTESİ

Çizelge 4.1 Doğrusal ayrılabilir fonksiyonlar	51
Çizelge 5.1 Çizelgeleme seviyelerinin sınıflandırılması	69
Çizelge 5.2 İşlem süreleri ve tezgah rotaları	74
Çizelge 7.1 İş merkezleri tezgah sayıları	103
Çizelge 7.2 Her iş merkezinde bulunan girdi/çıkış (G/Ç) alanı büyüklüğü	104
Çizelge 7.3 İş merkezleri arası mesafe (metre)	104
Çizelge 7.4 Parça tipi ihtimal dağılımları	104
Çizelge 7.5 Parça tipi operasyon sayısı	104
Çizelge 7.6 Her bir parça tipi için iş merkezi ismi	105
Çizelge 7.7 Parça tipi 1 hazırlık ve işlem zamanları	106
Çizelge 7.8 Parça tipi 2 hazırlık ve işlem zamanları	106
Çizelge 7.9 Parça tipi 3 hazırlık ve işlem zamanları	106
Çizelge 7.10 Parça tipi 4 hazırlık ve işlem zamanları	106
Çizelge 7.11 Parça tipi 5 hazırlık ve işlem zamanları	106
Çizelge 7.12 Örnek setinde kullanılan konfigürasyon örnekleri	108
Çizelge 7.13 Benzetim işlemi sonucundaki örnek çıktı	115
Çizelge 7.14 Benzetim çıktısı örneği	116
Çizelge 7.15 EETT1 YSA eğitim ve test sonuçları	120
Çizelge 7.16 EETT2 YSA eğitim ve test sonuçları	120
Çizelge 7.17 EETT3 YSA eğitim ve test sonuçları	121
Çizelge 7.18 EKİZ1 YSA eğitim ve test sonuçları	121
Çizelge 7.19 EKİZ2 YSA eğitim ve test sonuçları	122
Çizelge 7.20 EKİZ3 YSA eğitim ve test sonuçları	122
Çizelge 7.21 İGİG1 YSA eğitim ve test sonuçları	123
Çizelge 7.22 İGİG2 YSA eğitim ve test sonuçları	123
Çizelge 7.23 İGİG3 YSA eğitim ve test sonuçları	124
Çizelge 7.24 M1 YSA eğitim ve test sonuçları	125
Çizelge 7.25 M2 YSA eğitim ve test sonuçları	126
Çizelge 7.26 M3 YSA eğitim ve test sonuçları	126
Çizelge 7.27 M4 YSA eğitim ve test sonuçları	127
Çizelge 7.28 Örnek değerlendirme çizelgesi	128
Çizelge 7.29 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 1)	130
Çizelge 7.30 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 2)	131
Çizelge 7.31 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 3)	132
Çizelge 7.32 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 4)	133
Çizelge 7.33 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 5)	135
Çizelge 7.34 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 6)	136
Çizelge 7.35 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 1)	137
Çizelge 7.36 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 2)	139
Çizelge 7.37 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 3)	139
Çizelge 7.38 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 4)	140
Çizelge 7.39 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 5)	141
Çizelge 7.40 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 6)	141
Çizelge 7.41 Ortalama akış süresi (F) etki alanı (1.yaklaşım)	144
Çizelge 7.42 Ortalama akış süresi (F) etki alanı (2. yaklaşım)	144
Çizelge 7.43 Ortalama gecikme süresi (T) etki alanı (1.yaklaşım)	145
Çizelge 7.44 Ortalama gecikme süresi (T) etki alanı (2. yaklaşım)	145
Çizelge 7.45 Maksimum tamamlanma süresi (C_{max}) etki alanı (1.yaklaşım)	146
Çizelge 7.45 Maksimum tamamlanma süresi (C_{max}) etki alanı (1.yaklaşım) (devamı)	147

Çizelge 7.46	Maksimum tamamlanma süresi (C_{max}) etki alanı (2.yaklaşım).....	149
Çizelge 7.47	İş merkezi1 kullanım oranı (μ_1) etki alanı (1.yaklaşımı).....	150
Çizelge 7.47	İş merkezi1 kullanım oranı (μ_1) etki alanı (1.yaklaşım) (devamı)	151
Çizelge 7.48	İş merkezi1 kullanım oranı (μ_1) etki alanı (2.yaklaşım).....	152
Çizelge 7.49	İş merkezi 2 kullanım oranı (μ_2) etki alanı (1.yaklaşım).....	154
Çizelge 7.49	İş merkezi 2 kullanım oranı (μ_2) etki alanı (1.yaklaşım) (devamı)	155
Çizelge 7.50	İş merkezi 2 kullanım oranı (μ_2) etki alanı (2. yaklaşım).....	155
Çizelge 7.51	İş merkezi 3 kullanım oranı (μ_3) etki alanı (1.yaklaşım).....	157
Çizelge 7.51	İş merkezi 3 kullanım oranı (μ_3) etki alanı (1.yaklaşım) (devamı)	158
Çizelge 7.52	İş merkezi3 kullanım oranı (μ_3) etki alanı (2.yaklaşım).....	159
Çizelge 7.53	İş merkezi 4 kullanım oranı (μ_4) etki alanı (1.yaklaşım).....	161
Çizelge 7.53	İş merkezi 4 kullanım oranı (μ_4) etki alanı (1.yaklaşım) (devamı)	162
Çizelge 7.54	İş merkezi 4 kullanım oranı (μ_4) etki alanı (2.yaklaşım).....	162



ÖNSÖZ

Akademik hayatın en önemli dönüm noktalarından biri olan doktora tez çalışmamın bu son satırlarını yazarken, hem tezimin son aşamasına gelmiş olmamın sevincini, hem de bu tezin son aşamalarında mutluluğumuza daha da bir mutluluk katan ve bu tezin ayrı bir hediyesi olarak algıladığım, doğacak olan ikizlerimin keyfini birlikte yaşadım. Tez savunmasına gireceğim zamanla, ikizlerimin dünyaya gelme zamanlarının hemen hemen aynı tarihlere gelmesi benim için ayrı bir heyecana ve mutluluğa yol açmakta...

Bu heyecanı yaşarken, bu tezin ortaya çıkmasında en az benim kadar emek veren ve teşekkür etmem gereken kişiler var mutlaka...

En başta danışmanım Prof. Dr. Turay GÖKÇEN'e, bana verdiği destek ve beni yönlendirmek için hiçbir zaman unutamayacağım yardımları için;

Yine tezdeki yardımlarından dolayı Öğr. Gör. Akif KURT ve Arş. Gör. Mümtaz İPEK'e;

Eşim Yasemin AKKAYA'ya, doktora tezim süresince gösterdiği anlayışa, hoşgörüye ve tezin yazım aşamasında yaptığı yardımlara;

Benim bugünlere gelmemde en büyük katkısı olan ve sevgilerini hayatım boyunca yanımda hissettiğim annem ve babam, Belgüzar ve Hüseyin AKKAYA ve kardeşlerime;

Ve son olarak, doktora tezimin tesliminden yaklaşık üç ay sonra ailemize katılacak olan ikizlerime, çalışmalarında bana sağladıkları mutluluk ve motivasyon için teşekkür ederim.

Bu doktora çalışmasının ilerideki çalışmalara kaynak olabilmesi ve yol göstermesi dileklerimle...

ÖZET

Bir sistemin davranışlarını iyi bir şekilde temsil kabiliyetine sahip bir modelleme teknağı olan benzetim, yapay zeka teknikleriyle beraber ele alındığında etkili bir karar verme yapısı oluşturulabilir. Bu çalışmada, Atölye çizelgeleme tasarımında benzetim ve yapay sinir ağı kullanarak, tezgah seçimi, malzeme taşıma sistemi seçimi ve sistemde kullanılacak öncelik kuralı seçimi için bir sistem geliştirilmiştir. Yapay sinir ağı modeli olarak geriye yayma algoritması (GYA - backpropagation) ağı kullanılmıştır.

Çalışmada önce gerçeğe uygun sonuçlar üretebilen, eğitilmiş yapay sinir ağı elde edilmiştir. Ağı eğitmek ve gerekli olan örneklerin elde edilmesi için bilgisayar ortamında benzetim tekniğinden faydalanılmıştır. Daha sonra, karar verme ve bu kararın geçerli olduğu değerler aralığını bulma ve yorumlama yer almaktadır. Performans kriterlerinin belirli bir değerler kümesini sağlayabilecek olan, donanım konfigürasyonu ve çizelgeleme stratejisini belirleyebilmek için, eğitilmiş yapay sinir ağlarından yararlanılmıştır. Yapay sinir ağı tarafından teklif edilen sonuç ya da sonuçların benzetimi sonunda, performans kriterlerinin beklenen değerlerinden sapma miktarları hesaplanarak karar verme işlemi için bir çizelge halinde sunulmuştur. Performans ölçümünde kullanılan kriterler; ortalama akış zamanı, ortalama gecikme, maksimum tamamlanma zamanı ve tezgah merkezleri kullanım oranlarıdır.

Son olarak, hem karar verme işleminde yardımcı olabilecek hem de verilen kararların geçerli olduğu alanı belirleyebilecek bir etki analizi ile sistemin farklı durumlar karşısında nasıl bir tepki verebileceği ortaya koyulmuştur. Bunların sonucunda da karar verici, sistem hakkında daha etkili kararlar alabilecektir.

Anahtar kelimeler: Atölye Çizelgeleme, Benzetim, Yapay Sinir Ağları.

ABSTRACT

Simulation, being capable of representing a system's behavior in an effective way, when combined with the neural networks, can provide an efficient decision making structure. In this research, a system is developed in order to determine the machine, the material handling system and the priority rule that will be used in the system by using Simulation and neural network techniques in Job-Shop scheduling design. The backpropagation algorithm is chosen for the neural network model.

In this research, first, a neural network that is capable of providing realistic results is obtained. Simulation technique is used in order to obtain the samples to train the neural network in computer environment. The next step includes the decision-making, determination of the ranges where the selected decision remains valid and the related comments. Trained neural networks are used in order to determine the hardware configuration and the scheduling strategy that are capable of providing a determined set of performance criteria. After the simulation of the result(s) that is (are) proposed by the neural network, the deviations of the performance criteria from their corresponding expected values are calculated and proposed in a tabular format. The criteria used in the performance measurement are the average flow time, average tardiness, maximum completion time and machine center usage ratios.

Finally, the possible reactions of the system is presented by an "effect analysis" that may provide an aid both in decision making process and determining the ranges where the decision remains valid. Depending on these decisions and ranges the decision maker may be able to make more efficient decisions.

Keywords: Job – Shop Scheduling, Simulation, Artificial Network..

1. GİRİŞ

Bu çalışma bir atölye imalat sisteminde planlama, çizelgeleme ve analiz etme olarak kullanılan, benzetim-yapay sinir ağı temelli bir tasarım sistemidir. Atölye çizelgeleme tasarımı, tasarımların birçoğu gibi çok sayıda tasarım alternatifini değerlendirmeyi gerektiren karmaşık bir karar verme prosesidir. Bir tasarımın performansı değişik metotlarla analiz edilebilir. Örneğin; sistemin küçük ölçekte bir fiziksel modelini yapmak, analitik araçlar uygulamak, benzetim vb.

Bir atölye çizelgeleme sisteminin dinamik davranışını analiz etmek için en fazla kullanılan araç benzetimdir. Yeni bir atölye çizelgeleme tasarımı için benzetim, bütün sistemin tasarımında ve tasarım alternatiflerini değerlendirmede kullanılabilir (Law ve Kelton, 1992). Bir atölye çizelgeleme benzetiminin tipik sonuçları: Kapasite değerlendirme, teçhizat kullanım oranlarını belirleme, sistemdeki darboğazları belirleme, alternatif tasarımların performanslarını karşılaştırma, atölye çizelgeleme tasarımında önemli olmayan unsurları görme ve çizelgeleme stratejileri geliştirmedir (Pinedo, 1995).

Benzetim, yapay zeka teknikleriyle beraber ele alındığında etkili bir karar verme yapısı oluşturabilir. Yapay zeka teknikleri, tasarımda gerekli olan bilgi kaynakları setinin geliştirilmesi ve tasarım prosesinin organizasyonu gibi tasarım proseslerinde problem çözme kabiliyetini artırma özelliklerine sahiptir. Bir sinir ağı yalnız olarak veya başka bir uygulama programı ile beraber uygulanabilir. Sinir ağları, ürün ve proses tasarımı, çizelgeleme ve üretim sistemi tasarımı gibi problemlerin çözümünde uygulanmaktadır.

Bu tezde, atölye çizelgeleme tasarımında benzetim ve yapay sinir ağı kullanarak, tezgah seçimi, malzeme taşıma sistemi seçimi ve sistemde kullanılacak öncelik kuralı seçimi için bir sistem geliştirilmiştir. Bu sistemde, atölye çizelgeleme tasarımı şu elemanlardan oluşur: Alternatif tasarımların benzetimi, benzetim sonuçları ile çıkarım yapan yapay sinir ağı, model tasarımını seçmek için bir karar verici.

Sinir ağı, ileri doğru beslenen bir girdi katmanı, bir gizli katman ve bir çıktı katmanına sahiptir. Girdi katmanındaki düğüm sayısı, oluşturulmak istenen sistemin istenen performans ölçülerine bağlıdır. Bu durum, sinir hücreleri belirlenen atölye sisteminin performans ölçülerini (ortalama akış zamanı, ortalama gecikme, maksimum tamamlanma zamanı, tezgah merkezleri kullanım oranları) temsil eden değişkenlere karşılık gelir. Çıktı katmanındaki sinir hücreleri, uygulanması gereken tasarım düzenlemelerini (iş merkezi tezgah sayıları) gösterir.

Benzetim işlemi tarafından üretilen veriler düzenlenerek yapay sinir ağı girdileri elde edilir. Performans ölçütlerini (girdi katmanı) ve tasarım değişikliklerini (çıkı katmanı) temsil eden sinir hücreleri belirlenir. Belirli bir atölye çizelgeleme modeli için eğitime örnekleri seti oluşturulur. Sinir ağı eğitimi için Geriye Yayılım Algoritması (GYA) kullanılmıştır.

Yapay sinir ağı öncelik kuralı temelli ve iş merkezi temelli olmak üzere iki ayrı yaklaşım şeklinde ele alınmıştır. İş merkezine dayalı olan ikinci yaklaşım ile kararların daha kolay ve daha kesin olarak ve ek işlemler gerektirmeden alınabildiği belirlenmiştir. Öncelik kuralına dayalı olan birinci yaklaşım ise, karar verme sırasında yetersiz olmasıyla beraber, diğer yaklaşımda gözden kaçabilecek bazı noktaları yakalayabilme ihtimali, diğer yaklaşımın değerlendirilmesi ve en önemlisi etki analizinin belirlenen konfigürasyona göre yapılabilmesi için çalışmalara dahil edilmiştir.

Sistemde kullanılacak taşıyıcı (Otomatik Güdümlü Araç (OGA)) sayısı ve öncelik kuralı çiftinin kombinasyonuna göre elde edilen eğitime örneklerinden oluşan setler kullanılarak yapay sinir Ağları eğitilmiştir. Eğitime örnekleri seti önceki benzetim koşullarından elde edilmiştir.

Benzetim sonuçları hakkında çıkarım, girdi katmanındaki sinir hücreleri için girdi değerleri vektörü kullanılarak yapılmıştır. Girdi değerleri vektörü, ele alınan ortalama akış zamanı, ortalama gecikme ve maksimum tamamlanma zamanı kriterleri ile, tasarlanan sistemin birinci iş merkezi tezgah kullanım oranı (μ_1), ikinci iş merkezi tezgah kullanım oranı (μ_2), üçüncü iş merkezi tezgah kullanım oranı (μ_3), dördüncü iş merkezi tezgah kullanım oranı (μ_4) elemanlarından oluşmaktadır. Gizli katmandaki ve çıkı katmanındaki sinir hücrelerinin çıktısı hesaplanarak iş merkezlerinde kullanılması gerekli olan tezgah sayıları sonuç olarak elde edilmiştir.

Eğitilmiş olan yapay sinir ağı kullanılarak, istenen hedef performans değerlerine ulaşmak için en uygun tasarım konfigürasyonu alternatifleri elde edilmiştir. Elde edilen bu farklı atölye çizelgeleme tasarımları ile bir benzetim sınıfı örneği oluşturulmuştur. Bu yeni benzetim sınıfının ayrı ayrı benzetim işlemi yapılarak hedeften sapmaları minimum olan veya istenen hedef değerlere en yakın olan tasarım kabul edilmiştir.

Ayrıca, hem karar verme işleminde yardımcı olabilecek hem de verilen kararların geçerli olduğu alanı belirleyebilecek bir etki analizi ile sistemin farklı durumlar karşısında nasıl bir tepki verebileceği ortaya koyulmuştur. Bu tezde teklif edilen bu analiz ile karar verici sistemi kuş bakışı görebilecek ve etkili kararlar alabilecektir.

2. YAPAY ZEKA NEDİR?

Marvin Minsky'ye göre yapay zeka, "insanlar tarafından yapıldığında zeka gerektiren şeyleri yapan makineler oluşturma bilimidir." Yirmi yıl önce Minsky ve meslektaşları bilgisayar dünyasının aykırı örnekleriydiler. Diğerleri, bilgi işlem tekniklerini mü'endisliğe ve iş dünyasına uygularken, yapay zeka uzmanları etten bir makine olarak gördükleri insan beyninin yerine daha etkin elektronik modeller yerleştirmekten, dünyada yeni türler oluşturmaya yakın şeylerden söz ediyorlardı. Bugün yapay zeka alanındaki çalışmalar hala tartışma konusudur, ama bu çalışmalara daha saygın bir gözle bakılmaktadır. Günümüz toplumlarında bilgisayara bağımlılık arttıkça, makinenin sınırları hakkındaki karamsarlık da azalmaktadır. Yirmi yıl önce, Pisagor kuramını yeniden keşfetmek ve basit düzeyde satranç oynamak için programlar yazmaya zaman harcayan yalnızca birkaç bilim adamı vardı. Bugün bu insanlar ve takipçileri, ABD ve Avrupa'daki üniversitelerde araştırma merkezleri ve şirketler kurmaya başlamışlardır.

Savunma Bakanlığı ve diğer geleneksel bilimsel araştırma fonu kaynaklarının sağladığı cömert desteklerden, yapay zeka alanındaki çalışmaların giderek daha akılcı hale geldiği anlaşılmaktadır. Bu alana yatırım yapanlar, yapay zekaya sahip bilgisayarların büyük kazançlar sağlayacağı ve güçlü yeni silahlar üreteceği vaadinin cazibesine kapılmış gibi görünmektedir. Robotlar günümüzün montaj hatlarından daha karmaşık işlerde insanların yerini alacaktır. Akıllı bombalar hedeflerini daha büyük doğrulukla bulacak, programlar soruları yanıtlayacak, doğrudan şirket yöneticileri ve askeri görevlilerce verilen emirlere uyacaktır. Genel olarak, iş dünyası ve silahlı kuvvetlerde oldukça popüler olan işletim planlamasının büyük bir kısmı bilgisayarlar tarafından gerçekleştirilecektir.

Yıllardır gazetelerde ve popüler magazinlerde bilgisayarların mucizeleri hakkında yazılar okuyoruz. Yıllardır bilimkurgu filmleri ve kitapları, herşeyi bilen düşman bilgisayarlardan, robotlardan söz ediyor. Bilgisayar konusunda uzman olmayan insanların çoğu yapay zekanın zaten gerçekleştirildiğine inanıyorlar. Bu inanışa göre bazı bilgisayarlar konuşmayı anlayabilmekte, gazete okuyabilmekte, bir TV kamerası yardımıyla araba sürebilmekte ya da kendilerinden daha yetenekli bilgisayarları tasarlayabilmektedir. Aslında, günümüzde hiçbir bilgisayar programı bu işlerin kaba birer taklidinden daha fazlasını yapamamaktadır. Yapay zeka projesinin önündeki gerçek problemler bugün çözülebilen sorunlara göre çok daha zordur. Bu gibi problemler üzerinde yıllardır çalışan mükemmel programcılar bile şu ana kadar çok az ilerleme kaydedebilmiştir.

Bilgisayarların yerine getirmesi düşünölen işlevler (matematik ve simgesel mantık) ile insanların sahip olduđu yeteneklerin çeşitliliđi arasındaki büyük uçurum dikkat çekicidir. Programcılar ise, hala makinelerin verili yetenekleriyle sınırlanmış olarak çalışmak zorunda kalmaktadırlar.

Bilgisayarlar dama, tavla, satranç gibi iyi tanımlanmış ve mantığa dayanan oyunları oynamakta oldukça başarılılardır. Bilgisayarlar bu oyunlardan ilk ikisini hemen hemen bir insan kadar iyi oynayabilmekte ve satrançta amatörleri yenebilmektedir. Bununla birlikte, yapay zeka programcılarını için satranç oynamak ve mantığa dayanan bilmeceleeri çözmek hala çok mekanik ve sınırlı bir alanı kapsamaktadır. Yapay zeka programcılarını, makinelerin tartışmasız bir şekilde insanın yaptıklarına yakın şeyler yapabilmesini istemektedir. Bu yüzden, bilgisayarlara konuşma yeteneđi kazandırılması amaçlanmakta, öyküleri ve haberleri okuyan, bunları hatırlayan ve sonra okuduklarıyla ilgili soruları yanıtlayan programlar yazılmakta ya da TV kamerası tarafından oluşturulmuş bir şekilde geometrik şekillerini tanımak için tasarlanan programlarla mekanik görüntü problemleri üzerinde çalışmalara girişilmektedir. Tıp alanında teşhis amacıyla; jeolojide bir taş oluşumunun petrol bulundurup bulundurmadığını söylemek için; kimyada organik bileşiklerini belirlemek için kullanılan, önceden tasarlanmış yüzlerce hatta binlerce pratik kurala dayanan uzman sistemler diye bilinen programlarda çıkarım yapmak için otomatik tekniklerden yararlanılmaktadır. Bu arada, robotlar iş dünyasında giderek artan oranda kullanılmakta ve ticari robot tasarımcıları son yirmi yıldır yapay zeka uzmanları tarafından yapılmış araştırmaları değerlendirmektedir. Bu alanda değışen tek şey, belki de bu tür robotların birçok yapay zeka uzmanının ilgisini çekecek kadar kullanışlı hale gelmiş olmasıdır. Yapay zeka teknikleri pratik uygulamaları olmasına karşın, yine de bu uygulamalar olayın daha az ilginç olan yanıdır. Yakın zamandaki başarılarla rağmen yapay zeka, ekonomi ve askeri alanlarındaki vaatlerini gerçekleştirmekten hala yıllar, belki de on yıllarca uzaktadır. Yakın gelecekte, yalnızca bilimsel ve endüstriyel robotlar ile oldukça özelleşmiş uzman sistemler önemli bir etkiye sahip olacaktır. Diğer taraftan, bilgisayar satrancında yapılacak bir atılım yalnızca çok az sayıda usta ve büyük ustayı işsiz bırakabilecektir. Şu an, basit İngilizce komutlara karşılık veren bilgisayarlar mevcuttur ve şüphesiz bu alan daha da genişleyecektir. Günümüz kelime işlemcilerinin geleceğine bakıldığında da, kısa bir süre sonra bilgisayarların gelişkin bir görsel ve sözel iletişim aracı olabileceđi görölebilir.

Yapay zeka, pratik başarılarından çok, programların altında yatan, insanları doğanın sayısal

bilgisayarları olarak gören metafor nedeniyle tartışmaya açık ve çekici bir alandır. 1940'ların sonlarında prototiplerin yapılmasından bu yana, bilgisayarlar daha önce yalnız insanlara ait olan alanları istila etmeye başlamışlardı. Mühendisler ve uygulamalı matematikçiler için diferansiyel denklem çözer hale gelmişlerdi. İş dünyasında hesap çıkartma, envanter ve rapor alma gibi büro işlerini üstlenmişlerdi. Endüstriyel süreçlerde makineleri kontrol etmeye başlamışlardı. Bilgisayarların ortaya çıkmasından önce tüm bu işler insanlar tarafından yapılmaktaydı. Bu çalışanlar sürekli olarak bu makinelerle beraber çalışmak zorundaydılar. Endüstri devrimiyle ortaya çıkan yönelim, makineleri daha kendine yeterli hale getirmek ve insanı kontrol edilen süreçten uzaklaştırmak yönündeydi. Bu yönelim, yüzyıllar önce yel değirmenleri, su çarkları ve mekanik saatle değilse bile, buhar makinesiyle başladı, dokuma tezgahı, buharlı gemi, demiryolu, dinamo ve montaj hattı ile devam etti. Bilgisayar, kendi kendine yeten makine için bir aşamadır. Bilgisayarla çalışan programcı sadece programı yükler, bilgisayar kendi çalışmaya devam eder. Sonra da dakikalarca ya da saatlerce verilen komutlar herhangi bir dış müdahaleye gerek duymadan uygular ve yine programcıdan bağımsız olarak çeşitli sonuçlar üretir. Bir bilgisayar programı, pilotun hiçbir yardımı olmaksızın bir uzay mekiğini yörüngesinden çıkarıp yere indirebilir. Bir diğeri, bir mektubu vurgulanacak yerleri koyu harflerle ve altını çizerek mükemmel bir şekilde yazabilir. Başka bir program, sık sık bir takım tehlikeli hatalar yaparak programcısını bile şaşırtabilir. Karmaşık görevleri insan kontrolüne gerek duymadan yerine getirebilmesi, bilgisayar metaforu, milyonlarca profesyonel ve özel amaçlı programcı tarafından karşı konulmaz hale getirilmektedir. Bu makinelerin insana benzemeyen yönleri nelerdir? İnsanın temel özelliklerinden sayılan yetenekler makineye kazandırılabilirse, bu insan doğasının mekanik olduğu anlamına mı gelmektedir

Bu belirsizlikler, makinelerin tümüyle insana benzemesini hedefleyen, yapay zeka projesini ortaya çıkarmıştır. Yapay zeka projesinin hedefi matematiksel işlemler yapabilen, oyun oynayabilen, diğer makineleri kontrol edebilen günümüz bilgisayarlarına zihnin ve duyumların diğer fonksiyonlarını da kazandırmaktır. Herbert Dreyfus, Joseph Weizenbaum gibi yapay zeka karşıtları bilgisayar metaforunun absürd ve tehlikeli olduğunu düşünürken, içlerinde Nobel ödülü sahibi Herbert Simon'ın da bulunduğu, yapay zeka savunucuları konuyu, üzerinde tartışmaya değer bularak programlarının gelecek birkaç yıl içinde çok şey yapabileceği gibi mübalağalı iddialarda bulunmuşlardır. Fakat, aradan geçen yıllara rağmen bu iddiaların çoğu gerçekleştirilememiştir.

Bu konudaki en ünlü iddia, Alan Turing tarafından ortaya atılmıştır. Bilgisayar metaforu

tarafından büyülenen ünlü mantıkçı Turing 1950'de, yüzyılın sonunda bir bilgisayarın, insan yeteneklerini taklit ederek bizleri aldatabileceğini öngörmüştür. Alan Turing şu testi önermiştir; bir kişiyi, bir odada bilgisayar terminalinin başına koyalım ve terminali ikinci bir odaya bağlayalım. İlk odadaki adam ikinci odada kim veya ne olduğunu bilmeyecektir. İkinci odadaki terminalin başında oturan başka bir insan ya da bir sayısal bilgisayar olabilir. Birinci odadaki adam terminale sorular girer ve ikinci odadan yanıtları alır. İsteddiği her soruyu sormakta serbesttir. Örneğin, matematik problemleri ya da İngiliz edebiyatından sorular sorabilir veya politika hakkında bir tartışma başlatabilir. Kabloların diğer ucundakinin bir insan mı yoksa bir makine mi olduğuna karar vermek için beş dakikası vardır. Bu testin koşulları yapay zeka üzerinde çalışanlar tarafından hala onaylanan bir alıntı durumundadır. Yapay zeka uzmanları bilgisayarın kendisini meydana getirenleri başarılı bir şekilde taklit edebileceğini ispatlama hedefine olan bağlılıklarını hala korumaktadırlar.

Bu yönüyle, yapay zeka projesi eski bir geleneğin uzantısıdır. Yüzyıllardır, mühendislerin çabaları günümüzdeki projeler kadar ciddi olarak ele alınmamış olsa da insanları taklit edecek makineler ortaya çıkarmıştır. Zeki bilgisayar programlarının habercileri ilk elektromekanik aygıtlardır. Örneğin ellilerde, bir fizyolog olan Ian Grey Walter, yerdeki engelleri çevresinden dolaşarak aşan elektromekanik bir "kaplumbağa" tasarlamıştır. Bu kaplumbağanın atalarından biri de İspanyol teknoloji uzmanı Torres Quevedo tarafından yapılmış olan elektro mekanik bir satranç oyuncusudur. Bu tarihten önce ise, tümüyle mekanik otomatlar yapılmıştır. Bu otomatların 18. yüzyıldaki en ünlü örnekleri, Jacquet-Droz'un dolmakalem ve mürekkeple yazı yazan erkek çocuğu ve hareket eden parmaklarıyla harprikod çalan kız çocuğu, Vaucanson'un vakvaklayan ve yemek yiyebilen ördeğidir. 17. ve 18. yüzyıllarda, Avrupa'daki bahçeler akan sularla harekete geçebilen Neptün, Orpheus ve pastoral akımın örnekleri olan çoban ve peri heykelleriyle donatılmıştır. Aslında, hareket eden insan ve hayvan figürlerinin ilk örnekleri, 13. yüzyılda Avrupa'da icat edilen mekanik saatler üzerine çizilen figürlerdir.

Yapay zeka programları gibi, tüm bu hidrolik, mekanik ya da elektromekanik aygıtlar da, insan veya hayvan doğasının bazı yönlerini taklit etmek için yapılmış denemelerdir. Yapay zeka programcıları, bu ilk denemelerin insanın dışsal özellikleri ve kesintili hareketleriyle sınırlı olmalarından ötürü değersiz olduklarını belirtirler. Jacquet-Droz'un dış görünüşü insana benzeyen yazı yazan erkek çocuğu, bir kerede yalnız bir ya da bir kaç mesajı tekrarlayabildiğinden hiçbir önemi yoktur. Bu deneme, mühendisliğin veya bilimin önemli bir başarısı değil, üstün bir hüner gösterişidir. Dişliler problem çözme ve dil kullanımı gibi insan

doğasının temel niteliklerini ifade etmek için yanlış teknolojiydi. Elektromekanik aygıtlar bile, insan doğasını ifade edebilecek tümüyle elektronik bir teknolojiye yalnızca yol gösterebilirdi.

Bu iddia bir yönüyle doğrudur. Çünkü bilgisayarlar, bir saate ya da elektromekanik anahtar tablosuna göre daha esnek makinelerdir. Bununla birlikte, her teknolojinin metafor olarak kendine ait tuhaf bir gücü vardır. İnsan ve hayvan vücutlarının saatin mekanizmasıyla karşılaştırılması düşüncesi Descartes ve Leibniz gibi önemli düşünürler için bile etkileyiciydi. Descartes ve Leibniz'in felsefi tezlerinin mekanik örnekleri Jacquet-Droz ve Vauconson'un otomatları tarafından verilmiştir. Günümüzde yapay zeka uzmanları da buna benzer bir iş görmektedir. Oyuncak yapımcıları, o günün kusursuz mekanik teknolojisinin coşkusu, endüstri devriminin makinelerinde daha parlak bir ifadesini bulan bir coşkuyu dile getiriyorlardı. Aynı şekilde, yapay zeka programları da yeni elektronik teknoloji olanaklarının örnekleridir (Çakar, 1997).

2.1 Bilim Olarak Yapay Zeka

Doğal olarak, yapay zeka uzmanları kendilerini önceki yüzyıllarda Avrupa hanedanlarını eğlendiren oyuncak yapımcılarının modern meslektaşları olarak görmemektedir. Bilgisayarla yaptıkları çalışmaların oyuncak yapımının gerektirmediği bir çeşit eğitim ve entellektüel özen gerektirdiği şüphesizdir. Günümüz toplumu bilim ve teknolojiye, endüstri devriminin ilk zamanlarında olduğundan çok daha fazla bağımlıdır ve teknoloji uzmanlarına daha iyi bir toplumsal konum verilmektedir. Yapay zeka programcılarının çalışmaları diğer bazı bilgisayar uzmanları tarafından sorgulandığından toplumsal konumları konusunda kaygı duymaktadırlar. Onlar bilgisayar denilen saygın, güçlü, pahalı oyuncaklarla uğraşan oyuncak yapımcılarıdır ve hakettikleri toplumsal konumun kendilerine verilmesini istemektedirler. Yazılarında da yapay zekayı yeni oluşmakta olan bir kavrama (cognition) bilimi veya düşünen sistemlerin araştırma alanı olarak göstermeye çalışmaktadırlar. Yapay zeka uzmanları yalnızca yapay zekanın hafıza, öğrenme, dil ve çıkarım yapma (human inference) kuramları için kusursuz ve denenebilir bir model sağlayabileceğini iddia ediyorlar. Yine bu uzmanlara göre, iyi bir program salt metafor değildir. Aynı zamanda insan ve makine arasında bir karşılaştırmayı da ifade etmektedir. Gerçekten de, iyi bir program bir bakımdan insan düşüncesi hakkında ya da genel olarak insan ve bilgisayarın birer örnekleri olduğu kavrama süreçleri hakkında, işleyişini inceleyerek bilgi edinebildiğimiz bir insana benzer.

Bilgisayara bir yapay zeka programı yükleyelim ve çalıştıralım. Şimdi bilgisayar neyin bir modelidir? Elektronik beyin terimi bir zamanlar bilgisayar metaforunu adlandırmak için

kullanılan yaygın bir ifadeydi. Ama bilgisayar insan beyninin bir modeli midir? Bu soru, bilgi işlem tarihinin başlangıç evrelerinde de gündeme gelmiştir. Turing ve bu alanın diğer öncüsü John Von Neumann insan beyninin bir donanımı olan bilgisayarı, beyniyle karşılaştırmayı düşünmüştür. Daha 1940'larda, Warren Mc Culloug ve Walter Pitts elektronik modellerle insan sinir hücresi arasında analogi kurarak, bir sinir ağının matematiksel özelliklerini çıkarmışlardır. O zamanlar, sinir hücreleri hala elektriksel bilginin işlendiği oldukça basit birimler olarak görülmekteydi. Bir sayısal bilgisayarın beyindeki sinir ağına benzer bir işlev görebileceği düşünülüyordu. Ama bu umut gerçekleştirilemedi. Çalışmalar süresince konunun matematikselleştirilmesi konusunda önemli buluşlar yapıldıysa da insan beynini model alan ilk makineler bir oyuncaktan farksızdır. Bilim adamları beynin, "Dendritic" bağıntı ormanları ve karmaşık kimyasıyla, ayrı ayrı kısımlardan oluşan mantık bilişenlerinin basit bir ağı olmadığını da farketmeye başladılar. Bilgisayar ve beynin sihirli işlevlerini elektriksel sinyallerle gerçekleştirmesi gibi ilk bakışta parlak görünen bir benzerliğe rağmen, sayısal bilgisayarlar tek başlarına insan beyninin yeterli bir modeli değildir. 1950'lerin ortalarından bu yana yapay zeka uzmanlarının çoğu çalışmalarını fizyolojiden psikolojiye kaydırmışlardır. Günümüzde bilgisayar programlarının hedefi beynin kendisinden çok (bellek, konuşma yeteneği, çıkarım gibi) insan zihninin bazı yönlerini modellemektir.

Yapay zeka projesi gerçekte, bilimsel model kavramını diğer olgun bilimlerden ödünç almıştır. Bununla birlikte, bilgisayar modellerinin statüsü fizik ve kimyadaki modellerle aynı değildir. Fizikçinin modeli, elle tutulabilir varlıklar ya da motorlar değil, bir seri matematiksel denklemler ve matematiksel ifadelerdir. Bilim adamları, denklemlerin ne anlama geldiğini tartışmak konusunda kendilerine yardımcı olması ya da sonuçlarını sıradan insanlar tarafından daha kolay anlaşılır hale getirmek için sık sık metaforlara baş vururlar. Örneğin bir teldeki elektrik akımı bir borudan akan suya benzetilir. Başka bir örnekte atomun, elektronların gezegenler gibi çekirdeğin etrafında döndüğü minyatür bir güneş sistemi olduğu söylenir ya da bazen, Einstein'ın dört boyutlu evreninin üç boyutlu bir kürenin yüzeyiyle ortak olarak özellikleri üzerinde durulur. Yine de fizikçi için model, matematik ve onun yorumlanmasıdır; tanıdık bir fiziksel nesne ile karşılaştırma işi ikincildir. Şaşırtıcı olan bu fazlasıyla soyut matematik modellerin gerçek hayatta da geçerli olarak, bilim adamlarının olayları öngörmelerini ve kontrol etmelerini sağlamalarıdır. Galileo'nun dediği gibi doğanın kontrolünün gizemi bilim felsefecilerinin araştırma alanında kalmıştır. Burada belirtmek istediğimiz basit olarak, yapay zeka tarafından kullanılan bilgisayar modellerinin, fiziğin matematik modellerine benzemediğidir. Bilgisayar modelleri fizik modelleri gibi analitik

değildir; derin bir matematiğe dayanmaz, makinelerden ayrıldıklarında anlamlarını yitirirler ve hiçbir entellektüel inandırıcılık taşımazlar.

Fizikçinin ve yapay zeka programcısının bilgisayarı çalışmalarında nasıl kullandığını düşünün. Fizik için denklemler ve yorumları, bir kalem ve kağıttan daha fazlasını kullanmaksızın inceleyebileceği, değiştirebileceği soyut bir dünya oluşturur. Fizikçi sık sık denklemleri bir bilgisayar programına dönüştürme yolunu seçer. Modelin değişik varsayım ve girdilerle nasıl işlediğini görür. Yapay zeka uzmanı da çalışmasına makinenin başına oturmadan başlayabilir. Kalem ve kağıtla, zihnin bilgi birimlerini nasıl depoladığını, işlediğini ve yeniden yüklediğini göstermek için bir algoritma tasarlayabilir. Sonra bu algoritmayı bir programa çevirir, bilgisayarda dener ve bunları yaparken bilgisayarın kafasındaki, zihnin nasıl çalıştığına dair kavramsal çerçeveye daha sadık hale getirmeye çalışarak yeni incelikler ekler. Ama yapay zeka uzmanı ile fizikçi arasında çok önemli bir fark vardır; yapay zeka uzmanı kuramlaştırmasına zihnindeki bilgisayarla başlar. Denklemleri çok karmaşıksa ya da bilgisayara aktarılması olanaksız türdence fizikçi teorisini bilgisayara uygulanabilir hale getirmeyebilir. Bu durum, modeli tanım gereği bilgisayara uygulanabilir olan yapay zeka uzmanı için söz konusu değildir. Yapay zeka da, sayısal bilgiyle çalışan ve tanımlanmış programlara dönüşmeyen hiçbir şey, insan zihninin bir açıklaması olarak kabul edilmez. Başka bir deyişle, fizikçi için bir bilgisayar, modelini incelemeye yardımcı olabilir; yapay zeka programcısı için bilgisayar (özel bir programı çalıştırmak) modelin kendisidir. Bir yapay zeka programının komut listesini okumak da son derece anlamsız bir iştir.

Programın varlığı makineyle mümkündür. Elde edilen başarılar kadar yapılan hatalar, düşülen şaşkınlıklar ve kullanılan dilin uygun olmaması gibi durumlarda da program etkileyicidir. Bu nedenlerle yapay zeka projesi bilimsel bir modelden çok teknolojik bir metafora dayanır. Bu metaforu, el altında bir bilgisayar olmaksızın kullanmak da olanaklıdır. Gerçekten de psikologlar bugün sürekli olarak görsel "girdi" ve sözel "çıkıtı" dan, depolama ve yeniden yükleme için zihinsel stratejilerden, mesaj kodlama ve kod çözmekten metaforik (mecazi) olarak söz ediyorlar. Ama bu metaforların arkasında hep makinenin kendisi var. Toplumumuzun ihtiyacı olan teknolojiler konusunda pek çok şey öğrenildi: Tarımda kullanılan kimyevi gübre, böcek zehirleri ve fosil yakıtlar, taşımacılık için demiryolları, otomobiller ve uçaklar, ulusal güvenlik için titanyum. Günümüzde, insan zihnine bakışın bütünsel bir yönteminin kaynağı olan teknoloji de işte burada yatmaktadır. Bilgisayar ve bu teknolojinin oluşturduğu etki (heyecan) ortadan kaldırılırsa yapay zeka projesi de inandırıcılığını ve tehdit edici yönünü yitirecektir (Rich ve Knight, 1991).

2.2 Benzetim ve Gerçeklik

Yapay zekaya sahip bir bilgisayar, fizikçinin denklemlerinin, atomun doğasını açıkladığı anlamda, insan zihnini açıklamaz. Bunun yerine, zihnin bir taklidini oluşturur. Bu taklidin önemi, sebepleri hakkında düşünmeksizin insan zihniyle görünürde aynı sonuçları üretmesidir. Yapay zeka tanımını anımsayalım: "İnsanlar tarafından yapıldığında zeka gerektiren şeyler yapan makineler oluşturma bilimi." Sonuca yöntemden daha fazla önem veren bir tanım. Yapay zeka programcısı şu tanıma göre çalışır: Bir bilgisayar programı insan zihni gibi davranıyorsa bu program ciddiye alınabilir.

Yapay zeka karşıtlarından Searle bir yazısında iddiasını sağduyu düzeyinde ifade eder: "Hiç kimse bilgisayarla yapılan bir yangın benzetiminin bir mahalleyi yakıp yok edeceğini, ya da yağmur fırtınası benzetiminin bizi sırlıslıklam ıslatacağını düşünmez. Peki neden bazıları çıkıp, bilgisayarla benzetimi yapılan, anlama sürecinin gerçekten de herşeyi anladığını düşünmektedir? Zaman zaman, acı duyan veya aşık olan bilgisayarlar yapmanın çok çok zor olduğu söylenir, ama aşk ve acı, kavrama ya da diğer insana özgü yönlerden ne daha zordur, ne de daha kolay. Benzetim için sadece, doğru girdi, doğru çıktı ve ikisi arasında girdiyi çıktıya dönüştüren bir program gereklidir. Bilgisayar herşeyi bu özelliği kullanarak yapar. Acı, aşk, kavrama, yangın ya da fırtına durumlarında benzetimle, benzerini oluşturmayı birbirine karıştırmak hep aynı yanılmanın ürünüdür."

Gerçekten de, yapay zeka projesi, burada sağduyudan sapmıştır. Yapay zeka projesi, bilgisayar metaforunun kazandırdığı yeni ruh nedeniyle sağduyuyu terk etmek zorundadır ve tüm bilgisayar programcıları da benzeri bir sapmanın çekiciliğine kapılmıştır. Bilgisayarın doğasında var olan her kalıba girebilme özelliği yüzünden, programcılardaki eğilim, benzetimi, özdeşini elde etmeyle birbirine karıştırmak yönündedir. Makine genel olarak, fiziksel etki açısından olmasa da tasarım bakımından diğer makineleri ya da doğanın pek çok yönünü taklit edebilir. Bilgisayar bir jet motorunun çalışmasını, bir şehrin trafik akışını ya da bir bakteri kolonisinin üreme oranlarının benzetimini yapabilir. Bir dizi birbirinden ayrı olaya bölünebilen herhangi bir görüngünün (fenomen), bilgisayar tarafından az çok başarıyla benzetimi yapılabilir. Her önemli program bir benzetim, dış dünyanın bazı görünümelerini bilgisayarda yeniden oluşturmak için girişilen bir denemedir. Üstelik benzetim, kesin ve kanıtlanabilir bir çözüm isteyen matematikçinin ya da geleneksel filozofun bakışına değil, daha özgür bir bakış açısına dayanır. Benzetim, planlı bir deneme yanılma, bir seçeneği diğerine göre ayarlama, zararı en düşük düzeye indirme ve kazancı en yüksek düzeye çıkarma sorunudur. Programcı koşulları belirler ve sonra bir

dizi gerçekleşmesi olası durumu inceler. Y şehrinde bir rafineri inşa edersek, bu C ilçesinde hava kirliliğine yol açar mı? Rafineriden vazgeçersek ya da iki rafineri inşa edilirse, bu Y şehrinde işsizlik sorununu nasıl etkiler? Gerçek yaşamda gelecekle bu şekilde oynayamayız, ama bilgisayar dünyasında bunu yapabiliriz. Benzetimin doğduğu bakış açısıyla, gerçek dünyayı, benzetim programının çalıştırılmasından sonra ulaşılan durumlar dünyası olarak görmeye başlarız.

Yapay zeka projesi, insan zihninin, bilgisayarın bu ilgi çekici görünümüne dayanılarak benzetimi doğrultusundaki bir aşamasıdır. Bir yönüyle, yapay zeka programcısı oldukça pratiktir; yalnızca Çince sorulara Çince’de iyi yanıtlar veren bir program ister. Diğer taraftan, dış dünyanın "gerçekliğiyle" makinenin içindeki benzetim arasındaki, Çince’yi anlama ile bu anlamayı benzetim yoluyla elde etmek arasındaki farkı kasten görmezlikten gelir. Bilgisayarlar kendi küçük ve benzetimi yapılmış dünyalarının elemanları üzerinde işlem yapmakta giderek daha başarılı hale geliyor. Gittikçe daha fazla insan bilgisayarla çalışıyor ve benzetimin dayandığı derin nedenleri araştırmaktan çok, deneme yanılma, süreç ve sonuçları gibi kavramların oluşturduğu iş mantığını benimsiyor. Yapay zeka bu felsefi tartışmayı ancak karşıtlarını kendisinden uzaklaştırarak, filozoflarla onların kendi alanında çarpışmayı reddederek kazanabilir. Yapay zeka, daha sonra şu büyük felsefi soruya kendi yanıtını verecektir: Bilgi nedir? Bilginin, semboller üzerinde formal kurallara göre işlemler yapmak olduğu yolundaki yanıt, analitik felsefeciler için olduğu kadar daha eski felsefeciler için de oldukça yetersiz kalacaktır. Sorun, bilgisayarın sağladığı işlemsel terimlerle yeniden konulmalıdır. Fakat hiç şüphe yok ki, soru, geçmişte bir çok kez tekrarlanmıştır. Şimdi soruyu sorma sırası bilgisayar çağının teknoloji uzmanlarıyla, bu uzmanların diğer disiplinlerdeki müttefiklerindedir (Çakar,1997).

2.3 Felsefe Mühendisliği

O halde yapay zeka, bir simgesel mantık pratiğidir. Ama fiziksel bilimleri karakterize eden matematik ve deney arasındaki ilişkiye dayanmaz. Yapay zeka ile modern felsefe ve dilbilimi arasında da önemli bağlar vardır, ama yapay zeka, kavramlar arasında ayırım çizgileri çizmekte kullandığımız ayrıntılı dil analizi anlamında felsefe değildir. Yapay zeka, mantık, teknoloji ve felsefenin “felsefe mühendisliği” olarak adlandırılabilir ilgi çekici bir bileşimdir. "Felsefe mühendisliği" kavramında anlamı kuvvetlendirmek için zıt kelimeler bir araya getirilmiştir. Mühendisler fazlasıyla pratik insanlardır, ama elektrik mühendisliğinin büyük başarısı olan bilgisayar aynı zamanda makinelerin en felsefi olanıdır. Bilgisayar,

sinyallerin çip boyunca yollarını bulması için gerekli olan saniyenin çok küçük kesirleri üzerinde kafa yoran bilgi işlem çipi tasarım mühendisleri, zaman üzerine düşünen felsefecilere dönüştürür. Yine bilgisayar, bir matematik veya iş problemini adım adım programlanmış bir çözüme ayrıştırmaya uğraşan programcıları mantıkçılara çevirir. Bilgisayarın, mantık teoremlerini teller ve transistörler yoluyla somutlaştırmasındaki belirsizlik, felsefe mühendisliğinin kafasındaki belirsizliklerin de kaynağıdır. Yapay zeka çalışanları da tümüyle belirsizlik içinde ve felsefi insanlardır. Bazıları felsefeyi bilgisayar programlarına devretmeyi alçaltıcı bulmakta, diğerleri ise bilgisayarın felsefi tartışmayı yeniden canlandıracağını düşünmektedir. Her iki durumda da gelişme önemlidir. Bilgisayar tarafından oluşturulan gerçek insancıldan satranç programları veya öykü okuyan programlar değil, bizzat bilgisayar tasarımcıları, programcılar, bilgi mühendisleri ve yapay zeka uzmanlarıdır.

İnsanın kendi doğasını kavrayışı, bir yönüyle çağın teknolojisi tarafından belirlenir. Bu konuda en iyi tarihsel örnek, saatin ve saatin iç mekanizmasının, 17. ve 18. yüzyılda mekanikçi filozoflar üzerindeki etkisidir. Eski Yunan'dan başlayıp günümüze kadar uzanan Avrupa tarihinde ve şüphesiz diğer kültürlerde başka örnekler de bulunabilir. Geçmişte, filozoflar, teknolojiiden metaforlar alırlardı. Plato ve Aristo Yunan zanaatkarlarından, Descartes ve Leibniz de saat yapımcılarından metaforlar almıştı. Bugün de önemli makineler sayesinde, teknoloji uzmanları filozofun alanını işgal etmiştir. Bilgisayar uzmanı, epistemoloji, semantik ve psikolojinin soracağı soruları belirlediğini söyleme cesaretine sahiptir. Teknolojinin kendisi kadar eski olan teknoloji yoluyla düşünme pratiği kültürümüzde hiçbir zaman baskın düşünme yöntemi olmadı. Ancak, yapay zeka programcısı, bilgisayarı birçok modern makineden biri olarak düşünmeyi bırakmamız ve bilgisayarı insan doğasının doğru teknolojik yansıması olarak görmemiz gerektiğini önerir.

2.4 İnsan Zihninin Bilgisayara Yansıtılması

Başka bir bakış açısıyla insanın bilgisayara yansıyan görünümünün en çarpıcı özelliği, insan düşüncesinin gizemli yapısını formal semboller ağına indirgemesidir. Bilgisayar, çoğunlukla büyük bir başarıyla, tüm problemleri bu gibi sembol kalıplarına indirger. Ama buradaki düşünce, insan zihninin bir ürünü ya da en azından bir buluşu olan formel mantığın şimdi yapay zeka uzmanları tarafından insan zihnini açıklamak için kullanılmasıdır.

Bilgisayar metaforunda düşünme, bir süreç, programlar yoluyla gelişen bilgisayarın sistematik ilerlemesidir. Ama geçmişe ait düşünceler, anılar ve duyular düşünen bilgisayarımızın üzerinde iş gördüğü veriler olduğundan, makinenin içinde statik olarak

ifade edilmek zorundadırlar. Elektronik veriler, yalnızca bir ve sıfır dizileri olan "ikilik birimler" ve enformasyon "bitlerinde" oluşur. Anılar ve düşünceler, bu gibi dizilerle nasıl ifade edilebilir? Bitlere, onları anlamlı kılacak bir yapı, belirli bir sıra verilmeli ve içinde anlam kazanacakları bir bütünlük sağlanmalıdır. Gerçekten de uygun veri yapısını belirlemek, bilgisayarın birleri ve sıfırları nasıl yorumlayacağına karar vermek programcının başlıca görevidir. Bu, yapay zeka programcısı kadar matematikçi ve şehir planlamacısı için de geçerlidir. Hepsi, verilerini düzenlemenin uygun yollarını bulmak zorundadır. Bununla birlikte yapay zeka programcısı, zihnindeki düşüncelerin ve duyumların karşılıklı etkileşimini yansıtacak olağandışı veri yapılarını bulmakla yükümlüdür. Yapay zeka programcısı, bilgisayarın, birbirinden ayrı veri elemanlarını birbirine bağlayabilme yeteneğinden yararlanır. Çeşitli yöntemlerle makinenin içindeki bir dizi bit, diğer dizinin yerine, bu dizi de diğer dizilere ait bilgi taşıyabilir. Bilgisayarın belleği veri elemanlarının kasabalar, göstericilerin (pointer) bir kasabayı diğerine bağlayan otobanlar olduğu bir karayolları haritası gibi, düzenlenebilir. İnsan zihnine benzer bir yapı kurmaya uğraşan yapay zeka programcısının göstereceği ustalık, kasabalar ve otobanların, veri elemanları ve elemanlar arasındaki bağlantıların doğru bileşimini bulmaktır.

Örneğin, programcımızın makinesine alışverişe çıkma gibi yaygın bir günlük etkinliği öğretmeyi istediğini varsayalım. Programcı kurduğu yapıyı bir kağıda çizerek başlayabilir. En üste bir çember ve içine alışveriş yazar. Sonra alışveriş çeşitlerini düşünür. Yiyecek, giyecek, otomobil v.b. Her çeşit, kendi çemberine sahiptir ve her çember ilk çizilen çembere bir çizgiyle bağlanmıştır. Programcı her alışveriş çeşidini alt bölümlere ayırarak ilerler. Alışveriş yapan, yiyecek alışverişi için arabayla markete gider, bir sepet alır, sepeti sebze ile doldurur v.b. Çemberler çoğalmaya ve çemberleri birleştiren çizgiler çapraz hatlar çizmeye başlar. Alışveriş yapan, markete olduğu gibi giyim mağazasına ya da eczaneye de gidebilir. Bu yüzden burada çeşitli çizgiler birbirine yaklaşır. İşlem, etkinlikleri daha küçük alt etkinliklere ayırarak devam eder. Arabayla markete gitmek için araba çalıştırılır, garajdan çıkarılır ve ana caddeye döndürülür vb. Programcının daha önce alışveriş sırasında edinilmiş deneyimleri de kapsamak istediğini varsayalım. Bu deneyimler, büyüyen ağda uygun yerlere bağlanan başka çemberlerle temsil edilebilir. Programcının kağıdı, çemberlerin etkinlikleri simgelediği ya da denetimler ve kesişen çizgilerin bu etkinlikler ile deneyimler arasındaki birleştirici ve mantıklı bağıntılar olduğu alışveriş etkinliğinin bir yol haritası haline gelmiştir. Şu ana kadarki adım, yapıyı bilgisayara aktarmaktır. Şüphesiz, yapay zeka programlarında, yol haritaları üzerinde yapılan düzenlemeler; verilen örnekteki yol haritası üzerinde yapılan düzenlemeye göre çok

daha karmaşıktır. Ama tüm yapay zeka programları, bir sayısal bilgisayarın bu tür bilgiyi işleyebileceği başka bir yöntem olmadığından, çemberler ve çizgiler, veri elemanları ve göstericiler prensibine dayanırlar.

Çemberlerin her birinde ne vardır? Çemberlerin içinde "alışveriş", "otomobil" ya da "sebze" sözcüğünü simgeleyen bir etiketten ya da bitlerden başka birşey yoktur. Her çember kendi başına anlamı olmayan formal bir simgedir. Bu, bilgisayara uyarlanmış yol haritasında anlam, tamamen bu simgeler arasındaki bağlantılardadır. Bu simgeler ağı ancak, ağ üzerinde işlem yapan programlar çalışırken anlamlı hale dönüşür. Örneğin, bir markette neyin alınabileceği ya da alınamayacağı hakkında soruları yanıtlayan bir program, bu simgeler ağını anlamlı kılar. Yapay zekanın düşünen programları, bir simgeden diğerine atlayarak kolayca bağlantıları izler, bu sırada da ağı değerlendirir. Ağ üzerinde değişiklik yapar ve ağı genişletirler. Ama programlar çoğunlukla ustalık gerektirirler ve hatta ustalığın zaferidirler. Yapay zeka uygulamaları, sadece mantıksal ilkelere dayanan uzman sistemlerden (hastalıkları teşhis eden sistemler), bir muhafazakar seçmenin politik görüşlerini benzetim yoluyla elde eden programlara kadar uzanan bir çeşitliliğe sahiptir. Yapay zeka uzmanları, uygun yapı, sembollerin seçimi ve programlama stratejisi üzerinde şiddetli anlaşmazlıklara düşebilirler. Bazıları katı bir hiyerarşiyi, diğerleri daha dağınık kalıpları kullanmayı tercih ederler. Bazıları, çok karmaşık ağlar üzerinde yavaş bir şekilde ilerlemeye dayanan stratejileri seçerler, diğerleri programların daha karmaşık, veri yapılarının daha basit olduğu biçimler üzerinde dururlar.

Bununla birlikte, her yapay zeka programında zihin bir grafiktir ve gizli ya da söze dökülmemiş olan herşey bu grafiğin çatlakları arasında kaybolup gider. Heyecanlar ve duygular da mantıksal kavramlar kadar kolay bir şekilde ağına parçası olabilir. Ama bunlar da grafikteki diğer elemanlara açık bağlantılarla bağlı olmak zorundadırlar. Bu, insan zihninin kavranışında devrimci bir yoldur. Şüphesiz, devrimden tek başlarına bilgisayar ve yapay zeka sorumlu değildir; bu alanda, antropoloji ve dilbilim alanındaki yapısalcı yaklaşımlar, davranış psikolojisi ve sembolik mantığın da önemli katkıları olmuştur. Bilgisayar, 20. yüzyılda birçok düşünsel ekolün yöneldiği teknolojik odağı oluşturmuştur. Bu gelişmelerin ortaya çıkardığı sonuç, kolayca gözardı edilemeyecek bir psikoloji bilimidir.

Bir an için yapay zekanın oluşturduğu psikoloji bilimiyle, belki de son yüzyılın en etkili psikoloji bilimi olan psikanaliz arasındaki ayrımlar dikkate alınsın. Psikanaliz insan eylemlerinin arkasındaki derin nedenleri bulmak üzere dış görünüşün altında yatan

gerçekliği araştırmak için eski yöntemlerle yapılan analizdir. Psikanalizin amacı, bastırılmış çocukluk anılarını açığa çıkarmaktır. Psikanalist, insan zihnini, bir bilgi işlem birimi olarak değil, tersine içgüdüsel gücün derin bir kaynağı ve tutku ile akıl arasında el değiştiren bir savaş meydanı olarak açıklar. Freud'un yazılarında, içgüdülerin ve idin hiçbir zaman tümüyle aydınlatılamaması ve rasyonel hale dönüştürülememesi nedeniyle, psikanalizin sona ermeyecek bir süreç olduğu iması vardır.

Bu sınırsız ve çoğunlukla karamsar bakış açısıyla yapay zeka uzmanının bakış açısı arasında büyük bir karşıtlık vardır. Bilgisayar programındaki simgeler ağı, bu simgeler herhangi bir derinliği ifade etmediğinden, derin insan güdülerini açığa çıkaramaz. İnsanı bir bilgi işlemcisi ya da semboller üzerinde uğraşan bir varlık olarak düşünmeye başladığımız anda bu tür bir derinlik duygusu ortadan kalkar. Semboller üzerinde işlem yapma oyunu, programcı için ustalık isteyen ve gerçekten sinir bozucu olabilecek bir süreçtir. Ama karşılaşılan sorunlar psikanalizin yüzyüze geldiği sorunlardan tümüyle farklıdır. Bu farklılıkların tek nedeni, psikiyatrisi ruh hastalıklarıyla, olağandışı ve anormal zihinlerle uğraşması değildir. Yapay zeka programcıları da, bazen bilgi işlemin özel, belki de patolojik bir uyarlaması olarak gördükleri şizofreni ve paranoya benzetimlerini denerler. Burada, alışveriş örneğimizde olduğu gibi belirsizliklerden tümüyle uzak olan semboller ağı ve birbirini izleyen durumların işlemsel tanımı, insan zihninin gösteriminde, derinlik ve nedensellik ilkesinin yerine geçen şeylerdir. Hatta, yapay zeka uzmanlarının zaman zaman derinlik fikrini pek de ciddiye almadıkları görülür. Marvin Minsky, "bana göre, zeka, saygı duyduğumuz, ama anlamadığımız eylemlerin karmaşıklığından biraz daha fazlasını ifade etmektedir." diye yazar. Matematikteki derinlik sorunu da benzeri bir yön taşır; teoremin ispatı anlaşıldıktan sonra teoremin içeriği önemsiz hal gelir. Gerçekte, bilim ya da psikolojide derinlikli kavrayış, gizemli olan yönleri hiç de ortadan kaldırmaz. Bir fizikçi, atom altı parçacıkları daha iyi anladıkça, ilişki kurduğu dünyaya yabancılığı da artar ve daha çoğunu araştırmak zorunda kalır. Freudcu psikologlar da, zihni daha derin bir şekilde inceledikçe, giriştikleri işin sonsuz karakterlerinin daha iyi farkına varırlar. Fakat yapay zeka uzmanının deneyimi Minsky'nin betimlediği gibidir. Karmaşık insan deneyimleri, programlarının işleyebileceği bir veri yapısına dönüştürüldüğünde, yapay zeka uzmanı işin en derin kısmına ulaşmıştır bile.

Yapay zeka uzmanları yaptıkları işte oldukça ilerlediler. Nasıl herhangi bir icat belli bir oranda mucidinin zeka ve karakterini yansıtıyorsa, bilgisayar da insan doğasının aynasıdır. Fakat, bilgisayar kusursuz bir ayna değildir; insanın belli yeteneklerini (bilgi işlem olarak

kolayca karakterize edilen yeteneklerini) vurgulayarak ve diğerlerini gözardı ederek bakışımızı etkiler ve belki de çarpıtır. Herşeye rağmen insan zihninin ve bilgisayarın karşılaştırılması; tüm çarpıklıkları ve doğru yönlerine rağmen etkileyici olmaya devam etmektedir. Bilgisayarların gerçekten düşünüp düşünemeyeceği sorusu üzerindeki kısır tartışmaya katılmak zorunda değiliz. Bunun yerine, metaforun hangi yönlerden uygun olduğu, hangi yönlerden başarısızlığa uğrayabileceğini sorabiliriz. İnsan zihnine bakışımız, şairlerin ve filozofların yapıtları ve geçerli teknik metaforlar gibi toplumsal standartlara bağlı olarak, çağdan çağa değişir. Şüphesiz çağımızda yapılması gereken, bilgisayar metaforunun ne tam reddine ne de kör bir kabullenişine izin veren bir kavram olan, yeni elektronik teknolojinin gereklerine ayak uydurma zorunluluğudur (Bolter, 1989).



3. ÖĞRENME

Öğrenme, yapay zeka içindeki en önemli araştırma konularından birisidir. Genel anlamda öğrenme zeki sistemlerdeki bilgi ve yeteneklerin artışı olarak tanımlanabilir. Öğrenme işlemi yeni tanımlamaları, belirlemeleri, sağduyuyu, teknik ve teorik bilginin elde edilme işlemini içine alır. Ayrıca yeni tekniklerin gelişmesi, bilginin organizasyonu ve etkili bir temsil yeteneğini de içine almak zorundadır.

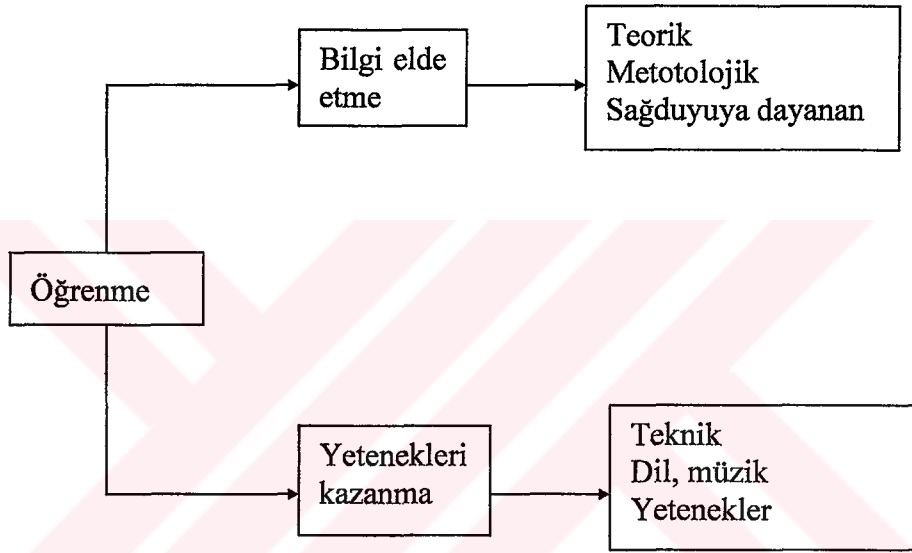
Simon (1983) öğrenmeyi şöyle tanımlamıştır: Öğrenme, aynı örnek kütledeki benzer ya da farklı işlerin tekrarı durumunda bu işlerin daha etkin ve verimli olarak yapılmasına izin veren sistemsel bir değişimdir. Bu değişim geri çevrilemeyecektir ve böylece öğrenme hemen ortadan kalkmayacaktır. Simon, öğrenme ve keşfetme arasındaki ilişkiye dikkat çekmektedir. Ayrıca öğrenme ve anlama arasındaki bağlantıdan söz etmekte ve bunların tamamının doğal dil problemini içine aldığını belirtmektedir. İnsanlar öğrendiklerinin çoğunu diğer insanlardan öğrenirler. Aradaki bağlantı ise doğal dil ile sağlanır.

Michalski (1986) ise öğrenmeyi, elde edilen tecrübelerin inşasının ya da değiştirilmesinin temsil edilmesi şeklinde tanımlamıştır. Bu tespitler, Simon'ın "performans" kriterinden Michalski'nin "tecrübenin temsil edilmesi" ölçüsüne kadar iyi bir referans noktası teşkil eder. Diğer taraftan Lenat, öğrenmenin göstergesi olarak "sinerji" prensibini önermiştir. Yani öğrenmenin, anlamanın ve zekanın birbiriyle ilişkili kavramlar olduğunu söylemiştir.

3.1 Makine Öğrenmesi

Öğrenme işleminin bilgisayarda modellenmesi makine öğrenmesi olarak tanımlanabilir. Makine öğrenmesi araştırmalarında iki temel düşünce vardır; kendi kendine anlayabilme ve öğrenebilme kabiliyetini bilgisayarla sağlama. Uzun zamandan beri yapay zeka araştırmalarının hedefi düşünebilen bilgisayar sistemleri yapabilmektir. İnsan öğrenmesinin yeterince anlaşılması sonucu makine öğrenmesinde yeni ilerlemeler olacak ve yeni bilimsel hedeflere ulaşılabilecektir. Makine öğrenmesinin temel bilimsel amaçlarından birisi de alternatif öğrenme mekanizmalarını araştırmak ve bulmaktır. Bunun dışında öğrenci ile bir iletişim sağlamak, çalıştırma verilerinin sisteme uygulanabilmesini ve birçok alanda tekniklerin genel uygulanabilirliğini keşfetmektir. Bu sistemlerin inşası ve test edilmesiyle birlikte bunların etkinliği ve limitleri belirlenebilir. Makine öğrenmesinin diğer bir hedefi de zeki araştırma yardımcıları ya da zeki kişisel yardımcılar inşa etmektir. Öğrenmeye, bilginin

elde edilmesi ve yeteneklerin mükemmelleştirilmesi olarak iki değişik şekilde bakılabilir. Bilginin elde edilmesi oldukça zor ve ciddi bir işlemdir. Bilginin elde edilmesi teknik, metodolojik ve teorik bilginin elde edilmesinin yanında sağduyunun da elde edilmesini kapsar. Bilgi tespitler, modeller, şemalar ve kurallar şeklinde elde edilir. Yetenek kazanma ve geliştirme, otomatik ve kavramaya dayanan yeteneklerin pratik yapılarak kademe kademe geliştirilmesini kapsar, mesela bisiklete binmeyi öğrenmek gibi. Yetenek geliştirme bilgi elde etmenin tersine çoğunlukla daha düşük kavrama seviyesinde yer alır. Carbonell (1987) bilgi elde etmenin yapay zeka araştırmalarının alanında olduğunu, yetenek elde etmenin ise alt sembolik bir proses olduğunu söylemiştir. Şekil 3.1’de bu durum şematik olarak gösterilmiştir.



Şekil 3.1 Öğrenmenin temel şekilleri

3.1.1 Makine Öğrenmesinin Tarihi Gelişimi

Makine öğrenmesine ait araştırmalar üç aşamada gelişmiştir.

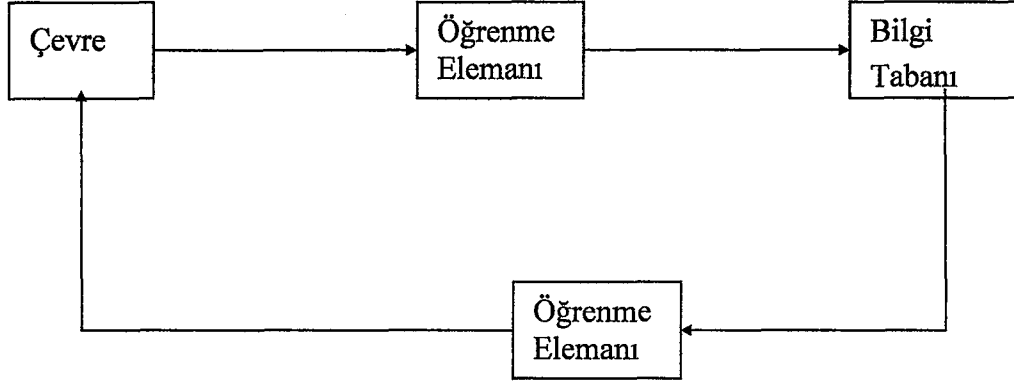
- Yapay sinir ağına dayalı modelleme teknikleri
- Sembolik öğrenme
- Bilgi ağırlıklı öğrenme

Öğrenen sistemlerin belirli bir ilk yapıları yoktur. Yapay sinir ağlarının inşası "tabula rasa" yaklaşımını temel alır ve bunların ilk yapıları tamamıyla veya kısmen tesadüfidir. Bu sistemlerde öğrenme sistem içinde yayılan sinyallerin yeni olasılıklarının artımlı değişmelerinden ibarettir. Bu sistemlerde uygulanan bir alıştırma seti sonucunda, sistem üzerinde geri besleme yapılarak sistemin kendisini düzenlemesi istenir ve sistem bu

düzenlemelerle optimum sonuca ulaşır. Yapılan ilk yapay sinir ağı çalışmalarında sınırlı sayıda başarı elde edilmiştir. İlk çalışmalar perceptron (algılayıcı) tipi öğrenme olduğundan çok önemli zorluklarla karşılaşmıştır. Fakat daha sonra yapay sinir ağı çalışmaları hızla ilerlemiş ve önemli başarılar elde edilmiştir. Makine öğrenmesinin tarihi gelişim sürecinin ikinci aşamasında yani 1960'lı yıllarda kavram kökenli öğrenme ortaya çıkmaya başlamıştı. Bu metot istatistik metotlar yerine, semantik ağ ve yüklem mantığı (predicate logic) gibi teknikleri kullanmıştır. Bu sistem, verilen bir kavram setini sembolik bir gösterimle öğrenir ve bu kavramların örnekleri ve ters örnekleri analiz edilir. Bu tür çalışmalar insanın kavram elde etmesi üzerindeki araştırmaları da içine almıştır. Öğrenme işlemi üzerindeki bu araştırmalar aşama aşama devam ederken öğrenmenin karmaşık bir işlem olduğu anlaşılmıştır. Sonuç olarak herhangi bir bilgi, başlangıcı olmaksızın yüksek seviyeli kavramsal bir öğrenmeyi, yüksek seviyeli kavramsal bir öğrenmeyi öğrenen sistemlerden beklemek gerektiği ortaya çıkmıştır. Bu çalışmaların üçüncü aşaması 1970'lerin ortasında başladı. Bu aşamada örneklerden öğrenilen ve izole ederek yapılan öğrenme çalışmaları araştırmacıların ilgisini çekmiştir. Dolayısı ile bilgi zengini sistemleri baz alan bir çok öğrenme metodu araştırılmaya başlanmıştır. Bunun dışında tümevarımla öğrenme (inductive learning), gözlemle öğrenme (learning from observation), keşifle öğrenme (discovery) ve kıyaslamayla öğrenme (learning by analogy) gibi metotlar da araştırmacıların ilgisini çekmiştir (Barrow, 1989).

3.2.2 Öğrenmenin Elemanları

Simon, 1983 yılında kullanışlı bir öğrenme sistemleri modeli üzerinde çalışmalar yapmaya başlamıştır. Simon'ın öğrenme modelini esas alan öğrenme elemanları şunlardır: Çevre, öğrenme birimi, bilgi tabanı ve performans birimidir. Çevre biriminin görevi öğrenme birimine bilişim sağlamaktır. Bu bilişim bilgi tabanında iyileştirmeler yapmak için kullanılmaktadır. Performans birimi de kendi görevlerini yerine getirmek için bu bilişimi kullanır. Aktivite sürerken elde edilen bilişim öğrenme birimine geri besleme olarak sunulabilir ve çevre birimini değiştirebilir. Şekil 3.2'de Simon'ın öğrenme modeli görülmektedir.



Şekil 3.2 Simon'ın modelini esas alan öğrenme elemanları

Bu basit model, öğrenme sistemlerini onların elemanlarının ilişkilerine göre sınıflandırmamıza izin verir. Sağlanan bilişimin genellik derecesi ve kalitesi öğrenme sistemlerinin tasarımında en önemli faktördür. Genel yada soyut bilgi geniş bir problem sınıfıyla ilgilidir. Özel bilgi ise bir tek probleme aittir. Buradaki öğrenme işlevinin amacı genellik seviyeleri ve boşluk arasında bir bilişim oluşturmaktır. Bu bilişim çevreyle sağlanır ve herbir seviyedeki performans birimi işlevini yerine getirmek için bilişimi kullanılır. Öğrenme birimi genellik seviyeleri ve boşluk arasında köprü kurabilmek için hipotezleri şekillendirebilmek zorundadır. Ayrıca hipotezleri değerlendiren gerekirse düzeltebilen bazı geri beslemeleri de (feedback) kabul etmek zorundadır. Dolayısıyla öğrenen sistemlerin görevlerini yerine getirebilmeleri için, kendilerine verilen bilgileri detaylarıyla öğrenmeleri gerekmektedir. Bunun yanında daha geniş yapılarda performans gösterebilmesi için özel bir bilişimi de genelleştirebilmek zorundadır. Makine öğrenmesi birkaç değişik bakış açısıyla farklı şekillerde sınıflandırılabilir. Bu sınıflandırmaları bilgi gösterimi, uygulama alanları, öğrenme seviyeleri ve öğrenme metotlarına göre yapabiliriz. Burada ana kriterimiz öğrenmeyi sonuç çıkarma stratejisine göre sınıflamaktır. Bazı öğrenme stratejileri çıkarsamayı oldukça fazla kullanırken bazıları kullanmaz. Burada çıkarsama miktarı performansın artma kapasitesini belirler. Böylece öğreticinin yükü ve dışsal çevredeki yük azalır. Genel öğrenme stratejileri şunlardır:

- Ezberleme ile öğrenme (Rote learning)
- Eğitimle yada bilgilendirme ile öğrenme (Learning from instruction)
- Tümdengelimle öğrenme (Deductive learning)
- Tümevarımla öğrenme (Inductive learning)
- Kıyaslama ile öğrenme (Learning by analogy)

Ezberleme ile öğrenmede yapılacak görevle ilgili her seviyede bilgi çevre birimi tarafından sağlanır. Böylece herhangi bir çıkarsamanın yapılmasına gerek kalmaz. Ezberleme ile öğrenme; programlama ile öğrenme ve hafızalama ile öğrenme olmak üzere iki farklı şekilde incelenebilir. Bilgilendirme yada eğitimle öğrenmede çevreden genel yada soyut bilgi elde edilir. Burada öğrenme birimi birkaç detayı tamamlamak için çıkarsama yapmak zorundadır. Makine öğrenmesinin görevlerinden birisi de eğitimi ya da bilgiyi kabul edebilen sistemler inşa edebilmektir. Ayrıca bilgileri depolayıp gerektiğinde etkin bir şekilde kullanabilmelidir. Öğrenme sistemleri bu stratejiyi beş aşamada uygular;

- Bir uzmandan gerekli bilgilerin alınması
- Verilen bilgileri kodlanacak şekle getirmek
- Verilen bilgileri uygun şekilde kodlamak
- Bu bilgileri mevcut bilgi tabanıyla birleştirmek
- Performans biriminin işlem sonuçlarını ölçmek

Tümdengelimle öğrenmenin de belirli bir stratejisi vardır. Bu strateji, bilginin yeniden formüle edilmesini, bilginin derlenmesini, makro operatörlerin çalışma kriterlerini ve açıklama tabanlı öğrenmeyi içine alır.

Gözlem ve keşifle öğrenmeye denetimsiz (unsupervised) öğrenme de diyebiliriz. Bu tip öğrenme metotları, kantitatif yada kalitatif metotları, teori formasyonunu, teori revizyonunu ve sınıflandırma kriterine göre sınıflandırmayı içerir. Gözlem ve keşifle öğrenme değişik metotları birleştirir. Tümevarım ve tümdengelim metotlarının birleştirilmesi buna bir örnektir. Tümevarımla öğrenme, örnek setinden yararlanarak öğrenmeyi temel alır. Bu şekilde bir öğrenme daha fazla çıkarsama gerektirir. Gözlemle öğrenme, çevre birimi ile etkileşim derecesine göre iki alt sınıfa ayrılır: Aktif tecrübe ve pasif gözlem. Kıyaslamayla öğrenmede bilişim benzer işleri yapan ilişkili bir çevreden sağlanır. Bu öğrenme sistemleri ilgili benzerlikleri, benzerlik kurallarını ve hipotezleri bulmak zorundadır. Kıyaslama ile öğrenmede değişik tipte çıkarsamalar gerekir. Bu öğrenme tekniği tümevarımla ve tümdengelimle öğrenme metotlarını birleştirir. Parçalardan bütüne giderek benzer bir yapı bulma tümevarıma dayanan bir çıkarsamayı içine alır. Halbuki benzerlik taslağının önceden şekillendirilmesi tümdengelimle dayalı bir çıkarsamayı gerektirir (Kocabaş, 1991; Langley ve Simon, 1995).

3.2 Değişik Seviyelerdeki Öğrenme Metotları

Öğrenme metotlarını üç seviyede inceleyebiliriz.

1. Bilgi seviyeli öğrenme (Knowledge-level learning)
2. Sembolik seviyeli öğrenme (Symbol level learning)
3. Makine seviyeli öğrenme (Device level learning)

Bu sınıflamada, öğrenme metotları sonuç bulma ve temsil seviyelerindeki uygunluğa göre incelenebilir. Bilgi seviyeli öğrenmede, herhangi özel bir sembol yada makine (device) referans alınmaksızın öğrenme metodu ve sonuçları tesbit edilebilir. Sembolik seviyeli öğrenmede, öğrenme metodu ve sonucu özel tipte sembol veya veri yapısına göre, özel bir makine referans alınmaksızın öğrenme metodu ve sonucu belirlenebilir (Kohonen, 1988). Makine seviyeli öğrenmede, öğrenme metodu ve sonuçlar bir makine referans alınarak belirlenir. Soyutlama, kıyaslama, benzerlik tabanlı genelleme, açıklama tabanlı genelleme ve kavramsal sınıflama bilgi seviyeli öğrenme kapsamı içindedir. Sınırlayıcılar ve genetik algoritmalar sembol seviyeli öğrenmenin kapsamı içine girer. Bağlayıcı (connectionist) sistemler ve yapay sinir ağları makine seviyeli öğrenme olarak kabul edilirler. Bu sınıflandırmayı kesin bir sınıflandırma olarak kabul edemeyiz. Fakat bu sınıflandırma değişik öğrenme metotları arasındaki farklılıkları açık olarak ortaya koymaktadır. Zeki sistemlerde bu farklı seviyeli öğrenmeler başarıyla tasarlanabilirler. Carbonel ve Langley (1987) değişik öğrenme metotları bulmuşlardır. Bunlar örneklerle öğrenme (learning from examples), arama metodu ile öğrenme (learning search heuristic), kıyaslama ile öğrenme (learning by analogy), gramer elde etme (grammar acquisition), keşifle öğrenme (learning by discovery). Carbonel öğrenmeyi; öğrenme stratejileri, öğrendiği bilgi tipleri ve uygulama alanlarına göre üç sınıfta incelemiştir (Carbonel, 1987; Langley, 1995).

3.2.1 Bilgi Seviyeli Öğrenme Metotları

Bu metotlar benzerlik tabanlı öğrenme (Similarity-based learning), açıklama tabanlı öğrenme (explanation based learning) ve kavramsal sınıflama (conceptual clustering) olarak ele alınmıştır.

3.2.1.1 Benzerlik Tabanlı Öğrenme

Açıklama tabanlı öğrenme bir sınıftaki örnekler arasındaki paylaşılmış özellikleri araştırır.

Aynı sınıfa ait değişik örneklerin benzer sebep ve açıklamalarının sebebini arar. Ayrıca bu öğrenme şekli farklı özellikleri dikkate alarak örnekler arasındaki farklılıkları genelleştirir ya da kavramları formüle ederek aynı işi yapar. Bu metot örnek altsetinden kavram belirlemede tümevarıma dayalı sapmaları kullanır. Bu metot çalıştırma örneklerindeki benzer özellikleri araştırmayı temel alır. Buna da benzerlik tabanlı genelleştirme denir. Mesela $a_1, \dots, a_n$ bireyler olsun (Kuzgun, Güvercin, Devekuşu), C bireylerin ait olduğu sınıf olsun (Kuş), P ise bu bireylerin sahip olduğu özellikler seti olsun P (tüylülük, iki ayak, kanatlar), P_c ortak özellikler seti olsun.

$$P_c \in P$$

$$C(a_1), \dots, C(a_n)$$

$$P = [P_1, \dots, P_n] \quad \text{Bireysel özellik seti}$$

$$P_c = [P_i, \dots, P_k] \quad \text{Ortak özellikler seti}$$

Burada genelleştirilmiş kavram $C_g(x) = P_c(x)$ olarak belirlenir.

Benzerlik tabanlı öğrenme deneysel ve veri ağırlıklı bir öğrenme metodudur. Bu metot, doğru bir genelleştirme yapmak ve aramayı sınırlandırmak için çok sayıda alıştırma seti kullanır.

3.2.1.2 Açıklama Tabanlı Öğrenme

Açıklama tabanlı genelleştirme açıklayıcı bilginin önemini özellikle vurgular. Ayrıca sistemin daha önceki bilgilerini, olayların yüksek seviyeli kavramsal açıklamasını formüle etmek için kullanır. Benzerlik tabanlı genelleştirme çok sayıda alıştırma örneğine güvenir ve doğru bir genelleştirme yapabilmesi için aramayı sınırlamak amacıyla tümevarıma dayalı bir sapmaya ihtiyaç duyar. Açıklama tabanlı genelleştirme, alan bilgisine ve çalışmadaki kavram bilgisine dayandığından aramayı sınırlar. Açıklama tabanlı genelleştirmede bir tek örnek seti analiz edildikten sonra sistem bilgisine dayanarak verilen örnek tündengelimine dayanan bir doğrulamayla genelleştirilebilir. Açıklama tabanlı genelleştirme, örneklerin kavram belirlemesini nasıl tatminkar hale getirdiğinin açıklamasını, çalışma örneklerini analiz ederek sağlar. Açıklama tabanlı öğrenme şunları gerektirir :

1. Hedef kavram belirlenmesi
2. Çalışma setinin oluşturulması

3. Alan teorisinin belirlenmesi

4. Öğrenilen kavramların kesinleştirilmesi, uygun formun tesbiti ve işlem kriterlerinin belirlenmesi

Açıklama tabanlı genelleştirme sistemi, çalışma koşulları altında çalıştırma örneğinin bir kavram örneği olduğunu kendi kendisine açıklamak zorundadır. Bir kavramın tesbiti bir örneğin kavram örneği olması için gerekli koşulları ve gerekliliği belirler. Açıklama tabanlı genelleştirme sembolik anlamda operasyonel açıklamanın genelleştirmesinin kapsanması ilişkisi olarak karakterize edilebilir.

$L(f(C,e,T,0),E)$ veya

$E = \text{gen}(f(C,e,T,0))$

C = Hedef kavram O = operasyonel kriter

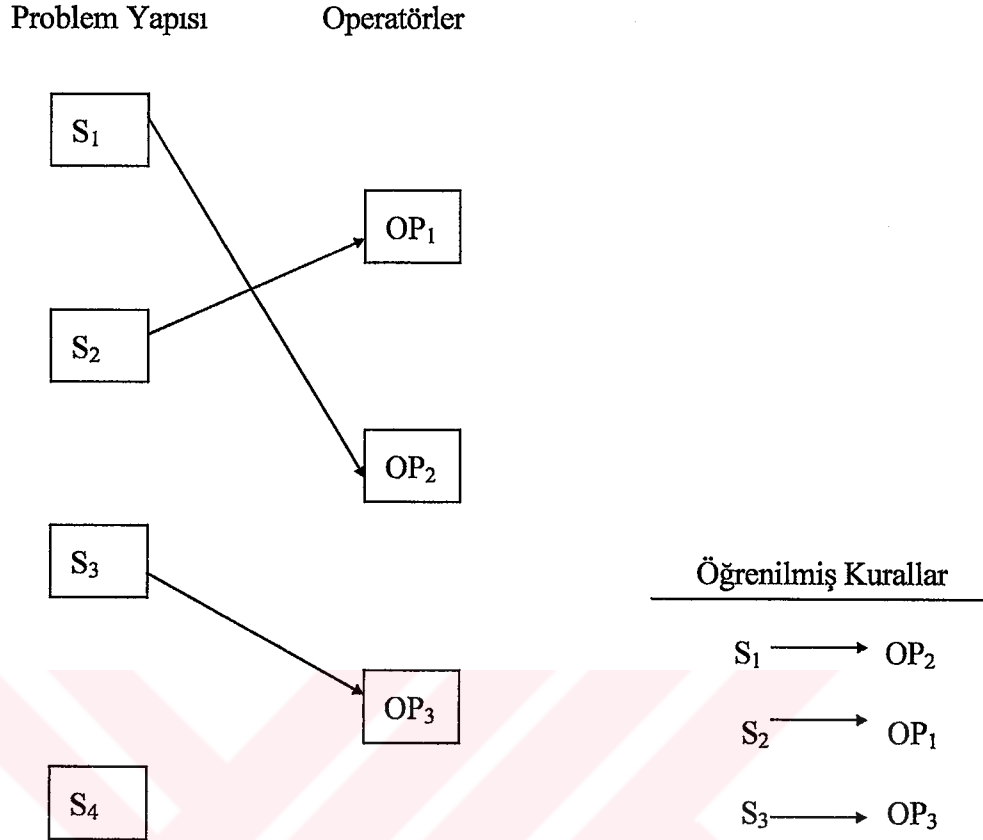
e = çalıştırma örneği E = açıklama

T = alan teorisi

Açıklamayla öğrenmenin sonucu genelleştirilmiş kavram belirlemesidir. Burada e çalıştırma örneği girilirse, sistem S açıklamasını bulur. C ise hedef kavramdır. S şöyle belirlenir,

$S = f(C, e, T, O)$

Açıklama tabanlı öğrenme benzerlik tabanlı öğrenmenin en önemli kısıtlamalarının üstesinden gelebilir. Bu öğrencinin alan bilgisinin, hedef kavram ve operasyonel kriter kabul etmesi varsayımı ile başarılır. Halbuki benzerlik tabanlı öğrenme bu verilerin hiçbirine güvenmez. Açıklama tabanlı öğrenmede, LEX sistemi gibi sezgisel algoritmalar da uygulanabilir. Bu sistem şekil 3.3’de görüldüğü gibi problem yapısını operatörle birleştirerek öğrenme işlemini gerçekleştirir.



Şekil 3.3 LEX'in sezgisel algoritma yapısı

LEX sistemini integral deyimini basitleştirmek için kullanabiliriz. Bu program bir operatörler setine sahiptir. Bu operatörler integral deyimlerinin üzerinde işlem yapabilecek kapasiteye sahiptir. Mesela OP_3 operatörü sabitleri integralin dışına çıkarır. Burada LEX'in hedefi integralin hangi sınıfa dahil olduğu ve integrale hangi kuralın uygulanacağıdır

$$R3 : \int r f(x) dx \Rightarrow r \int f(x) dx$$

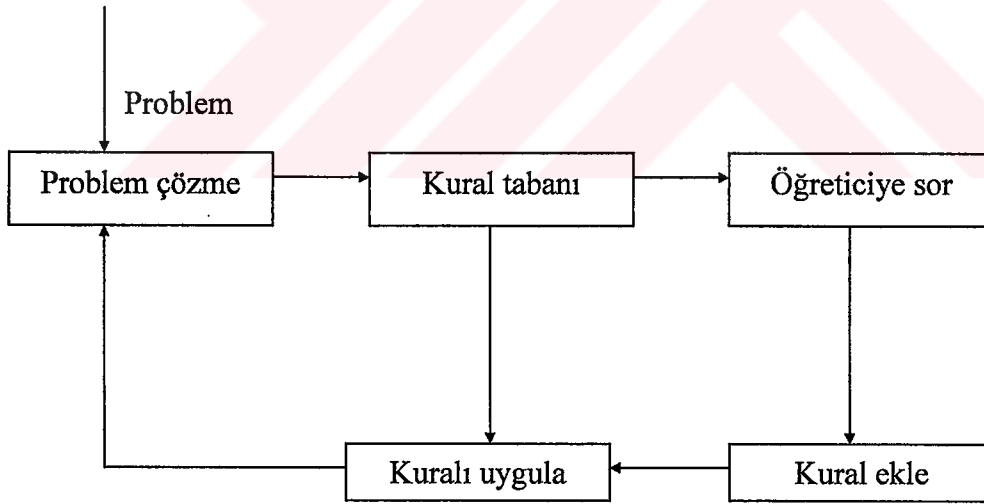
Mesela $\int 7x^2 dx$ verildiğinde, integral $7 \int x^2 dx$ şeklinde değiştirilir. $7 \int x^2 dx$ burada $\int x^2 dx$ 'nin operasyonel genelleştirmesini bulmak ve OP_3 ile birleştirmek zorundadır. Açıklama tabanlı öğrenmede en önemli kısıt, öğrenmenin çok güçlü bir biçimde öğrenicinin eski bilgisine bağlı olmasıdır. Bu, alan teorisindeki kuralların herbiri ile genelliğe şiddetle bağlı olacak özel bir çalışma seti için üretilmiş genelleştirme derecesinin bir sonucu olacaktır. Açıklama tabanlı öğrenme çok az bir problem yapısını içine alır. Bunlar :

1. Eksik teorili problemler
2. Kontrolsüz (Intractable) teorili problemler

3.Tutarsız teorili problemler

Bu sebeple, eğer alan teorisi eksik, kontrolsüz ve tutarsız ise açıklama tabanlı genelleştirme operasyonel kavram belirleme işlemlerini yapamaz. Açıklama ve benzerlik tabanlı sistemlerin birleştirilmesi bu sistemlerin karşılıklı zayıflıklarını azaltır. Açıklama tabanlı öğrenmenin biraz daha ıslah edilmiş bir şekli de episodik öğrenmedir. Bu metotta öğrenme sistemi problem yapılarını operatörlere nasıl birleştireceğinden başka, içinde operatörler kullanılan işlemlerin sırasını da belirler. Kısaca bu metot öğrenme ve problem çözmeyi birleştirir. Burada herbir kuralın bir yapısı vardır. Bu kurallar açıklama tabanlı genelleştirmeye hızla öğrenilir ve episodlar ile birleştirilir. Bu birleştirmelerle problemin çözüm kuralı öğrenilmiş olur (Mitchell ve Keller, 1986).

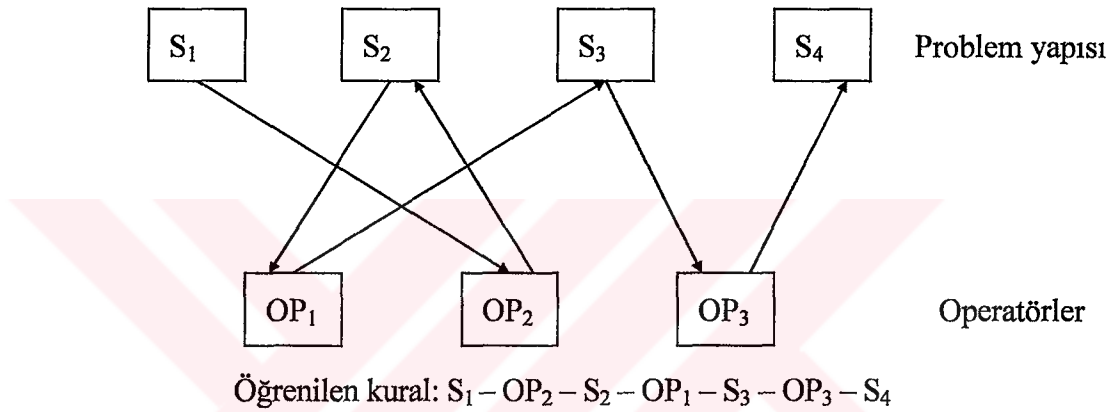
Porter & Kibler (1986), episodik öğrenmeyi prolog'da yazılmış PET denilen bir sistemde uygulamışlardır. PET, kuralları oluşturduğu zaman, kuralların herbirine puan verir. Kurallar, yüksek puandan düşük puana doğru sıralanır. PET çalışmaya başladığında problemi çözmek ve hedefe varmak için operatörler setini problemlere uygular ve PET çalışırken uygulanabilir kural arar. Tezat olan durumlar en yüksek puanlı kurallar ile tekrar çözülür. Şekil 3.4 PET'in kontrol stratejisini göstermektedir.



Şekil 3.4 Episodik Öğrenmede PET'in Kontrol Yapısı

PET ayrıca doğrusal eşitliklerin ve integrallerin nasıl çözüldüğünü öğrenmek için de uygulanmıştır. Bu sistem birkaç operatöre sahiptir. Bu operatörlerden ikisi eşitlikteki değişken terimleri birleştirir. Bir diğeri sabit terimleri birleştirir. Bir diğeri eşitlikteki sıfır katsayılı terimleri siler. Kalan operatörler bir sabit ile eşitliğin çarpılması, toplanması ve çıkartılması gibi cebirsel işlemleri çözerler. Bu alan içinde PET'in görevi eşitlikteki terim sayısını

azaltarak problemin yapısını basitleştirmektir. Burada herbir operatör basitleştirici işlemleri yapar ve daha sonra kurallar öğrenilerek bilgi tabanına eklenir. Böylece sistem, problemlerin çözülmesinde kullanılacak operatörler için bir kurallar ağı kurar. PET girilen bir episod ile birleşmedikçe bir kuralı şekillendiremez. Bu kısıtlayıcı politikanın avantajı sadece bu kuralların bilinen kullanımları ile kural tabanına eklenmesidir. Diğer bir avantajı ise PET yanlış bilgilendirme karşısında şaşırmaz. İntegral hesaplamada PET 18 basit operatör kullanır. Şekil 3.5 PET'in problem yapısını sırasıyla operatörlerle nasıl birleştirdiğini göstermektedir. Episodik öğrenmenin en önemli kısıtı kurallarının spesifik olmasıdır (Porter ve Kibler, 1986).



Şekil 3.5 PET'in problem yapılarını episodlarda operatörlerle birleştirmesi

3.2.1.3 Kavramsal Sınıflama

Sınıflandırma yapma işlemi gözlemle öğrenmenin bir şeklidir. Kavramsal sınıflama metodu objeleri sınıflandırır. Geleneksel metotlardan farklı olarak sınıflandırma yapılırken Cluster analizini ve nümerik sınıflama (taxonomy) metodunu baz alır. Kavramsal sınıflama objeleri bir hiyerarşi ile kavramsal bir ağa yerleştirir. Kavramsal sınıflamada sınıflandırma problemi aşağıdaki gibi belirlenir.

Verilenler :

1. Obje seti
2. Objeleri karakterize etmede kullanılacak özellikler(attributler)
3. İstenilen sınıf (cluster) sayısı
4. Problem kısıt bilgisi, özelliklerin değeri, kriterlerin değerlendirilmesi,

Bulunanlar :

1. Obje sınıflarının bir hiyerarşisi
2. Alt sınıflar ve kriterlerin eniyilenmesi

3.2.2 Sembol Seviyeli Öğrenme Metotları

Sembol seviyeli sistemler temel olarak bilgi seviyeli sistemlerden bilgi gösterim şekilleri ve öğrenme metotlarından dolayı farklılık gösterir. Bu sistemler sabit bir alfabe, sabit uzunluk vektörü veya sıralanmış bir liste kullanırlar. Bu onların bilgi gösterim şeklidir. Burada iki tür sembolik seviyeli öğrenmeyi inceleyeceğiz. Sınırlandırıcılar ve genetik algoritmalar.

3.2.2.1 Sınırlandırıcılarda Öğrenme

Sınırlandırıcılar tümevarıma dayalı öğrenme sistemleridir. Bu sistemlerde bilgi bir matris, vektör veya parametreler seti olarak temsil edilebilir. Tümevarımla öğrenme bir sınıfa ait nesnelerin tek tek tanımlanmasından, bir sınıfın tamamının tanımlanmasına kadar devam eden süreçte bir muhakeme prosesi olarak tanımlanabilir. Kavram, eğer bir veri kümesi için uygulandığında doğru bir sınıfı tanımlıyorsa doğru, diğer hallerde yanlış olan sembolik bir tanımlamadır. Burada bir eğitim seti, karakteristik değerler vektörü ve ona karşılık gelen sınıftan oluşur. Öğrenilmiş bir kavram tümevarımla öğrenme prosesiyle belirlenmiş bir kural olarak tanımlanabilir. Eğer yeni bir veri bu kuralın şartlarına uygun olursa, karşılık gelen sınıfa dahil olur (Pham ve Aksoy, 1994).

3.2.2.2 Genetik Algoritmalarda Öğrenme

Genetik algoritmalar adaptif ve genel amaçlı arama teknikleridir ve popülasyon genetiği prensibini temel alır. Popülasyon kelimesi burada anlam olarak bir probleme ait alternatif çözümler kümesini ifade eder. Genetik algoritmalar genetik operatörler setinden ibarettir. Temel genetik operatörler, değiştirme operatörü (mutation), çaprazlama operatörü (crossover) ve yeniden popülasyon oluşturma operatörü (reproduction) dır. Değiştirme operatörü karakter içerik pozisyonunun rasgele değişmesini sağlar. Yani kromozom üzerinde değişiklik yapılmasını sağlar. Çaprazlama operatörü, karakter çifti ile seçilmiş parça arasındaki yer değişmedir. Başka bir ifadeyle, iki kromozom arasında bilgi alışverişini sağlar. Yeniden popülasyon oluşturma, özel karakterleri değerlerine göre kopyalayarak yeniden popülasyon oluşturur. Genetik algoritmalarla öğrenmeyi bir örnekle açıklayalım: Sıralanmış bir özellik seti ve obje setini ele alalım. Buradaki

herbir obje bu özellikler setinin bir alt setine sahiptir. Yani her bir obje bu altset ile karakterize edilebilir. Objeler {b, c ve d}, özellikler seti {p, q, r, s, t ve u} olsun. Burada özellikler [p, q, r, s, t, u] şeklinde sıralanmıştır. Değişirme operatörü b' ye uygulandığında [0, 1, 1, 1, 0, 0] elde edilir. Burada üçüncü pozisyon kavramsal olarak belirlenebilmesi için değişmeye uğrar. Burada [0, 1, 1, 1, 0, 0] ayrıca [0, 1, 0, 1, 0, 0]'ın özelleştirilmiş bir halidir. Burada kavram genelleştirme için 0 yerine 1 alınmıştır. Benzeri olarak çaprazlama operatörü iki objenin aynı özelliklerini birleştirerek yeni kavramlar genelleştirir. Mesela b ve c arasındaki bir yer değişirme d'yi üretebilir. Böylece q, s, t özelliklerini kazanır. Kural tabanlı sistemlerde en zor problemlerden birisi makul yeni kuralların otomatik olarak genelleştirilmesidir. Holland 1986 yılında kural tabanlı sistemlerle genetik algoritmaları, adaptif sistem denilen sınıflayıcı sistemleri geliştirebilmek için birleştirmişti. Burada girdi arakesiti bir çevreden özellikleri arar ve onları mesajlara çevirir. Böylece mevcut çevresel yapı mesaj şeklinde kabul edilir. Bu mesajlar mesaj listesinde saklanır. Mesaj listesindeki mesaj setleri koşullar ile eşleştiği zaman kurallar aktif hale gelir.

Genetik algoritmaların çalışma prensibini şöyle belirleyebiliriz :

1. Sınıflandırıcı setinden çiftler seçilir.
2. Yeni sınıflandırıcılar meydana getirmek için çiftlere genetik operatörler uygulanır.
3. En zayıf sınırlandırıcı sonucu ile birlikte yerleştirilir.
4. Operatörlerle yeni kurallar üretilip test edilir ve çevre karşısında geliştirilir.

Genetik algoritmalarla, yapay sinir ağları birleştirilerek değişik çalışmalarda kullanılmışlardır (Holland, 1975; Davis, 1994).

3.2.3 Makine Seviyeli (Device Level) Öğrenme

Yapay sinir ağları (neural networks) makine seviyeli sistemlerdir. Sembol seviyeli öğrenmeden en önemli farkı ise bilginin gösterim şeklidir. Yapay sinir ağları daha sonraki bölümlerde ayrıntılı olarak anlatılacaktır.

3.3 Bilgi Gösterimi ve Öğrenme Arasındaki İlişki

Bilgi sistemlerinde aşağıda belirtilen genel bilgi gösterim şekilleri veya onların kombinasyonları kullanılır: Kural tabanı, çatı (frame), yüklem mantığı (predicate logic),

semantik ağ, sınırlandırıcılar ve yapay sinir ağları. Bu metotların herbiri uygulandığı yerlere göre avantaj ve dezavantajlara sahiptir. Kural tabanlı sistemlerde bilgi basit olarak temsil edilir. Bu sistemler ileri ve geri zincirleme tekniğini kullanırlar. Bunlar gerçek bilgi seviyeli sistemlerdir. Kural tabanlı sistemler yapılandırılmış bir bilgi temsiliinde bulunamaz. Yükleme mantığı (Predicate logic) kullanan sistemlere benzemezler ve basit sonuç çıkarma mekanizmasını temel alırlar. Çoğu kural tabanlı sistemler, çatı ve mantıksal tabanlı özellikleri birleştirirler.

Çatı (frame) sistemleri, etkili bir çıkarsama yapabilmek için genellikle kural tabanlı ve mantıksal tabanlı metotları birleştirir. Mantıksal tabanlı sistemlerde, bilgi basit ve karmaşık yükleme (predicate) yapıları ile temsil edilir. Mantıksal tabanlı sistemler tımdengelimli öğrenme metotlarına destek veren sistemlerdir. Ayrıca soyutlama, monotonik olmayan düşünme ve teori revizyonu için de destek teşkil eden sistemlerdir. Semantik ağ bilgi temsiliinin önemli olan diğeri bir şeklidir. Bu sistemlerde bilgi düğümleri ve bağlantıları şeklindedir. Burada düğümleri, objeleri, sınıfları ve olayları gösterir. Yükleme mantığı ve semantik ağ arasında çok güçlü bir yapısal paralellik vardır. Semantik ağlar da belirleyici bilgi kavram hiyerarşisinin içinde organize edilebilir. Burada is-a, has-a gibi bağlantıları vardır. Sınıflandırıcı ve sınırlandırıcı sistemler genetik algoritmaları ve tümevarımla öğrenme sistemlerini temel alırlar.

3.4 Bilgi Sistemleri ve Makine Öğrenmesinin Sınırları

Yapay zeka sistemlerinin yapabileceği işleri fazla ve çok karmaşıktır. Dolayısıyla bu alanda çok fazla sayıda bilginin temsil edilebilme ihtiyacı doğmaktadır. Mevcut sistemlerdeki benzer problemlerden kaçmak için bilgi alanı genişletilmek zorundadır. Eğer bir sistem iyi performans gösteriyorsa bu şartı sağlamış demektir. Amarel, Meta-DENDRAL gibi fizik ve biyoloji alanındaki geniş teori formasyonlu sistemleri geliştirmenin önemini vurgulamıştır. Çünkü bu tür sistemlerin geliştirilmesi öğrenme ve keşfetme problemlerine deneysel bir yaklaşım sağlar. Amarel, çeşitli öğrenme metotlarını birleştirmesi sonucu daha iyi sonuçlar alınabileceğini söylemiştir. Ayrıca Lenat da bütünsel öğrenme sistemlerinin gerekliliğini belirtmiştir. Bu sistemlerin etkileşimli çalışması daha gerçekçi insan öğrenme metotlarının inşası edilmesi için gerekli hale gelmiştir. Diğeri yanda makine öğrenmesi araştırmalarının hedefi geniş sistemlerde biraraya getirilebilecek yeni metotları bulabilmektir. Fakat açık ve net metodolojiler olmaksızın bu metotları biraraya getirmek zor olacaktır. Makine öğrenmesi metotlarının biraraya getirilmesi ve geliştirilmesi büyük bilgi tabanlı sistemlerin inşasında ve

bilgi elde edilmesinde çok önemlidir. Mesela CYC gibi genel bilgi sistemleri değişik öğrenme metotlarını biraraya getirir. Simon insan öğrenmesinin yavaş bir proses olduğunu ileri sürmüştür. Buna sebep olarak insan öğrenmesinde, bilginin elde edildikten sonra organizasyon aşamasının geniş olması gösterilebilir. Bu belirlemeyi doğru olarak kabul edersek geniş tabanlı makinelerle öğrenme hızının arttırılamayacağı ortaya çıkacaktır. Ayrıca Simon bir gerçeğe daha dikkat çekmektedir; hiçbir bilgisayarda insan öğrenmesinin bir kopyası yoktur. Bir insanın beynine depolanmış olan bilgi asla bir başka insanın beynine direk olarak transfer edilemez. Çünkü insanların değişik bilgi organize etme yolları vardır. Buna rağmen aynı konuda uzman iki kişinin bilgi organizasyonu aynı olabilir. Dolayısıyla aynı konuda uzman iki kişi birbirlerinden çok hızlı bilgi alıp verebilirler (Simon, 1983; Forsyth ve Naylor, 1985; Lenat, 1983).

Lenat ve Feigenbaum bir sistemin bilgi ve öğrenme performansı arasındaki ilişkiyi araştırmışlardır. Bir prensibe giriş yapmışlardır. Bu prensip öğrenme konusunu tek bir sözle özetliyordu, "Bilgi Güçtür". Bir sistemin zeki anlamasını ve faaliyetini, getirebildiği özel bilgi ve yüksek seviyeli yeterliliği gösterir. Lenat ve Feigenbaum bilgideki açıklık prensibini vurgulamışlar ve "Kesin Bilgi" prensibini ortaya koymuşlardır. Buna göre zeki bir sistemdeki bilgi çokluğu, bilginin kesin bir şekilde temsil edilmesine ihtiyaç duyar. Bu prensibin etkinliğinin sistemdeki bilgi organizasyonuna bağlı olduğu söylenebilir. Burada öncü sayabileceğimiz bazı araştırmacılar bilgi elde etme konusundaki araştırmaların kısıtlılığına dikkat çekmektedir. Bilgi organizasyonu konusunda ciddi araştırmalar yapılmadıkça da bu problem kalıcı olacaktır (Lenat, 1983). Simon öğrenme konusundaki araştırmalar için sırasıyla şu tavsiyelerde bulunmaktadır.

- Araştırma, insan öğrenmesini anlama ve benzetim yönünde olmalıdır.
- Araştırma, insan öğrenmesinin niçin çok yavaş ve etkisiz olduğu hakkında olmalı ve buna karşılık hızlı makine öğrenme metotları tasarlanmalı ve ihtimaller ele alınıp denenmelidir.
- Araştırma, bilgi bazlı sistemler için doğal dil arakesitini ele almalıdır.
- Eksik bilgilendirme ve program üzerinde araştırma yapılmalıdır.
- Keşfetme programları üzerinde araştırma yapılmalıdır.

Simon, makine öğrenmesinde kullanışlı bir araştırma yapabilmek için seçilen hedefin kesinliği ve açıklığının gerekliliği üzerinde önemle durmaktadır. İnsan öğrenmesinin daha derin bir

şekilde anlaşılması, makine öğrenmesinde nelerden kaçılacağı ve nelerin taklit edileceği konusunda önemli anahtarlar sağlar. Eğer makine öğrenmesinde bu şekilde bir yönelim olacaksa çalışmalar insan öğrenmesini anlama yönünde olacaktır (Dayhoff, 1990). Lenat ve Feigenbaum yapay zeka araştırmalarında deneysel araştırmaların yol göstericiliği prensibini ortaya atmışlar ve buna "Deneysel sorgu hipotezi" demişlerdir. Bu prensibe göre yapay zeka araştırmalarında başarılı olmanın en kısa yolu hipotezlerimizi programlarda şekillendirmektir ve programları çalıştırarak veri elde etmektir. Lenat ve Feigenbaum üç aşamalı bir araştırma programını ileri sürmüşlerdir:

1. Dikkatli bir şekilde ve elle kodlanmış geniş ve büyük bilgi tabanı oluşturulması,
2. Yeterince bilgi temsil edildiğinde, veritabanının sindirilmesi ve okunmasının daha hızlı olması,
3. İnsan bilgisinin sınırlarını aşmak için, keşfetme (Discovery) ile öğrenmenin temel alınması.

Bunlara ek olarak, Lenat ve Feigenbaum yapay zeka araştırmacılarının zamanlarının önemli bir kısmını kendi alanları ile ilgili yeterlik düzeyi yüksek programlara ayırmaları gerektiğini söylemişlerdir. Buna delil olarak da Stefik ve Friedland'ın iki yıl çalışarak geliştirdikleri moleküler biyoloji ile ilgili MOLGEN programını ortaya koymuşlardır (Lenat, 1983).

4. YAPAY SİNİR AĞLARI

4.1 Neden Yapay Sinir Ağları?

Yapay sinir ağlarına olan ilgi son yıllarda büyük bir artış göstermiş hatta doyma noktasına gelmiştir. Mühendislik, tıp, felsefe, psikoloji gibi alanlarda çalışanlar yapay sinir ağlarını kendi uzmanlık alanlarına taşıyarak, kendi alanlarında uygulamalarını geliştirmeye başlamışlardır. Bu ilgi teorik ve uygulama alanlarındaki başarılarla daha da pekiştirilmiştir. İnsan zekasına has gibi görünen bazı alanların sayısal olarak ifade edilebileceği ve böylece makinelerin insan zekasına şaşırtıcı derecede benzer yollarla öğrenme ve hatırlama işlerini yapabileceği görülmüştür (Haykin, 1994).

4.1.1 Yapay Sinir Ağlarının Özellikleri

Yapay sinir ağları, insan sinir sisteminin biyolojik yapısından esinlenmiş bir yapıya sahiptir. Yani, yapay sinir ağları, insan sinir sistemindeki sinir hücrelerinin fonksiyonunu gören temel elemanlardan oluşmaktadır. Bu elemanlar insan beyninin anatomisine benzer şekilde organize edilmişlerdir. Bu büyük benzeşmenin yanı sıra, yapay sinir ağları, şaşırtıcı derecede insan beyninin birçok özelliklerini taşırlar. Örneğin, tecrübe ile öğrenirler, daha önce öğrenilen bilgileri genele indirgeyerek yeni çıkarsamalar yaparlar. Bu yeni çıkarsamalardan gereksiz kısımları atarak önemli olan öz kısmını alırlar. Fakat bütün bu fonksiyonel benzeşmelere rağmen, yapay sinir ağlarının yakın bir gelecekte insan beyninin fonksiyonlarını aynen yerine getirebileceği söylenemez. Bugün bu alanda geline nokta, insan zekasının derinlemesine bir incelemesinin, yapay sinir ağları alanında devrim niteliğinde uygulamaları da beraberinde getireceğini göstermektedir.

4.1.1.1 Öğrenme

Yapay sinir ağları, çevreden aldıkları girdiye (bilgi) çıktı (cevap) olarak gösterdikleri davranışlarını değiştirebilirler. Öğrenme işleminde ağa belli bir girdi verildiği zaman, ağ tutarlı cevaplar üretebilmek için kendi üzerinde değişiklik yapmak durumundadır. Yapay sinir ağlarını eğitebilmek için çeşitli öğretme algoritmaları geliştirilmiştir. Bu algoritmaların her birinin kendine göre zayıf ve güçlü olduğu noktalar vardır. Burada cevaplanması gereken en önemli iki soru “ Bir yapay sinir ağına neler öğretilir?” ve “Öğretme işlemi nasıl olmalıdır?”

4.1.1.2 Genelleme

Öğretme işlemi bitmiş bir ağ, bir dereceye kadar ona verilen girdilerdeki ufak değişikliklere karşı duyarsız olabilir. Yani sürekli aynı tepkiyi gösterir. Bu yetenek, gerçek dünyadaki, çevreden gelen faktörlerle ufak bozulmalara uğramış girdileri farkedebilmek açısından önemlidir. Bu durum, klasik bilgisayarlarda kullanılan mantığı aşarak; içinde yaşadığımız mükemmel olmayan dünyayı anlayabilmek için geliştirilmiş bir sistemdir. Yapay sinir ağları genelleme işini kendi yapılarından dolayı otomatik olarak yaparlar.

4.1.1.3 Özetleme

Bazı yapay sinir ağları, girilen belli bir girdi dizisinin özünü ayıklama, çıkarma yeteneğine sahiptir. Örneğin, bir ağa girdi olarak “A” harfinin çeşitli bozuk şekillerini verebiliriz. Yeterli öğretmeden sonra, girilen bozuk bir “A” harfine cevap olarak ağ düzgün bir “A” harfini verebilecektir. Yani bir anlamda ağ, daha önce görmediği, öğrenmediği bir şeyi üretmeyi öğrenmiş olacaktır. Burada ideal prototipler çıkarabilmek yeteneği insanın önemli ve yararlı bir özelliğidir ve günümüzde bu özelliğin yapay sinir ağlarında kullanılması gündeme gelmiştir. Yapay sinir ağları, elbette, klasik bilgisayarların yapmakta olduğu maaş bordrosu hesaplamak gibi işler için uygun değildir. Fakat bugün klasik bilgisayarların yapmakta zorlandığı veya yapmakta çok sınırlı kaldığı model tanıma, algılama (Pattern-recognition) alanında, yapay sinir ağlarının geniş bir kullanım alanı mevcuttur (Zurada, 1992).

4.1.2 Yapay Sinir Ağlarının Tarihi Gelişimi

İnsanlar daima kendi düşünceleri hakkında merak sahibi olmuşlardır. Bu kendi kendini sorgulama işlemi, yani beynin kendisini düşünmesi, insanoğluna has bir özelliktir. “Düşüncenin doğası” hakkında felsefeden, anatomik yapısına kadar yapılan spekülasyonlar oldukça boldur. Filozoflar ve din bilimciler genellikle psikologlar ve anatomicilerin bu konudaki fikirlerine karşı çıkmışlar ve bu konu, hiçbir yarar sağlamayan, hiçbir sonuç vermeyen ateşli tartışmalara sebep olmuştur. Bu konuda sadece düşünmekten, tartışmaktan öteye gitmeyenlerin vardığı sonuçlar pozitif bilimin sunduğu sağlamlık ve inanılabilirlikten yoksun kalmıştır. Bu konuda deney yapanlar beynin ve sinir sisteminin gözlemlenmesinin ve incelenmesinin oldukça zor olduğu görüşünde birleşmişlerdir. Sonuçta, fiziksel gerçeklere bakış açımızı değiştiren bilimsel metotlar ne yazık ki insanın kendisini anlamasında yavaş kalmıştır.

Nörobiyologlar ve nöroanatomistler bu alanda çalışmalar yapmakta ve önemli ilerlemeler kaydetmektedirler. İnsan sinir sisteminin yapısı ve fonksiyonlarının özenli bir haritasının çıkarılmasıyla, beynin bağlantısı (bir anlamda kabloları) ve yapısı hakkında oldukça önemli bilgiler elde edilmiştir. Fakat aynı ilerleme beynin işleyişi konusunda gösterilememiştir. Bu alandaki bilgi arttıkça, karmaşıklığın ağır ağır çözüldüğü görülmüştür. Sayıları milyarları bulan ve birbiriyle bağlantılı sinir hücreleri, en gelişmiş bilgisayarları dahi çaresiz bırakan bir sistemi oluşturmaktadırlar. İnsan beyni, insanoğlunun en fazla üzerinde durduğu ve cevap bulmaya çalıştığı soruların cevabını yavaş da olsa ortaya çıkarmaktadır. Sinir hücrelerinin fonksiyonları ve birbirleriyle olan bağlantıları daha iyi anlaşıldıkça, araştırmacıların teorilerini test etmek amacıyla matematiksel modeller geliştirmelerine imkan sağlanmıştır. Artık deneyler dijital bilgisayarlar üzerinde insan veya hayvan deneklere ihtiyaç duyulmadan çözülebilmektedir. İlk çalışmalar, bu modellerin beynin fonksiyonlarını taklit etmelerinin yanısıra; kendi başlarına bazı yararlı fonksiyonları da yapabildiklerini göstermiştir. Böylece sinir modellemenin günümüzde de geçerliliğini koruyan, karşılıklı olarak birbirini destekleyen iki ana hedefi belirlenmiştir: Birincisi, insan sinir sisteminin fizyolojik ve psikolojik işleyişini anlamak, ikincisi, beyine benzer fonksiyonlar gerçekleştiren yapay sinir ağları elde etmek. Nöroanatomisi ve nöropsikoloji alanındaki gelişmelerle birlikte, psikologlar da insanın öğrenme yetisini modelleme çalışmalarına başlamışlardır. Bu modellerden en verimli olanı Hebb tarafından geliştirilmiştir. Hebb, 1949'da, yapay sinir ağları "öğretme algoritmaları" için başlangıç noktası sayılan bir öğrenme kanunu ileri sürmüştür. Günümüzde üzerine katkılar yapılmış olan bu model, bir sinir hücresi ağının öğrenme işini nasıl gerçekleştireceği konusunda fikir vermiştir. 1950'li ve 60'lı yıllarda, bir grup araştırmacı bu biyolojik ve psikolojik gelişmeleri birleştirerek ilk yapay sinir ağını elde etmişlerdir. İlk olarak elektrik devresi şeklinde elde edilen ağ, daha sonra bilgisayar benzetimi haline dönüştürülmüştür. Bu ilk başarılar bir aktivite ve iyimserlik patlamasını da beraberinde getirmiştir. Marvin Minsky, Frank Rosenblatt, Bernard Widrow ve daha birçok bilim adamı tek yapay sinir hücresi katmanından oluşan ağlar geliştirmişlerdir. Algılayıcı (perceptron) adı verilen bu ağlar, hava tahmini, elektrokardiogram analizleri ve yapay görme gibi çeşitli kullanım alanları bulmuşlardır. Bu dönemde bilim adamları "zeka"nın anlaşıldığı, çözüldüğü düşüncesine kapıldılar ve beynin bir benzerini üretmenin, sadece yeterince büyük bir ağı oluşturulmasından ibaret olduğu iyimserliğini taşımaktaydılar.

Bu iyimser düşünceler kısa sürede sona erdi. Çünkü ağlar, o güne kadar çözebildiği problemlere genel hatlarıyla benzer farklı problemleri çözmei başaramamıştı. Bu başarısızlık

yoğun bir analiz ve inceleme dönemini başlattı. Marvin Minsky, matematiksel teknikleri kullanarak, ağ işlemleriyle ilgili sağlam teoriler geliştirdi. Onun bu araştırmaları “perceptrons” (Minsky ve Papert 1969) adlı kitabın basımını beraberinde getirdi. Bu kitapta Minsky ve Papert, o zaman kullanımda olan tek katmanlı ağların, teorik olarak birçok basit problemi çözemeyeceğini ispatlamışlardır. Sonuç olarak Minsky yapay sinir ağlarının başarılı olabileceği görüşünü taşııyordu.

Minsky’nin zekası ve prestiji kitaba büyük inanılrlık sağlamıştır ve onun vardığı sonuçlara hiç kimse karşı çıkmamıştır. Cesareti kırılan bazı araştırmacılar başka alanlara kaymışlar, devlet desteğini çekmiş ve bunların sonucu olarak yapay sinir ağları 20 yıllık bir karanlık döneme girmiştir.

Fakat yapay sinir ağları ile ilgili çalışmalar, Teuvo Kohonen, Stephen Grossberg ve James Anderson gibi birkaç bilim adamı tarafından sürdürülmüştür. O dönemde kıymeti bilinmeyen ve fazla destek görmeyen bu araştırmacıların eserlerini yayınlıyacak yayıncı bulmakta zorlanılmıştır. Bu nedenle 1970’li yıllarda ve 1980’li yılların başında yayınlanan araştırmalar birçok gazete ve dergiye dağılmış haldedir. Karanlık dönem denilen bu dönemde yapılan çalışmalar, bugünün çok katmanlı ağlarına teorik bir temel teşkil etmişlerdir. Böylece Minsky’nin fazla kötümser olduğı ispatlanmıştır. Minsky’nin kitabında bahsettiğı birçok problem bugün ağlar tarafında rahatlıkla çözülebilmektedir.

Son yıllarda, yapay sinir ağlarının teori kısmı uygulamaya dönüştürülerek, bu teknolojinin ticari kısmıyla ilgilenen şirketler ortaya çıkmıştır. Bu alanda yapılan araştırmalarda büyük bir patlama olmuştur. 1987’de yapay sinir ağları konusunda dört büyük toplantı yapılmış ve ayrıca beşyüzün üzerinde araştırma yayınlanmıştır. Daha yakın tarihlere gelindiğinde ise yapılan çalışmaların ve toplantıların daha da arttığı gözlemlenmiştir. Bu dönemden çıkarılacak en önemli ders, sorgulamadan bazı gerçekleri kabullenmenin bilimsel yaklaşıma zarar verdiğidir. Dolayısıyla Minsky’nin bilimsel çalışmasının yapay sinir ağları alanındaki gelişmeye köstek vurduğunu söylemek yanlış olmaz. Fakat yapay sinir ağları alanındaki belli bir olgunlaşma dönemine de Minsky’nin “Perceptrons” adlı kitabının verdiğı şokla girildiğı gözlemlenmiştir (Haykin, 1994).

4.1.3 Günümüzde Yapay Sinir Ağları

Yapay sinir ağlarının yeteneklerini gösteren birçok ilginç örnekler mevcuttur. Bunlardan;

- Yazılı bir metni sese çeviren ve daha sonra bu sesin başka yöntemlerle konuşma haline

getirildiği ağılar (Sejnowsky ve Rosenberg, 1987)

- Elyazısı harfleri tanıyabilen ağılar (Burr, 1987)
- Sinir ağılarına dayanan görüntü sıkıştırma sistemleri (Cottrell, Munro ve Zipser, 1987)

bazılarıdır. Yakın zamana gelindiğinde bu gibi çalışmaların yüzlercesinin yapılmış olduğu görülmür. Disiplinler arası bir konu olduğunda yapılacak çalışmaların sayısı da artacaktır. Bu çalışmaların hepsinde mevcut algoritmaların en başarılısı olan GYA uygulanmıştır. Bu algoritma üzerinde birbirinden bağımsız ve değişik zamanlarda üç araştırmacı çalışmıştır. Bunlar sırasıyla; Werbos 1974, Parker 1982 ve Rumelhart, Hinton, Williams 1986. Rumelhart ve arkadaşlarının yaptığı çalışma ortaya çok güçlü teorik ve matematiksel temeli olan bir algoritma koymuştur. Yani çok katmanlı ağıları eğitmek için sistematik bir yöntem sunulmuş ve böylece Minsky tarafından ifade edilen sınırlamaların üstesinden gelinmiştir. Aslında GYA'nın da bazı problemleri vardır. Birincisi, ağın sonlu bir zaman dilimi içerisinde eğitileceği garantisi yoktur. Ağı eğitme çabaları başarısızlıkla da sonuçlanabilir. Her ne kadar daha iyi sonuç alınabileceği garantisi yoksa da, bazı parametreler ya da eğitme seti değiştirilerek öğretme girişimi tekrarlanmalıdır. Ayrıca ağın, mümkün olan en iyi konfigürasyonla eğitileceği garantisi de yoktur.

Performansını ispat etmiş, bazı sınırlamaları ve cevapsız kalmış sorularıyla yeni bir araştırma sahası ile karşı karşıya kalınmıştır. Bu konuya ister istemez ihtiyatlı bir iyimserlikle yaklaşılmaktadır. Yazarlar başarılarını yayınlarında gösterdikleri sorumluluğu, ne yazık ki başarısızlıklarını yayınlamakta pek gösterememişler (birkaç istisna hariç) ve böylece gerçeği tam olarak yansıtmayan bir izlenim oluşturmuşlardır. Riskli bir iş için sermaye arayanlar, daha önceki başarılarını inandırıcı bir şekilde ispatlamak zorundadırlar. Bu yüzden yapay sinir ağlarının, zamanından önce aşırı taleple karşı karşıya kalmak ve sunamayacağı hizmetlerin sözünü vermek tehlikesi ortaya çıkmaktadır. Bu da, bu sahanın inandırıcılığını kaybetmesi ve 1970'lerin karanlık dönemine tekrar girilmesi demektir. Mevcut ağıları geliştirmek için daha sağlam ve somut çalışmalara ihtiyaç vardır. Bu sahanın gerçek potansiyelini gösterebilmesi için, öncelikle yeni tekniklerin geliştirilmesi, varolanların iyileştirilip, güçlendirilmesi ve teorik tabanın genişletilip sağlamlaştırılması gerekmektedir. Bunlara yönelik çalışmalar da son yıllarda iyice artmıştır (Lakhmi ve Vemuri, 1999).

4.1.4 Gelecekte Beklentiler

Yapay sinir ağıları, savaş alanı idaresinden bebek bakımına kadar çok çeşitli alanlarda kullanılabilir. Potansiyel uygulama alanları, insan zekasının kolaylıkla üstesinden geldiği fakat klasik bilgisayarların çok hantal ya da yetersiz kaldığı alanlar diye tarif edilebilir. Uygulama alanları en az geleneksel bilgisayarlarındaki kadar geniş olan bu saha, geleneksel bilgisayarlarla aynı önemde gelişecektir. Ama bu durum, teorik yapılanmanın yetersiz olduğu günümüzde, bu alanda yapılan araştırmaların hızla yerini almasıyla gerçekleşecektir (Crooks, 1992).

4.1.5 Yapay Sinir Ağları ve Uzman Sistemler

Son yıllarda yapay zeka alanında mantık ve sembol gösterimine dayanan disiplinler hakim hale gelmiştir. Örneğin uzman sistemler gündeme gelmiş ve önemli başarılarla imza attıkları gibi başarısızlıkları da olmuştur. Bazıları yapay sinir ağlarının günümüz yapay zekasının yerini alacağını düşünmektedir. Fakat yapılan uygulamalardaki eğilim, yapay sinir ağları ve uzman sistemlerin, herbiri kendi uzmanlık alanına giren işi yapmak üzere, birleşip aynı sistemler üzerinde bulunacaklarını göstermektedir. Aslında bu eğilim sadece uzman sistem ve yapay sinir ağı ikilisi için değil, aynı zamanda tüm yapay zeka teknikleri için geçerlidir. Oluşturulan bu sistemlere melez (hybrid) sistemler adı verilmektedir.

Bu bakış açısı insanoğlunun çalışma biçimiyle de desteklenmektedir. Hızlı cevap veya tepki bekleyen aktiviteler model-algılama (pattern-recognition) mekanizmaları tarafından yönetilmektedir. Bu tür hareketler hızlı ve bilinç dışı üretildiği için, vahşi hayatta veya düşman varlıkların olduğu çevrelerde hayatta kalabilmek için anında ve hızlı bir tepki vermek açısından önemlidir. Örneğin insanın üzerine gelen bir vahşi hayvan karşısında doğal refleks göstermesi veya mantık tahlili yaparak hareket etmesi tercihindendir hangisi daha doğru olacaktır? Eğer model-algılama sistemimiz, belli bir zaman aralığında, mutlak olmayan doğru bir yorumda bulunamıyorsa, o zaman problem bir üst merteye olan akıl fonksiyonuna havale edilir. Dolayısıyla bu daha fazla bilgi ve daha fazla zaman olmakla birlikte daha kaliteli sonuç demektir.

Bu iş bölümünün uygulandığı bir yapay sistem göz önünde canlandırılabilir. Cevap ve tepkileri büyük bir çoğunlukla yapay sinir ağı verecektir. Bu ağlar verilen kararların eminlik derecesini bilebildikleri için, “bilmediklerini bildikleri” zaman, çözüm için uzman sisteme başvuracaktır. Bir üst mertebeye verilen kararlar daha somut ve daha mantıklı olmakta, fakat

bir sonuca varmadan önce daha fazla bilgiye ihtiyaç duyulmaktadır. Bu iki sistemin birlikte hareket etmesi, tek başlarına olmalarından daha kuvvetli bir sistem oluşturmakta ve biyolojik benzerleri tarafından sunulan modeli başarılı bir şekilde temsil edebilmektedir.

4.2 Yapay Sinir Ağlarının Esasları

4.2.1 Biyolojik Prototip

Yapay sinir ağları çok değişik şekilde gelişme göstermişlerdir. Görünen bu farklılıklara rağmen aslında sinir ağları birçok ortak nokta içerir. Bu bölümde bu noktalar ayrıntılı olarak tanımlanacak ve tartışılacaktır.

Yapay sinir ağları, biyolojik sinir hücrelerinden esinlenerek düşünülmüştür. Araştırmacılar ağ şeklini ve algoritmasını düşünürken beynin organizasyonunu incelemişlerdir. Ancak beynin çalışma sistemi ile ilgili bilgiler çok sınırlıdır ve bu konuda çalışanlara yol gösterme olanağı çok kısıtlıdır. Bu bakımdan ağ tasarlayıcıları mevcut biyolojik bilgiden daha ileriye gitmeye ve yararlı fonksiyon bulmaya yardımcı olacak kavramlar aramaya başlamışlardır. Birçok durumda bu önemli değişim biyolojik gerçekleri göz ardı eder, beyin sanal olur, ağlar beyin anatomisi ve fonksiyonları hakkında organik olarak uygulanamaz. Bu inceliğe rağmen, biyolojik olarak çok benzemese de, yapay sinir ağları ile beyin arasında yapılan karşılaştırmalarda benzerlikler ortaya koyulabilmektedir. Yapay sinir ağı fonksiyonları, insanın algılamasını anımsatır. Bu sebeple benzetme yapılabilmektedir. Fakat bu tür karşılaştırmalara olumlu bakılmamaktadır. Çünkü hayal kırıklığına uğranılabilecek gerçek dışı bulgular ortaya çıkabilir. İyi analiz edilmeyen her konuda aynı tehlikeyi yaşamak kaçınılmazdır.

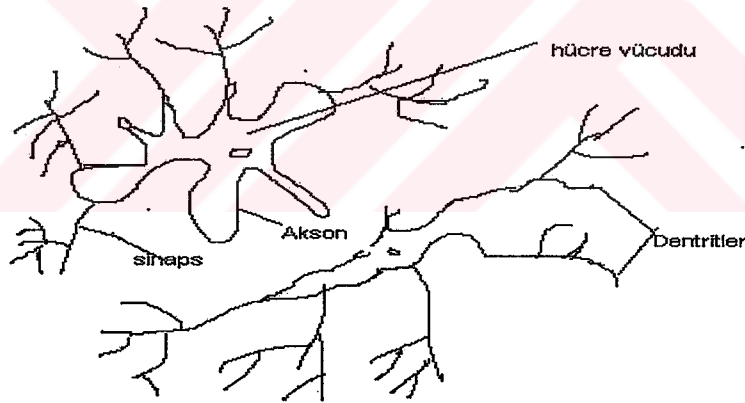
İnsan sinir sistemi sinir hücrelerinden oluşur. Beyinde olduğu tahmin edilen 100 milyar sinir hücresinin yaklaşık 1 katrilyon bağlayıcısı vardır. Her sinir hücresi vücuttaki diğer hücrelerle birlikte ortak birçok karakteristikler paylaşır. Fakat beynin haberleşme sistemini oluşturan sinir hücrelerinin görevi sinyal alma, işlem yapma ve elektrokimyasal sinyallerin sinir ağları içinde iletimini sağlamaktır. Şekil 4.1 tipik biyolojik sinir hücresi çiftinin yapısını göstermektedir. Dendritler sinyalleri hücre vücudundan sinaps diye adlandırılan birleşme noktasından alırlar. Sinapsların alıcı olan kısımları ile hücre vücudu irtibat halindedir. Burada toplandıklarında bazı girdiler hücreyi etkileme eğilimi gösterir. Diğerleri eğilimlerin uyarılmasını engeller. Hücre, vücudu içinde kümülatif uyarma eşiğini aştığı zaman uyarılır ve aksondan diğer sinir hücrelerine sinyal gönderir. Bu basit fonksiyonel çıkış birçok

karmaşıklığı ve kabulleri kapsar. Fakat yine de çoğu yapay sinir ağı bu tür basit karakteristikleri kendisine model alır.

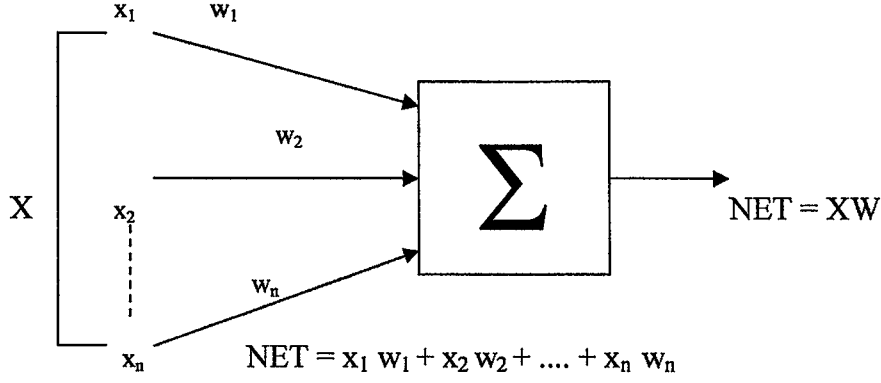
4.2.2 Yapay Sinir Hücresi (Nöron)

Yapay sinir hücresi, biyolojik sinir hücresinin, girdi, işlem ve çıktı karakteristiğini taklit etmek için tasarlanmıştır. Burada herbir girdi kendi ağırlığı ile çarpılmakta ve bu çarpımların hepsi toplanmaktadır. Bu toplam sinaptik kuvvete benzetilebilir. Bu toplam, sinir hücresinin aktivasyon seviyesini belirlemek için kullanılır. Şekil 4.2 bu modeli göstermektedir.

Ağ dizileri çok çeşitli olmasına rağmen çoğunluğu şekil 4.2'deki gösterime dayanır. Burada girdiler seti $X_1, X_2, \dots, X_n$ olarak gösterilip yapay sinire uygulanmıştır. Bu girdiler toplanarak X vektörü oluşturur ve biyolojik sinapslara sinyal olarak gönderilir. Her sinyal bağlı olduğu ağırlık değeri $W_1, W_2, \dots, W_n$ ile çarpılarak Σ işareti ile gösterilen bloğa aktarılır. Her ağırlık tek bir biyolojik sinaptik bağlantının gücünü temsil eder. Burada ağırlıklar kümesinin toplamı W vektörünü oluşturur.



Şekil 4.1 Biyolojik sinir hücresi



Şekil 4.2 Yapay sinir hücresi

Toplam blok kabaca biyolojik hücrenin yapısını gösterir. Ağırlıklı girdilerin cebirsel toplamı NET adı verilen bir çıktı meydana getirir. Bu olay vektör notasyonunda şu şekilde gösterilir:

$$NET = XW \quad (4.1)$$

4.2.3 Aktivasyon Fonksiyonları

NET sinyali genellikle aktivasyon fonksiyonu F ile işlem görerek sinir hücresinin çıkış sinyalini oluşturur. Bu çıkış sinyaline ÇIKTI diyebiliriz. Bu basit doğrusal bir eşik fonksiyonu olarak,

$$ÇIKTI = K(NET) \quad (4.2)$$

şeklinde ifade edilebilir. Burada K sabit bir eşik fonksiyonudur.

$$ÇIKTI = 1 \text{ EĞER } NET > T$$

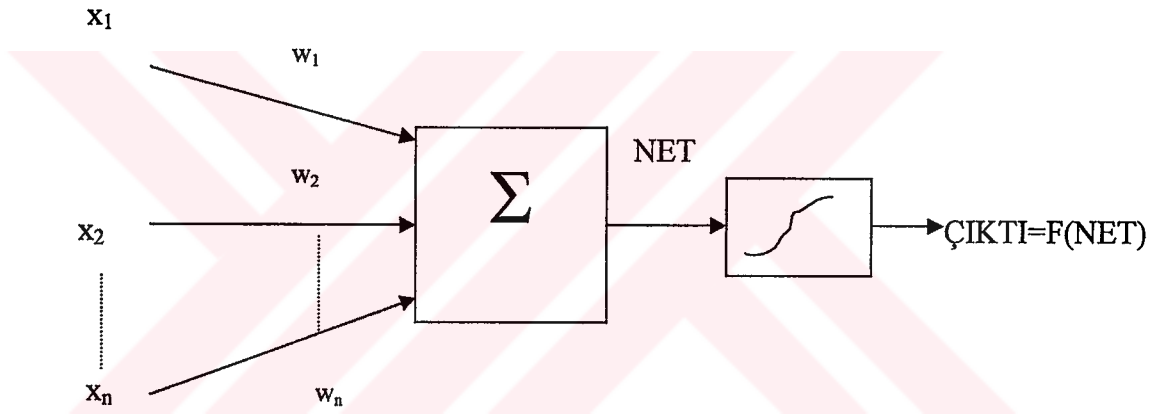
$$ÇIKTI = 0 \text{ diğer durumlar}$$

Burada T sabit eşik değeri veya fonksiyonu olup, biyolojik sinir hücresinin doğrusal olmayan karakteristiklerini temsil eder ve daha genel ağ fonksiyonlarına izin verir. Şekil 4.3'de F ile gösterilen blok net çıktısını kabul ederek ÇIKTI sinyalini üretir. Böylece F üretim bloğu net aralığını daraltır. Böylece F fonksiyonuna sıkıştırma fonksiyonu adını da verebiliriz. Sıkıştırma fonksiyonu genelde mantıksal bir fonksiyon veya S şekilli başka bir fonksiyon olarak seçilebilir. Bu durum Şekil 4.4a'da gösterilmiştir. Bu fonksiyon matematiksel olarak $F(x) = 1 / (1 + e^{-x})$ şeklinde ifade edilir. Buradan NET değeri aktivasyona uygulanırsa

$$ÇIKTI = 1 / (1 + e^{-x}) \quad (4.3)$$

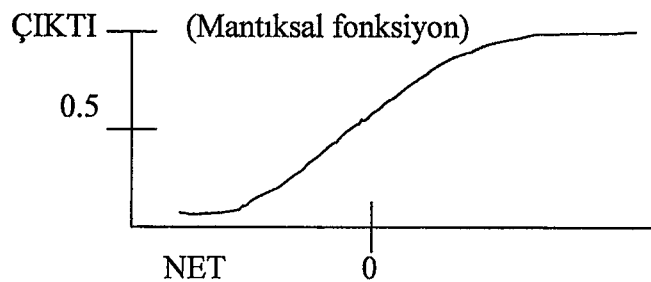
bulunur. Aktivasyon fonksiyonu yapay sinire doğrusal olmayan bir kazanç sağlar. Bu kazanç

ÇIKTI'daki değişimin NET'deki değişmeye oranı ile hesaplanabilir. Burada kazanç belirli bir değişme seviyesindeki eğrinin eğimidir. Buradaki değerler çok büyük negatif ya da pozitif değerler olduğunda sıkıştırma fonksiyonunu ÇIKTI değerlerinin aynı büyüklükte değişmesini önler. Grossberg (1973) bu doğrusal olmayan kazanç karakteristiğinin ses çözünürlüğü ikilemini çözdüğünü bulmuştur. Burada dikkat edilmesi gereken nokta nasıl aynı ağın hem küçük hem büyük sinyalleri alabildiğidir. Küçük giriş sinyalleri ağda yüksek kazançla sebep olur. Küçük girdilere kullanılabilir çıktı üretmek için büyük kazançlara ihtiyaç vardır. Bazen büyük çıktı sinyalleri yüksek kazanç seviyelerini gerektirir ve tekrar eliminasyon ile herhangi kullanılabilir çıktı sağlar. Mantıksal fonksiyonun merkezi, yüksek kazançlı alanı küçük sinyallerin üretim problemlerini çözer. Bu esnada pozitif ve negatif uçlardaki azalan kazancın alanları geniş uyarılar için uygundur. Bu yolla, bir sinir hücresi geniş girdi seviyelerinde yüksek kazanç sağlar.



Şekil 4.3 Aktivasyon fonksiyonlu yapay sinir hücresi

$$\text{ÇIKTI} = 1 / (1 + e^{-\text{NET}}) = F(\text{NET}) \quad (4.4)$$



Şekil 4.4a Sigmoidal aktivasyon fonksiyonu

Genel olarak kullanılan aktivasyon fonksiyonlarından birisi de hiperbolik tanjant

fonksiyonudur. Mantıksal fonksiyon olarak şekil bakımından benzerdir ve biyologlar tarafından sinir-hücre aktivasyonunun matematik modeli olarak sıkça kullanılır. Kullanılan yapay sinir ağı aktivasyon fonksiyonu aşağıdaki gibidir:

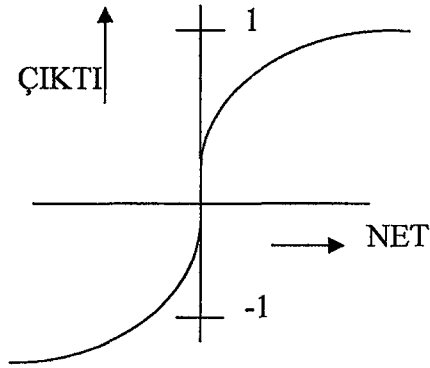
$$\text{ÇIKTI} = \tanh(x) \quad (4.5)$$

Mantıksal fonksiyondaki gibi hiperbolik tanjant fonksiyonu da S şeklindedir ve orijine göre simetriktr. Görüldüğü gibi NET değeri sıfır olduğunda ÇIKTI değeri de sıfır olur (Şekil 4.4b). Mantıksal fonksiyondan farklı olarak hiperbolik tanjant fonksiyonu ÇIKTI için çift kutupludur. Karakteristiği de ağlarda kullanılabilir bir yapıya sahiptir.

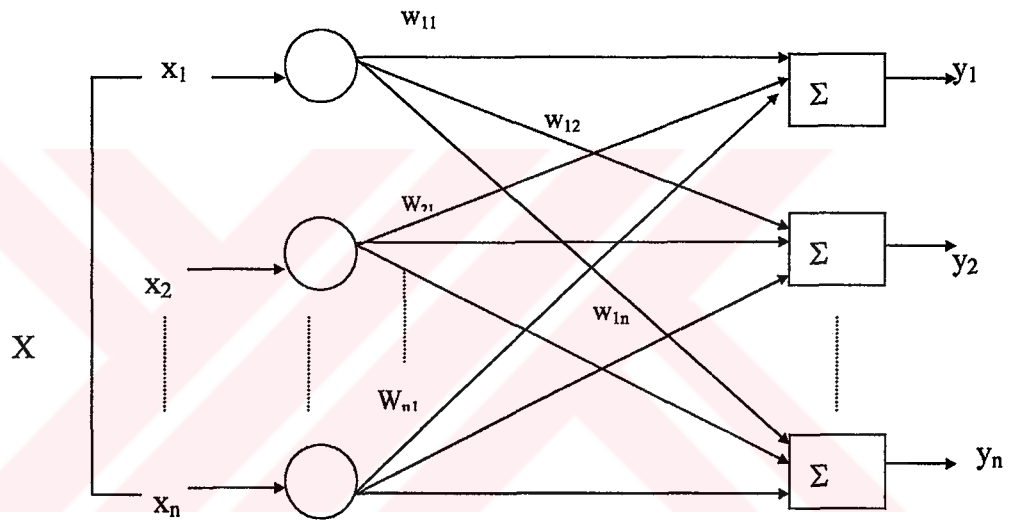
Ağların biyolojik sistemlerle kuvvetli benzerlikleri vardır. Belki de biyolojik bir sinir hücresinin doğasına dair önemli kısımlar biyolojik sinir hücresi gibi davranır veya bu benzerlik sadece tesadüftür. Bunu ancak zaman ve araştırmalar gösterecektir (Lin ve Lee, 1995).

4.2.4 Tek Katmanlı Yapay Sinir Ağları

Yapay bir sinir hücresinin ve hücrelerinin birbirine bağlanmasından oluşan bir yapay sinir ağının gücü ve başarısı gerçek bir sinir sistemininki kadar mükemmel olmayacaktır. En basit ağ yapısı şekil 4.5’de görüldüğü gibidir. Yuvarlak gösterimler sadece girişlerin sinir hücresine dağılmasını sağlar. Herhangi bir çözümleme yapmaz. Bu nedenle onları sinir hücrelerinden farklı olarak ifade edebilmek için yuvarlak olarak göstermek gerekir. Burada girdilerin kümesi olan X’in her elemanı bir ağırlık değeri ile yapay sinir ağına bağlanır. Oluşturulmuş ilk yapay sinir ağları bu kadar karmaşık değildi. Her sinir hücresi basit bir hesaplamayla girdilerin ağırlıklı toplamını ağa veriyordu. Yapay ve biyolojik sinir ağlarının bazı bağlantıları silinmiş olabilir. Fakat yapay sinir ağının yapısını genelleştirmek için eksiksiz bağlantılar tercih edilir. Burada W ağırlık matrisidir, m girdi sayısı, n ise sinir hücresi sayısıdır. Örneğin 3. girdiden 2. sinir hücresine bağlantı yapan ağırlık W_{32} olarak ifade edilir. Dolayısıyla burada NET hesabı $\text{NET} = XW$ olacaktır.



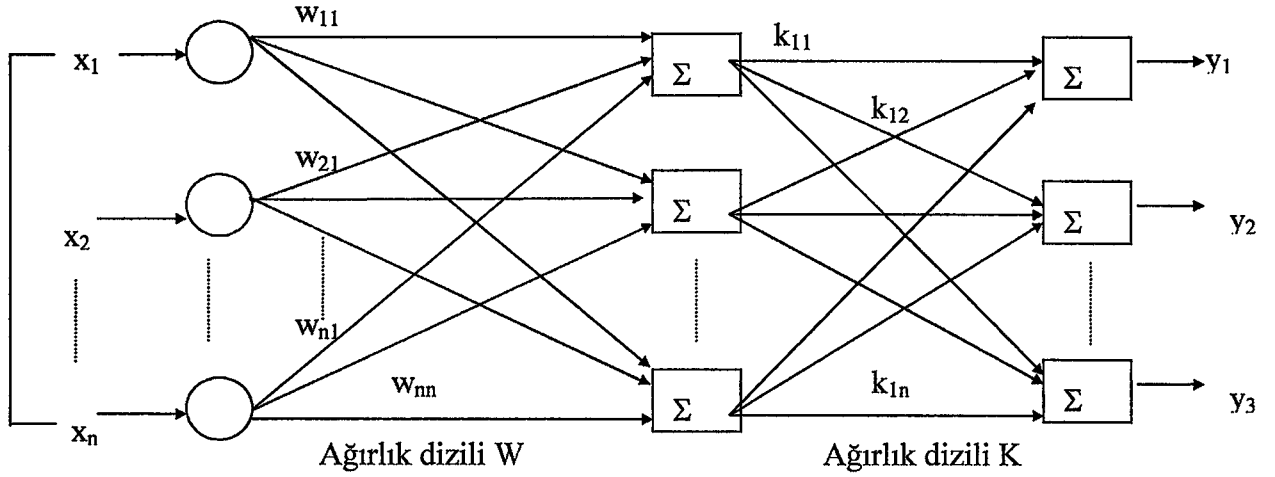
Şekil 4.4b Hiperbolik tanjant fonksiyonu



Şekil 4.5 Tek katmanlı sinir ağı

4.2.5 Çok Katmanlı Yapay Sinir Ağları

Daha geniş ve daha karmaşık ağların hesaplama kapasiteleri daha fazladır. Ağlar beynin bir parçasını yansıtacak şekilde inşa edilmişlerdir. Çok katmanlı ağların tek katmanlı ağlara oranla daha mükemmel oldukları ispatlanmıştır ve tüm çalışmalar bu ağlar üzerinde yoğunlaşmıştır. Çok katmanlı ağlar, tek katmanlı ağların bir araya gelmesinden oluşur. Şekil 4.6 çok katmanlı ağ yapısını göstermektedir.



Şekil 4.6 Çift katmanlı sinir ağı

4.2.6 Doğrusal Olmayan Aktivasyon Fonksiyonu

Çok katmanlı ağlarda, eğer aktivasyon fonksiyonu doğrusal ise X girdi vektörü olmak üzere W_1 birinci ağırlık matrisi, W_2 ikinci ağırlık matrisi olmak üzere çıktıyı şu şekilde hesaplayabiliriz:

$$(XW_1)W_2 \quad (4.7)$$

Matris çarpımı birleşme kadar terimler tekrar gruplandırılır ve yazılır:

$$X(W_1W_2) \quad (4.8)$$

Burada iki katmanlı doğrusal ağ, iki ağırlık vektörünün çarpımını ağırlık vektörü olarak alan bir ağ ile eşit durumdadır. Burada tek katmanlı ağların ve doğrusal aktivasyon fonksiyonlarının hesaplama güçlerinin sınırlı olduğunu söyleyebiliriz.

4.2.7 Tekrarlanan Ağlar

Bu noktaya kadar incelenen ağlarda geri besleme bağlantısı yoktur. Bu olay bir tabakanın çıktısının aynı ya da daha evvelki bir tabakaya girdi olarak bağlanmasıdır. Bu özel ağlar, tekrarlanmayan veya ileri beslemeli ağlar olarak adlandırılır ve geniş uygulama alanı bulur.

Daha genel ağlar geri besleme bağlantısı içerir ve tekrarlanan adını alır. Tekrarlanmayan ağların belleği yoktur. Çıktıları aktif olan girdileri ve ağırlıkları ile belirlenir. Bazı gösterimlerde tekrarlanan ağlar geçmiş çıktılarını tekrar çevrime sokarlar. Bu açıdan onların çıktıları hem aktif girdi hem de daha önceki çıktılardır. Bu nedenle tekrarlanan ağların çıktıları kısmen geçmiş girdilere dayandığından, insan hafızasına benzer davranış sergilerler (Haykin, 1994).

4.2.8 Yapay Sinir Ağlarının Eğitilmesi

Yapay sinir ağlarının en önemli özelliği öğrenme yeteneğidir. Öğrenmeleri insanların zeka gelişimiyle paralellik içerir. Bu açıdan öğrenme olayının belli başlı kısımlarını incelemek durumundayız. Ancak bu olayın sağlıklı olmasına çok dikkat edilmelidir. Çünkü yapay sinir ağlarında öğrenme sınırlıdır ve doğru yolda olsak bile çözümlemeden önce birçok zorlukla karşılaşabiliriz. Ağın eğitimi, girdi kümesinin uygulanması ile erişilmek istenen çıktıların elde edilmesidir. Her girdi ve çıktı vektörle gösterilir. Eğitilme, girdi vektörlerinin sırayla uygulanması ile yapılır. Bu arada ağın ağırlıkları daha önceki prosedüre göre uygulanır. Eğitim esnasında ağ ağırlıkları büyüklük olarak değerlere dönüşür ve her girdi vektörü beklenen çıktı vektörüne dönüştürülür.

Eğitim algoritması denetimli, denetimsiz ve destekli olarak üç grupta incelenir. Denetimli eğitimde, her girdi vektörünün beklenen çıktıyı gösteren bir hedef vektörüyle eşlenmesi gerekir. Bu ikili eğitim çifti olarak adlandırılır. Genel olarak ağ, bu çiftlere dayanarak eğitilir. Girdi vektörü uygulanır. Ağın çıktısı ona ait hedef vektörü ile değerlendirilir ve fark (hata) geri beslemeye tabi tutulur. Böylece ağırlıkları da hatayı minimize etmeye yönelik değiştirilir. Eğitim seti vektörleri sırayla uygulanır ve hatalar hesaplanır. Her vektöre ait ağırlıklar hatanın eğitim kümesi için kabul edilebilir seviyede olacağı biçimde ayarlanır.

Denetimli eğitim birçok uygulama başarısına rağmen biyolojik olarak imkansız olduğu söylenerek eleştirilmiştir. Beklenen ve oluşan çıktıları karşılaştırıp geri beslemeye tabi tutan ve düzelten bir beyin düşünmek oldukça zordur. Eğer bu beynin işlem süreci olsaydı beyne arzu edilen bu çıktılar nereden gelecekti? Bir çocuk ilk gelişme seviyesinde ortaya çıktığı bilinen düzeltmeleri nasıl yapacaktı? Denetimsiz eğitim, biyolojik sisteme çok daha iyi uyum sağlayan bir modeldir ve Kohonen (1984) ve diğer birçok araştırmacı tarafından geliştirilmiştir. Herhangi bir hedef vektörüne ihtiyaç duymaz ve böylece daha önce belirlenmiş ideal durumlarla mukayeseye tabi tutulmaz. Eğitim seti sadece girdi vektöründen oluşur. Eğitim algoritması uyumlu çıktılar içermek için ağ ağırlıklarını düzenler. Yani bir vektör ya da benzeri diğer bir vektör uygulaması yaklaşık aynı çıktıyı verir. Eğitim prosesi, bu açıdan eğitim setinin istatistiksel özelliklerini çıkartır ve benzer vektörleri gruplar. Belirli bir sınıftan vektörün uygulanması, belirli bir çıktı vektörü oluşturacaktır. Ancak önceden herhangi bir çıktı tipinin hangi sınıftaki vektörden elde edileceğinin tahminini kesin olarak yapmak zordur. Bu tür bir ağın çıktılarının eğitim prosesine yardım edecek bir duruma dönüştürülmesi zor değildir. Genel olarak girdi-çıkı ilişkisinin tanımlanması oldukça kolay bir olaydır.

Günümüzde birçok eğitim algoritması Hebb'in belirttiği konulardan türetilmiştir. Hebb bu konuda sinaptik gücün (ağırlığın) yükseltildiği denetimsiz bir model ortaya koymuştur. Böylece ağ genel olarak kullanılan yolla kuvvetlendirilir ve tekrarla öğrenme konusu açıklanabilir. Yapay sinir ağı Hebb'in eğitimini kullanıyorsa ağ ağırlıklarını kaynak ve hedef sinir hücresinin aktifleşme seviyesinin sonucunu kullanarak yükseltebilir. Her ne kadar son 20 yıldır daha verimli eğitim algoritmaları geliştirilse de ağlar hala Hebb'in eğitimini kullanan sistemlere göre oluşturulur (Haykin, 1994).

4.3 Yapay Sinir Ağlarında Bilgi Gösterimi

Herhangi bir yapay sinir ağı uygulamasının temel görüşlerinden biri de ağın gerçekten ne yaptığını anlamaktır. Örneklerden öğrenilen ve bir problemi çözmek için uydukları özel algoritmaları bulunmayan ağlar için ağ tasarımcısı eldeki problemin ağ çözümüne yeterliliğini belirlemelidir. Bu, çoğunlukla ağda bilginin nasıl saklandığını anlamayı kolaylaştırır.

Yerleştirilen bilgi dahilinde ve çalıştırılmış bir sinir ağı tarafından kullanılan süreç; belirleme, sonuç çıkarma, analiz ve tahmin elemanlarına sahiptir. Uzman sistemlerce bilinen kurallar ve gerçekler listesini çıkarmak kolay bir işlem değildir. Yine de çalıştırılan ağdaki birleştirilmiş bilgiyi mukayese ve geçerlilik için çıkarım, şüphesiz aldatıcı ve zaman alıcı olabilir. Nörobiyolojistlerin beyin işlevlerini anlaması gibi, ağların bir problemi nasıl çözdüğüyle ilgilenen yapay sinir ağı araştırmacıları da enformasyon dedektif gibi davranmak durumunda kalmışlardır. Sıradan bir bilgisayar programında bilginin belirli bir depolama şekli vardır. Bir program tarafından korunan iki çeşit bilgi vardır: Programın kendisini içeren bilgilerin listesi ve program tarafından kullanılan değişkenlerin değerleri. Verilen herhangi bir zamanda bilgisayarın ne bildiğini tamamen bulma, geçerli bilgi listesini ve bütün değişkenlerin değerlerini çıkarmakla başarılabilir. Bu, programda kapsanan bütün bilgilerin tam ve kusursuz bir sayımını sağlar.

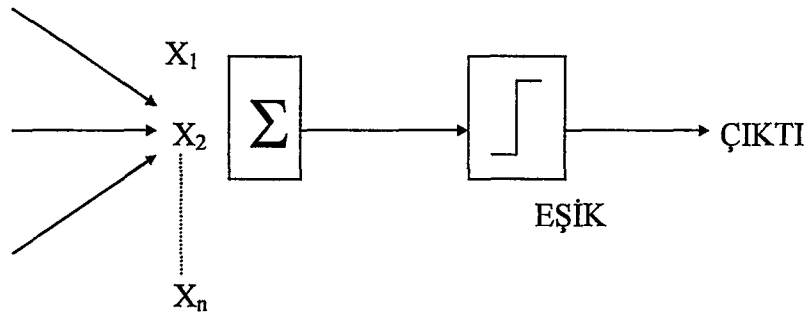
Bir yapay sinir ağında, ağırlıklara ve bağlantılara kolayca erişilebilirken, bir uygulamacının verdiği kararın güvenilirliği ile ilgili kararın problemi belirlilikten uzak olabilir. Bilgisayar programında bulunan kolayca anlaşılmış bilgiye karşıt olarak, sinir ağının bilgisi bir kişiye karmaşık ve anlaşılmaz gelebilir. Uygulamalarda kullanılan bir çok sinir ağı üç veya daha fazla tabakaya sahiptir. Bu tabakaların ne yaptığı hakkındaki açıklama şu şekildedir: Girdi tabakası ağ boyunca modeli dağıtır, çıktı tabakası uygun cevabı meydana getirir ve orta tabaka detektörlerinin özelliklerini toplama rolünü oynar. Orta tabakadaki her özellikli detektör sinir hücresini, modeldeki özelliği veya özellikleri araştırır ve birini bulduğunda

kuvvetlice tepki verir. Bu durumda çıktı tabakası, orta tabakanın meydana çıkardığı özelliklerin belirli birleşimlerini temel alan uygun çıktı modelini kurabilir.

Yapay sinir ağlarının çalışmasına, çıktı tabakasından alınmak istenen cevaba uygun olarak orta tabakada uygun bir özellik detektör kümesini kuran proses olarak bakılabilir. Ağı eğitmek için kullanılan eğitime setinde girdi ve çıktılar arasında sabit bir ilişki varsa, böyle bir sistem için ağı eğitmeye gerek yoktur. Böyle bir ilişkiler seti yapay sinir ağının uygulama mantığına ters düşer. Eğitime setinde önemli özellikler çok iyi vurgulanmalıdır. Böyle olursa ağın eğitimi daha mükemmel gerçekleşir. Eğer ağ tasarlayıcıları ortaya konan problemi çok iyi biliyorsa, ağı çok mükemmel olarak tasarlayabilirler. Bunun sonucunda da sonuca daha çabuk ulaşabilirler. Eğer ağ tasarımcısı problemi iyi bilmiyorsa, ağı eğitmek zaman alır. Ama verilen ağ tasarımı eksik bile olsa ağın eğitimi tamamlanır. Ağ eğitildikten sonra, onun verilen problemi çözüp çözemeyeceği incelenecektir. Bunun için ağın test edilmesi gerekir. Ağın öğretilmiş olay hakkında neyi bilip bilmediğini düşünmesi için ağı sonuçları bilinen olaylar verilmekte ve sonuçlar incelenmektedir (Zurada,1992).

4.4 Algılayıcı ile (Perceptron) Öğrenme

Yapay sinir ağlarında ilk anlamlı çalışmalar 1940'lı yıllarda başlamıştır. Araştırmacılar insan beyninin biyolojik bir sinir hücresi olarak modellendirmeyi düşünmüşlerdir. Bu arada insan beyninin çalışma sistemi araştırılmıştır. Mc Culloch ve Pitts (1943) yapay sinir ağları hakkındaki ilk sistematik çalışmayı yayınlamışlardır. Bunların çalışmalarının bir çoğu sinir hücresi modeliyle kısıtlı kalmıştır. Şekil 4.7'de basit bir sinir hücresi modeli görülmektedir.



Şekil 4.7 Algılayıcı (perceptron) sinir hücresi

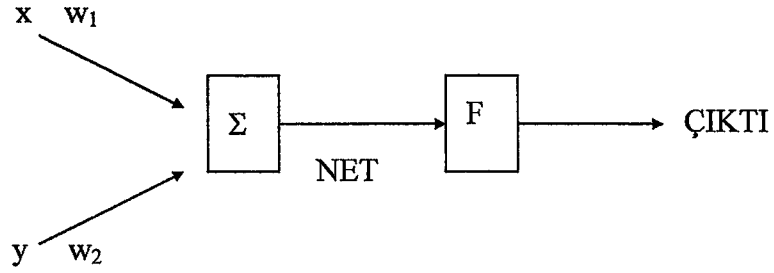
Toplama birimi herbir girdiyi, verilen ağırlıkla çarpar ve sonuçları toplar. Eğer bu toplam önceden belirlenmiş eşik değerinden büyükse çıktı değeri 1, diğer durumda ise 0'dır. Bu sistemler ve bunların çeşitli türleri de algılayıcı (perceptron) adını alır. Bu sistemler bir tek katmandan ibarettir. Ama daha karmaşık sinir ağları da aynı isimle adlandırılmaktadır.

1960'lı yıllarda bu algılayıcılar, büyük ilgi ve ümit meydana getirdiler. Roseblatt 1962'de algılayıcı öğrenme hakkında dikkate değer bir teorem ortaya koydu. Widrow ve Holf (1960), Widrow ve Angell (1962), Widrow ve Angell (1963) algılayıcı türü sistemlerini ortaya koydular. Araştırmacılar da bu sistemlerin üzerinde büyük bir istekle durdular. Minsky, bu problemi büyük bir çabayla ve kararlılıkla analiz edip, tek katmanlı algılayıcının ne yapabileceği ve ne öğrenebileceği konusundaki önemli kısıtları gösterebilmiştir. Burada çok katmanlı sinir ağları hakkında yeterli bilgi olmadığından araştırmacılar başka alanlara yönelmişlerdir. Bunun sonucunda daha önce de belirttiğimiz gibi yapay sinir ağları konusundaki çalışmalar bir duraksamaya uğramıştır. Daha sonra yeni metotların keşfedilmesiyle ilgi ve araştırma çabaları yeniden canlanmıştır.

Minsky'nin çalışmaları belki algılayıcı ile ilgili gayretleri azaltmıştı. Ama yapay sinir ağları konusunun gelişmesinde bir geçiş sağlamıştır. Ayrıca Minsky'nin analizleri çürütülememiş ve önemini de korumuştur. Araştırmacılar son yıllarda algılayıcıları geniş bir şekilde kullanmaya başlamışlardır. Algılayıcı ile öğrenme birçok yapay sinir ağı uygulamaları için bir temel teşkil etmiştir. Yani yapay sinir ağı çalışmaları için mantıksal başlangıç noktasıdır (Haykin, 1994).

4.4.1 Algılayıcı ile Gösterim

Roseblatt'ın algılayıcı ile öğrenme teoreminin ispatı, uygun bir şekilde temsil edilen herşeyin algılayıcı tarafından öğrenilebileceğini göstermiştir. Öğrenme ve bilginin temsil edilmesi arasında büyük bir bağlantı vardır. Bilginin uygun gösterimi algılayıcının ya da diğer sinir ağlarının öğrenme yeteneğine referans teşkil eder. Ayrıca öğrenme işlemi de, yapay sinir ağlarının ağırlıklarının ayarlanabilmesi için sistematik bir prosedürü gerektirir. Örneğin, 0'dan, 9'a kadar numaralanmış bir kart dizisini ele alalım. Elimizde bir makine olduğunu ve bu makinenin de tek ve çift numaralı kartları ayırt edebildiğini varsayalım. Mesela tek numaralı kartları görünce makinenin lambasının yandığını düşünelim. Bu makineyi biz bir algılayıcı ile temsil edebiliriz. Bilginin temsil edilebilme yeteneği açısından bakacak olursak, tek katmanlı algılayıcının bu konuda ciddi kısıtları vardır. Burada iki girdi değeri söz konusudur. Bu değerler 0 veya 1 olabilir. Burada bir çıktı değeri vardır ve bu çıktı değeri de 0 veya 1 olabilir (Haykin, 1994).



Şekil 4.8 Tekli sinir hücresi sistemi

4.4.2 Doğrusal Ayrılabilirlik

Çok geniş sayıda fonksiyon tek katmanlı sinir ağı ile temsil edilemez. Bu tip fonksiyonlara doğrusal ayrılamaz fonksiyonlar denir ve bunlar tek katmanlı sinir ağlarının kapasitelerini belirler. Eğer doğrusal ayrılabilirlik varsa, geometrik ayrılabilen noktalar setindeki sınıflama problemleri için tek katmanlı sinir ağları kullanılabilir. Burada verilen iki girdi durumunda ayıraç olarak düz bir çizgi kullanılmaktadır. Üç girdi durumunda sonuçlar üç boyutlu uzayda olacaktır. Daha fazla girdi durumunda ise gösterim işlemi yapılamayacaktır. Doğrusal ayrılabilirlik bir algılayıcının gösterim yeteneğini sınırlar. Bu bakımdan verilen fonksiyonun ayrılabilir olmasının bilinmesi çok önemlidir. Fakat çok sayıda değişken varsa bunu tespit etmek kolay değildir. n adet ikili (binary) girdi içeren bir yapay sinir ağı iki değişik girdi modeli içerebilir. Her bir girdi modelinin 0 ve 1 olmak üzere iki farklı çıktı üretebilmesi sebebiyle n tane değişkenin $(2^2)^n$ adet farklı fonksiyonu ortaya çıkacaktır. Çizelge 4.1’de tesadüfi olarak seçilmiş doğrusal ayrılabilir fonksiyon sayısı görülmektedir.

1960’ların sonunda doğrusal ayrılabilirlik problemi çok iyi anlaşılmıştır. Tek katmanlı yapay sinir ağının çok ciddi gösterim sınırlamaları vardı. Bunun da üstesinden katman sayısı çoğaltılarak gelinmiştir. Mesela iki katmanlı yapay sinir ağı, tek katmanlı iki yapay sinir ağının birleştirilmesiyle şekillenmiştir. Bu işlem daha genel ayrıştırma ve sınıflama işleminde, konveks açık ya da kapalı alanlar içindeki noktalarda başarı gösterebiliyordu. Konveks bir bölgede alınan iki nokta bir çizgi ile birleştirilebilir. Alanın dışına çıkmadan bu işlem yapılabilir. Konveksliği anlayabilmek için basit iki katmanlı bir sinir hücresini ele alalım. İlk katmanda iki sinir hücresini ve buna bağlı ikinci katmandaki tek bir çıktı sinir hücresini ele alalım. Bunun eşik değeri 0.75 ve ağırlıklar 0.5 olsun. Bu durumda çıktı katmanı bir tek sinir hücresine ihtiyaç duyar. Buradan da çıktı üretir. Sonuç olarak burada algılayıcı, mantıksal bir fonksiyon olarak çalışır (Zurada, 1992).

Çizelge 4.1 Doğrusal ayrılabilir fonksiyonlar

N	$(2^2)^n$	Doğrusal ayrılabilir fonksiyonların sayısı
1	4	4
2	16	14
3	256	104
4	65536	1882
5	$4.3 \cdot 10^9$	94572
6	$1.8 \cdot 10^{19}$	5.028.134

4.4.3 Algılayıcı (Perceptron) Öğrenme Algoritması

Yapay sinir ağlarının öğrenme yeteneği onun en çok ilgi çeken özelliğidir. Yapay sinir ağları aynı biyolojik sinir ağları gibi modellenirler. Bir bütün olarak bir sinir ağı kendini sürekli olarak yeniler ve düzenler. Yukarıda bahsettiğimiz doğrusal ayrılabilirlik kriterini kullanarak tek katmanlı yapay sinir ağının istenilen fonksiyonu temsil edip etmediğine karar vermek mümkündür. Eşik değeri ve ağırlıklar için ihtiyaç duyulan modeli elde etmenin bir yolu yoksa, doğrusal ayrılmazlık sorusuna evet cevabı bizim için faydalı olacaktır. Eğer yapay sinir ağı pratik değerler içinse, sistematik bir metoda ihtiyaç duyulacaktır. Burada öğrenme, denetimli (supervised), denetimsiz (unsupervised) ya da destekli (reinforcement) olabilir. Denetimli öğrenme sisteminin istenilen şekilde olabilmesi için, direk olarak birbiri ardınca düzenlenmesi gerekir. Denetimsiz öğrenmede öğreticiye ihtiyaç yoktur. Sistem istenilen değişiklikleri üretmek için kendi kendini organize eder. Algılayıcı (perceptron) öğrenme denetimli bir öğrenme şeklidir. Yapay sinir ağının çalıştırılması işlemine eğitim adı verilir. Bu işlem sonunda sinir ağı öğrenme işlemini tamamlar. Destekli öğrenmede ağı, üretilen çıktının doğru ya da yanlış olduğu söylenir.

Algılayıcı ile öğrenmede model seti olarak girdiler alınır ve istenilen çıktı değeri elde edilinceye kadar eğitme işlemine devam edilir. Bir örnek olarak rakam tanıma işlemini ele alalım. Her rakamı 12 adet kutucuk içinde düşünelim. Sayıları bu şekilde yerleştirdikten sonra içi dolu olan kutu için 1, boş olan kutu için 0 sayısını kullanalım. Mesela, “3” rakamını bu

kutucuklara yerleştirirsek, boş kalan kutucuklara 0, dolu olanlara 1 verebiliriz. Dolayısıyla 0 ve 1'den oluşan bir model elde edebiliriz. Burada X girdi vektörü olarak ele alınacaktır. Doğal olarak vektördeki sayıların her biri 0 ve 1 rakamlarıyla kodlanmış olacak ve herbiri bir girdi değerinden işleme sokulacaktır. Daha sonra bir başlangıç ağırlık seti verilecek ve öğrenme işleminin sonunda ağırlıklar tamamlanacaktır. Algoritmayı özetleyecek olursak;

1. Girdi modelini uygula ve çıktı değerini hesapla,
2. a) Eğer çıktı doğru ise tekrar 1. adıma git
b) Eğer çıktı yanlış ve 0 ise herbir girdiyi uygun ağırlıklarla topla
c) Eğer çıktı yanlış ve 1 ise herbir girdiyi uygun ağırlıklardan çıkar.
3. 1. adıma git (Schalkoff, 1997).

4.4.4 Genelleştirilmiş Delta Kuralı

Algılayıcı, öğrenmenin önemli algoritmalarından biridir. Sürekli girdi ve çıktı durumlarında kullanılabilen bir tekniktir. Burada yapay sinir ağlarının hedef çıktısı D ve gerçek çıktısı da A olsun. Bu durumda hata δ ;

$\delta = D - A$ ile bulunur. Burada;

$\delta = 0$ ise 2a adımına,

$\delta > 0$ ise 2b adımına,

$\delta < 0$ ise 2c adımına gidilir.

$$\Delta_i = \lambda * \delta * X_i$$

$$W_{i(n+1)} = W_{i(n)} + \Delta_i$$

burada Δ_i , ağırlıkları düzeltme katsayısıdır. Δ_i 'yi hesaplamak için, hata değeri δ ile X_i değeri ve öğrenme katsayısı λ çarpılır. Bu değer bir önceki ağırlığa eklenir ve yeni ağırlıklar hesaplanır (Kennedy ve Chua, 1988).

4.5 Geriye Yayma Algoritması (GYA - Backpropagation)

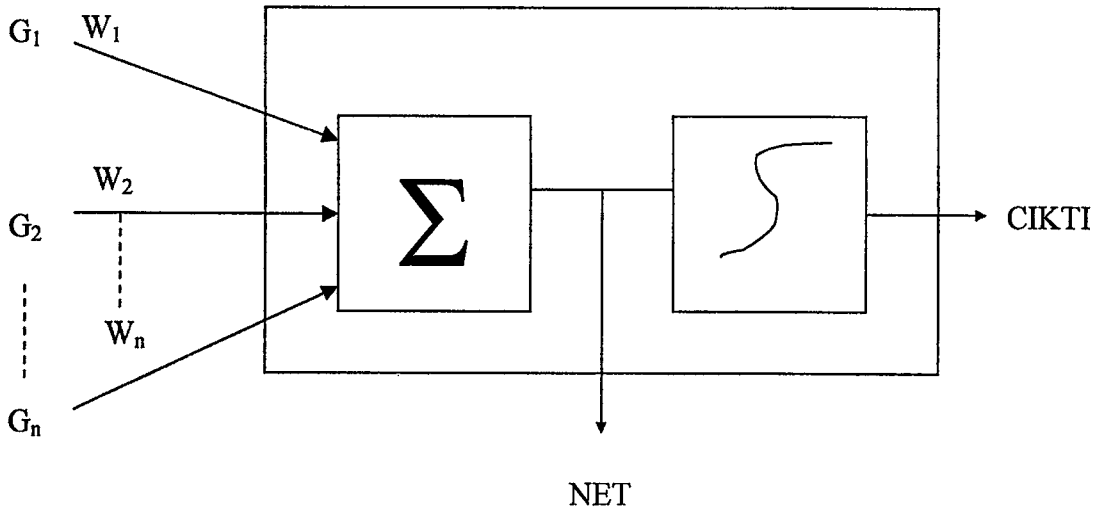
Yıllarca çok katmanlı yapay sinir ağları ile ilgili çalışmalardan bir sonuç alınamadı. Tek

katmanlı yapay sinir ağıları ancak birkaç sınırlı imkan tanıdı. Fakat daha ötesine gidilemedi. GYA'nın keşfi yapay sinir ağlarına olan ilgiyi tekrar canlandırdı. GYA çok katmanlı yapay sinir ağları için sistematik bir metottur ve çok güçlü bir matematik temele sahiptir. Bunun yanında her türlü kısıta rağmen GYA yapay sinir ağlarının uygulanabildiği problem aralığının genişlemesine yol açmıştır ve çok sayıda başarılı ve güçlü uygulamaların üretilmesine imkan vermiştir.

GYA algoritmasının ilginç bir tarihsel gelişimi vardır. Rumelhart, Hinton ve Williams 1986 yılında GYA algoritmasını açık ve net biçimde ortaya koymuşlardır. Bu araştırmaya ilk başlayanlardan birisi Parker olmasına rağmen, bu algoritmayı ilk olarak 1974 yılında Werbos ortaya koymuştur. Rumelhart ve Parker, Werbos'un bu çalışmasını geliştirmek için büyük bir çaba harcadılar ve böylece algoritma büyük bir gelişim sağlamış oldu. Yapay sinir ağları konu itibariyle disiplinler arası olması sebebiyle daha ciddi bir yapıya sahiptirler. Disiplinler arası olması, her yere kolayca adapte edilebilmesi nedeniyle çok çabuk gelişmiş ve konuyla ilgili kayda değer çalışmalarda bulunulmuştur. Bu da konunun gelişimini otomatik olarak hızlandırmıştır (Haykin, 1994).

4.5.1 Yapay Sinir Ağlarının Teknik Yapısı

Şekil 4.9'da yapay sinir hücresi GYA ağları için yapısal bir kuruluş bloğunu göstermektedir. Şekilde görüldüğü gibi yapay sinir ağına bir veri seti uygulanır. Bunların herbiri belirli ağırlıklarla çarpılır ve sonuçlar toplanır. Bu sonuçların toplamı NET olarak isimlendirilir ve bu ağıdaki herbir yapay sinir için hesaplanmak zorundadır. NET hesaplandıktan sonra bunu düzeltmek için F aktivasyon fonksiyonu uygulanır ve böylece ÇIKTI sinyali elde edilir.



Şekil 4.9 Aktivasyon fonksiyonu ile bir yapay sinir hücresi

$$NET = G_1 W_1 + G_2 W_2 + + G_N W_N = \sum_{I=1}^N G_I W_I \quad (4.9)$$

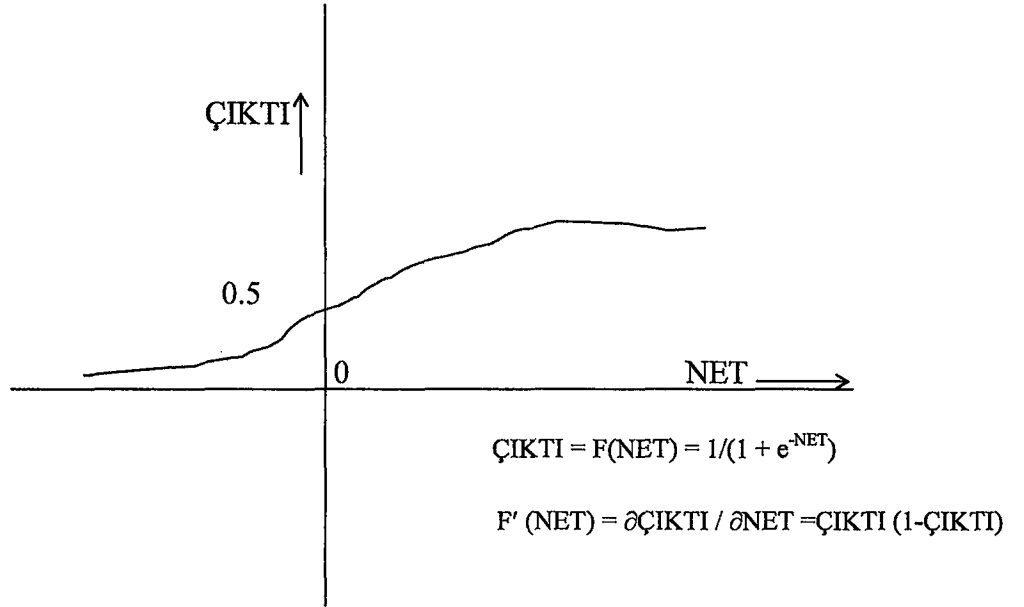
$$\text{ÇIKTI} = F(NET) \quad (4.10)$$

$$\text{ÇIKTI} = \frac{1}{1 + e^{-NET}} \quad (4.11)$$

$$F'(NET) = \frac{\partial \text{ÇIKTI}}{\partial NET} = \text{ÇIKTI}(1 - \text{ÇIKTI}) \quad (4.12)$$

Şekil 4.10'da GYA için kullanılan bir aktivasyon fonksiyonu görülmektedir. Görüldüğü gibi aktivasyon fonksiyonu sigmoidal bir fonksiyondur. Bu fonksiyonun istenilen şekilde türetilmesi de kolaydır. Bu fonksiyonu GYA'da öğrenme eğrisi olarak kullanmaktayız. Bu fonksiyon lojistik fonksiyon olarak da adlandırılır. Sigmoid fonksiyon NET'in aralığını daraltır. Bu nedenle ÇIKTI 0 ile 1 arasında değişir.

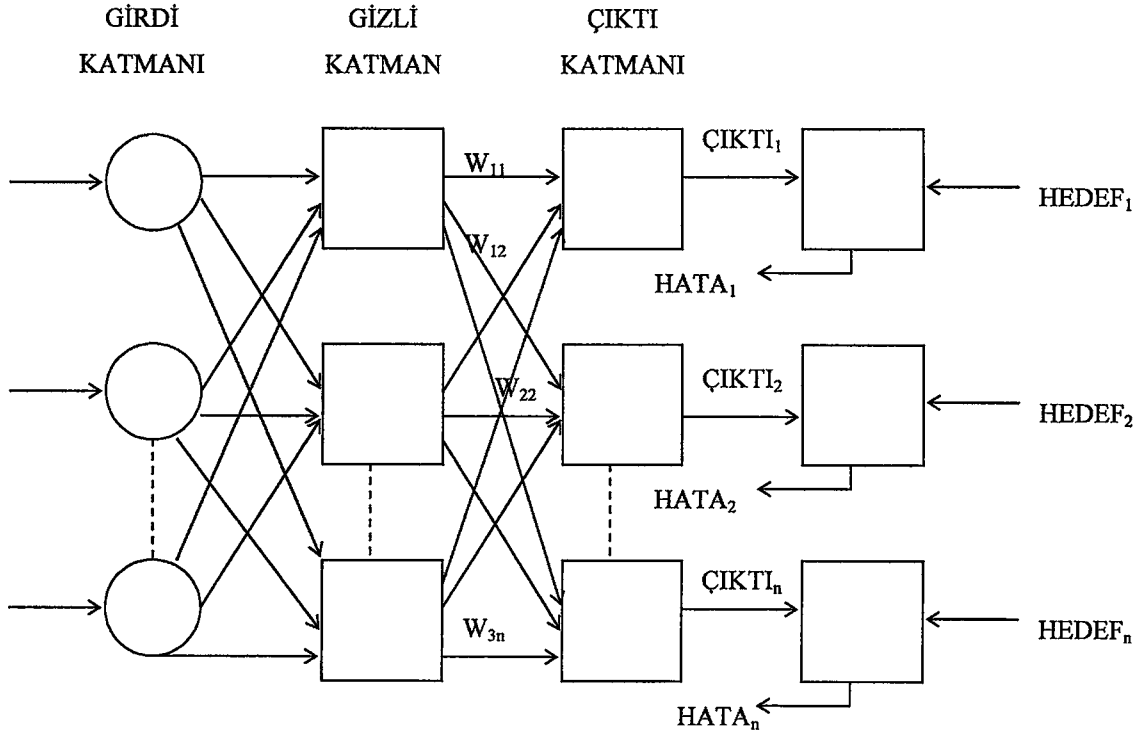
Eğer çok katmanlı yapay sinir ağlarına, doğrusal olmayan bir giriş varsa tek katmanlı ağlara göre çok daha etkili bir temsil gücüne sahiptirler. Burada ihtiyaç olan yapıyı sigmoid fonksiyon karşılamaktadır. Aktivasyon fonksiyonu olarak daha farklı bir fonksiyon da kullanılabilir. Fakat GYA her yere uyum sağlayıp, değişebilen bir fonksiyon yapısı gerektirir. Sigmoidal fonksiyon ise bu ihtiyaçlara cevap verebilecek niteliktedir. Sigmoidal fonksiyonun otomatik bir kontrol yapısı olduğundan bazı avantajları mevcuttur. Şekil 4.10'da görüldüğü gibi eğrinin eğilimi küçük sinyaller yani NET değeri sıfıra yakın değerler için dikine hızla artan şekildedir ve yüksek çıktı değerleri üretir. Sinyalin değeri daha da arttığı zaman fonksiyon dengeye gelir. Yani geniş sinyaller doyma noktasına gelmeden yapay sinir ağları tarafından birbirine uyumlu hale getirilir. Bu sayede sinyalin aşırı zayıflamasına imkan vermeksizin sinyalin içeriden geçmesine izin verilir (Haykin, 1994).



Şekil 4.10 Sigmoidal aktivasyon fonksiyonu

4.5.2 Çok Katmanlı Yapay Sinir Ağları

Şekil 4.11’de GYA ile çalışan çok katmanlı bir yapay sinir ağı görülmektedir. Burada ilk sinir hücre seti sadece dağılım noktalarını gösterir. Herhangi bir girdi toplamı yapmazlar. Girdi sinyalleri kendi çıktıları üzerindeki ağırlıkların içinden geçerler. Sonradan gelen katmanlardaki herbir sinir hücresi yukarda anlatıldığı gibi NET ve ÇIKTI sinyallerini üretir. Katman sayısını tesbit etmede en iyi yöntem şu an için deneme yanılma yöntemidir. Burada dikkat edilecek olursa şekildeki yapay sinir ağı iki katmandan oluşmaktadır. Sinir hücreleri ağırlık setiyle birleştirilir. Girdi katmanı sıfırıncı katman olarak belirlenmektedir.



Şekil 4.11 İki katmanlı geriye yayma algoritması

4.5.3 Yapay Sinir Ağı'nın Çalıştırılması

Yapay sinir ağı'nın çalıştırılma amacı GİRĐİ setini uygulamak suretiyle istenilen ÇIKTI'ları almaktır. Yani yapay sinir ağı'nda kullanılan ağırlıkları elde etmektir. Ağırlıkları elde etmek ise, öğrenme işlemi'ni tamamlamak demektir. Yapay sinir ağı'nın yapısından da anlaşılacağı gibi ancak uygun ağırlıklarla istenilen çıktı setleri elde edilebilir. Burada GİRĐİ ve ÇIKTI setlerini bir vektör olarak ifade edecek olursak yapay sinir ağı'nın çalışma sistemi herbir GİRĐİ vektörünü, bir ÇIKTI vektörüyle eşleştirme şeklindedir. Olayı bir örnekle açıklayalım. Alfabetik harfler 0 ve 1 şeklinde kodlayalım ve bunun içindeki harfleri çizgilerle ayıralım ve içinden çizgi geçen kutu için 1, geçmeyen kutu için 0 değerini verelim. Mesela A harfini temsil etmek için sinir hücresinin girdileri sıfır ve birlerden ibaret olacaktır. Burada çıktı olarak A harfini temsil edecek bir sayı kullanabiliriz. Eğer alfabedeki tüm harfleri kullanacak olursak harf sayısı kadar eğitim setine (training set) ihtiyacımız olacaktır. Yapay sinir ağı çalıştırılmadan önce ağırlıkların hepsine çok küçük rassal sayılardan ibaret değerler verilmelidir. Tabii bunun amacı yapay sinir ağı'nın büyük ağırlık değerleriyle şişmesini önlemektir. Mesela tüm ağırlıkların değerleri eşit olsa, yapay sinir ağı öğrenme işlemi gerçekleştirilemeyecektir. Bir yapay sinir ağı'nda GYA'nın çalışması aşağıdaki adımları gerektirir:

1. Eğitim setini hazırla ve girdi vektörünü yapay sinir ağına uygula

2. Yapay sinir ağının çıktı değerini hesapla
3. Yapay sinir ağının çıktısı ile istenilen çıktı arasındaki hatayı hesapla
4. Hataları minimize edecek şekilde hatayı geriye yay ve yapay sinir ağı ağırlıklarını hesapla
5. Hatalar kabul edilebilir seviyede olana kadar iterasyonu sürdür.

Burada hesaplamalar katman katman yapılacaktır. Şekil 4.11 temel alınırsa ilk katmandaki yapay sinirin ilk çıktıları hesaplanır ve bunlar sonraki katmanın girdisi olarak kullanılır. Bu katmanın da yapay sinirinin çıktıları hesaplanır ve bunlar da bu yapay sinir ağının çıktı vektörüdür. 3. adımda yapay sinir ağının çıktılarından herbiri şekil 4.11’de olduğu gibi ÇIKTI olarak etiketlenir. Bundan sonra HATA hesaplaması yapılır. Bu hata 4. adımda ağı ağırlıklarını hesaplamak için kullanılır. Burada itme ve çekmeyle değişen ağırlıklar eğitim setiyle tespit edilir. 4.adım yeteri kadar tekrar edildikten sonra gerçek çıktılar ile hedef çıktılar arasındaki hata kabul edilebilir bir değere indirgenir. Böylece yapay sinir ağı öğrenme işlemini tamamlayarak, ağırlıklar son halini bulmuştur. Artık yapay sinir ağı tanıma işlemi için kullanılabilecektir. Kolayca anlaşılabileceği gibi 1. ve 2. adımlar ileri doğru bir geçiş (forward pass) sağlarlar. Yani sinyaller yapay sinir ağının girdi katmanından çıktı katmanına doğru toplanırlar. 3. ve 4. adımlar ise tersine bir geçiştir (reverse pass). Burada hesaplanan hata sinyali ağı içinden geriye doğru yayılır ve nihai ağırlıklar hesaplanır.

4.5.3.1 Yapay Sinir Ağı Üzerinde İleri Doğru Geçiş (Forward Pass)

1. ve 2. adım, vektör formunda aşağıdaki gibi açıklanabilir. X girdi vektörü uygulanır ve çıktı vektörü elde edilir. Görüldüğü gibi hesaplama çok katmanlı yapay sinir ağı üzerinde aşama aşama yapılır ve hesaplama girdilere en yakın vektör üzerinden başlanır. Herbir sinir hücresindeki NET değeri, sinir hücresinin girdilerinin ağırlıklı toplamı olarak hesaplanır. F aktivasyon fonksiyonu bu aşamadaki herbir sinir hücresi için ÇIKTI değerini üretmek amacıyla kullanılır. İlk olarak bir katman için çıktı seti bulunur. Bu set girdi olarak diğer katmana sunulur. Bu işleme, en son bize sonuç olacak çıktı seti üretilene kadar katman katman tekrar edilir.

Bu işlem kısaca bir vektör notasyonunda yazılabilir. Sinir hücreleri arasındaki ağırlıklar bir W matrisi olarak kabul edilebilir. Mesela 8. sinir hücresinden 3. katmandaki 5. sinir hücresine $W_{8,5}$ diyelim. N katmanı için NET vektörü X ve W’nin bir sonucu olarak ortaya konulur. Yani

vektör notasyonu olarak $N = XW$ kullanılır. Burada NET vektörü olan N , F fonksiyonuna uygulanır. Böylece ÇIKTI vektörü hesaplanır. Bir katman için hesaplanan ÇIKTI vektörü, diğeri için GİRİDİ vektörü olmaktadır. Sonuç katmanındaki ağırlıkların hesaplanması 4.13 eşitliğinin tüm katmanlara uygulanmasına bağlıdır.

$$\mathcal{C} = F(XW) \quad (4.13)$$

4.5.3.2 Yapay Sinir Ağı Üzerindeki Geriye Doğru Geçiş (Reverse Pass)

Çıktı katmanının ağırlıklarının hesaplanması:

Elde çıktı katmanındaki herbir sinir hücresi için hedef bir değer olması sebebiyle, düzeltilmiş ve birleştirilmiş ağırlıklar Delta kuralının düzeltme etkisi kullanılarak kolayca elde edilebilir. İçerideki katmanlar gizli katmanlar (Hidden layer) olarak alınabilir ve bunların çıktıları hedef değer olarak kullanılamaz. Bu, işlemi daha da karmaşık hale getirir.

Şekil 4.12 gizli katman j 'deki p sinir hücresinden, k çıktı katmanındaki q sinir hücresine tek bir ağırlık için çalışma prosesini göstermektedir. k katmanındaki bir sinir hücresinin çıktısı HATA sinyalini üretmek için onun hedef değerinden çıkarılır. Bu katmanın sinir hücresi için ÇIKTI (1-ÇIKTI) şeklinde hesaplanmış aktivasyon fonksiyonunun türevi ile çarpılır. Böylece δ HATA değeri bulunmuş olur.

$$\delta = \text{ÇIKTI} (1-\text{ÇIKTI}) (\text{HEDEF} - \text{ÇIKTI}) \quad (4.14)$$

Daha sonra δ , j sinir hücresinin ÇIKTI'sı ile çarpılır. Bu sonuç dönüşte öğrenme oranı dediğimiz katsayı ile çarpılır. Bu sonuç da ağırlığa eklenir.

$$\Delta W_{pq,k} = \lambda \delta_{q,k} \text{ÇIKTI}_{pj} \quad (4.15)$$

$$W_{pq,k} (n+1) = W_{pq,k} (n) + \Delta W_{pq,k} \quad (4.16)$$

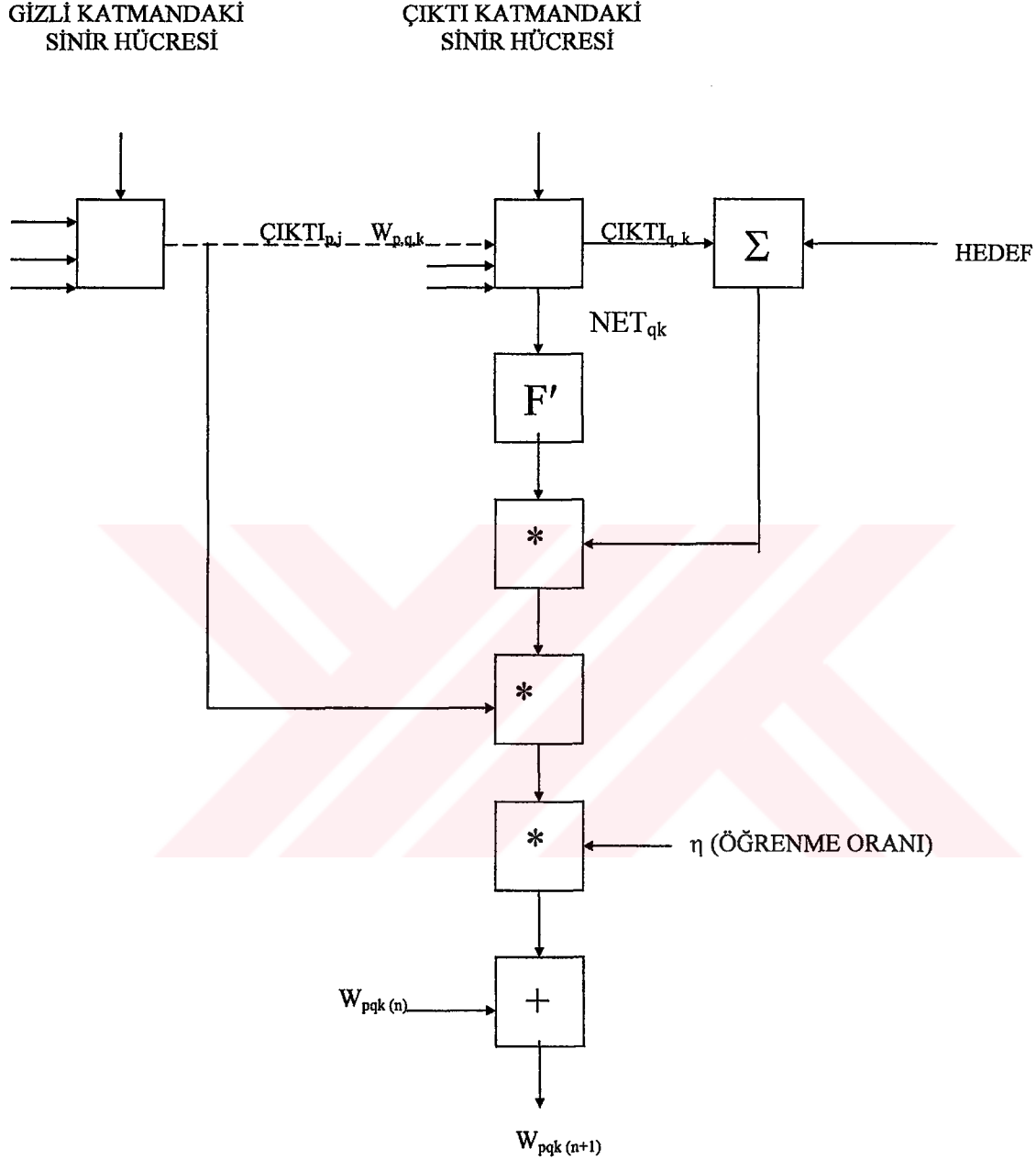
$W_{pq,k} (n)$ = Düzeltilmeden önceki n . adımda gizli katmandaki p sinir hücresinden, çıktı katmanındaki q sinir hücresine geçerken verilen ağırlığın değeri. Burada k altındisi onun gideceği katman ile birleştirildiğini ifade eder.

$W_{pq,k} (n+1)$ = $(n+1)$. adımdaki ağırlığın düzeltilmeden sonraki değeri

$\delta_{q,k}$ = k çıktı katmanındaki q sinir hücresi için sigma katsayısı

ÇIKTI_{pj} : j gizli katmanındaki p sinir hücresinin çıktı değeri

Burada p ve q altındisleri özel bir sinir hücresini referans alır. j ve k altındisleri ise bir katmanı temsil eder.



Şekil 4.12 Çıktı katmanındaki bir ağırlığın çalıştırılması

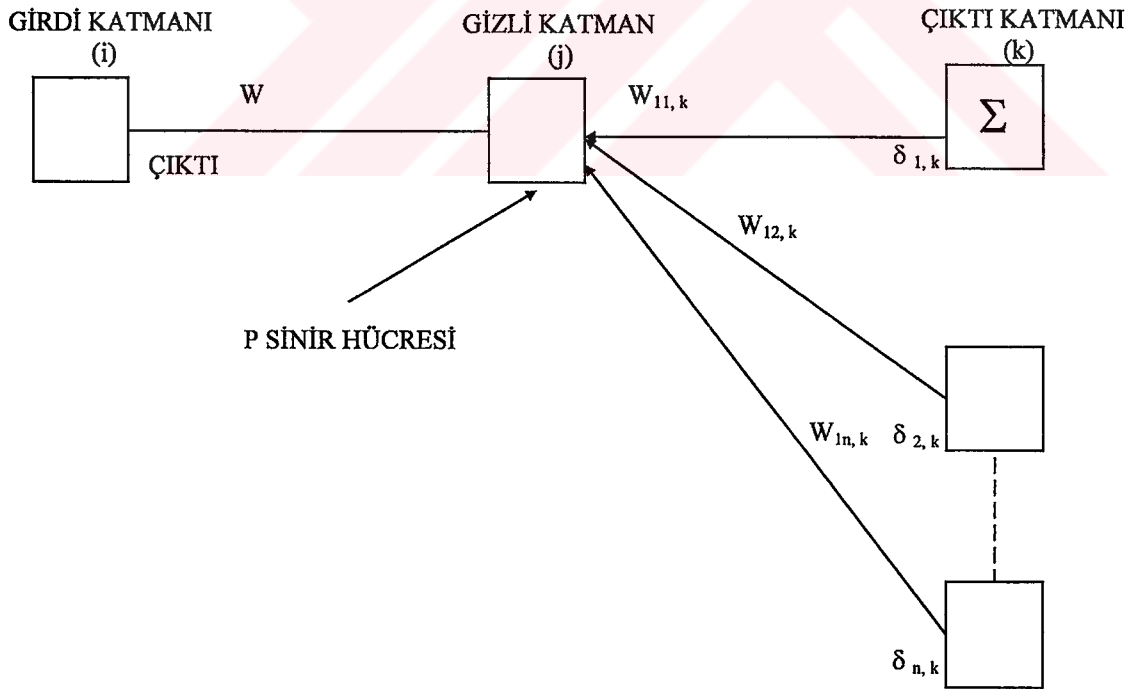
Gizli katmanın ağırlıklarının düzeltilmesi:

Gizli katmanlar hedef bir vektöre sahip değildirler. Burada geriye yayma, gizli katmanları ile çalışır, çıktıların hataları tersine bir işlemle aşama aşama geriye işletilir ve her bir katmanda düzeltilmiş ağırlıklar elde edilir. Hata δ , çıktı katmanındaki her bir sinir hücresi için hesaplanır. Bu, çıktı katmanının içine doğru verilen ağırlıkları düzeltmek için kullanılır. Daha sonra ilk gizli katmandaki hata (δ) için, her bir sinir hücresinde bir değer üretmek amacıyla

aynı ağırlıklar geriye doğru yayılır. δ değerleri bu gizli katmandaki ağırlıkları düzeltmek için kullanılır. Benzer yolla tüm önceki aşamalar için geriye yayılır. Çıktı katmanından önceki gizli katmanda bir sinir hücresi olduğunu kabul edelim. Geriye dönüşte bu sinir hücresi, çıktı değerlerini çıktı katmanındaki sinir hücrelerine ağırlıkları bağlayarak yayar. Bu işlem devam ederken çıktı katmanından gizli katmana geriye doğru δ değeri geçirilerek, ağırlıklar geriye doğru işleme tabi tutulur. Çıktı katmanına bağlayan sinir hücrelerinin δ değeri ile bu ağırlıklar çarpılır. Aktivasyon fonksiyonunun türetilmesiyle birlikte tüm sonuçlar ve çarpımların toplanması sonucu gizli katman sinir hücresinde ihtiyaç duyulan δ değeri hesaplanır.

$$\delta_{pj} = \text{ÇIKTI}_{pj} (1 - \text{ÇIKTI}_{pj}) (\sum \delta_{q,k} W_{pq,k}) \quad (4.17)$$

Şekil 4.13'e bakıldığında, eldeki δ değeri ile ilk gizli katmana verilen ağırlıklar (4.15) ve (4.16) eşitlikleri kullanılarak düzeltilebilir. Ayrıca doğru katmanları göstermek için indisler düzenlemeye tabi tutulur. Verilen gizli katmandaki her bir sinir hücresi için δ hesaplanmak ve tüm ağırlıklar düzeltilmek zorundadır. Bu işlem, geriye doğru tüm ağırlıklar düzeltilinceye kadar tekrarlanır. Vektör notasyonu ile geriye doğru hata yayılması hızlandırılabilir.



Şekil 4.13 Gizli katmandaki bir ağırlığın çalıştırılması

D_k çıktı katmanındaki δ_N setini ele alalım ve W_k çıktı katmanı için ağırlık seti olsun. D_j 'ye (gizli katman için δ vektörü) ulaşmak için iki aşamayı takip edeceğiz.

1. W_k çıktı katmanına gizli katmanı bağlayan ağırlık matrisinin transpozesi ile çıktı

katmanı D_k 'nın δ vektörü çarpılır.

2. Gizli katmandaki ilgili sinir hücresi için aktivasyon fonksiyonunun türetilmesi ile elde edilen sonuçların herbir elemanı çarpılır.

Sembolik olarak gösterecek olursak;

$$D_j = D_k W_k' [O_j (I - O_j)] \quad (4.18)$$

Burada $\$$ işareti iki vektörün eleman çarpımını gösterir. O_j , j katmanının çıktı vektörüdür.

Uygun bir sapmayla sinir hücresi elde etmek istenilen bir durumdur. Bu lojistik fonksiyonunun orijini denkleştirir. Bu sayede çalışma prosesinin sonuca yaklaşım hızı artar. Bu özellik algoritmayla kolayca birleştirilir.

Rumelhart, Hinton ve Williams GYA'da işlem zamanını iyileştirmek için proses kararlılığını arttıran bir metot belirlediler. Momentum denilen bu metot, bir önceki aşamadaki ağırlık değişimlerinin miktarı olan düzeltilmiş katsayıların toplamını içerir. İlk olarak yapılan düzeltme bir hatırlatma olmakta ve tüm sonradan yapılan ağırlık düzeltmelerini yeniden şekillendirmek için hizmet vermektedir. Bu düzeltme eşitliği aşağıdaki gibi belirlenir.

$$\Delta W_{pq,k(n+1)} = \eta(\delta_{q,k} \text{ÇIKTI}_{P,j}) + \alpha [\Delta W_{pq,k(n)}] \quad (4.19)$$

Ağırlık değişimleri;

$$W_{pq,k}(n+1) = \Delta W_{pq,k}(n) + \Delta W_{pq,k}(n+1) \quad (4.20)$$

Bu metot sayesinde bir yandan diğer bir yana hızlı bir geçiş sağlanır. Bu metot bazı problemlerde oldukça etkilidir. Fakat her problemde etkili olmayabilir. Sejnowski ve Rosenberg 1987'de benzer şekilde üstel düzeltmeyi temel alan bir metot belirlediler. Bu metot bazı durumlarda çok çok iyi sonuçlar vermiştir. Bu üstel düzeltme eşitliği aşağıdaki gibidir:

$$W_{pq,k}(n+1) = \alpha \Delta W_{pq,k}(n) + (1 - \alpha) \delta_{q,k} \text{ÇIKTI}_{P,j} \quad (4.21)$$

Ağırlık değişimleri;

$$W_{pq,k}(n+1) = W_{pq,k}(n) + \eta \Delta W_{pq,k}(n+1) \quad (4.22)$$

Burada α , 0 ile 1 arasında değişen üstel düzeltme katsayısıdır. Eğer $\alpha = 0$ ise düzeltme minimumdur. Ham ağırlıkların düzeltilmesi yeni hesaplanan değişimlerden meydana gelmektedir. Eğer $\alpha = 1$ ise yeni hesaplama göz önüne alınmaz, bir önceki hesaplama

tekrarlanır. λ öğrenme oranı katsayısıdır ve bu ortalama ağırlık değerinin ölçüsünü ayarlamak için kullanılır.

4.5.4 İleri Algoritmalar

Bir çok araştırmacı temel GYA'yı genişletmeyi ve ıslah etmeyi düşünmüştür. Parker 1987'de geriye yayma algoritmasının yaklaşım hızını iyileştirmek için bir algoritma belirlemiştir. Bu algoritma literatürde Second-order Backpropagation olarak adlandırılmaktadır. Bu algoritma, doğru ağırlık aralıklarının daha mükemmel bir tahminini yapmak için ikinci türevi kullanır. İkinci türevin dışında alınacak türevler tahmini iyileştirmeyecektir. Stornette ve Huberman (1987) GYA'nın karakteristik çalışmasını iyileştiren basit bir algoritma geliştirmişlerdir. Bunlar konvansiyonel olarak girdilerin 0-1 dinamik aralığının ve gizli sinir hücresinin çıktılarının optimum olmadığını belirlemişlerdir. Bunu gerçekleştirmek için de aşağıdaki aktivasyon fonksiyonunu önermişlerdir:

$$\text{ÇIKTI} = -\frac{1}{2} + \frac{1}{1 + e^{-NET}} \quad (4.23)$$

Bu aktivasyon fonksiyonu yaklaşım süresini %30 ile %50 oranında azaltmıştır. Bu algoritmanın performansı pratik olarak yapılan düzenlemelere verilen bir örnektir. Pineda (1988) ve Almeida (1987) yapay sinir ağının çıktılarını aynı ağa girdi olarak uygulamışlardır. Sonuç olarak öğrenme işleminin bu gibi sistemlerde oldukça hızlı ve kararlılık kriterinin oldukça tatmin edici olduğunu görmüşlerdir (Zurada, 1992; Haykin, 1994; Dağlı, 1994; Tsoukalas ve Uhrig, 1997; Shalkoff, 1997).

4.5.5 Uygulama

1. Başlangıç ağırlıkları ve sapma (bias) değerleri rassal olarak ele alınsın ve NET değer elde edilsin

$$\text{GİRDİ} = [0.9 \quad 0.1 \quad 0.9] \quad \text{ÇIKTI} = [0.65]$$

$$W_{ij}^1 = \begin{bmatrix} 0.058 & -0.249 \\ -0.281 & -0.331 \\ 0.352 & 0.471 \end{bmatrix} \quad (\text{Başlangıç ağırlıkları matrisi})$$

$$\theta^1 = [-0.187 \quad 0.478]$$

$$NET = [0.9 \quad 0.1 \quad 0.9] \begin{bmatrix} 0.058 & -0.249 \\ -0.281 & -0.331 \\ 0.352 & 0.471 \end{bmatrix} - [-0.187 \quad 0.478]$$

$$NET = [0.5279 \quad -0.3114]$$

2. Daha sonra toplama fonksiyonu (NET) sigmoidal aktivasyon fonksiyonuna uygulanırsa çıktı değerleri elde edilir.

$$\zeta_1 = \frac{1}{1 + e^{-0.5279}} = 0.629 \quad \zeta_2 = \frac{1}{1 + e^{-(-0.3114)}} = 0.429$$

$$\text{ÇIKTI}^1 = [0.629 \quad 0.429] \text{ (gizli katmanın çıktıları)}$$

3. Çıktı katmanının çıktıların hesaplanması

$$W_{ij}^2 = \begin{bmatrix} 0.231 \\ -0.399 \end{bmatrix} \text{ (Başlangıç ağırlığı)}$$

$$\theta^2 = [0.399] \text{ (Başlangıç sapma değeri)}$$

$$NET = [0.629 \quad 0.429] \begin{bmatrix} 0.231 \\ -0.399 \end{bmatrix} - [0.399] = 0.422478$$

$$\text{ÇIKTI}^2 = \frac{1}{1 + e^{-(-0.422478)}} = 0.396$$

$$f(NET) = \frac{1}{1 + e^{-NET}} \quad f'(NET) = \frac{e^{-NET}}{(1 + e^{-NET})^2}$$

$$\delta_j = (\text{ÇIKTI} - \zeta_j) f'(NET_j)$$

$$\delta_j = (0.65 - 0.396) \frac{e^{-0.422478}}{(1 + e^{-0.422478})^2} = 0.061$$

4. Gizli katman için ağırlık düzeltmeleri

$$\delta_j = \sum (\delta_k W_{kj}) f'(NET_j)$$

$$\delta_j = 0.061 \begin{bmatrix} 0.231 \\ 0.399 \end{bmatrix} \begin{bmatrix} f'(0.5279) \\ f'(-0.3114) \end{bmatrix} = \begin{bmatrix} 0.003 \\ -0.006 \end{bmatrix}$$

$$\Delta W^2 = 0.5 * 0.061 * \begin{bmatrix} 0.629 \\ 0.429 \end{bmatrix} = \begin{bmatrix} 0.019 \\ 0.013 \end{bmatrix}$$

$$\Delta W_{14} = 0.5 * 0.003 * 0.9 = 0.0013$$

$$\Delta W_{15} = 0.5 * -0.006 * 0.9 = 0.0027$$

$$\Delta W_{24} = 0.5 * 0.003 * 0.1 = 0.00015$$

$$\Delta W_{25} = 0.5 * -0.006 * 0.1 = 0.0003$$

$$\Delta W_{34} = 0.5 * 0.003 * 0.9 = 0.0013$$

$$\Delta W_{35} = 0.5 * -0.006 * 0.9 = 0.0027$$

$$\Delta W^1 = \begin{bmatrix} 0.0013 & -0.0027 \\ 0.00015 & -0.0003 \\ 0.0013 & -0.0027 \end{bmatrix}$$

$$W_{ij}^{yeni} = W_{ij}^{eski} + \Delta W_{ij}$$

$$W_{ij}^1 = \begin{bmatrix} 0.058 & -0.249 \\ 0.281 & -0.331 \\ 0.352 & 0.471 \end{bmatrix} + \begin{bmatrix} 0.0013 & -0.0027 \\ 0.00015 & -0.0003 \\ 0.0013 & -0.0027 \end{bmatrix} = \begin{bmatrix} 0.0593 & -0.2517 \\ -0.2808 & -0.3323 \\ 0.3533 & 0.4683 \end{bmatrix}$$

5. ÇİZELGELEME

“Çizelgeleme”, günlük hayatımızda çok sıkça kullanılan bir terimdir. Aslında, çoğu konuşmalarda kullanılan bu temel kavram, “çizelgeleme” değil, “çizelge”dir. Çizelge, bir sınıf veya ders çizelgesi gibi elle tutulur, anlaşılır bir plan veya yazılı belgedir. Bir çizelge, çoğu zaman bir şeylerin ne zaman gerçekleşeceğini anlatır. Belli faaliyetlerin zamanlanması için bir plan gösterir ve “bir şeylerin ne zaman harekete geçeceği” sorusuna cevap verir. Çizelgeleme de, bir anlamda, çizelge oluşturma süreci olarak düşünülebilir (Baker, 1974).

Bir hizmetin veya ürünün üretilmesi için gerekli olan faaliyetler bütününe “iş”, işin gerçekleştirilmesindeki her bir basamak veya görev “işlem”, bir işe ait herhangi bir görevin gerçekleştirilebilmesi için gerekli olan süre “işlem süresi”, tüm bunları gerçekleştirmek için gerekli olan “kaynaklar” ve sağlanması gereken “kısıtlar” bir çizelgeleme probleminin temelini oluşturmaktadır. Bu kavramların tümüyle günlük hayatta karşılaşılmaktadır. Karşılaşılan bu problemlerin büyük bir çoğunluğu bir çizelgeleme problemidir. Bunlara örnek olarak;

- Evdeki günlük çamaşır, bulaşık, yemek ve temizlik işlerinin gerçekleştirilmesi,
- Havaalanına inmek için kuyrukta bekleyen uçaklar,
- Farklı derslerden verilen projelerin gerçekleştirilmesi,
- Bir eğitim kurumundaki haftalık ders programı,
- Alınan siparişlerin tezgahlara yüklenmesi,
- Bir tamir atölyesinde araçların tamir edilmesi,
- Bir bilgisayar yazılımının oluşturulması,
- Bir işyerinde veya gemide görevli kişilerin çalışma saatlerinin düzenlenmesi,
- Bankaya gelen müşterilerin hizmet görmesi,

vb. gibi verilebilecek daha pek çok problem, birer çizelgeleme problemi olarak ele alınabilir.

Tüm bu anlatılanlar paralelinde çizelgelemenin bir tanımını vermek gerekirse, “çizelgeleme, sistem kaynaklarının çeşitli işlere, görevlere veya faaliyetlere zaman bazında tahsis edilmesidir (Sabuncuoğlu, 1998)”. Amacı, hedefleri başarmak için kaynakların/kapasitenin

etkin bir biçimde kullanımını sağlamaktır. Çizelgeleme, sisteme gelen ve sistemden ayrılan varlıkların zamanlanması ile ilgilenir. Faaliyetlerin nerede ve ne zaman gerçekleştirileceğine karar vermek anlamına gelir. Çoğu organizasyonda, sistem içine yayılmış ve her yerde kendini gösteren bir süreçtir (Pinedo, 1995).

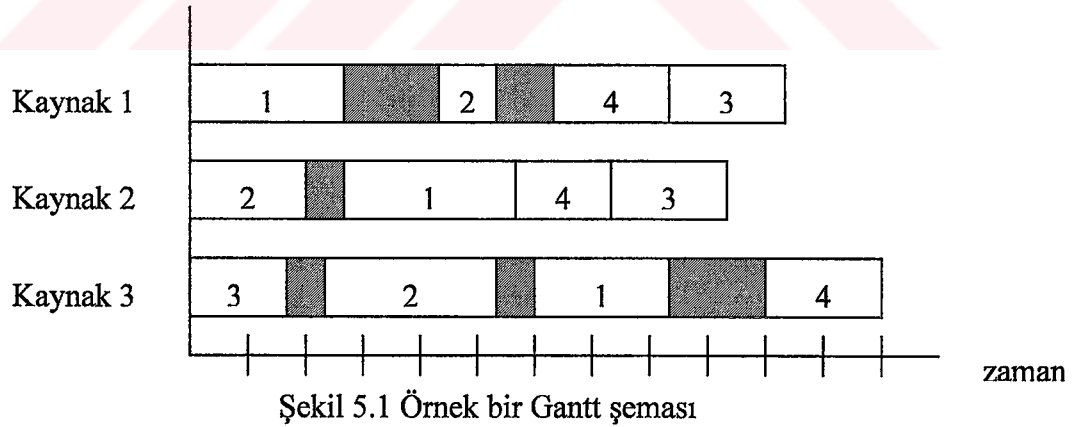
Her zaman çeşitli kaynakları kullanan belli sayıdaki işlerin belli bir zaman dilimi içerisinde başarılması ile ilgilenir. Kaynaklar sınırlı miktardadır. Başarılması gerekenler “işler” veya “projeler”, “faaliyetler”, “işlemler” ve “görevler” olarak adlandırılan basit kısımlardan oluşmaktadır. Her bir faaliyet, “işlem süresi” olarak isimlendirilen belirli bir zaman aralığında, belli miktardaki kaynağa ihtiyaç duymaktadır (Morton ve Pentico, 1993). İş parçaları, öğrenci grupları, bilgisayar yazılımları vb. benzeri şeyler de “işler” olarak yorumlanabilir. Kaynaklar da, “tezgahlar”, “hücreler”, “taşıma araçları”, “insanlar” ve benzeri şekilde adlandırılan basit, temel kısımlara sahiptir (Morton ve Pentico, 1993). Bir başka ifadeyle, karşılaşılan çizelgeleme problemine bağlı olarak, tezgah takımları, demiryolu rayları, sınıflar, bilgisayarlar, çamaşır makineleri, vezne, uçak pistleri vb. şeyler “kaynaklar” olarak ele alınabilir.

Çizelgeleme problemleri, çoğu zaman faaliyetlerin kendi aralarındaki, kaynaklar ile faaliyetler arasındaki veya sisteme dışarıdan müdahale eden oluşumlar arasındaki ilişkilerden ortaya çıkan fazlaca sayıdaki kısıtlarla oldukça karmaşık bir hal almaktadır. Yukarıda açıklananlar ışığında, çizelgeleme, kısıtlı eniyileme problemleri (KEP) olarak adlandırılan geniş problem sınıfına girmektedir. Daha açık ifade etmek gerekirse, çizelgeleme, problemin verilen kısıtlarına bağlı olarak belli bir amaç fonksiyonunu eniyileyen bir birleşimin bulunması ile ilgilenmektedir. Örneğin, atölye tipi ve akış tipi üretimin çizelgelenmesinde, amaç süreç içi stok zamanını veya tüm işlerin birlikte tamamlanma zamanını (makespan) enküçükleme ve kısıtlar ise tesisin maksimum kapasitesi, elverişli makine sayısı vb. olacaktır. Rotalama, akış planlama vb. içeren kısıtlı eniyileme problemleri olarak ele alınabilecek bir çok üretim problemleri bulunmaktadır. Ayrıca fabrika yerleşimi ve yükleme problemleri de kısıtlı eniyileme problemleri olup, diğer problemlerle yakın ilgileri bulunmaktadır (Özkazanç, 1999).

Sonuç olarak, karşılaşılan çizelgeleme problemlerinin tümü, yerine getirilmesi gereken görevler kümesini ve bu görevleri gerçekleştirmek için gereken bir elverişli kaynaklar kümesini içermektedir. Dolayısıyla, çizelgeleme sürecinde her bir kaynağın çeşidini ve miktarını belirlemek gerekmektedir. Böylece görevlerin uygun olarak ne zaman başarılabileceği belirlenmiş olur. Kaynaklar belirlendiğinde etkin bir şekilde çizelgeleme

probleminin sınırları çizilmiş olur. Bununla beraber, her bir görev, kaynak gereksinimi, süresi, başlama ve bitiş zamanları gibi bilgiler olarak açıklanır. Bu görevler arasında herhangi bir teknolojik kısıt varsa bu da eklenir. Kaynaklar ve görevler hakkındaki bilgi, bir çizelgeleme problemini tanımlar. Bununla beraber, bir çözüm bulmak karmaşıktır ve biçimsel çözüm yaklaşımları yararlı olmaktadır (Baker, 1994).

Biçimsel modeller, öncelikle problemlerin anlaşılmasında ve daha sonrasında da iyi bir çözüm bulunmasında yardımcı olmaktadır (Baker, 1994). Çizelgeleme faaliyeti sonunda elde edilen çizelgenin görüntülenmesi için oldukça geniş kullanılan bir şema, Henry Gantt ismi ile özdeşleşmiştir (Pinedo, 1995). Gantt şeması, basit olarak, her bir tezgah üzerinde işlere ait işlemleri zaman bazında görüntülemektir. Şekil 5.1’de de gösterildiği gibi Gantt şeması, her bir kaynağı/tezgahı temsil eden yatay barlardan oluşur ve boş veya dolu olduğunu gösteren zaman dilimleri içerir. Gantt şemasının incelenmesi, çizelgenin kaynak/tezgah boş zamanları, toplam üretim zamanı ve iş bekleme süreleri bakımından değerlendirilmesine imkan verir (French, 1982). Şemanın amacı, bir görsel yardım ile verilecek karar için bir odaklama, daha iyi anlama ve geliştirme imkanı sunmaktır. Gantt şeması mükemmel bir görüntüleme aracı olmasına rağmen, bir çizelgeleme tekniği olarak ciddi zayıflığa sahiptir. Çünkü iyileştirilmesine herhangi yapısal bir yaklaşım getirmez. Analizci, iyileştirilmiş bir çizelge bulabilmek için, kendi sezgilerini kullanmak zorundadır.



Özetleyecek olursak, üretimin gerçekleştirildiği ve fayda sağlayan bir şeylerin ortaya konduğu ve buna bağlı bazı kararların verilmesi gerektiği her ortamda planlama söz konusudur. Diğer bir ifadeyle, eldeki mevcutlar veya sonradan elde edilmesi düşünülen kaynaklar ile şimdi ve/veya gelecekte nelerin yapılabileceği kararının verilmesi planlama olarak adlandırılacak olursa, bu bağlamda, çizelgeleme, bu planlama sürecinin her aşamasında kendini gösteren, kaynakların gerçekleştirilmesi gereken faaliyetlere belli bir zaman aralığında tahsis edilmesi sürecidir. Dolayısıyla, çizelgeleme, planlama sürecinin bir parçasıdır ve farklı seviyelerde

ortaya çıkabilmektedir. Genel anlamda kesikli-parça imalatının olduğu ortamlarda var olan üretim çizelgeleme, zaman düzleminde yapılması gereken görevler için bir veya daha fazla amacı eniyileyecek şekilde kısıt kaynakların tahsis edilmesiyle ilgilidir (Pinedo,1995). Çizelgeleme probleminde iki tür olurluluk kısıtından bahsedilebilir:

1. Tezgah kapasite kısıtları
2. Teknolojik kısıtlar.

İşte çizelgeleme probleminin çözümü bu iki tip kısıtın olurlu bir çözümüdür. Sonuçta elde edilen çözüm şu iki soruyu cevaplayabilmelidir:

1. Yerine getirilecek her bir görev için hangi kaynak tahsis edilecek?
2. Her bir görev ne zaman yerine getirilecek?

Dolayısıyla, geleneksel olarak, çoğu çizelgeleme problemi kısıtlara bağlı eniyileme problemi olarak görülmektedir. O halde tipik bir üretim çizelgesinin özelliklerinin neler olduğu sıralanmalıdır. Bir çizelge;

1. Belli bir zaman içinde işletilebilmeli,
2. Kafiye büyüklüğünü belirlemeli, hazırlıkları ve sıraya-bağlı hazırlık zamanları ile maliyetlerini hesaba katmalı,
3. Proses rotalamada hayli esnek olmalı,
4. Stokastik çizelgeleme ortamını temsil edebilme; yani rassal olayları ve rahatsızlıkları içermeye yeteneğine sahip olmalı,
5. Çeşitli performans kriterlerini dengelemeli ve bünyesinde tutmalıdır (Baker,1994)

5.1 Çizelgeleme Seviyeleri

Çizelgelemeye olan ihtiyaç, parçaların ve alt parçaların satın alınmasının çizelgelenmesi, yapım ve montaj sürecinde işlerin ve tezgahların çizelgelenmesinden, paketlemenin ve sevkiyatın çizelgelenmesine kadar imalat sürecinin her seviyesinde kendini göstermektedir (Cheung, 1994).

Üretim çizelgeleme problemlerinde de, maliyetleri en küçükmek için tahsis edilen işgücü, malzeme ve donanım kavramları imalat ile ilgili tipik kaynaklardır. Problem, işleme süresi,

sıralama düzeni, kaynakların elverişliliği ve talep gibi kısıtlara sahiptir (Blazewicz ve diğ., 1996). Organizasyonlar iki genel çizelgeleme türüne sahiptirler. Ana üretim çizelgelemesi, tüm tesis genelinde üretim seviyesini planlar ve çoğu zaman üç aydan bir yıla kadar değişen bir planlama dönemine sahiptir. Detaylı çizelgeleme daha alt organizasyon seviyelerini planlar ve günlük işlemler ile ilgilenir. Atölye bazında işlerin çizelgelenmesi, tezgahlara yüklenmesi faaliyetlerini ele alır. Detaylı çizelgeleme girdilerin tahsis edilmesini yaparken, ana üretim çizelgelemesi bunlar için gerekli girdileri planlar. Detaylı çizelgeleme, ana üretim çizelgelemesi kapsamında yürütülmektedir (French, 1982).

Ortaya konan beklenti ve hedeflere bağlı olarak, planlama dönemi değişken olmakla beraber, çizelgeleme seviyeleri Morton ve Pentico (1993) tarafından Çizelge 5.1’de beş farklı seviyede ele alınmıştır.

Çizelge 5.1 Çizelgeleme seviyelerinin sınıflandırılması

Seviye	Problem Örnekleri	Dönem
Uzun Dönem Planlama	Tesis genişlemesi, tesis yerleşimi, tesis tasarımı	2 – 5 yıl
Orta Dönem Planlama	Üretimi sorunsuz hale getirme (production smoothing), lojistik destek	1 – 2 yıl
Kısa Dönem Planlama	İhtiyaç planı, atölye emirleri, tedarik zaman düzenlemesi	3 – 6 ay
Çizelgeleme	Atölye içi rotalama, montaj hattı dengeleme, süreç yığın büyüklüğünün tespiti	2 – 6 ay
Tepkisel Çizelgeleme/Kontrol	Anlık işler, arızalı tezgahlar, malzeme gecikmeleri	1 – 3 gün

Bu sınıflandırma, daha genel olmakla birlikte, aslında diğer çizelgeleme sınıflandırmaları ile ele alınma açılarından ortak özellikler göstermektedir. Tüm bu yaklaşımlar, planlama süreci ve/veya planlama dönemi açısından incelendiğinde;

- Stratejik planlama, uzun dönem planlaması
- Üretim planlama, orta dönem planlama
- Ana üretim çizelgelemesi ve malzeme gereksinim planlama, kısa dönem planlaması
- Detaylı çizelgeleme ise çizelgeleme ve tepkisel çizelgeleme/kontrol ile ortak seviyelerde yer aldığı söylenebilir.

5.2 Çizelgeleme Problemlerinde Kabuller

Çizelgeleme problemlerinde belli başlı bazı kabuller vardır. Bunları aşağıdaki gibi sıralamak mümkündür:

1. Her iş belirli operasyonlardan oluşmuştur. Aynı işin iki işlemi aynı zamanda işlenemez. Böylece bitmiş bir ürünün, montaj önceliğine göre üretilmiş parçalarının ortaya çıkardığı problemler ihmal edilir.
2. Kesinti yok; bir operasyon o tezgahda başlamadan önce, daha önce başlamış olan operasyon tamamlanmalı
3. Her tezgahda 1 adet olmak üzere herbir işin m sayıda belirli operasyonları vardır. Bir işin aynı tezgahda iki işlem gerektireceği olasılığına izin verilmemektedir. Aynı şekilde herbir işin tüm tezgahlarda işlenmesi gerekmektedir. Bir iş, bir veya daha fazla tezgahı atlayamaz.
4. Erteleme yok; her iş tamamlanıncaya kadar devam etmelidir.
5. İşlem sırasında ara stoklara izin verilebilir. Yani işler bir sonraki tezgahın boşalmasını bekleyebilir.
6. Tezgahlar boş kalabilir.
7. Hiçbir tezgah aynı anda birden fazla iş yapamaz.
8. Teknolojik kısıtlar önceden biliniyor ve sabittirler.
9. Hiçbir rasgelelik yoktur;
 - a. İşlerin sayısı biliniyor ve sabit
 - b. Tezgahların sayısı biliniyor ve sabit
 - c. İşlem zamanları biliniyor ve sabit
 - d. Hazırlık zamanları biliniyor ve sabit
 - e. Bir problemi belirtmek için gerekli tüm sayısal veriler biliniyor ve sabit (Başlıgil, 1988).

5.3 Çizelgeleme Ortamı Çeşitleri

Farklı üretim sistemleri, farklı çizelgeleme problemlerine ve ihtiyaçlarına sahiptir. Çizelgeleme ortamı, küçük, karmaşık, tek müşteriye çalışan atölyelerden, yüksek hızlı üretim yapan transfer hatlarına, kesikli parça imalatından, sürekli proses akışlarının olduğu pek çok farklı durumda ele alınabilmektedir. Bu bağlamda çizelgeleme ortamları çeşitli sınıflarda incelenebilmektedir (Morton ve Pentico, 1993).

- Atölye tipi üretim
- Açık atölye tipi üretim
- Parti tipi üretim
- Akış tipi üretim
- Parti/akış tipi üretim
- İmalat hücresi
- Montaj atölyesi
- Montaj hattı
- Esnek iletim hattı

5.4 Performans Ölçütleri ve Çizelgeleme Problemlerinin Sınıflandırılması

Bir çizelgeleme problemi için enbüyüklenecek veya enküçüklenecek iyi bir amaç bulmak zor olabilir (French, 1982). Verilmesi gereken iki önemli karar bulunmaktadır. Birincisi, sıralama - işlerin kaynaklar üzerindeki gerçekleşme sırası- ve ikincisi de, zamanlama, bir başka ifadeyle zaman boyutunun eklenmesidir. Bu iki faaliyetin bir arada gerçekleştirilmesi de "çizelgeleme"dir.

Sıralama + Zamanlama = Çizelgeleme

Oluşturulan çizelgenin değerlendirilmesi için bir amacın ve buna bağlı olarak da bir başarı ölçütünün belirlenmesi gerekmektedir. Ancak bundan, önce bazı temel tanımlamaların yapılması gerekmektedir.

k : tezgah/kaynak sayısı ($k = 1, 2, \dots, m$)

i : iş sayısı ($i = 1, 2, \dots, n$)

j : i. işin işlem sayısı ($j = 1, 2, \dots, g_i$)

ijk : i. işin j. işleminin k. tezgahda yapılması

r_i : i. işin işleme hazır olduğu zaman

P_{ij} : i. işin j. işleminin süresi

d_i : i. işin teslim zamanı

a_i : i. işin yapılması için eldeki süre (allowance) ($a_i = r_i - d_i$)

W_{ij} : i. işin j. işleminden önceki bekleme süresi

C_i : i işinin tamamlanma süresi ($C_i = r_i + W_i + P_i$)

F_i : i işinin akış süresi ($F_i = C_i - r_i = W_i + P_i$)

L_i : i işinin gecikme süresi (lateness) ($L_i = C_i - d_i = F_i + r_i - d_i = F_i - a_i$)

T_i : i işinin geçliği (tardiness) ($Enb[0, L_i]$)

E_i : i işinin erkenliği (earliness) ($Enb[0, -L_i]$)

Oluşturulacak çizelgenin değerlendirilmesi için bir başarı ölçütünün belirlenmesi gerekmektedir. Oldukça fazla sayıda başarı ölçütü olmakla birlikte, başarı ölçütleri 3 temel amaç çerçevesinde ele alınmaktadırlar (French, 1982). Bunlar sırasıyla şunlardır:

- Kaynak kullanım ölçütlerinden birinin eniyilenmesi: Herhangi bir anda kuyrukta bekleyen ortalama iş sayısının veya tamamlanmamış ortalama iş sayısının enküçüklenmesi süreç içi stok maliyetleri ile dolaylı da olsa ilişkilendirilebilir. Benzer şekilde tezgah boş zamanlarının enküçüklenmesi ile de kaynakların etkin kullanımı sağlanmaya çalışılabilir.
- İş gecikme ölçütlerinden birinin eniyilenmesi: Yapılan anlaşmalardan en az zararla çıkmak açısından geciken iş sayısının enküçüklenmesi, planlama dönemi açısından işlerin mümkün olan en az gecikme ile tamamlanması istenebilir.
- İş akış zamanı ölçütlerinden birinin eniyilenmesi: İşlerin atölye içinde mümkün olduğunca az kalması ve çabuk tamamlanmaları arzu edilebilir. Toplam tamamlanma zamanının (makespan), enbüyük akışın, ortalama akışın veya

ortalama tamamlanma süresinin enküçüklenmesi ile ilgilenilir.

Bununla birlikte, ilgilenilen çizelgeleme problemini belli standart bir gösterimle ifade edilmektedir. Bunun için, $n/m/A/B$ şeklinde, 4 terimden oluşan bir gösterim kullanılmaktadır (French, 1982).

n : İş sayısı

m : Tezgah sayısı

A : Üretim ortamı içindeki akış desenini veya disiplini açıklamaktadır. $m = 1$ için A terimi boş bırakılır.

F : Akış tipi üretim içindir, tüm işler için tezgah sırası aynıdır.

P : Permütasyon akış tipi üretim içindir. Burada tüm işler tezgah sıraları aynı olmakla kalmaz, her tezgah içinde aynı iş sırasının çizelgede sağlanması kısıtı vardır.

G : Teknolojik sınırlamaların olmadığı atölye tipi üretimdir. Gelişigüzel sırada üretilebilmesidir.

B : Başarı ölçütünü açıklamaktadır.

5.5 Atölye Tipi Üretim Ortamında İşlerin Çizelgelenmesi

Bir atölye tipi üretim, üretim emirlerini veya işlemleri içeren işlerin gerçekleştirildiği bir dizi genel amaçlı tezgahlardan oluşmaktadır. Özel bir siparişin sonucu olarak ortaya çıkan bu işler benzersizdir. Her ne kadar geleneksel anlamda, atölyeyi bir tezgah atölyesi olarak düşünsek de, atölye tipi üretim süreci, bakım faaliyetleri vb. faaliyetler için temel modeldir (Johnson ve Montgomery, 1974).

Atölye tipi üretim çizelgeleme problemi, tezgahların bazı performans ölçütlerini eniyileyecek şekilde yapacak sıralama düzeninin belirlenmesinden oluşmaktadır. Spesifik bir atölye tipi üretim çizelgeleme problemini tanımlamak ve sınıflandırmak için 4 faktör sunulmaktadır (Johnson ve Montgomery, 1974).

1. İşlerin gelişi düzenlidir. Eğer n adet iş (eş zamanlı olarak/aynı anda) boş ve işin gerçekleştirilmesi için anında elverişli olan atölyeye eş zamanlı olarak giriyorsa, bu çizelgeleme problemi statik demektir. Eğer işler, rassal bir sürece göre olasılıklı olarak belli aralıklarla geliyorsa, çizelgeleme problemi dinamiktir.

2. Atölyeyi oluşturan tezgah (m) sayısının belirtilmesi gerekmektedir.
3. İşlerin tezgahlardan akış süreci belirtilmelidir. İşlerin hangi öncelik kısıtları dahilinde tezgahlardan geçeceğinin bilinmesi gerekmektedir.
4. Atölyenin performansını değerlendirmede kullanılan kriterler, çizelgeleme sürecinde önemli bir rol oynar. Çoğu zaman başvurulanan bir performans ölçütü, toplam tamamlanma zamanı (makespan) veya bütün işlerin tümüyle tamamlanması için gerekli olan toplam süredir. Dolayısıyla, bu süreyi enküçükleyecek bir çizelgeleme prosedürü geliştirmektir. Biz her ne kadar belli bir sınırlı sayıdaki başarı ölçütleri kullanacak olsak da, çizelgeleri değerlendirmek için termin zamanı, iş başına düşen ortalama gecikme süresi, ortalama akış süresi gibi pek çok farklı performans ölçütleri kullanılmaktadır.

Örnek olması açısından işlem sürelerini ve öncelik kısıtlarını içeren bir probleme ait veriler çizelge 5.2'de verilmiştir.

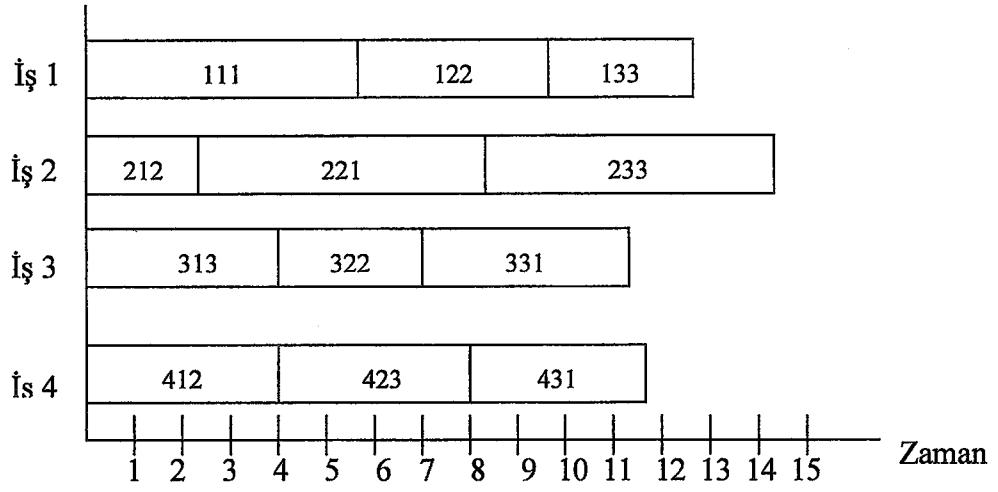
Çizelge 5.2 İşlem süreleri ve tezgah rotaları

İşler	İşlemler		
	1	2	3
1	4	3	2
2	2	4	4
3	3	2	3
4	3	3	1

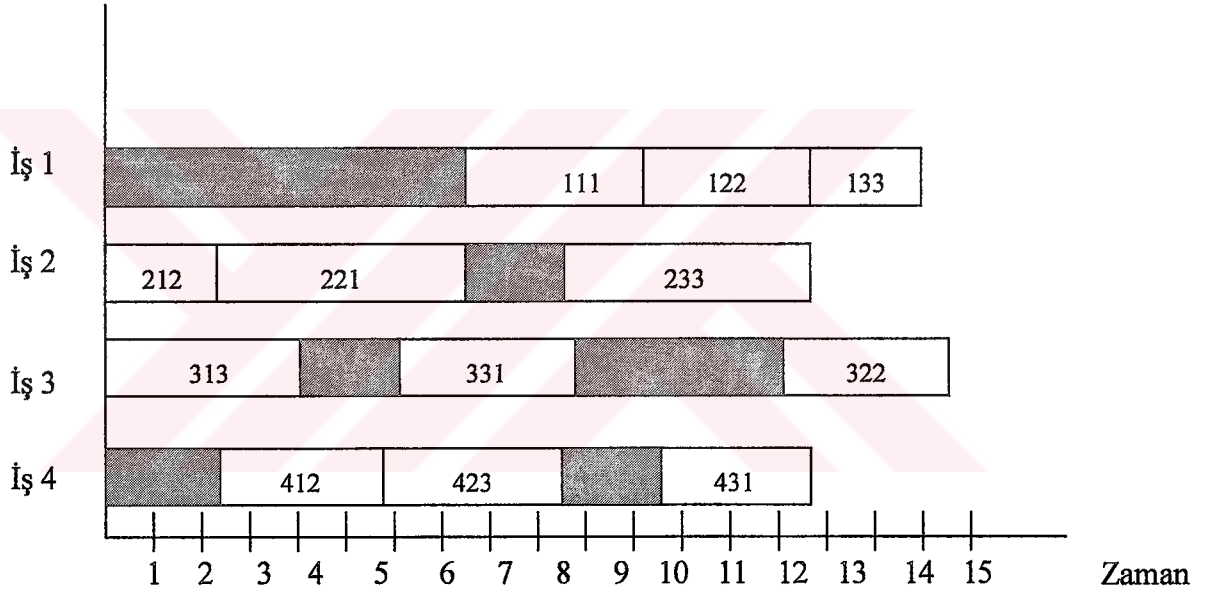
İşler	İşlemler		
	1	2	3
1	1	2	3
2	2	1	3
3	3	2	1
4	2	3	1

Örneğin, 3 nolu tezgah üzerinde çizelgelenmesi gereken işler olarak, 13, 23, 31, 42 gözükmemektedir. Belli bir sabit sıralama kuralı $ijk(133) = 13$, $ijk(233) = 23$, $ijk(313) = 31$ ve $ijk(423) = 42$ olabilir.

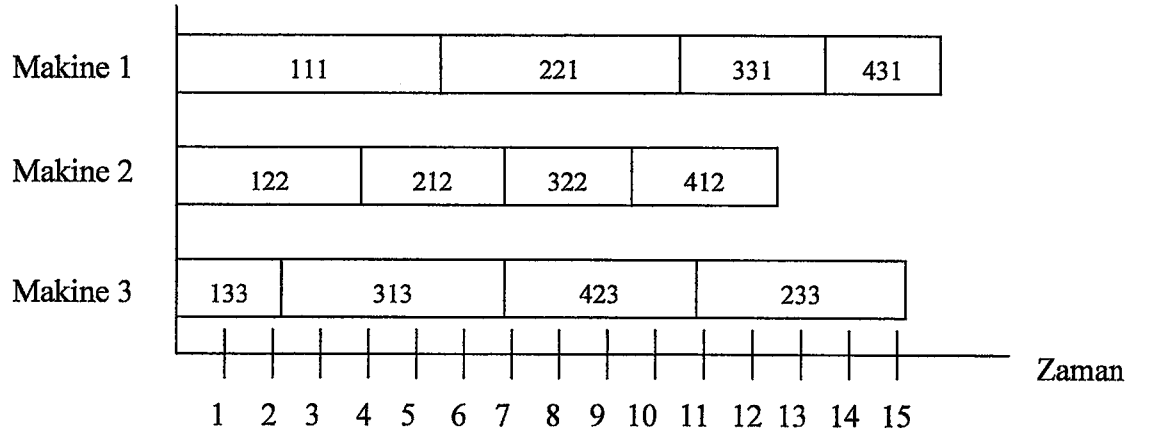
Bir akış tipi üretim ortamında, n adet iş m adet tezgah üzerinde çizelgeleneceğinden $(n!)^m$ kadar olası çizelge bulunmaktadır. Bu çizelgelerden de bir kısmı uygun olmayan çizelgeleri, başka bir ifadeyle öncelik kısıtlarını sağlamayanları kapsamaktadır. Farklı çizelgeler oluşturulabilmektedir. Bunlara örnek oluşturması bakımından, Şekil 5.2, Şekil 5.3, Şekil 5.4 ve Şekil 5.5'deki örnek Gantt şemalarında da gözükmemektedir.



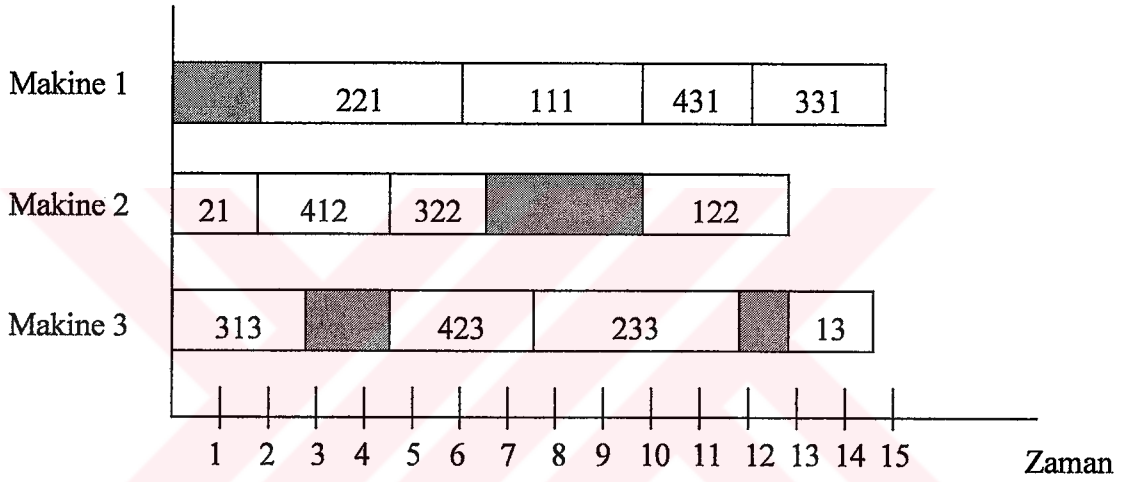
Şekil 5.2 İşlere ait uygun olmayan Gantt şeması



Şekil 5.3 İşlere ait uygun Gantt şeması



Şekil 5.4 Tezgahlara ait uygun olmayan Gannt şeması



Şekil 5.5 Tezgahlara ait uygun Gannt şeması

Oluşturulabilecek tüm çizelgeler -uygun olmayanların da dahil olduğu- kümesinde 3 çeşit çizelge ile karşılaşmaktadır. Daha açık bir ifadeyle, uygun olanlar içinde, 3 çeşit çizelge bulunmaktadır (French, 1982; Foo ve Takefuji, 1988a; Morton ve Pentico, 1993). Bunlar sırasıyla; aktif, yarı aktif ve gecikmesiz (nondelay) çizelgelerdir.

Tek bir kaynak üzerinde işlemlerin daha erken başlama/tamamlanması gerçekleştirilebiliyorsa, uygun çizelgede yerel sola kaydırma (local left shift) yapılabilir. Eğer hiçbir yerel sola kaydırma mümkün değil ise, bu çizelge "yarı aktiftir" (semi active) denir.

Klasik bir atölye tipi üretim probleminde (Morton ve Pentico, 1993);

- Verilen bir çoklu permütasyon sıralamasında, sadece karşı gelen tek bir yarı aktif çizelge vardır,
- Belli bir amaç için, en azından bir yarı aktif çizelge optimumdur,

- Sonlu sayıda yarı aktif çizelge bulunmaktadır.

Üzerinde iyileştirme yapılan çizelgeler, yarı aktif çizelgelerdir. İyileştirilmesi biraz daha kolaydır. Bir faaliyet hemen sola doğru hareket ettirilemiyor olmasına rağmen, bloklar üzerinden sola doğru atlatılması mümkün olabilir ve daha iyi bir çizelge elde edilebilir.

Benzer şekilde, eğer bir işlem, çizelgedeki daha erken bir boşluğa kaydırılabiliyor, tüm kaynaklar üzerinde iyileştirme sağlanabiliyor ve uygunluk korunabiliyor ise, bir bütünsel sola kaydırma (global left shift) ortaya çıkar. Tüm çizelgeler içinde, bütünsel sola kaydırmanın yapılamayacağı küme, aktif çizelgeler kümesi olarak adlandırılmaktadır.

Son olarak da, hiçbir tezgahın/kaynağın , işlemlerin yerine getirilmesi esnasında herhangi bir anda boş/aylak kalmadığı çizelge gecikmesiz çizelgedir (Morton ve Pentico, 1993).

n adet işin m adet tezgahda eniyi biçimde çizelgelenmesi olarak ifade edilen atölye tipi üretim ortamında çizelgeleme işleminde, n adet işin her biri, işlem olarak adlandırılan m adet alt işten oluşur. Dolayısıyla, her tezgah üzerinde olası $n!$ kadar sıralama vardır. Eğer bu olası $n!$ adet sıralama her bir tezgah için bağımsız olduğu düşünülürse, kısıtları sağlamayan/uygun olmayan, toplam $(n!)^m$ adet olası çizelge oluşturulabilir. Uygun çizelge iki tür kısıtı sağlamalıdır. İlk olarak, öncelik kısıtlarının, yani her işe ait işlemlerin belirlenen sıralama dahilinde gerçekleştirilmesi sağlanmalıdır. İkinci kısıt ise, kaynak kısıtı olup, bir tezgahda aynı anda birden fazla işlemin gerçekleştirilemeyeceğini ifade etmektedir. Problem bir doğrusal karar modeli ile ifade edilmek istenirse;

n : iş sayısı

m : tezgah (işlem) sayısı

S_{ik} : i . işin k . tezgahta başlama zamanı

t_{ijk} : i . işin j . işleminin k . tezgahta yapılması için gerekli olan süre

$$y_{ipk} = \begin{cases} 1, & \text{eğer } k. \text{ tezgahda } i. \text{ iş } p. \text{ işten önce gerçekleştirilirse,} \\ 0, & \text{diğer durumlarda} \end{cases}$$

H : tüm işlerin gerçekleştirilmesi için gerekli toplam süre olmak üzere:

$$S_{ik} - S_{ih} \geq t_{i,j-l,h}$$

$$1 < j \leq m, 1 \leq i \leq n$$

$$S_{pk} - S_{ik} + H(1 - y_{ipk}) \geq t_{ijk}$$

$$1 \leq k \leq m, 1 \leq i < p \leq n$$

$$S_{ik} - S_{pk} + H y_{ipk} \geq t_{pik} \quad 1 \leq k \leq m, 1 \leq i < p \leq n$$

$$y_{ipk} = 0, 1 \quad 1 \leq k \leq m, 1 \leq i < p \leq n$$

$$S_{ik} \geq 0 \quad 1 \leq i \leq n, 1 \leq k \leq m$$

kısıtları altında

$$EnkZ = \sum_i^n \sum_k^m S_{ik}$$

şeklinde modellenebilir (Van Hulle, 1991; Pinedo, 1995).

Modelin çözümü için $(m-1)n$ adet öncelik kısıtı, $mn(m-1)$ adet kaynak kısıtı, mn adet negatif olmama kısıtı söz konusudur. Ayrıca, mn adet sürekli değişken ve $mn(n-1)/2$ adet 0-1 tamsayı değişken bulunmaktadır. Sadece 4 işin ve 4 tezgahın bulunduğu küçük boyutlu bir problem için bile toplam 76 adet kısıt ve 16 reel ve 24 adet 0-1 tamsayı değişken gerekmektedir. $(n!)^m$ adet olası çizelge olduğunu da düşünecek olursak, 331776 farklı şekilde çizelge oluşturulması gerekmektedir. Sürece sadece bir tek işin daha eklenmesiyle bu sayı $(5!)^4 = 2.1 \cdot 10^{28}$ olmaktadır. Benzer şekilde, değişken ve kısıt sayısı da hızlı bir şekilde artmakta ve buna bağlı olarak problem çözüm uzayı da büyümektedir. Gerçek hayat problemleri düşünüldüğünde, bu türdeki problemlerin çözümünün en hızlı işlemcilerde bile oldukça uzun süreceği açıktır. Bu nedenle, probleme farklı çözüm yaklaşımları aramak kaçınılmaz olmaktadır.

5.6 Çözüm Yaklaşımları

Atölye tipi üretim ortamında işlerin çizelgelenmesi probleminin çözümü için getirilen yaklaşımlar iki temel başlık altında incelenebilir (Blazewicz et al., 1996)

- Kesin Yöntemler
- Sezgisel Yaklaşımlar

Kesin yöntemler, analitik yöntemler olarak da adlandırılabilir (Sim et al., 1994). Tamsayılı programlama, dal ve sınır algoritmalarını içermektedir. Problem, doğrusal programlama yöntemleri ile modellenir. Optimum çözümün bulunmasını garanti eden modellerdir. Ancak, problemin NP-zor yapıda olması nedeniyle gerekli hesaplama süresinin problem boyutuna bağlı olarak üstel büyümesi, yöntemlerin gerçek hayat problemlerinde

uygulanmasını zorlaştırmaktadır (French, 1982). 20 iş ve 10 tezgahdan oluşan bir problem için bile günlerle ifade edilebilecek bir hesaplama süresine ihtiyaç duyulmaktadır (Morton ve Pentico, 1993).

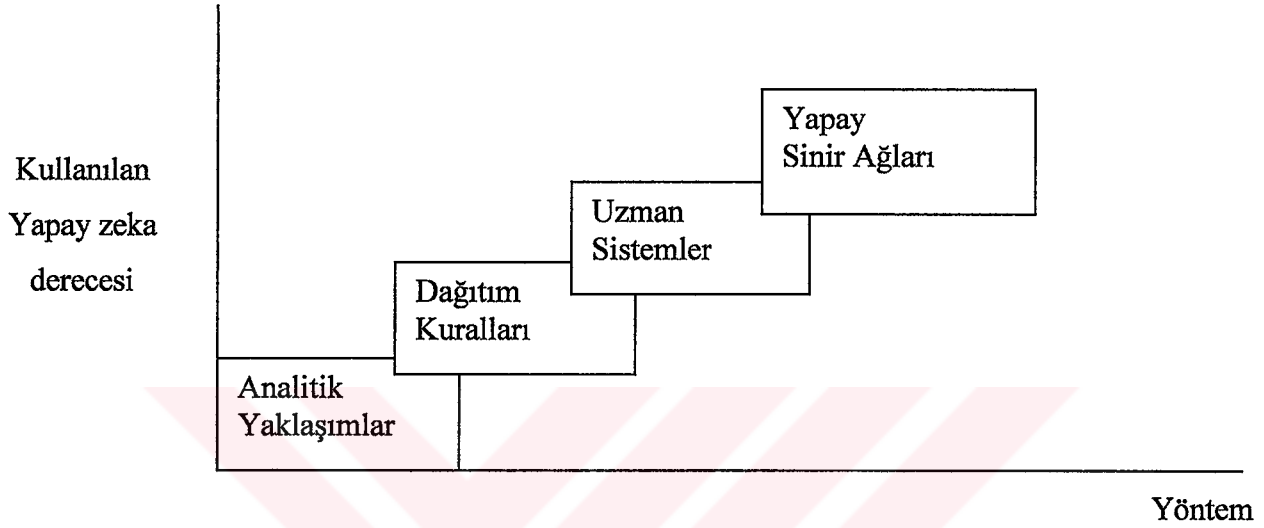
Sezgisel yöntemler , problemin çözümüne getirdiği uygulama kolaylığı ve az karmaşa dolayısıyla çok sık başvurulanan yaklaşımlardır ve optimum sonucu garanti etmezler. Bu kapsamda, kullanılan yaklaşımlardan biri de öncelik kurallarını kullanan sezgisel algoritmalar.. Bu öncelik kurallarının bazıları şunlardır (Blazewicz vd., 1996):

1. SOT (shortest operation time) : Tahsis edilecek ilgili kaynak üzerindeki en kısa işleme süreli işlem
2. LOT (longest operation time) : Tahsis edilecek ilgili kaynak üzerindeki en uzun işleme süreli işlem
3. LPRT (longest remaining processing time) : Kalan en uzun işleme süreli işe ait işlem
4. SPRT (shortest remaining processing time) : Kalan en kısa işleme süreli işe ait işlem
5. LORPT (longest operation remaining processing time) : Kalan işlemlerden işleme süresi en uzun olan işlem
6. Rassal
7. FCFS (First come first served): İlk giren ilk işlem görür.
8. SPT (shortest processing time): Toplam işleme zamanı en kısa olan iş
9. LPT (longest processing time) : Toplam işleme süresi en uzun olan iş
10. LOS (longest operation successor) : Bir işlemi takip eden en uzun süreli işlem
11. EDD (Earliest Due Date) : En erken teslim tarihi

Bu kurallardan biri veya birkaçının birleşimi birlikte kullanılarak ilgilenilen probleme çözüm aranır. Bu tezde SPT (EKİZ- En Kısa İşlem Zamanı), FCFS (İGİG – İlk Gelen İlk İşlem Görür) ve EDD (EETT- En Erken Teslim Tarihi) kuralları kullanılmıştır. Amaç hızlı bir şekilde bir uygun çizelge oluşturmaktır. Geliştirilen pek çok sezgisel ve arama yöntemi bu sınıflandırma içinde bulunmaktadır. Shifting bottleneck (darboğaz açma), local search (yerel

arama), bunlardan birkaçıdır. Bu yöntemlere ilişkin daha detaylı bir inceleme, Blazewicz ve diğerleri (1996) ve Morton ve Pentico (1993) tarafından ele alınmıştır.

Bununla birlikte, yapay zeka teknikleri de bu sınıflandırma içinde yer almaktadır. Problemin çözümünde yapay zeka tekniklerine olan ihtiyaç Sim ve diğerleri (1994) tarafından oluşturulan şekil 5.6'da da görülmektedir.



Şekil 5.6 Farklı yaklaşımlarla gerekli olan yapay zeka derecesi

Problemin boyutunun artması ile hesaplama süresi ve çözüm kalitesi yönlü sıkıntılar, yapay zeka teknikleriyle aşmaya çalışılmıştır. Genetik algoritma, yasaklı arama, tavlama benzetimi, yapay sinir ağları, uzman sistem tabanlı pek çok yeni yaklaşımlar ile atölye tipi üretim ortamında çizelgeleme problemlerine çözüm aranmıştır (Blazewick, 1996; Cheung, 1994; Sim vd., 1994).

5.7 Yapay Sinir Ağları ile Atölye Çizelgeleme

Hopfield ve Tank (1985) kısıtlı eniyileme problemlerinin (KEP) yapay sinir ağları ile çözülebileceğini gösterdikten sonra, buna uyan çizelgeleme problemleri yapay sinir ağı yoluyla çözülmeye başlanmıştır. Ancak ağı yerel optimuma düşmesi gibi problemler ile karşılaşmıştır. Bu problemi gidermek için yapılan çalışmaların birinde, Arizono ve diğerleri (1992) stokastik sinir ağı modellerinin geliştirilmesini amaçlamışlar ve toplam gerçek akış süresini en küçük yapacak bir çizelgeleme problemini stokastik sinir ağı modellerinden biri olan Gaussian motoru modeliyle çözmüşlerdir.

Benzer bir çalışmada, Sabuncuoğlu ve Gürgün (1996) Hopfield ve yarış-tabanlı ağları kullanarak Parallized Neural Network (PNN) adı verilen bir ağı tasarımını sunmuşlardır. Amaçlanan modelde, verilen bir tezgahdaki işlerin sırası her bir satırın bir işi ve herbir sütunun

sıradaki bir pozisyonu gösterdiği bir $n \times n$ neuron matrisi tarafından temsil edilmiştir. İş atölyesi durumunda ise yukarıdakine ilaveten, m tezgah sayısı olmak üzere $m \times n \times n$ büyüklüğünde üç boyutlu bir matris vardır. Ağ tasarımı aktivasyon fonksiyonu sigmoid ve çizelgelenen operasyonların en büyük tamamlanma süresi o ağın enerjisi olarak kullanılmıştır.

Sonuçta, ağın pek çok iş atölyesi problemi için optimum çözümler ürettiği yazarlar tarafından gösterilmiştir.

Taşgetiren ve diğerleri (1995) çalışmalarında, uzman çizelgeleme sisteminin iş atamasında hangi öncelik kuralının kullanılacağına ilişkin karar verme sürecinden esinlenerek geliştirilen bir hatayı GYA ağı ile çizelgeleme yaklaşımı sunmuşlardır. Uzman sistemlerdeki çıkarım mekanizmasının işlevini üstlenen BP ağı ile bu sistem, uzman sistemlerden daha basit bir işlevsel yapı gerektirir. Sonuçlar, bu sistemin klasik öncelik kurallarından daha iyi sonuç verdiğini ve sistemin olurluluğunu göstermektedir.

Tank ve Hopfield ağı geliştirilerek, oldukça zor eniyileme problemlerinden klasik atölye çizelgeleme problemine bir çözüm getirilmeye çalışılmıştır (Foo vd., 1995). Atölye çizelgeleme problemleri için yapay sinir ağları kullanılmıştır. Bir doğrusal enerji fonksiyonu kullanarak, burada sunulan uygulamada klasik problem çözümlerinden kaçınmaya çalışılmıştır. Bu çalışmada bir çok yapay sinir hücresine ve birbirleri ile bağlantılarına gereksinim duyulabilir. Ama her bir sinir hücresi çok basit aktivasyon fonksiyonuna sahiptir. Böylece burada teklif edilen yapay sinir ağı kendi kendini kontrol altında tutabilmekte ve karmaşık hesaplamalara ihtiyaç duymamaktadır.

Bir başka çalışmada parti hacmi kısıtı altında atölye çizelgeleme için yapay sinir ağı geliştirilmiştir (Chang ve Jeng, 1995). Bu çalışmada birçok sinir hücresi kullanılarak işlerin parti hacimlerine bölünmesi sağlanmıştır. Ama bu oldukça karmaşık ve zor bir uygulamadır. Bu çalışmada, bir çok yapay sinir hücresi parti hacimleri için kullanılmıştır. Sonuçlar, kalite ve hız açısından oldukça iyi çıkmıştır. Bu sinir ağı kısaca JSSNN (Job shop scheduling neural networks) olarak isimlendirilmiştir. Bu ağın iki önemli birimi vardır. İlki, zaman ölçüm birimi (TAU (Time Accumulator Unit)), diğeri de tezgah atama birimi (MAU (Machine Assignment Unit)). Bu araştırmada, hızlı bir şekilde optimum ya da optimuma yakın çizelgeleme sonuçları elde edilmiştir. Bunlar kalite ve hız açısından oldukça iyi sonuçlardır. Bununla beraber bu metot hiç bir zaman tek bir optimum sonuç bulmayı garanti etmez. Bu uygulamada, m -tezgah n -iş problemini çözmek için $mn \times (m+7)$ tane sinir hücresine ihtiyaç duyulmuştur. Bu sayı diğer çalışmalarda kullanılan sayılara göre daha ekonomiktir.

Yang ve Wang (2000), genel atölye çizelgeleme problemleri için CSANN (Constraint Satisfaction Adaptive Neural Network) ve sezgisel algoritmaların birleştirilmesi ile bir uygulama yapmışlardır. Bu çalışmada atölye çizelgeleme problemi için önerilen yaklaşımlarda birkaç sezgisel algoritma ve CSANN algoritması birleştirilmiştir. CSANN, mümkün çözümleri elde etmek için çalışması süresince kısıtlardaki bozulmaları ortadan kaldırmaktadır. Böylece mimarisini oluşturarak özel bir atölye çizelgeleme probleminin kaynak kısıtları ve sıralama sorunlarına kolay bir çözüm getirmiş olmaktadır. CSANN'nin uyarlamalı olma özelliği, diğer tatmin olmuş kısıtlı sinir ağları yapılarından onu ayırmaktadır. CSANN atölye çizelgeleme probleminde kullanıldığı zaman, elde edilen mümkün çözümlerin kalitesi, çizelgeleyicinin istediği amaca ulaşabilmesi için, biraz da beklenen toplam akış zamanının tercihine bağlıdır. Beklenen toplam akış zamanı uygun bir seçimse, o zaman istenilen amaca ulaşabilmektedir. Bu çalışmada, CSANN'nin performansını ve ürettiği çözümlerin kalitesini geliştirmek için CSANN bir sezgisel algoritma ile beraber çalıştırılmıştır. Bu sezgisel algoritmanın eklenmesiyle CSANN daima iyi çizelgeler üretmiştir.

Yine Yang ve Wang (2001), atölye çizelgeleme problemleri için yeni bir uyarlamalı yapay sinir ağlarını iki sezgisel algoritma ile birleştirerek kullanmışlardır. Burada CSANN mümkün çözümler elde etmek için kullanılmıştır. CSANN'nin özelliğini geliştirmek için iki tane sezgisel algoritma kullanılmış ve daha iyi sonuçlar elde edilmiştir. Benzetim sonuçları, atölye çizelgeleme problemi için önerilen bu melez yaklaşımdan iyi sonuçlar elde edildiğini göstermiştir. Ayrıca hesaplama hızı ve çözüm kalitesinde de iyi sonuçlar üretilmiştir.

Atölye çizelgeleme problemlerinin çözümü için diğer bir yaklaşım da yapay sinir ağları ile genetik algoritmanın birlikte kullanılarak melez bir çözüm tekniği elde edilmesidir. Buradaki sinir ağının yapısı da kısıtlı bir ağ yapısıdır (CNN – Constraint neural network). CNN burada atölye başlangıç zamanlarını ve başlangıç sıralamasını bulmak için kullanılmıştır. Buradan elde edilen değerler, daha optimum bir sıralama bulmak için genetik algoritma ile birleştirilmiştir. Bu çalışma pratik uygulamalar da ve birçok benzetim çalışmalarında test edilmiş ve çok iyi sonuçlar elde edilmiştir (Yu ve Liang, 2001).

Atölye çizelgeleme yaklaşımları için bir başka çalışmada da, klasik atölye benzetim paket programları olan Siman, Promodel ve Arena programları ile yapay sinir ağlarının bir karşılaştırılması yapılmıştır. Buradaki yapay sinir ağının mimarisinde çok seviyeli ağ yapısı kullanılmış ve ağın eğitimi de hatayı GYA ile sağlanmıştır. Sonuçta yapay sinir ağının ürettiği çıktıların, Siman, Promodel ve Arena gibi benzetim paket programlarından elde edilen sonuçlardan kötü olmadığı ortaya çıkmıştır. Ama işlerin tezgahlara atanması ve sıralanması,

ortalama akış zamanlarının bulunması yapay sinir ağı modelinde daha hızlı bir şekilde olmuştur. Ayrıca yapay sinir ağı modelinin, daha büyük ve daha geniş modellere daha rahat bir şekilde uygulanabileceği görülmüştür (Fonseca ve Navarrese, 2002).

Başka bir çalışmada , bir üretim faaliyet çizelgelemesinin tasarlanması, geliştirilmesi ve uygulanması için çok katmanlı algılayıcı (ÇKA) (MLP – Multi-Layered Perceptron) yapay sinir ağı kullanılmıştır. Burada, bir işin işlem sırası ve işlem zamanını bulmak için bir veri tabanından yararlanılmıştır. Yerel minimum çözümleri kontrol etmek için GYA kullanılmıştır. Ayrıca başlangıç üretim çizelgelemesi için bir sezgisel algoritma geliştirilmiştir. Bu çalışmada sunulan üretim faaliyet çizelgeleme sistemi, gerçek bir üretim ortamında da test edilmiş, faydalı ve mümkün çözümler elde edilmiştir. Bu sistem aynı zamanda şirketlerin, üretim faaliyet kontrollerini, müşteri hizmetlerini ve karlılığını geliştirmede de yardımcı olmaktadır (Feng vd., 2003).



6. ATÖLYE ÇİZELGELEME VE BENZETİM

6.1 Benzetimin Tanımı ve Temel Amaçları

Benzetim, dinamik bir sistemin özelliklerini ve davranışlarını bilgisayar aracılığıyla taklit eden ve sonuçlarını sayısal olarak değerlendiren bir tekniktir. Sonuçta elde edilenler, istenen model karakteristiklerine ait birer tahmindir. Diğer bir tanımla benzetim, incelenen bir gerçek hayat sisteminin belli bir zaman diliminde istenilen gerçek karakteristiklerini tahmin etmek amacıyla sistemin matematiksel, mantıksal bir modelinin geliştirilmesi ve bu sistem üzerinde deneyler yapılması sürecidir.

.Günümüzde endüstriyel problemlerin doğasındaki karmaşıklık maalesef pek çok analitik çözümü olarak dışı bırakmaktadır. Problemlerin yapısı değişen teknolojiyle birlikte karmaşılaşmakta ve bütünleşik sistemlerin sayısı hızla artmaktadır. Analitik yaklaşımların aksine benzetim modelleri, karmaşık problemlerin modellenmesi ve çözümünde daha başarılı olurlar. Değişkenler arasındaki etkileşimi benzetim modellerinde gözlemek daha kolaydır.

Benzetim modeli, sadece matematik denklemlerine değil, aynı zamanda denemelere dayanır ve model optimum sonuçlar ortaya çıkarmaz. Fakat benzetim modelleri yardımı ile alternatif çözümler ortaya koyarak, optimum sonuca en yakın çözümü verir. Benzetim, yalnız imalat ortamında değil, çevremizde görebileceğimiz pek çok alanda uygulanabilir.

Analitik yaklaşımların aksine benzetim modelleri, karmaşık problemlerin modellenmesi ve çözümünde daha başarılı olurlar. Değişkenler arasındaki etkileşimleri benzetim modellerinde gözlemek daha kolaydır. Ancak yoğun bilgisayar kullanımını gerektirir. Gerçek sistemden toplanan bilgiler, bilgisayarda geliştirilen modellere uygulanarak sayısal birtakım sonuçlara ulaşmak hedeflenir. Bunların değerlendirilmesi ve yorumlanması yapılarak sistem performans ölçütlerine ait birtakım tahminlerde bulunulur. Benzetim modelleri aracılığı ile en kötü durum senaryoları da incelenebilir.

Bir benzetim çalışmasının temel amaçları şöyle sıralanabilir:

- Bir gerçek hayat sistemini girdi ve çıktılarıyla matematiksel olarak ifade etmek,
- Gerçek sistemi, kurulan model üzerinden tanıyıp araştırmak, değişik kararları ve seçenekleri gerçek sistemde hiçbir değişiklik yapmadan deneyebilmek,
- Elde edilen bilgiler ışığında, sistemle ilgili ön görüşlerde bulunabilmek ve

uygulamaya esas olan kararları belirlemek.

Bilgisayar benzetimi, bir organizasyonun belirli işlevsel özelliklerini incelemek için, gerçek sistemden çok uygun bir bilgisayar modeli ile denemeler yapan deneysel bir yaklaşım olarak kabul edilebilir. Temelde benzetim, bir problemin içindeki başka yollarla taşınması fazlasıyla zor olan büyük miktarlardaki verileri işlemek ve analiz etmek için bir bilgisayarın gücünü kullanan deneysel bir metodolojidir. Benzetim, bir sistemi analiz etmek için etkin ve ekonomik hatta bazen tek mümkün yoldur. Doğrudan gerçek deneylerle kıyaslandığında, bilgisayar benzetimi yaklaşımının daha düşük maliyet, daha kısa süre, daha fazla esneklik ve çok daha düşük risk gibi üstünlükleri vardır. Sonuç olarak bu metodoloji hem akademik araştırmacılar, hem de uygulama mühendisleri tarafından imalat sistemleri alanında yaygın olarak kullanılmaktadır.

Mühendisler veya yöneticiler benzetimi, yeni sistemlerin planlanması ve tasarımının yanısıra mevcut imalat organizasyonlarının performansının geliştirilmesi için de kullanmaktadır. Örneğin, bugün herhangi bir yatırımcının yeni bir fabrika kurarken çeşitli benzetim araştırmaları yoluyla gelecekteki performansı yeterince incelemeyen, önemli miktarda para harcaması olasılığı düşüktür. Diğer yanda, akademik araştırmacılar; daha rasyonel organizasyon yapıları, daha etkin imalat stratejileri ve daha etkin üretim kontrol politikaları geliştirme çabası ile, çeşitli imalat sistemlerinin özelliklerinin daha iyi anlaşılması için benzetim tekniklerini kullanmışlardır.

6.2 Benzetimin Uygulama Alanları

Benzetimin kullanıldığı bazı uygulama alanları şöyle sıralanabilir:

- Üretim/imalat sistemlerinin tasarım ve analizi
- Montaj hattı dengeleme
- İşgücü planlaması
- Malzeme taşıma sistemleri
- Yeni askeri silah ve sistem taktiklerinin saptanması
- Bir envanter sistemindeki sipariş planlarının incelenmesi
- Satış tahminleri

- Üretim planlaması ve çizelgelemesi
- İletişim sistemlerinin ve bunlar için gerekli mesaj protokollerinin tasarımı
- Otoyollar, havaalanları, metrolar ve limanların tasarım ve işletimi
- Ambulans bulundurma noktalarının ve buralardaki araç sayılarının saptanması
- Yangın söndürme istasyonlarının yerlerinin ve buralarda bulundurulması gerekli minimum araç sayılarının saptanması
- Finansal veya ekonomik sistemlerin analizi
- Dağıtım kanallarının tasarımı
- Bir bilgisayar sisteminin donanım ve yazılım gereksinimlerinin belirlenmesi
- İşletme yöneticilerinin eğitilmesi (işletme oyunları/firmabenzetimi)
- Alınacak riskleri minimize etmek için uzay uçuşları denemeleri
- Tamir-bakım sistemleri (Halaç, 1998).

Endüstride problem çözümünün gitgide karmaşıklaştığı bilinmektedir, o kadar ki, yöneticiler karar vermede fazlasıyla etkin bir rehberliğe ihtiyaç duymaktadır. 1950'lerden bu yana bilgisayar, yönetimi iyileştirmede ve mühendislik kararlarında kilit bir faktör durumuna gelmiştir. Bununla beraber, burada bilgisayar kullanımı temel olarak durum bilgilerine hızlı ve doğru ulaşım ile sınırlandırılmıştır. Yalnızca özel bir kaç uygulamada bilgisayar, karar vermenin gerçek sürecini iyileştirmek için idari uzmanlık ile beraber kullanılmıştır. Bir olayın dinamik karakterini tahmin etmede temel bir yöntem olması ve böylelikle karar sürecinin temelini geliştirmesi ile bilgisayar benzetimi bu uygulamalardan biridir. Bir bilgisayar benzetim modeli ile, bir sistem analiz uzmanı veya yönetici, gerçek sistem ile deney yapma gerekliliği olmadan, bir prosesin davranışını gözleyebilir. Çeşitli bozucu etkiler altında sistemin performansını değerlendirmek veya darboğazları belirlemek için, örneğin farklı imalat akışlarını, yeni işlem koşullarını, yeni teçhizat yerleşimlerini veya farklı çevrim zamanlarını deneyebilir.

İmalat sistemleri alanında, iki temel benzetim uygulamasını görmek mümkündür. Sistem değerlendirmede değişen işlem koşullarına göre uyarlanması veya genişletilmesi gereken mevcut bir sürecin benzetimi yapılır. Sistem tasarımında, pahalı tekniklerden kaçınmak ve aşırı

işlem koşullarına tepkiyi gözlemek için yeni bir süreç incelenir (Wu, 1992).

6.2.1 İleri İmalat Sistemlerinin (İİS) Tasarımı

İleri imalat sistemleri (İİS)'nin (AMS - Advanced Manufacturing Systems) yerleştirilmesi, ilgili sistemlerin başarıyla tasarlanması, yerleştirilmesi ve işlemlerinden sorumlu olan sistem tasarımcıları ve mühendisler için büyük çaplı sorunlar ortaya çıkarır. Genel olarak İİS gerçekten tanıtılmadan önce, bir İİS uygulamasının olası etkilerinin yoğun şekilde analiz edilmesi gereklidir. Yalnızca bu yolla bir taahhüt yapılmadan önce sözkonusu olan büyük sermaye toplamları için tatminkar bir hüküm verilebilir.

Sonuç olarak bilgisayar benzetimi, benzetim tekniklerinin çok yönlülüğü ile birleşen bilgisayar yeteneklerinin sözkonusu işlevleri yerine getirmek için çok güçlü ve esnek bir araç oluşturması nedeniyle İİS tasarımında çok geniş bir uygulama alanı bulmuştur. Bir bilgisayar benzetimi modeli, sistemlerin karmaşıklığı ile savaşılabılır ve ek olarak işlemlerini muhtemel yatırımcılara açıklayabilmek için etkin bir iletişim aracı sağlayabilir. Benzetim kullanımı ile, İİS girişiminin riskleri önemli ölçüde azaltılabilir.

Bilgisayar benzetimi kullanılarak, alternatif İİS performanslarının değerlendirilmesi için iki seviyede sistem tanımı yapılabilir. İlkinde, brüt işlem karakteristiklerinin bir değerlendirmesi yapılır. Örnek olarak, iş istasyonu kullanımının ortalama ve varyansları, üretim oranı, parça bekleme süresi ve taşıyıcı kullanımı gibi. Taşıyıcı sayısı, palet sayısı, darboğaz problemler ve teçhizata müdahalelerin etkisi gibi teçhizat gereksinimlerinin belirlenmesi için de daha detaylı benzetim modelleri kullanılabilir. Bu tür bilgiler, sistemin performansını eniyeleyecek işlem stratejilerini formüle etmek için tasarım ekibine gerekli olacaktır. Ayrıntılı modeller kaba modellerden çok daha fazla hesaplama zamanı gerektirir ve benzetim akışları gerçek üretimin bir kaç günü veya haftası ile sınırlandırılmak zorundadır. Gerçekte tasarımda ve bir İİS'nin işlemlerinde sözkonusu olan problemler aşağıdaki üç seviyeli hiyerarşi ile açıklanabilir:

Seviye 1 : İİS'nin temel bileşenleri

Seviye 2 : Sipariş süreçleme ve planlama

Seviye 3 : Çizelgeleme ve tezgah kontrolü

Bilgisayarlı benzetim teknikleri bu problem seviyelerinin her birinde etkin olarak kullanılabilir. İmalat hücrelerinin tasarımında benzetimin yararlarını incelemek için çok sayıda araştırma yapılmıştır. Benzetimin, ayrıntılı tasarım dallarının belirlenmesinde güçlü bir araç olduğu iyi

bilinmektedir. Dolayısıyla bir çok firma, yeni bir tasarımın uygulamaya konması hakkında nihai bir karar vermeden önce önerilen imalat olanakları üzerinde benzetim çalışmalarının yapılmasına ihtiyaç duymaktadır. Bunlara rağmen, İİS'ye uygulandığında tasarım benzetiminin dezavantajları da yok değildir; bunlardan en önemlisi çok maliyetli olması ve pahalı kaynaklar kullanmasıdır. Firmaları benzetim kullanmaktan alıkoyan temel faktör olarak maliyet belirlenirken, toplam yatırımın % 2 - % 4'ü arası bir masrafın kabul edilebilir bir ölçü olduğu öne sürülmektedir (Thompson, 1994; Haddock, 1988).

6.3 Bilgisayar Benzetiminin Karakteristikleri

Karar vermeye bir yardımcı olarak, bilgisayar benzetim tekniği diğer yaklaşımlarla kıyaslandığında istenen özelliklere ve maalesef bazı dezavantajlara sahiptir.

6.3.1 Bilgisayar Benzetiminin Kazançları

Bilgisayar benzetiminin en önemli üstünlüklerinden biri çok esnek bir yaklaşım olmasıdır. Makul esneklikteki bir yapıda bir model oluşturulduktan sonra hızla ve ucuz şekilde yeni özellikler kazandırılabilir. Sıklıkla, aynı modelin ek seçenekleri değerlendirilebilmesi için yalnızca girdi verilerinin değiştirilmesi gerekir. Analitik bir yaklaşımla bir sistemi yeniden modellemek çok daha uzun sürer.

Çoğu sistem fazlasıyla karmaşıktır ve analitik modeller yalnızca basitleştirici kabuller kullanılarak oluşturulabildiğinden sıklıkla sadece durgun olay durumlarını tahmin edebilirler. Gerçekte bir imalat süreci sisteme kısa, düzensiz aralıklarla sürekli verilen parçaları içerir. Böyle bir durumda zaman zaman sistemin geçici davranışının analiz edilmesi gerekir. Bu işlevi yerine getirmek için bilgisayar benzetimi çoğu zaman tek uygun araçtır.

Bilgisayar benzetiminin en önemli üstünlüklerinden biri de incelenen süreci canlandırabilme yeteneğidir. Canlandırmanın sonucunda, modeli kuran için kazançlar, modeli kuran ve kullanan arasında karlı iletişim ve kullanıcılara ve yönetime sunušta kazançlar elde edilir. Canlandırmanın sağladığı iletişim kolaylığı nedeniyle model kullanıcısı aktif olarak bir benzetim modeli ile model geliştirme çevrimine katılır. Bunun sonucunda modelde ve kullanıcının kazançlarında önemli iyileşmeler gerçekleşir. Bir grafik gösterim, benzetimi yapılan sürecin mantıksal davranışı hakkında çok iyi fikir verebilir. Böylelikle programcı,

sistemde hal deęişikliklerinin mantıksal sırasını takip edebilir. Dinamik bir grafik gösterimi izlenerek özel bir deneyin verimli olup olmadığını görmek kolay olduğundan dolayı iyi tasarlanmış grafikler model ile etkin deneyler yapmaya yardımcı olur.

Bunların yanı sıra, uygun gösterime sahip bir canlandırma çeşitli yazılı dokümantasyonlar gerektirebilecek bilginin çok kısa bir sunuş süresinde aktarılmasını sağlar. Sonuç olarak bu kullanıcılara inanabilecekleri bir şey sunulduğunu hissettirir. Bir benzetim çalışmasının ulaşılabilirliğini ve güvenilirliğini geliştirme amaçlı bir araç olarak canlandırmanın kullanılmasıyla benzetimin bir değer olarak daha hızlı büyümesi ve imalat sistemleri mühendisliğinde bir teknik olarak kabul edilmesi olasıdır.

Bilgisayar benzetiminde modeller geniş ve karmaşık olabilir ve kurulmaları uzun sürebilir. Bunlara rağmen halen diğer analitik yöntemlere göre bir çok üstünlükleri vardır. Diğer deneylere kıyasla, ki burada gerçek bir sistemin fiziksel olarak modellenmesi gereklidir; bilgisayar benzetiminin düşük maliyet, zaman tasarrufu, kesin cevap ve emniyet gibi avantajları vardır. Bir bilgisayar benzetim modelinin genellikle dinamik veya geçici etkilerle başedemeyen matematik modellemeye göre de üstünlükleri vardır. Kuyruk teorisi modelleri, yalnızca basit sistemlerin sabit halde işlemlerini analiz etmede kullanılabilen ve yalnızca belirli olasılık dağılımlarını yönetebilen yaklaşımlara bir örnektir. Bu, incelenebilecek problemlerin çeşitliliğini kısıtlar. Diğer işlemsel araştırma yöntemleri de örneğin yalnızca çok fazla kısıtla ulaşılacak sonuçlar gibi, yönetim kararlarına yardım etmede sıklıkla yetersiz kalır. Analitik yöntemlerle bir atölyede işlerin çizelgelenmesi tipik bir örnektir.

6.3.2 Bilgisayar Benzetiminin Kısıtlamaları

Sistem tasarımı ve analizine yardımcı olan benzetimin belirli yetersizlikleri de vardır. Belki en önemlileri, uzmanlık gereksinimi ve benzetim modeli kurmak için gerekli zamandır. Verilen en iyi şartlarda bile benzetim projeleri daima zaman alıcıdır ve sıklıkla sonuçlar alınıncaya kadar aylar geçer.

Ayrıca benzetim kullanımının başka zorlukları da vardır. İlk problem, makul bir sürede benzetimi yapılabilecek tasarım seçeneklerinin sayısıdır. Benzetim çıktılarının istatistiksel özellikleri tercihen tüm akla uygun senaryolarla, her sistem düzenlemesi için çok sayıda benzetim çalışmasının gerekli olduğu anlamına gelir. Bu, tasarımcının daha alternatif tasarımlar oluşturma serbestliğini kısıtlar. Ayrıca, benzetim modelinin hızlı biçimde uygulanması için gereken çabanın hesaba katılmaması tehlikesi de her zaman vardır. Bir

çok standart benzetim aracı mevcut olmasına rağmen her uygulamada tek bir karar kuralları kümesi ve kendi çözümünü gerektiren bir yapı vardır. İkinci konu da benzetim modellerine ait verilerin elde edilebilirliğidir. Benzetim modelleri gerçek bir sistemi temsil eder ve bu yüzden mesela süreçleme ve talep bilgileri ve tasarım safhasında hazır olmayan işlem kuralları gibi tasarlanan sistemi detaylandıran farklı veriler sağlanmalıdır. Bir diğer problem de benzetim modelinin başlangıç safhasıdır. Planlama aşamasında tanımlanmamış bir çok problem olduğundan, benzetim modeline süreci temsil edeceği farzedilen keyfi bir aşama konulmalıdır. Bu problemlerden dolayı bir benzetim projesi ne tasarımcının ne de kullanıcının tasarım planının ilk aşamalarında tamamen tahmin edebildiği bir risk faktörü içerir.

Ek olarak, benzetim deneylerinin istatistiksel bir tabiatı olduğu ve sonuçları da buna göre yorumlanması gerektiği her zaman hatırlanmalıdır. Bu tür bir yaklaşımla bir problemin optimum çözümü garanti edilemez. Tüm bunlarla beraber, benzetim, planlama işlemi için tek seçenek değildir, her zaman bir ön tasarım safhasına gerek duyar (Law ve Kelton, 1991).

6.4 Benzetim Modellerinin Kurulması

Tüm benzetim projeleri şu adımları izler:

- Veri toplama
- Problem analizi
- Benzetim modelini belirleme
- Model programlama
- Modelin incelenmesi
- Benzetim işleminin yapılması
- Değerlendirme ve benzetim sonuçlarının yorumlanması
- Rapor oluşturma

Bir modelleme yaklaşımı seçiminde, problemdeki sistemin karakteristikleri gözönüne alınmalıdır. Bundan dolayı, bu alt başlıkla ilgili uygulamalarıyla beraber çeşitli benzetim yaklaşımları açıklanacaktır. Genel olarak kullanılan benzetim modelleri şu şekilde

sınıflandırılabilir:

- Statik/Dinamik
- Sürekli/Süreksiz (Ayrık-olaylı)
- Stokastik/Deterministik
- Toplu/Toplu olmayan

Statik benzetim modeli sistemin belirli bir zamandaki gösterimidir. Böyle modeller aktivitesiz yapıya sahip olarak tanımlanırlar ki bu benzetim prosesinin zaman içinde yayılmayacağı anlamına gelir. Monte Carlo benzetimi bu modelin tipik örneklerindendir

Statik sistemlerin tersine, dinamik benzetim modelleri yapısal bileşenleri aktivite ile birleştirir. Bunun anlamı, dinamik benzetim modeli zaman içinde yayılan sistemin bir gösterimidir. Bu tip modele tipik bir örnek, zaman içinde iş parçalarının sisteme girip çıktığı bir imalat sistemi benzetimidir. Bu tip bir dinamik benzetim modeli, örneğin, toplam maliyeti minimum yapmak üzere ne zaman ve ne kadar üretmeliyiz gibi karar değişkenlerinin yaklaşık değerlerini bulmaya yardımcı olur.

Sürekli benzetim, durum değişkenlerinin zaman içinde sürekli değiştiği bir modeldir. Sürekli modelleme çıkarma, ayrıştırma ve toplama üzerine kurulmuştur ve sistemde varolan ilişkileri anlatan bir veya daha fazla diferansiyel ve/veya cebrik eşitlikleri içerir.

Genellikle matematik eşitliklere dayanan sürekli benzetim modellerinin tersine, ayrık-olaylı (süreksiz) benzetim durum değişkenlerinin ani farklı olaylarla değiştiği bir modelleme sistemi gösterimidir. Ayrık-olaylı modellemeyi tanımlamanın genel bir standardı yoktur. Bununla birlikte, ayrık olaylı benzetimin birçok uygulaması, bir tip veya başka kuyruklama yapılarını içerirler ve bundan dolayı aşağıdaki terminoloji sıkça kullanılır.

Yapısal Elemanlar:

1. Varlıklar (veya Müşteriler): Bunlar bir sistemin benzetim modelindeki başlı başına tanımlanabilen ve işlenebilen geçici temel elemanlardır. Bir atölyede işlenmeyi bekleyen işler, imalat sisteminin tipik varlıklarıdır.
2. Özellikler: Sistemdeki her varlığın bir veya daha fazla özelliği olabilir. Özellikler varlıkla ilgili bilgidir. İş numarası, müşteri bilgisi, malzeme tipi, renk ve arabanın kaç kapısının olacağı bir imalat çevresindeki tipik özelliklerdir. Bu özellikler

benzetim prosesi esnasında birçok yönden faydalı olmaktadır. Mesela, sistemdeki her varlığı tanımlamak için veya varlıkların şimdiki durumlarını takip etmek için kullanılabilirler.

3. Kaynaklar: Bunlar model yapısının sürekli temel elemanlarıdır. Geçici elemanlara yani varlıklara servis sağlamakla sorumludurlar. Bunlar tipik olarak bir imalat sistemindeki tezgahları ve diğer iş istasyonlarını kapsar.
4. Kuyruklar: Kuyruk, sistemde varlıklar kaynak servisi beklerken, kaynaklar herhangi bir nedenle boş olmadığı zaman oluşurlar. Genelde, kaynaklar ve kuyruklar, varlıkların içinden geçtiği bir benzetim modeli ağı oluşturmak üzere birbiriyle birleştirilebilirler.

Dinamik Elemanlar:

5. Olaylar: Bir kaynağın bir varlıkla buluşup meşgul duruma geçmesi veya bir varlığın servisini bitirip tekrar serbest kalması gibi sistem durumundaki değişimlerin oluştuğu zaman içindeki noktalardır.
6. Aktiviteler: Her olayda başlatılan operasyon ve prosedürlere aktiviteler denir. Aktiviteler sistem varlıklarının durumlarının transformasyonundan sorumludurlar. Örneğin, tezgahda işleme aktivitesi hammaddeyi parçaya dönüştürür. Bir aktivite başlamadan önce belli kesin durumlar sağlanmalıdır ve bir kere başladığında, bir aktivitenin bitmesi genellikle zaman alır.
7. Prosesler: Proses, basitçe, bir sıralı olaylar grubudur. Mesela, bir hammadde bir iş istasyonuna gelir, işlenir ve sonra da bir sonraki iş istasyonuna gider. Genelde, bir proses bir varlığın sistemde ilerlediği süre içindeki tecrübesini tanımlar. Bazen benzetim prosesinin gerçekleşmesi için benzer prosesler faydalı olur.
8. Benzetim Saati: Bu, benzetim devam ettikçe benzetimi yapılan zamanın şimdiki değerini izlemek için kullanılan bir benzetim modeli değişkenidir. Bu değişkenin değerine göre, benzetim prosesinde yaklaşık aktiviteler başlar.

Ayrık-olaylı benzetimin anahatları yukarıdaki tanımlamalara göre çıkartılabilir. Sistem, herbiri Q_i adlı kuyruktan sorumlu, I tane kaynak S_i ($i = 1, 2, \dots, I$) içerir. Bu kaynaklar ve kuyruklar keyfi olarak birbirine bağlanabilirler. Dinamik benzetim prosesi sırasında, geçici varlıklar E_j ($j = 1, 2, \dots, N$) kaynaklara ulaşırlar; bir veya daha fazla aktivite

yer aldıktan sonra varlıklar yeni bir kuyruğa Q_k girerler ve buna bağlı olarak kaynak S_k 'ya Q_k 'nın durumuna göre (dolu ya da boş), S_k 'nın durumu ve E_j 'nin öncelikli özelliği, E_j , w_{jk} gibi bir bekleme gecikmesine maruz kalacaklardır (w_{jk} , 0 değerini alabilir). E_j 'nin kaynak S_k 'da işleme zamanı geldiğinde, E_j ilk sistem parametrelerinden biri olan P_{jk} gibi proses gecikmesine maruz kalacaktır. Eğer E_j S_k 'yı terkettiğinde S_k hala fonksiyonel kalmaya devam ediyorsa, bir sonraki varlık S_k 'da işlenebilir demektir. Eğer başka varlık yoksa S_k yeni bir varlık gelinceye kadar boş bekler. Herhangi bir kaynaktaki arızalar ek gecikmelere sebep olacaktır. Dinamik proses önceden belirlenmiş bir durumla karşılaşıluncaya kadar sürer. Örneğin, benzetim saati istenilen benzetim zamanına eriştiğinde; N adet varlığın tümü sistemden geçtikten sonra veya herhangi başka bir kriter sağlandığında.

Bir iş istasyonu iyi bir örnek teşkil eder. Bir iş parçası işlenmek için geldiğinde gerekli tezgah boş değilse tampon stokta bekleyecektir. Tezgah yeni bir iş almaya hazır olduğunda ve tamponda da bekleyen bir iş varsa işlem başlayabilir. Bundan dolayı, sistemin durumu zamanla durumdan duruma ayrılarak değişir. Bir sistem analiz uzmanının bakış açısından, tezgah ile ilgili tek münasebeti tezgah dolu olduğunda, boş olduğunda veya bozuk olduğundadır. Diğer hallerde tezgahın parçayı nasıl işlediği önemli değildir. Ayrılarak durum değişimi perspektifinden bakıldığında, bu tezgah istasyonu örneğinde bir dizi açık sistem olayları göze çarpıyor. Malzemenin gelişi, malzemenin boş bir tezgaha yüklenmesi, bitmiş iş parçasının tezgahdan sökülmesi, tezgahın yeni varlığa hazır hale gelmesi ve tezgah arızaları gibi olaylar açık sisteme örnek verilebilecek olaylardır. Tezgahın parçayı işleme zamanı rassal dağıtımla veya bir sabitle (deterministik benzetim) belirlenebilir. Bu sistemdeki diğer aktiviteler için de geçerlidir. Ayrılarak-olaylı bir benzetimde esas ilgi sistemin durum değişiminde ve zaman içinde varlıkların ve kaynakların kesişimlerdeki benzer değişimlerde.

Görülüyor ki ayrılarak-olaylı benzetim modellemenin genel problemi yukarıda bahsedilen yapısal ve dinamik elemanların kesişimlerinin nasıl temsil edileceğidir. Hemen hemen tüm benzetim dilleri bu prosesi gerçekleştirmek için birkaç temel yaklaşımın birini kullanırlar. En azından dört yaygın kullanılan yaklaşım vardır.

Bir ayrılarak sistemin kaynaklarının ve varlıklarının aktiviteleri arasındaki ilişkileri ve fonksiyonlarını açıklamak için, tüm elemanları temsil edebilecek bir tekniğe ihtiyacımız var. Aktivite çevrim diyagramı tekniği bu amaca uyan bir tekniktir.

Stokastik ve deterministik sistemleri birbirlerinden ayıran şey, eğer benzetim modeli sistem parametresi olarak hiçbir rassal değişken içermiyorsa bu sistem deterministiktir. Bir girdi kümesi verildiğinde deterministik benzetim modeli bir dizi sistem cevabı üretecektir. Diğer yandan, eğer sistem parametrelerinin bir ya da daha fazlası tahmin edilemiyorsa, fakat belli bir olasılık dağılımından elde edilebiliyorsa bu benzetim modeli stokastiktir. Bir girdi kümesi verildiğinde stokastik benzetim modeli sadece gerçek sistem cevaplarının bir tahminini verecektir. Çünkü çıktıların kendisi rassal değişkenlerdir. Stokastik benzetimin birçok örneğinde, gerçek sistem cevaplarını tahmin edebilmek için benzetimin bir dizi tekrarlı çalıştırılması gerekmektedir. Bu da tahmin edilen sistem değişkenliğini azaltır.

Bilgi toplama, bir benzetim modelinin tasarımında karar verilecek en önemli sistem değişkenlerinden birisidir. Bunun nedeni düzgün bir bilgi toplama bir benzetim modelinin geliştirilme ve korunma maliyetlerinde ve daha da önemlisi modelin yararlılığında (bu hem benzetim hem de analitik modeller için doğrudur) çok önemlidir. Bu kritik bilgi toplama seviyesi sorusu, ürün yapısı ve imalat sisteminin operasyonel yapısı analiz edilerek, geliştirilecek benzetim sisteminin amaçları doğrultusunda çözülebilir. Üretim planlama çalışmalarında açıkça tanımlanmamasına rağmen genellikle bilgi toplama seviyesi ile üretim planlama ve kontrol hiyerarşisi seviyesi arasında bir bağlantı vardır.

Kümülatif modeller genellikle ortaklık düzeyindeki uzun dönemli politika çalışmalarında kullanılırlar. Bu tür bir benzetim modelinden yararlanmak ortaklığın çevresi, stratejik kararlar ve geliştirilen planlarla ilişkilidir. Buradaki ana etkenler, örneğin ortaklığın pazar durumu, yeni ürünlerin geliştirimi, tesis alım satımı ve işgücü seviyelerinin uyumudur. Bu yüzden sistemdeki ürünler, işgücü, donanım ve diğer kaynaklar modelde kümülatif varlıklar olarak yer alabilir. Detaylı üretim süreci ve üretim çıktıları sınıflanmaz. Bu tür model, genel olarak imalat organizasyonunun üretim yönetiminde kullanışlı bir analiz aracı içermeyebilir.

Bundan farklı olarak, atölye düzeyi altında kısa döneme uygun planlı problemler vardır. Bazı modellerdeki varlıklar üretim için gerekli olan sünek malzemeler, parçalar ve alt montajın detaylı bir listesini oluşturabilir. Tezgahlar ve operatörler de uygun varlıklar olarak gösterilecektir. Benzetim süreci, simülatörün tüm işletme ihtiyaçlarını hesapladığı bu kategorinin örnek olduğu bir MRP sisteminden faydalanır. Bu işletme ihtiyaçları varlıklar, miktar ve ihtiyaç zamanı ile birlikte elverişli kapasite hakkındaki detaylı bilgi şeklinde sınıflandırılabilir. Yarı mamul hareketi daha sonraki günler veya haftalara ait kapasite ihtiyaçlarının en iyi şekilde belirlenmesi için benzetimi yapılabilir (Law ve Kelton, 1991; Askin ve Standridge, 1993).

6.4.1 Benzetim Modeli

Bir bilgisayar benzetim modeli geliřtirmek için, analize yönelik iki programlama süreci seçeneđi vardır: Model, özel-amaçlı benzetim dillerinden biriyle yazılabilir veya genel-amaçlı programlama dillerinden biriyle yazılabilir. 1960'ların başında çok sayıda benzetim modeli geliştirildiğinden beri, bilgisayar benzetiminin bazı genel prensipleri ortaya konmuştur. Yapım-kontrol prosedürleri (benzetimin her türü için), sıradaki prosedürleri (sıradaki model yapıları için) ve rassal süreç prosedürleri (stokastik benzetim için) gibi kesin birtakım genel mekanizmalara tüm benzetim süreçlerinde gerek duyulmuştur. Sonuçta bu benzetim prosedürlerini genelleyen ve benzetim modellerinin programlanmasını kolaylařtıran özel-amaçlı benzetim dilleri geliştirilmiştir. Bu benzetim dillerinin içerdiki etkenler şunlardır: Olay ve benzetim (eldeki) zamanı, varlıklar ve durumlar için model mantığı, ihtimal dağılım oluşumu, başlangıç model işlemleri, rapor-oluřturma işlemleri, interaktif benzetim kořumu, grafik animasyon, sonuç-analiz işlemleri ve diğerkontrol işlemleri ve hata mesajları.

Ayrıca, geleceğın fabrikasında Esnek İmalat Sistemleri (EİS) ve Bilgisayar Bütünleşik İmalat (BBİ) kavramları giderek önem kazandığından, giderek büyüyen sistemlerin benzetiminin yapılmasından dolayı bu tür bir benzetim için daha detaylı benzetim programlarını geliřtirmek önem kazanmıştır. Özetle burada iki kategoriden söz edebiliriz: Genel çalışma sistemleri ve özel çalışma sistemleri .

Genel çalışma sistemleri çeşitli oranlarda benzetim süreçlerinde kullanılabilirler. Temel benzetim fonksiyonları ve istatistiksel işlevleri vardır. Bu grubun avantajı meydana getirilmiş olan tüm modeller için uygun olmasıdır. Ancak, meydana getirilecek partiküler bir model için uzman modelciler gereklidir. Özel çalışma sistemleri problemlerin kesin kısımları için geliştirilmiştir. Bunlar modelleme ve programlama zamanını ve işini azaltırlar.

Seçim için karar süreci aşağıdaki sebeplerden birine bağılıdır: Birincisi, bilgisayar dünyasında genellikle doğru olan husus programlama dilinin seviyesi yükseldikçe, programın çalışmasının sınırlandığıdır. Özel amaçlı benzetim dillerinin önemli bir kısmında sistemler, yalnızca kesin türlere uyduğunda ticari olarak elverişlidir. Paketlerle elde edilen çıktı raporlarının biçimi dilin kullanımı yoluyla yazdırılır. Bu paketlerin çıktı raporları genellikle amaçlar ve standart sapmalar gibi sistem değışikliklerinin istatistiksel özetidir. Bu tür bir raporun yapısı yüzünden miktara bağılı olarak zamanla önemli düzeyde bir bilgi kaybı olabilir. Bazı durumlarda istatistiksel raporlardan farklı olarak sistem değışkenlerine bağılı zamanda, detaylı raporların oluşturulabilmesi de bir model için önemlidir. Bu özellikle model önüne

üretim planlaması için kullanıldığında önemlidir. Bazı raporlar yalnızca genel amaçlı program diliyle oluşturulabilir. Ayrıca, inceleme altındaki modellerin bazı özel işlevleri genel-amaçlı bir programlama dili kullanılarak daha iyi bir şekilde modellenebilir. Çünkü bu tür bir dil özel amaçlı pakete oranla daha esnektir.

İkinci olarak, eğer sistem analiz uzmanı bilgisayar programlama ve genel programlama dilleriyle modelleme konularının her ikisinde de tecrübeliyse bu, daha esnek bir modellemeye izin verecektir. Ayrıca, pratikte bir şirket genel amaçlı bir dilde belirli bir inceleme ve uzmanlaşma çalışması yapabilir. Bu dil, şirketin çeşitli departmanlarındaki tüm sistem mühendisleri tarafından kullanılan ana programlama dillerinden biri olabilir. Bu yüzden, bir benzetim modeli kullanıcı için kolay anlaşılır ve yürütülebilir olacaktır. Ayrıca gelecekte model gerekli olduğunda modelin taşınması mümkün olacaktır.

Üçüncü olarak da şirket için özel amaçlı bir benzetim paketinin kullanımı daha pahalı olabilecektir. Bir imalat işletmesinin geneli benzetim çalışmalarına tahsis edilmiş bir organizasyon değildir. Şirkette, paketlerin para ve zaman kullanımı araştırmasının yapılması gerekmebilir.

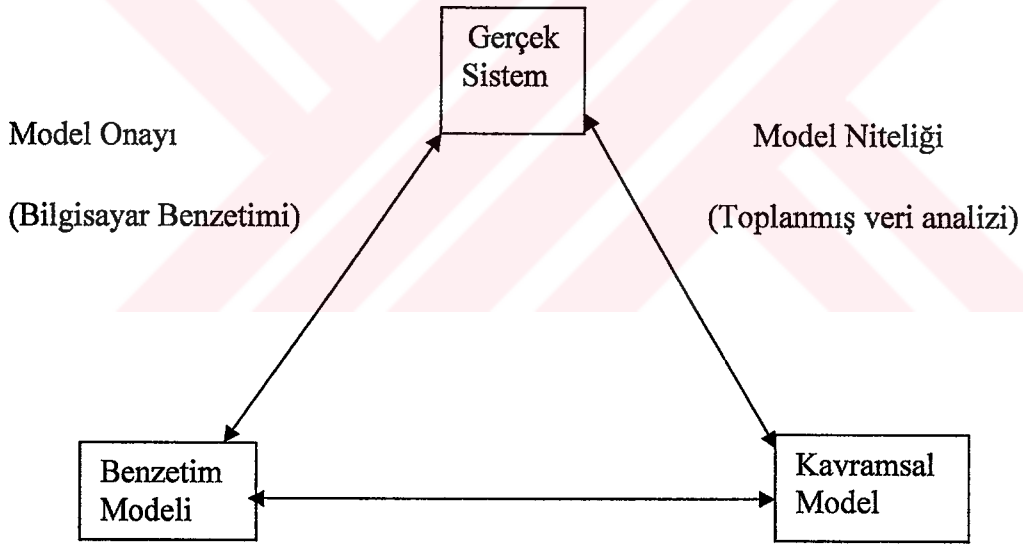
Yine eğer model özel amaçlı dillerden biriyle geliştirilecekse, programlama süreci ardındaki yapı ve mantık bilgisi benzetimin modelleme sürecinde modellemeyi yapana yardımcı olacaktır.

6.4.2 Benzetimin Geçerliliği

Burada dikkat edilmesi gereken bir ayrıntı verilerin analiz edilmesi ve model doğruluğudur. Bir sistem analiz projesi altındaki model geliştirme sürecinde görülen model geliştirme ve doğruluk analizinin önceki sonuçları Şekil 6.1' de grafik olarak gösterilen yapıda olmalıdır. Bu yapı üç temel elemandan oluşur. Model niteliği, geçerli sistem uyumunun kabul edilebilir bir seviyesinin oluşturulması için gerekli olan model yapısı, elemanları ve verisinin tanımlama prosedürüdür. Model analizi, sistem için geliştirilen modelin uygun çalışmasını sağlar. Bu tamamen modelin doğruluğuyla ilgilenir. Örneğin, eğer ilgilenilen model bilgisayar modelleme görevlerini kapsıyorsa, rutinlerin doğru programlandığı ve fonksiyonun tasarlandığı gibi mantıksal olduğu kesin olmalıdır.

Modelin nitelik safhası veri birikimi, açıklaması, parametre hesabı ve sistem yapısı ve fonksiyonlarının analizini içerir. Modelin temelini oluşturmak için kullanılan bilgi birikiminin

yetersizliğinden ötürü, model oluşturma görevlerinin ilk adımı veri birikiminden oluşur. Veri birikim süreci genellikle çok düzeyli ve model yapısındaki geniş ve yüksek maliyetli görevler gibi çok kapsamlıdır. Açıklayıcı ve sayısal nitelikteki gerekli bilgi, pratikte çeşitli kanallarla bir araya getirilebilir. Örneğin altında çalışılan sistemin çeşitli alanlarında yeralan anahtar personelle yapılan görüşmeler ve birikiminin yetersizliğinden ötürü, model oluşturma görevlerinin ilk adımı veri birikiminden oluşur. Veri birikim süreci genellikle çok düzeyli ve model yapısındaki geniş ve yüksek maliyetli görevler gibi çok kapsamlıdır. Açıklayıcı ve sayısal nitelikteki gerekli bilgi, pratikte çeşitli kanallarla bir araya getirilebilir. Normalde kanallar boyunca birikim için çok miktarda veri gereklidir ve bu daha sonra önemli bilgiyi belirlemek için analiz edilmelidir. Örneğin; ürünlerin özelliklerini, işletmenin şeklini, kapasitelerini ve imalat sistemlerinin yönetimi ile işlemsel politikasını açıklamak için bulunmalı, gruplandırılmalı, biriktirilmeli veya hesaplama yoluyla elde edilmelidir. Bu yetersizlik ve veri analizleri şekil 6.1’ de gösterilmiştir.



Şekil 6.1 Benzetim modeli gelişim süreci

Model yapısının ilk safhasında olduğu gibi modelleme prosesinin nitelik safhası da son modelin kalitesini garanti etmekte hayati bir rol oynar. Modelde veri girdisinin doğruluğunu kesinleştirmek için büyük dikkat gereklidir. Ayrıca modelin veri çıktısı döngüde doğrulanabilir. Bu sürece yardımcı olan elverişli veri analizi teknikleri mevcuttur. Örneğin, eğer ilgilendiğimiz model deterministik olmayan türden ise, elverişli veriler çoğunlukla önceki gözlemlere dayalı zaman serileri formundadırlar. Belki de deneyim amacıyla, bu verileri incelemenin son iki safhasında istatistiksel olarak yeniden üretilebilecek

olduklarından, bunları açıklamak üzere, örneğin normal, poisson dağılımı gibi uygun birtakım dağılımları kullanmak gerekebilir. Bu durumda, bu dağılımı modelde güvenli olarak kullanmadan önce uygun olduğunu düşündüğümüz dağılımın eldeki veri serileri ile ilişkisini test etmeliyiz (Haddock ve O'Keefe, 1990).

Model analizi ve doğruluğu, model kümülatif olduğundan, tanımlama problemi ile ilgilenirler. Model analizi temelde modelin kendisinin yapısıyla ilgilidir. Bilgisayar, program yazma ve çalıştırmadaki süreç tekrarını sağlar.

Doğruluk, her ne kadar modelin doğruluğu ile ilgilenirse de, model kurulmasında ve çalıştırılmasında farklı bir süreçtir. Gerçek sistem ile karşılaştırma gibi model davranışları ile de ilgilenir. Pratikte doğruluk incelemesi teknikleri geliştirmek için çalışmalar mevcuttur. Modelin doğruluğunun incelemesinin tümü birçok faaliyeti içerir. Bilgisayar benzetim modellerinde ise, modelin iyi kurulup kurulmadığının anlaşılması için bazı işlemler yapılmalıdır. Bunun için yüksek düzeyli doğruluğa sahip bir model geliştirilmelidir. Modeli eklemek için duyarlılık araştırması, tecrübe, gözlem veya elverişli diğer bilgilerle yapılabilenler kullanılmalı, araçların, varyansların ve dağılımların basit ampirik testleri elverişli veriler kullanılarak yapılmalı, turlama türü testler ve karmaşık istatistiksel testler elverişli veriler üzerinde koşulmalı, kuyruk teorisi sonuçları kullanılmalı ve özel veri birikiminden yararlanılmalıdır (Öztemel, 1996).

Her ne kadar tüm bu teknikler bir modelin doğruluğunun araştırılması için kullanılsa da, modellenmiş bir sistemin doğruluğunda karar vericilerin önemli bir etkisinin olduğu unutulmamalıdır. Ayrıca bir model, sadece modellemeyi yapanın modele etki gücüyle değil, derece için karar vericinin onun kullanışlı ve uygun olduğuna inanmasıyla da kullanılacaktır. Bu yüzden modeli geliştirirken, kullanıcıya gerekli olanlar modelin başarı araçları ve doğruluğudur. Buradan da, model geliştirimi süresince aşağıdaki soruların cevaplanması önemlidir :

1. Uzmanın fikri ile, modeldeki hipotezlerin başlangıç etkileri inanılır düzeyde mi?
2. Analiz çalışmalarının hassasiyeti, gerçek hayatta benzerleriyle karşılaştırılan çıktı örneklerini gösteriyor mu?
3. Model gerçek hayattaki gibi hareket ediyor mu?

Buradan, bir modelin doğruluk düzeyinin tespiti için çok sayıda yol mevcuttur. Bunlardan birincisi, modelin hala gelişimin ilk safhalarında iken, kullanıcıya giriş imkanı vermesidir.

Daha sonra kullanıcı, modelin büyüklük ve karmaşıklık içinde geliştirildiği gibi modelin doğruluğuna güveni artar. İkinci olarak, model içinde grafikler gibi daha verimli iletişim araçlarının kullanılmasıdır. Böylece kullanıcı, modelin çalışmasını izleyebilir ve buradan da davranışlarını değerlendirebilir. Bunların etkisiyle, analiz uzmanı ve müşteri arasında kapalı bir işbirliği oluşturulabilir (özellikle çeşitli ortaklık düzeylerinin yönetiminde). Böylece onların profesyonel fikri, modelin doğruluğunun ve modelde yapılacak değişikliklerin sürekliliği konusunda yarar sağlar.

Başlangıç Şartı : Bir benzetim çalışmasında benzetim, ortaya çıkan herhangi bir olay olmadan başladığında ve tüm kuyruklar boş olduğunda, geçici periyodun etkilerinin azaltılması önemlidir. Bunun sebebi bu periyod boyunca sistem, gerçek kararlı davranışlarını gösteremeyecektir (bu davranışlar, ilgilendiğimiz sistemin günü gününe operasyonlarında çalışan genel gerçek durumdur). Veri çıktısı analiz edildiğinde, model kararlı duruma ulaşmışken yalnızca sistem sonuçlarını kullanması arzu edilir.

Bu türdeki bir benzetim çalışmasındaki önemli detaylardan biri de modelin, benzetim boyunca kararlı duruma ulaştığına karar vermektir ve bu her zaman kolay bir iş değildir. Benzetim üzerindeki ilk etkileri azaltmak için kullanılan iki yöntem vardır. Bunlardan biri gruplandırılmış bir periyodda kullanılır. Bu yöntemle, benzetim ilk olarak herhangi bir çıktının alınmadığı gruplandırılmış bir periyod boyunca koşar. Veri çıktısı birikimi yalnızca geçici faz bitiminde görülen bu gruplandırılmış periyodun sonunda başlar. Diğer özellikler başlangıç şartları için kullanılır. Burada benzetim süreci sıfır olmayan şartlarda başlar. Böylece sistem, koşumun başında düzenli bir durumda olabilecektir.

Eğer periyoddaki koşum yeterince uzunsa, birçok durumda gruplandırılmış periyod yöntemi başlangıç şartı yöntemine tercih edilecektir. Çünkü düzenleyici yöntemle, veri kaydı süreci başlamadan önce düzenli konuma ulaşıldığına emin olunmalıdır. Bu son yöntemdeki yeterli büyüklükte etki, toplam başlangıç şartlarının seçimine bağlı iken mümkündür. Ancak, uzun gruplandırılmış periyodun kullanılmasıyla bilgisayar zamanında önemli miktarda bir kayıp problemi ortaya çıkar. Büyük modellerde bilgisayar zamanının verimli olarak kullanılması en önemli faktörlerden biridir.

Bilgisayar zamanını verimli kılmak için düzenli bir durumu sağlayan uzlaşık yöntemleri kullanmak da mümkündür. Benzetim denemesinin koşumu öncelikle düzenli duruma varılacak kadar uzun olmalıdır. Sistem, daha sonraki benzetimler için başlangıç sistem şartları olarak kaydedilip kullanılmalıdır. Bu yolla, etki her benzetim koşumunun ilerisindeki uzun

koşum periyodu gibi olacaktır. Bununla beraber bilgisayar zamanında da bir kayıp söz konusudur. Ayrıca bu yöntemle bazı etkilerin sıfırdan başlamayan şartlardan ötürü sonuçlanabileceği de unutulmamalıdır. Bu yolla kullanılan bir başlangıç şartı söz konusu benzetimin başlama noktasında iki zaman serisine birden bağlı olabilecektir (Birincisi, genelde başlangıç şartlarını çıkarmak için olan deneme serileri ; ikincisi ise geçerli benzetim için kullanılan gerçek serilerdir). Sonuçta bu noktada, bu süreksizlikte kesin bir düzeye girilebilir. Uygun durumlarda, bu süreksizlik sistemin dinamik sürecindeki geçicilik düzeyi benzetim çıktılarındaki sonuç etkilerinden ötürü kaçınılmaz olacaktır. Analiz uzmanı, hangi seviyedeki istenmeyen etkinin gözardı edileceğini belirlemelidir (Law ve Kelton, 1991).

6.5 Benzetim Tabanlı Atölye Çizelgeleme

Yaklaşık yirmi yıldan beri bilgisayarlar benzetim imalat sistemlerinin tasarım ve analizinde kullanılmaktadır. Son on yıl boyunca benzetimin sınırlı kapasiteli çizelgeleme konusunda kullanımı yaygınlaşmıştır. Sınırlı kapasiteli çizelgeleme, geleneksel bir benzetim analizinden bazı yönleri ile farklıdır: Geleneksel bir benzetim analizinde karar değişkenleri genellikle gerekli tezgah sayısı ya da kullanılması gereken malzeme taşıma teçhizatı gibi tasarımla ilgili kararlardır. Benzetim tabanlı atölye çizelgeleme sisteminde ise karar değişkenleri, işin atölyeye ne zaman gönderileceği, bir tezgahdaki işlerin hangi sırayla işleneceği, parçanın atölyede hangi rotayla dolaşacağı, çizelge zamanının aşılması gibi işlemsel kararlardır. Benzetim tabanlı atölye çizelgelemenin girdileri tipik iş emirleri ve teslim tarihleridir. Çıktısı genel benzetim özet istatistikleri değil, her tezgahta işlerin ne zaman başlayacağı, hangi siparişlerin zamanında teslim edilebileceği gibi bilgileri gösteren detaylı bir çizelgedir. Şunu belirtmek gerekir ki benzetim tabanlı atölye çizelgeleme optimum çizelgeyi oluşturmaz. Bunun yerine değişik “eğer-ne” senaryolarını deneyerek en uygun çizelgeyi oluşturmayı sağlar.

Bir benzetim tabanlı çizelgeleme, geleneksel benzetim modellerinden daha kısa bir süre için çalışır ve daha sık çalıştırılır. Ayrıca mevcut sistem bilgilerini alabilmek için imalat veri tabanları ile etkileşim içinde olması uygundur

Özellikle yüksek teknolojili imalat sistemlerinde benzetim tabanlı çizelgelemenin kullanılması faydalıdır. MRP, MRP-II gibi sistemlerden alınacak bilgiler ve atölyeden gelen anlık durum bilgilerini kullanarak alternatif çizelgelerin hazırlanması, bunların denenerek en uygununun seçilmesi gibi hızla yapılması gereken işlemler benzetim tabanlı çizelgeleme yardımıyla hızlı ve kolay şekilde gerçekleştirilir. Özellikle tezgahların boş beklemelerinin,

işlerin gecikmesinin çok önemli maddi kayıplar oluşturduğu pahalı ve yüksek teknoloji sistemlerde benzetim kullanılarak çizelgeleme yapılması karar vermeyi kolaylaştırarak maddi kayıpların en aza inmesini sağlar.

Bir atölyedeki sistemin genel yapısı şöyledir. Atölyeye gelen işler çizelgelenmiş tezgahların bekleme hattına alınırlar. Tezgah boşaldığı zaman öncelik kuralına göre seçilen işler tezgaha atanırlar ve işlem bittikten sonra işleneceği diğer tezgahın bekleme hattına taşıma aracı ile taşınırlar. İşlemleri tamamlanan işlerin gerekirse montajı yapılarak ürün elde edilir. Bu tip bir sistemi çizelgeleyebilmek için modeli oluşturmaktır.

Çizelgeleme ve kontrol amacıyla kullanılan modellere işlemsel (operational) modeller denir. Atölyenin benzetimi, benzetim saatini arttırarak atölyenin tüm çalışmasının taklit edilmesi işlemidir. Her arttırmada model parametrelerindeki gerekli değişiklikler yapılır. Her değişiklik bir olay anlamına gelir. Bir tezgah, operatör ya da iş istasyonu bir operasyonu tamamlayınca, daha sonra ne yapacağı konusunda bir karar vermelidir. Her varlık bir sonraki operasyonunu seçmelidir. Bu seçim bir yükleme ya da çizelgeleme kuralına göre yapılır. Atölye çizelgelemede kullanılan yükleme kurallarının etkinliği hakkında pek çok çalışma yapılmıştır. Çoğu yazılım sistemleri, kullanıcının seçebileceği standart yükleme kuralları seti sunarlar. Bunlardan bazıları şunlardır: En kısa işleme zamanı, boş zaman, en erken teslim tarihi gibi kurallardır. Fakat bu kurallar bazen gerçek sistemi tanımlamaya yetmemektedir. Bu gibi durumlarda kullanıcının tanımladığı kurallar kullanılmalıdır.

Benzetim tabanlı çizelgeleme ile oluşturulmuş “eğer-ne” tipi senaryolar karar vericiye şu alanlarda karar esnekliği sağlayabilir:

- İşlerin imalat alanına giriş sırasının ve zamanının belirlenmesi
- Parti büyüklüğünün belirlenmesi
- İşlerin belirli tezgahlarda sırasının belirlenmesi
- Alternatif iş rotalarının kullanılması
- Fazla mesai ile arttırılabilecek mümkün kapasitenin belirlenmesi

Bu ihtiyaçları karşılayacak bir model geliştirmedeki ana güçlükler şöyle gruplanabilir:

1. Üretim sistemine ait gerçek zaman (real time) verilerine ulaşabilme
2. Benzetim modelinin sonuçlarının yöneticiye zamanında ulaşabilmesi için bilgisayar

gücünün yeterli olması gerekir.

Eğer model sadece atölye düzeyinde stratejik karar verme amacıyla kullanılacaksa, çok az gerçek zaman verisi gerekir. Fakat model üretim çizelgeleme ya da kapasite analizi için gerçek zamanlı bir karar destek sistemi olarak kullanılacaksa gerçek zaman verilerine ulaşabilme önemli bir darboğazdır. Bu durumda imalat kontrol veri tabanının tamamının kullanılması gerekir (Dinçmen ve Demirci, 1990).



7. ATÖLYE ÇİZELGELEME TASARIMINDA YAPAY SİNİR AĞLARI İLE DENEYSEL ÇALIŞMA

7.1 Modelin Kurulması ve Tanıtımı

İncelenen atölyede 4 iş merkezi vardır ve her iş merkezi bir veya daha fazla aynı tezgah içerir. Tezgahlar bir anda yalnız bir parça işleyebilir ve o parçanın operasyonları bitirilinceye kadar diğer bir parçanın operasyonuna başlanamaz. Parçalar yükleme/boşaltma merkezinden sisteme girerler. Parçaların halihazırda paletler üzerinde ve (eğer gerekli ise) aparatlar (fixtures) üzerinde olduğu kabul edilir. Eğer bir parça rotası üzerindeki ilk iş merkezinin girdi/çıkış alanında (G/Ç) boşluk var ise, operasyonu yapılmak üzere merkezin girdi alanına yükleme/boşaltma merkezinden parça transfer etmek için bir taşıyıcı istenilir. İş merkezindeki bir tezgah kullanılabilir bir hale geldiği zaman, parça girdi alanından tezgaha otomatik olarak hareket ettirilir ve operasyon başlatılır. Bir parçanın operasyonu tamamlandığında, eğer merkezin çıkış alanında boş yer varsa, parça otomatik olarak tezgahın çıkış alanına hareket ettirilir. Eğer bir sonraki merkezin girdi alanında boşluk var ise parçayı doğrudan bir sonraki merkeze taşımak amacıyla bir OGA (Otomatik Güdümlü Araç) istenilir. Boşluk yoksa parçayı yükleme/boşaltma merkezine taşımak için bir OGA istenilir. Parçaya bütün operasyonlar bir kez icra edildikten sonra parça çıkış merkezine aktarılır ve burada sistemden ayrılır.

7.1.1 Atölye Planı

İş merkezi sayısı dört tanedir. Ayrıca bir yükleme-boşaltma merkezi ve bir merkezi stok alanı vardır.

7.1.1.1 İş Merkezleri Tezgah Sayıları

Her bir iş merkezindeki tezgah sayısı bir ile beş arasında değiştirilerek alternatif konfigürasyonlar elde edilmiştir.

Çizelge 7.1 İş merkezleri tezgah sayıları

İş merkezi	1	2	3	4	5	6
Tezgah Sayısı	1-2-3-4-5	1-2-3-4-5	1-2-3-4-5	1-2-3-4-5	1	-

7.1.1.2 Her İş Merkezinde Bulunan Girdi/Çıktı (G/Ç) Alanı Büyüklüğü

İş merkezi girdi/çıktı (G/Ç) alanı büyüklüğü her iş merkezi için birer adettir.

Çizelge 7.2 Her iş merkezinde bulunan girdi/çıktı (G/Ç) alanı büyüklüğü

İş merkezi	1	2	3	4	5	6
Girdi alanı	1	1	1	1	1	1
Çıktı alanı	1	1	1	1	1	1

7.1.1.3 İş Merkezleri Arası Mesafe

İş merkezleri arası mesafeler aşağıdaki gibidir.

Çizelge 7.3 İş merkezleri arası mesafe (metre)

İş Merkezi	2	3	4	5	6
1	10	20	30	10	20
2		10	20	20	15
3			10	35	15
4				40	15
5					10

7.1.2 Parça Bilgileri

7.1.2.1 Parça Tipi

Beş farklı parça tipi vardır. Belli bir parça tipi için bir talep olması ihtimali bir değişkendir. Çizelge 7.4 her bir sipariş için atanan parça tiplerinde kullanılan ihtimal dağılımlarını gösterir. İhtimal dağılımları bütün parça tipleri için eşittir.

Çizelge 7.4 Parça tipi ihtimal dağılımları

Parça Tipi	1	2	3	4	5
İhtimal	0.20	0.20	0.20	0.20	0.20

7.1.2.2 Her Bir Parça Tipi İçin Operasyon Sayısı

Çizelge 7.5 Parça tipi operasyon sayısı

Parça Tipi	1	2	3	4	5
Operasyon Sayısı	3	3	2	3	2

7.1.2.3 Her Bir Parça Tipi İçin İş Merkezi Sayısı

Çizelge 7.6 Her bir parça tipi için iş merkezi ismi

Parça Tipi	1	2	3	4	5
İş Merkezi	1 - 3 - 4	1 - 2 - 4	2 - 4	2 - 3 - 4	1 - 4

7.1.3 İşler

7.1.3.1 Parça Tipi

Parça tipi yukarıda da belirtildiği gibi eşit ihtimale sahip olan beş ayrı parçadan oluşmaktadır.

7.1.3.2 Sipariş Büyüklüğü

Her bir parça tipi eşit ihtimalle, gelişler arası süre ortalaması 5dk olan üstel dağılıma uygun olarak sisteme dahil olmaktadır.

Gelişlerarası Süre => Ort. = 5dk Üstel Dağılım

7.1.3.3 Teslim Tarihi

Her bir işin teslim tarihi “Toplam İş Hacmi (TİH)” ne göre belirlenmiştir. Bunun için ortalaması 300, standart sapması 40 olan normal dağılım kullanılmıştır.

$$\text{Teslim Tarihi} = t_{\text{geliş}} + 5 * N(300,40)$$

$t_{\text{geliş}}$: Geliş zamanı

$N(300,40)$: Ortalaması 300 ve standart sapması 40 olan normal dağılımdan elde edilen rastgele değeri ifade etmektedir. Bu değerler bir parçanın sistemde ortalama kalma süresine göre hesaplanmıştır.

7.1.3.4 Hazırlık ve İşlem Zamanları

Her parça tipi için hazırlık ve işlem zamanları aşağıdaki çizelgelerde gösterilmiştir.

Çizelge 7.7 Parça tipi 1 hazırlık ve işlem zamanları

Sıra	İş Merkezi	Hazırlık Zamanı	İşlem Zamanı
1	1	N (6,2)	N (9,3)
	2		
2	3	N (5,1)	N (11,2)
3	4	N (4,1)	N (6,1)

Çizelge 7.8 Parça tipi 2 hazırlık ve işlem zamanları

Sıra	İş Merkezi	Hazırlık Zamanı	İşlem Zamanı
1	1	N (5,1)	N (12,3)
2	2	N (5,3)	N (6,3)
	3		
3	4	N (4,1)	N (6,1)

Çizelge 7.9 Parça tipi 3 hazırlık ve işlem zamanları

Sıra	İş Merkezi	Hazırlık Zamanı	İşlem Zamanı
	1		
1	2	N (5,3)	N (7,2)
	3		
2	4	N (5,1)	N (8,2)

Çizelge 7.10 Parça tipi 4 hazırlık ve işlem zamanları

Sıra	İş Merkezi	Hazırlık Zamanı	İşlem Zamanı
	1		
1	2	N (4,2)	N (6,1)
2	3	N (5,1)	N (9,3)
3	4	N (5,1)	N (7,3)

Çizelge 7.11 Parça tipi 5 hazırlık ve işlem zamanları

Sıra	İş Merkezi	Hazırlık Zamanı	İşlem Zamanı
1	1	N (4,2)	N (10,3)
	2		
	3		
2	4	N (3,1)	N (5,2)

7.1.4 Öncelik Kuralı

İşlenmek üzere bekleyen parçalardan, öncelikle hangisinin işlem göreceğini belirlemek için kullanılan çizelgeleme stratejileri: En Kısa İşlem Zamanı (EKİZ), En Erken Teslim Tarihi (EETT) ve İlk Gelen İlk İşlem Görür (İGİG)'dir. Ayrıca, yükleme-boşaltma merkezi ve OGA kuyruğu için kuyruk öncelik kuralı İGİG'dir. En Kısa İşlem Zamanı Kuralı (EKİZ), en kısa işlem zamanına sahip parçayı öncelikli olarak sisteme alır. İşlem zamanı; operasyon zamanları ile hazırlık zamanlarının toplamıdır. Bir atölyede akış süresini (flow-time) minimize etmek için, mümkün operasyonlar içinde en kısa operasyon süresine sahip olan işi öncelikle tercih eden en kısa işlem zamanı (EKİZ) kuralı literatürde teklif edilmiştir (Baker, 1984). EKİZ sıralama, teslim tarihi belirlemede açık olarak kullanılsa bile, teslim tarihlerini karşılamada da çoğu defa etkilidir (böylece gecikmelerin ölçülmesinde de iyi olabilir). Akış süresi ve gecikmeler kriterlerinin her ikisinin de desteklediği özelliklerinden dolayı en kısa işlem zamanı (EKİZ) kuralı tasarım bilgi tabanının değişkenlerinden biri olarak seçilmiştir.

En Erken Teslim Tarihi Kuralı (EETT)'nda sisteme parça girerken parçanın teslim zamanı dikkate alınır. En erken teslim edilecek parça en önce sisteme alınır. Gecikmeler ölçüsü için teslim tarihine dayalı çizelgeleme kuralları vardır ve onlardan biri en erken teslim tarihi (EETT) dir.

İlk Gelen İlk İşlem Görür Kuralı (İGİG), bu stratejide işlem zamanları veya teslim zamanları göz önüne alınmaz, parçalar geliş sırasına göre sisteme girer.

7.1.5 Malzeme Taşıma Sistemi

Malzeme taşıma aracı olarak OGA (Otomatik Güdümlü Araç) tercih edilmiş ve konfigürasyon tasarımı sırasında üç ayrı değerde ele alınmıştır. Tasarımların bir kısmında bir OGA, bir kısmında iki OGA, bir kısmında üç OGA kullanılarak alternatif konfigürasyonlar oluşturulmuştur. Sistemde kullanılan OGA hızı 10 m/dk'dır.

7.1.6 Örnek Konfigürasyonlar

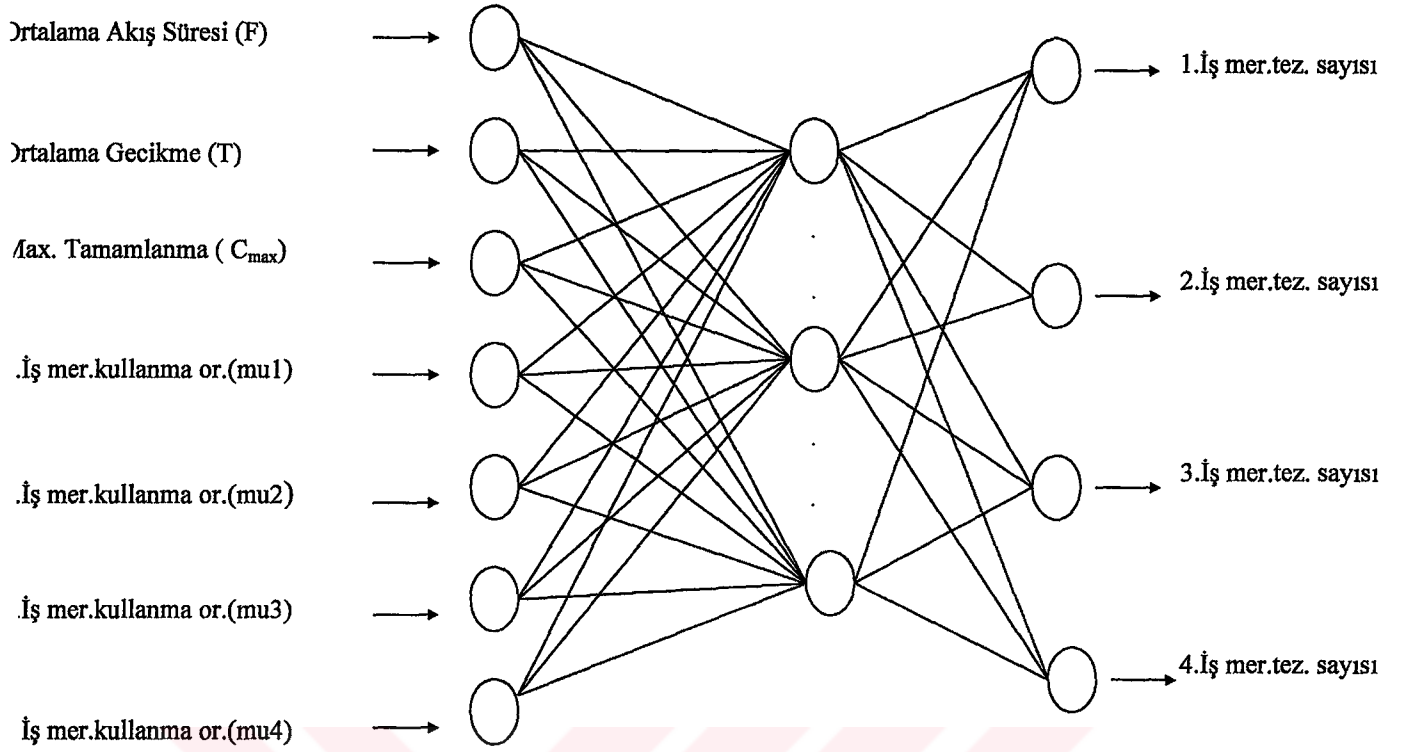
Örnek konfigürasyonların, her bir öncelik kuralı ve taşıyıcı sayısı çifti için, bütün sistemi temsil edecek şekilde tezgah sayısı alternatifleri ile belirlenmiştir.

Çizelge 7.12 Örnek setinde kullanılan konfigürasyon örnekleri

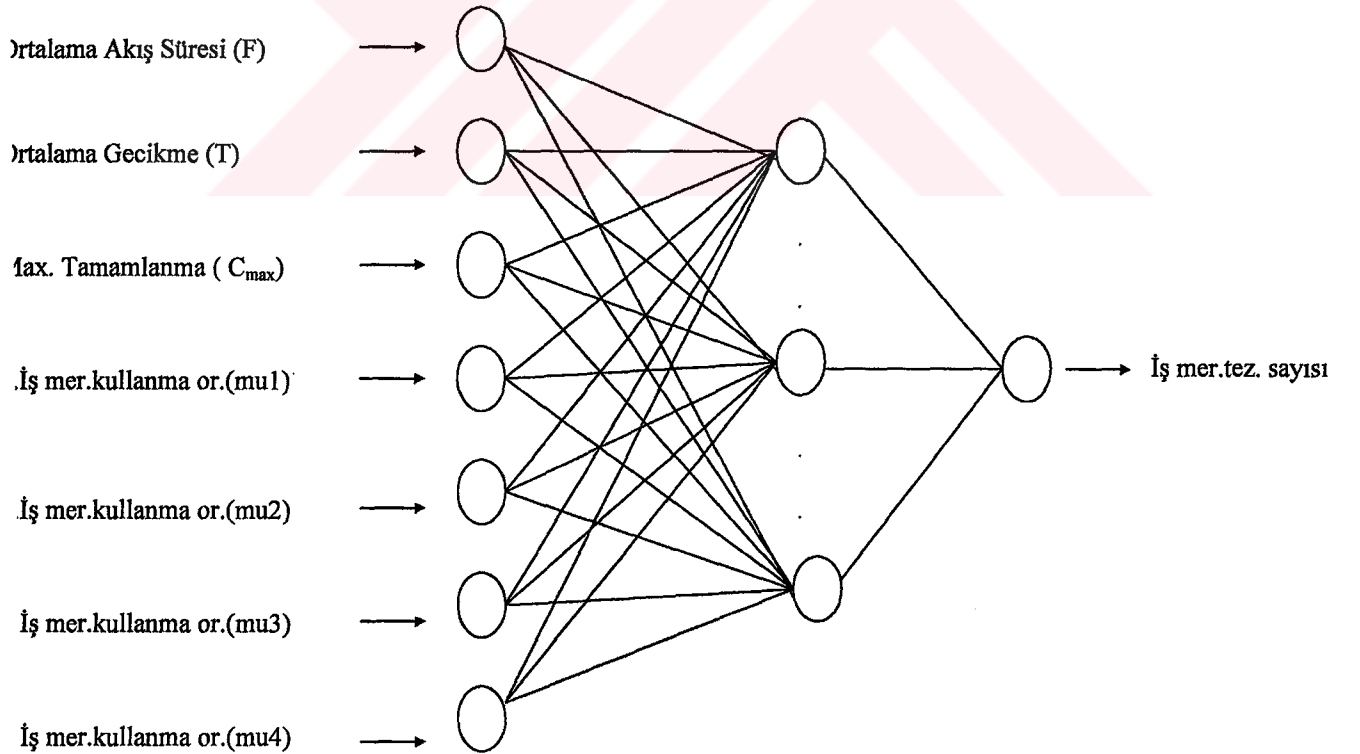
Taşıyıcı Sayısı = 1 OGA Öncelik Kuralı = EETT	Taşıyıcı Sayısı = 2 OGA Öncelik Kuralı = EKİZ	Taşıyıcı Sayısı = 3 OGA Öncelik Kuralı = İGİG
İş Merkezi Tezgah Sayısı	İş Merkezi Tezgah Sayısı	İş Merkezi Tezgah Sayısı
1 - 1 - 2 - 2	1 - 1 - 1 - 1	1 - 2 - 1 - 3
1 - 2 - 3 - 3	3 - 2 - 1 - 3	1 - 2 - 3 - 3
2 - 3 - 2 - 5	1 - 2 - 4 - 5	1 - 2 - 3 - 4
1 - 2 - 4 - 3	2 - 2 - 3 - 5	1 - 2 - 3 - 4
1 - 2 - 1 - 3	2 - 2 - 2 - 3	3 - 2 - 4 - 3
1 - 2 - 4 - 5	3 - 4 - 3 - 3	1 - 3 - 5 - 5
2 - 2 - 4 - 3	2 - 2 - 3 - 3	1 - 2 - 1 - 4
1 - 2 - 1 - 3	1 - 2 - 3 - 5	1 - 3 - 4 - 3

7.1.7 Yapay Sinir Ağı Yapısı

Bu konuda iki ayrı yaklaşım ele alınmıştır. Birinci yaklaşım, örnek setinde sadece seçilen bir öncelik kuralı ile OGA sayısı konfigürasyonu ile elde edilen değerlerin bulunduğu, öncelik kuralı-OGA sayısı kombinasyonuna göre yapay sinir ağı topolojisi oluşturmak ki, tez içinde "Öncelik Kuralına Dayalı Yaklaşım" olarak isimlendirilmiştir. İkinci yaklaşım ise, her iş merkezi için yapay sinir ağı topolojisi oluşturmak ki, bütün donanım-çizelgeleme stratejisi kombinasyonlarına ait değerlerin bir örnek setinde yer aldığı bu yaklaşım tezde "İş Merkezine Dayalı Yaklaşım" olarak isimlendirilmiştir. İki ayrı yaklaşım şeklinde ele alınan yapay sinir ağı yapıları şekil 7.1 ve şekil 7.2'de gösterilmiştir.



Şekil 7.1 Öncelik kuralına dayalı yaklaşım için yapay sinir ağı yapısı



Şekil 7.2 İş merkezine dayalı yaklaşım için yapay sinir ağı yapısı

7.1.8 Eğitim ve Test Setleri

Her bir öncelik kuralı ve taşıyıcı sayısı çifti ile örnek setleri belirlenir. Üç öncelik kuralı ve üç taşıyıcı sayısı alternatifi ele alındığından sistemde $3 \times 3 = 9$ adet örnek seti oluşur. İş Merkezine Dayalı Yaklaşım olarak isimlendirilen ikinci yaklaşımda, bu setler bir araya getirilerek dört ayrı set elde edilir. Eğitim için set büyüklüğü birinci yaklaşım için 75, ikinci yaklaşım için 550, test için set büyüklüğü birinci yaklaşım için 100, ikinci yaklaşım için 750 olarak alınmıştır.

7.1.9 Performans Kriterleri

Çizelgelemede amaçları ifade etmek her zaman kolay değildir. Amaçlar, çok karmaşıktır ve genellikle birbirleriyle bağdaşmazlar. Ancak, çizelgelemede ne derece başarılı olunduğuna karar vermek için bir takım kriterleri tanımlamak gereklidir. Aksi taktirde matematik olarak çizelge hakkında yargıya varmak zorlaşır.

Bazı durumlarda kararlaştırılmış teslim tarihlerine uymak tercih edilir. Aksi taktirde güvenilirlik kaybına uğranılacak ve maliyet söz konusu olacaktır. Bazen ise teslim tarihi çok önemli olmayabilir ve çizelgeleme periyodunun uzunluğu minimum yapılmak istenebilir. Çünkü bütün işler tamamlandıktan sonra tezgahlar başka işler için kullanılabilir. Böylece tezgahların aylak (boş) kalma zamanları minimum olacaktır. Aylak tezgah, aylak sermaye yatırımı demektir. Bunlarla beraber, stok maliyetleri minimum yapılmak istenebilir.

Özetle, amaçlar çok farklı, çeşitli ve birbirleriyle çelişir olabilir. Ama çizelgeleme probleminin etkinliğinin ölçülmesi için tanımlanması gereklidir. Çizelgenin aynı ortam için, donanım konfigürasyonunu değiştirilerek ve/veya farklı öncelik kuralları kullanılarak birkaç kez yapılması halinde, her konfigürasyonun etkinliğini ölçmek için bu tezde kullanılan "Performans kriterleri (etkinlik ölçütleri)" aşağıda ifade edilmiştir.

Akış süresi (F_i), işin atölyede harcadığı zaman olarak tanımlanır. İşin tamamlanma zamanından hazırlık süresi çıkarılmak suretiyle elde edilir.

$$F_i = C_i - r_i$$

Ortalama akış süresi (F), tüm işlerin akış sürelerinin aritmetik ortalamasıdır. n adet iş için;

$$F = \frac{1}{n} \sum_{i=1}^n F_i$$

şeklinde hesaplanır.

Gecikme (L_i), işin tamamlanması istenilen tarihi ile gerçekleşen bitirilme tarihi arasındaki farktır. Ortalama gecikme (L) ise, gecikmeler toplamının toplam iş sayısına bölünmesi ile elde edilir.

$$L_i = C_i - d_i$$

$$L = \frac{1}{n} \sum_{i=1}^n L_i$$

Eğer bir iş erken tamamlanırsa, L_i negatif değere sahip olacaktır. İstenen bitirilme zamanından sonra tamamlanması halinde ise pozitif değere sahip olacaktır. Çalışmalarımızda gecikmeler ölçüsü olarak pozitif gecikmeler göz önüne alınmıştır. Ortalama pozitif gecikme (T);

$$T = \frac{1}{n} \sum_{i=1}^n T_i$$

İşin tamamlanma zamanı, bütün işlemlerinin bittiği zamandır. İşin işlem süresi, bekleme süresi ve hazırlık süresinin toplamı şeklinde ifade edilir. Maksimum tamamlanma zamanı ($C_{\max}$), atölyedeki işler arasında en uzun süre atölyede kalış zamanına sahip olan işin tamamlanma zamanını ifade eder.

$$\text{Maksimum tamamlanma zamanı } (C_{\max}) = \max \{C_i\}$$

Tezgah Kullanım Oranları ($\mu_1, \mu_2, \mu_3, \mu_4$); ortalama tezgah kullanım oranı (U), atölyede bulunan ve işlemlerin gerçekleştirilmesinde kullanılan tezgahların kullanım oranlarının (U_i) ortalamasıdır. Bir tezgahın kullanım oranı, tezgahın çalıştığı süreler toplamının çizelgeleme süresine oranıdır. Çizelgeleme süresinin $C_{\max}$ olduğu göz önüne alınırsa, J . tezgahın kullanım oranı;

$$U_J = \frac{P_J}{C_{\max}}$$

olarak ifade edilir. Burada P_J ; J tezgahından geçen operasyonların toplam işlem süresi olup,

$$P_J = \sum_{i=1}^n P_{ji}$$

şeklinde hesaplanır. Bu durumda, m adet tezgah için ortalama tezgah kullanım

oranı;

$$\bar{U} = \frac{1}{m} \sum_{i=1}^m U_j \text{ olacaktır.}$$

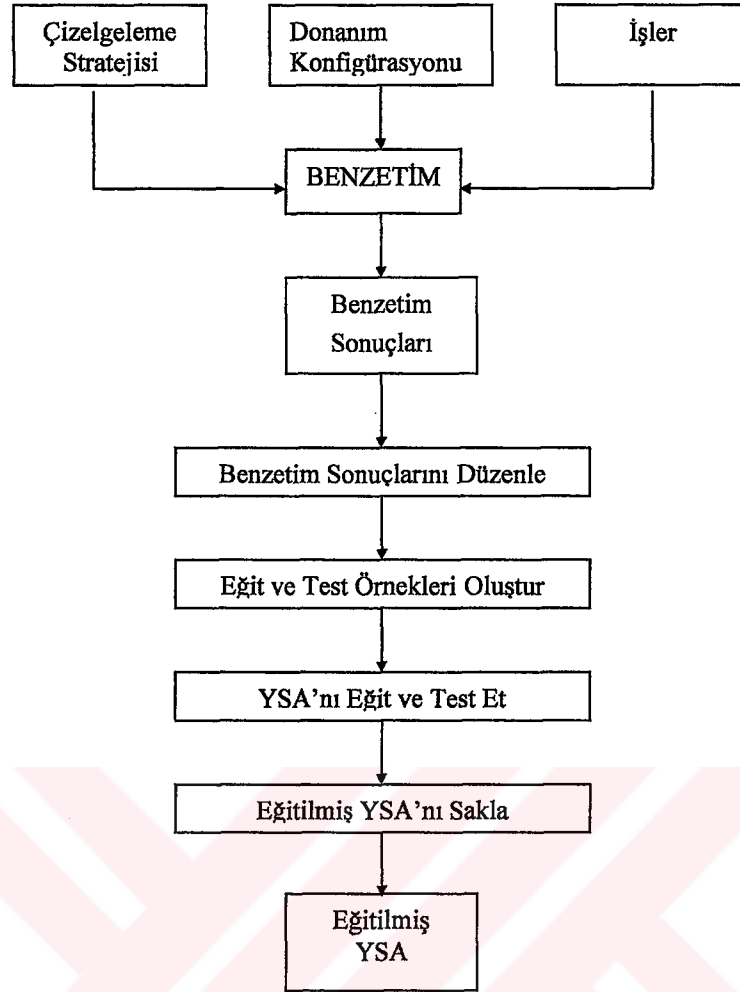
7.2 Uygulama ve Deneysel Çalışma

Çalışmanın birinci kısmını gerçeğe uygun sonuçlar üretebilen, iyi eğitilmiş yapay sinir ağı elde etmek şeklinde özetlemek mümkündür. Eğitilmiş yapay sinir ağı elde etmeden önce öğrenilmesi istenen yapının örneklerinin bulunması gereklidir. Bu örneklerin elde edilmesi için bilgisayar ortamında benzetim tekniğinden faydalanılmıştır.

Benzetim modülünün girdileri; atölye ortamında hangi tip tezgahdan ne kadar bulunduğu gibi bilgileri içeren donanım konfigürasyonu, işlerin atölye ortamına nasıl dahil olduğu, hangi iş merkezlerinde işlem gördüğü, bu işlemlerin ne kadar sürdüğü gibi bilgileri içine alan işlerin yapısı, ayrıca atölyeye dahil olan işlerin hangi sıra ile işlem göreceğini belirleyen çizelgeleme stratejisi gibi bilgilerdir. Benzetim işlemi için SIMAN programlama dili kullanılmıştır. Benzetim işleminin çıktıları sistemi değerlendirme ve karar verme işlemlerinde kullanılmak üzere seçilmiş olan performans kriterlerinin gerçekleşen değerleridir.

Yapay sinir ağını eğitme ve test etme işlemleri için gerekli olan örnekler benzetim çıktılarının düzenlenmesi ile elde edilir. Bu düzenlemede yapay sinir ağının girdileri, istenen performans kriterlerinin gerçekleşen değerleri, çıktıları ise iş merkezlerinde kullanılacak olan tezgah sayılarıdır. Yapay sinir ağları iki ayrı yaklaşımla ele alınmıştır.

Çalışmanın eğitilmiş yapay sinir ağı elde edilerek tamamlanan birinci kısmının algoritma şeması şekil 7.3'de gösterilmiştir.



Şekil 7.3 Yapay sinir ağının oluşturulması ve test edilmesi

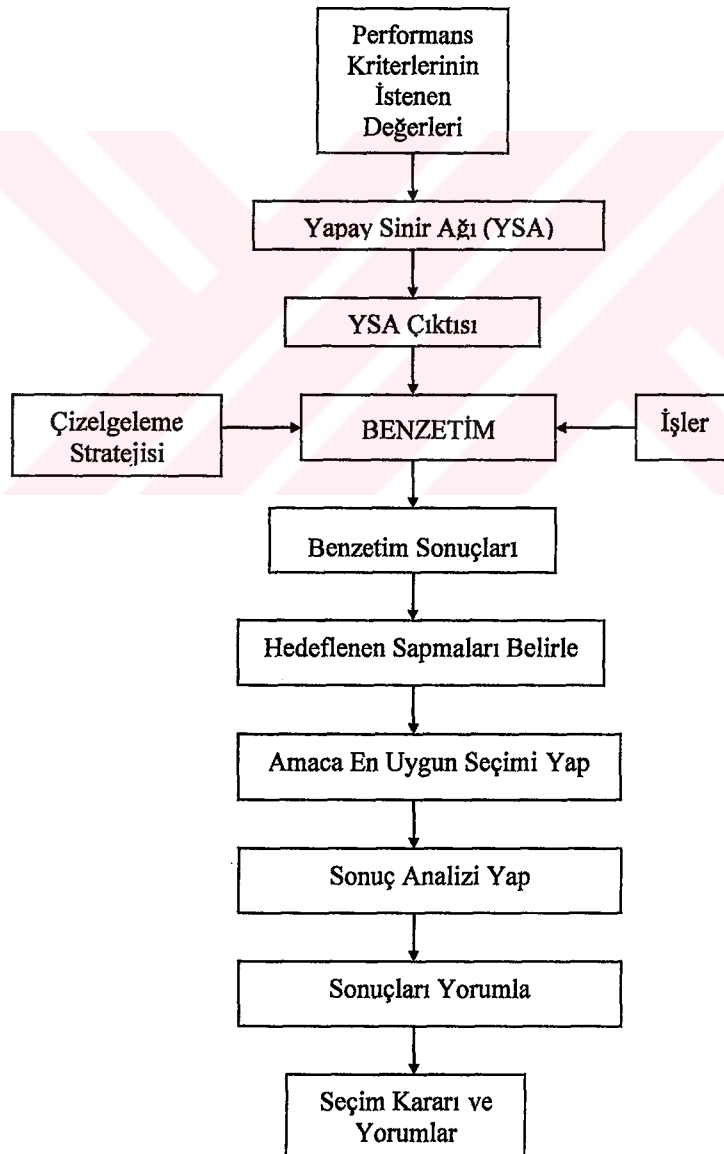
Çalışmanın ikinci kısmı ise, karar verme ve bu kararın geçerli olduğu değerler aralığını bulma ve yorumlama şeklinde özetlenebilir. Performans kriterlerinin belirli bir değerler kümesini sağlayabilecek olan donanım konfigürasyonu ve çizelgeleme stratejisini belirleyebilmek için eğitilmiş olan yapay sinir ağlarından yararlanılır. Performans kriterlerinin istenen değerlerini başarabilmek için gerekli olan en uygun konfigürasyon eğitilmiş yapay sinir ağı tarafından kolaylıkla belirlenebilecektir. Burada etkili ve güvenli bir karar yapısı oluşturabilmek için yapay sinir ağlarının iki ayrı yaklaşım ile ele alındığı unutulmamalıdır.

Yapay sinir ağı tarafından teklif edilen sonuç ya da sonuçlar, farklı alternatifler için benzetim yolu ile tekrar denenerek performans kriterlerinin beklenen değerlerinden sapma miktarları yüzde olarak hesaplanır. Hedeften sapma yüzdesi ile birlikte diğer hesaplanan değerler bir çizelge halinde karar verme işlemi için sunulacaktır. Karar verme işlemi karar vericinin belirlediği bir kritere göre yapılabilir. Örneğin; hedeften sapma yüzdelерinin ortalama mutlak değeri karar verici için bir kriter olabilir. Aynı zamanda birden fazla kriter göz önüne alınabilir. Ortalama gecikmelerin minimum değeri ve ortalama mutlak sapma

değerlerinin minimum değeri gibi.

Son olarak, hem karar verme işleminde yardımcı olabilecek hem de verilen kararların geçerli olduğu alanı belirleyebilecek bir etki analizi ile sistemin farklı durumlar karşısında nasıl bir tepki verebileceği ortaya konulmuştur. Böylece karar verici, sistemi kuş bakışı görebilecek ve etkili kararlar alabilecektir.

Yine bu konular ile alakalı geniş bilgi bu bölümde yapay sinir ağları başlıklı kısımda bulunmaktadır. Çalışmanın benzetim modülünde SIMAN benzetim programlama dili, yapay sinir ağları, karar sistemi ve etki analizi modüllerinde ise Neuro Solutions 4.21 paket programından yararlanılmıştır. Bu kısmın algoritma şeması şekil 7.4’de gösterilmiştir.



Şekil 7.4 Karar verme ve etki alanı analizi

7.2.1 Benzetim Deneyleri

Benzetim deneyleri, atölye çizelgeleme konfigürasyonu ve iş öncelik kuralı kararının, performans kriterleri (akış zamanı, gecikmeler, maksimum tamamlanma zamanı ve iş istasyonu kullanım oranları) üzerindeki etkisini görmek üzere tasarlanmıştır. İş merkezi tezgah sayısı, taşıyıcı sayısı ve iş öncelik kuralı parametrelerinin manipulasyonu yoluyla alternatif konfigürasyonlar elde edilmiştir.

Benzetim sonucunda üretilen raporda dikkate alınan değerler daha önce de anlatıldığı gibi ortalama akış zamanı, ortalama gecikme, maksimum tamamlanma zamanı ve iş merkezi tezgah kullanım oranlarıdır. Seçilen her alternatif konfigürasyon için benzetim yapılarak sonuçlar elde edilmiş ve düzenlenerek örnek setleri oluşturulmuştur. Aşağıdaki çizelge benzetim işlemi sonunda elde edilen bir çıktı örneğini göstermektedir.

Bu örnek çizelgeye göre performans değerleri aşağıdadır :

Çizelge 7.13 Benzetim işlemi sonucundaki örnek çıktı

Ortalama akış zamanı		8259.48		
Ortalama gecikme		4487.59		
Maksimum tamamlanma zamanı		16365.84		
İş Merkezi	1	2	3	4
Tezgah Sayısı	5	2	2	2
Tez. Kul. Oranı	$2.08/5 = .416$	$1.19/2 = .595$	$1.27/2 = .635$	$1.68/2 = .84$

Çizelge 7.14 Benzetim çıktısı örneği

SIMAN Özet Raporu Çalıştırma Numarası: 1						
Proje	: eett					
Date	: 30/ 08/2003					
Çalıştırma süresi	:1965E+05					
Hesaplanan Değerler						
No	Tanımlama	Ortalama	Standart Sapma	Minimum Değer	Maximum Değer	Gözlenen Parça Sayısı
1	PARCA AKIŞ SÜRESİ	7025.58	3513.18	456.30	12229.31	99
2	PARCA AKIŞ SÜRESİ	6645.84	3291	553.86	11669.60	98
3	PARCA AKIŞ SÜRESİ	10965.48	4702	180.15	16543.46	39
4	PARCA AKIŞ SÜRESİ	10667.98	4696	340.17	16662.97	104
5	PARCA AKIŞ SÜRESİ	6125.42	3358	218.77	11611.66	110
6	TOPLAM AKIŞ SÜRE	9165.36	4750	180.15	16662.97	500
7	GİR AR SURE	26.19185	31.00952	1.00000	498.62	499
8	ÇIK AR SURE	32.02824	19.79842	1.00000	92.44877	499
Kesikli Değişen Değerler (Discrete Change Variables)						
Nu	Tanımlama	Ortalama	Standart Sapma	Minimum Değer	Maximum Değer	Zaman Perivodu
1	MCI KUL	2.08	2.21	0.00	5.00	16597.28
2	MC2 KUL	1.19	0.89	0.00	2.00	16597.28
3	MC3 KUL	1.27	1.35	0.00	2.00	16597.28
4	MC4 KUL	1.68	0.49	0.00	2.00	16597.28
5	ROBOT KUL	0.04	0.29	0.00	1.00	16597.28
6	ORT KUL	0.41	0.72	0.00	3.00	16597.28
7	MCI KUY	0.34	0.43	0.00	1.00	16597.28
8	MC2 KUY	1.78	1.85	0.00	6.00	16597.28
9	MC3 KUY	2.45	2.39	0.00	9.00	16597.28
10	MC4 KUY	3.82	0.69	0.00	7.00	16597.28
11	M2 ARA BE K	0.06	0.21	0.00	2.00	16597.28
12	M3 ARA BEK	0.97	1.28	0.00	4.00	16597.28
13	M4 ARA BEK	58.82	29.89	0.00	99.00	16597.28
14	MI BEK	33.78	38.48	0.00	128.00	16597.28
15	M2 BEK	71.48	52.65	0.00	149.00	16597.28
16	M3 BEK	0.00	0.00	0.00	0.00	16597.28
17	ÖRT GEÇ	4487.56	3356.78	0.00	10718.35	16597.28
Savaşlar						
Nu	Tanımlama	Sayılan Parça	Sınır			
1	TOP. PAR. SAYISI	500	500			
2	GEÇ İS SAY	448	Sınırsız			
Run Time : 49 Saniye						

7.2.2 Yapay Sinir Ağı Eğitime ve Test Sonuçları

Yapay sinir ağları, benzetim amacının tersine olarak akış zamanı, gecikmeler, maksimum tamamlanma zamanı ve iş istasyonu kullanım oranlarının atölye çizelgeleme konfigürasyonu ve iş öncelik kuralı kararı üzerindeki etkisini görmek üzere tasarlanmıştır.

Benzetim sonuçlarından elde edilen değerlerle eğitime ve test örnekleri oluşturularak GYA ile yapay sinir ağları eğitilmiştir. Yapay sinir ağlarının eğitilmesinde iyi sonuçlar elde edilmiştir.

Yapay sinir ağlarının eğitime ve test işlemleri sonuçları aşağıda bu başlıklı konu içinde çizelgeler halinde sunulmuştur.

Bu konuda iki ayrı yaklaşım ele alınmıştır. "Öncelik Kurallı Temel Yaklaşım" olarak ele alınan birinci yaklaşımda, örnek setinde sadece seçilen bir öncelik kuralı ile OGA (Otomatik Güdümlü Araç) sayısı konfigürasyonu ile elde edilen değerlerin bulunduğu, öncelik kuralı-OGA sayısı kombinasyonuna göre yapay sinir ağı topolojisi oluşturulmuştur. "İş Merkezli Temel Yaklaşım" olarak isimlendirilen ikinci yaklaşımda ise, her iş merkezi için yapay sinir ağı topolojisi oluşturulmuştur. Bütün donanım-çizelgeleme stratejisi kombinasyonlarına ait değerlerin bir örnek setinde yer aldığı bu yaklaşımın, birinci yaklaşıma göre daha iyi sonuçlar verdiği görülmüştür. Yapay sinir ağlarının ayrı ayrı oluşturulması ile, her ağın ayrı ayrı eğitilerek hatanın daha da küçültülmesi amaçlanmıştır.

Her ağın girdi değerleri performans ölçütleri, çıktı değeri ise iş merkezi tezgah sayılarıdır. Eğitilmiş olan her ağ üzerinde aynı girdi değerleri (istenilen performans ölçütleri) girilerek, bu performans ölçütlerini gerçekleyebilecek olan iş merkezleri tezgah sayıları elde edilmiştir.

Elde edilen bu tezgah sayıları, OGA Sayısı-Öncelik Kuralı ikilisinin kombinasyonuna göre (1 OGA- EETT, 2OGA-EETT, 3OGA-EETT, 1 OGA- EKİZ, 2OGA-EKİZ, 3OGA-EKİZ, 1OGA-İĞİĞ, 2OGA-İĞİĞ, 3OGA-İĞİĞ), ayrı ayrı benzetim yapılarak istenilen performans ölçütlerinden minimum mutlak sapma gösteren donanım ve çizelgeleme stratejisi (Tezgah Sayısı-OGA Sayısı-Öncelik Kuralı) belirlenmiştir.

7.2.2.1 Öncelik Kurallı Temel YSA Yaklaşımı Eğitime ve Test Sonuçları

Güvenli bir karar verme sistemi için yapay sinir ağının iki ayrı yaklaşım ile ele alındığı daha önce ifade edilmişti. Bu şekilde ayrıca, farklı yorumlar yapılabilecek sonuçlar da elde edilebilecektir. Örneğin teklif edilen diğer sistem konfigürasyonları ile birlikte öncelik kuralının "eett", OGA sayısının "bir" veya "iki" veya "üç" olduğu durumların sistem üzerindeki etkisini incelemek gibi. Aşağıdaki çizelgeler bu yapay sinir ağı yaklaşımı için oluşturulan örnek setleri ile elde edilen, kendi kendine öğrenme ve seçilen test örnekleri ile test edilme işlemlerinin sonuçlarını gösterir. Bu sonuçlardan da görülebileceği gibi öğrenme başarı oranının %100'e çok yakın gerçekleştiği söylenebilir.

Öncelik kuralına dayalı olan birinci yaklaşım için öncelik kuralı ve OGA sayısı kombinasyonlarından oluşan dokuz adet yapay sinir ağı tasarlanmıştır. Bu yapay sinir ağlarının kısaltılmış olarak kullanılan isimleri ve açıklamaları ise şöyledir:

EETT1: Öncelik kuralı " En Erken Teslim Tarihi ", OGA sayısı "bir"

EETT2: Öncelik kuralı " En Erken Teslim Tarihi ", OGA sayısı "iki"

EETT3: Öncelik kuralı " En Erken Teslim Tarihi ", OGA sayısı "üç"

EKİZ1 : Öncelik kuralı " En Kısa İşlem Zamanı ", OGA sayısı "bir"

EKİZ2 : Öncelik kuralı " En Kısa İşlem Zamanı ", OGA sayısı "iki"

EKİZ3 : Öncelik kuralı " En Kısa İşlem Zamanı ", OGA sayısı "üç"

İGİG1 : Öncelik kuralı " İlk Gelen İlk İşlem Görür" , OGA sayısı "bir"

İGİG2 : Öncelik kuralı " İlk Gelen İlk İşlem Görür" , OGA sayısı "iki"

İGİG3 : Öncelik kuralı " İlk Gelen İlk İşlem Görür" , OGA sayısı "üç"

Bu dokuz yapay sinir ağının her biri ayrı ya da aynı sonucu üretebilir. Her yapay sinir ağının sonucu, istenen performans kriterlerini gerçeklemek için gerekli olan tezgah sayılarının bütün iş merkezleri için belirlenmesidir. Her bir eğitime seti için, deneme-yanılma ile belirlenen farklı iterasyon sayısı ve öğrenme oranları uygulanmış ve test edilmiştir. Sonuçlar aşağıdaki çizelgelerde sunulmuştur (Çizelge 7.15 – 7.23). Çizelgelerde sol taraftaki çizelge eğitim, sağ taraftaki çizelge test sonuçlarını göstermektedir. Bu çizelgelerde yer alan ifadeler kısaca aşağıda açıklandığı gibidir.

- **Girdi sayısı:** Bütün yapay sinir ağları için girdi sayısı "yedi" adettir. Bunlar seçilen performans kriterlerinin benzetim sonucunda elde edilen ve gerçekleşen değerlerini ifade eder. Bu kriterler; "Ortalama Akış Süresi", "Ortalama Gecikme Süresi", "Maksimum Tamamlanma Süresi", "Birinci İş Merkezi Kullanım Oranı", "İkinci İş Merkezi Kullanım Oranı", "Üçüncü İş Merkezi Kullanım Oranı" ve "Dördüncü İş Merkezi Kullanım Oranı" dır.
- **Çıktı sayısı:** Çıktı katmanında bulunan yapay sinir hücresi sayısını ifade eder. Dolayısıyla birinci yaklaşım için bu sayı "dört" tür.Yani, iş merkezlerinde kullanılması gerekli olan tezgah sayıları yapay sinir ağının çıktısı olarak elde edilir.
- **Gizli (hidden) birim sayısı:** Yapay sinir ağının ara (gizli) katmanında kullanılan düğüm sayısını ifade eder. Birinci yaklaşım için bu sayı deneme-yanılma yolu ile 35 olarak belirlenmiştir.

- **Öğrenme oranı (λ) ve Momentum katsayısı (α):** Öğrenme, her ikisi de 0 ile 1 değeri arasında değer alan öğrenme oranı ve momentum katsayısı ile kontrol edilir. Öğrenme işleminde son derece önemli olan bu katsayılar her ağ için ayrı ayrı olarak, yine deneme-yanılma yolu ile belirlenmiştir.
- **İterasyon sayısı (N):** Epoch, örnek setinin bir defa yapay sinir ağına gösterilmesidir. İterasyon ise her örneğin bir defa yapay sinir ağına gösterilmesidir. İterasyon sayıları genel olarak 200 epoch dan sonra öğrenme işleminde istenen seviye sağlanıncaya kadar kademeli olarak artırılarak elde edilen sayıları ifade eder.
- **İterasyon süresi:** Öğrenme işlemi sırasında geçen sürenin dakika cinsinden ifadesidir. İterasyon süresi için, bir karar verme işleminde uzun sayılmayacak sonuçlar elde edilmiştir.
- **Toplam örnek sayısı ve toplam grup sayısı:** Aslında grup sayısı ile ifade edilen yapay sinir ağı açısından örnek sayısıdır. Ancak, birinci yaklaşımda her örnekte bulunan dört iş merkezi sonucunun hepsi de ayrı ayrı önemli olduğundan başarı oranının bu açıdan hesaplanabilmesi için, her örnek grup ismi altında incelenmiştir. Böylece 75 grup (yani örnek seti büyüklüğü), her grupta 4 çıktı bulunduğundan 300 örnek olarak değerlendirilmiştir. Bu şekilde daha hassas sonuçlar elde edilmeye çalışılmıştır.
- **Hatalı örnek sayısı ve hatalı grup sayısı:** Hatalı örnek sayısı yapay sinir ağının 4 çıktısından kaç tanesinin gerçek değerden farklı olduğunun toplam sayısını, hatalı grup sayısı ise, bu şekildeki çıktıların kaç adet olduğunu ifade eder.
- **Hatalı örnek oranı ve Hatalı grup sayısı oranı:** Gerçek değerlerden farklı olan hatalı örnek veya grup sayılarının toplamı örnek veya grup sayılarına bölünmesi ile elde edilir.
- **Sapma toplamı:** Yapay sinir ağının çıktıları tam değer olarak ifade edildiğinden hatalı olmayan örnekler dahil olmak üzere bütün örnekler için çok küçük de olsa bir miktar sapma olabilecektir. Sonuç üzerinde büyük bir etkisi olmamasına rağmen ağın başarı oranı hakkında fikir vermesi açısından değerlendirmeye dahil edilmiştir. Çizelgelerden de görüleceği üzere çok küçük sapma değerleri elde edilmiştir.

- **Ortalama sapma ve Ortalama grup sapma oranı:** Hesaplanan sapma miktarının yukarıda açıklanan toplam örnek sayısı ve toplam grup sayısına bölünmesi ile elde edilir.

Çizelge 7.15 EETT1 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	499912
İterasyon süresi	5.54 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	300
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.00
Sapma toplamı	2.12
Toplam örnek sayısı	300
Ortalama sapma	0.007
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.028

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	499912
İterasyon süresi	5.54 dk
Hatalı örnek sayısı	4
Toplam örnek sayısı	400
Hatalı örnek oranı	0.00
Hatalı grup sayısı	4
Toplam grup sayısı	100
Hatalı grup sayısı oranı	0.04
Sapma toplamı	4.32
Toplam örnek sayısı	400
Ortalama sapma	0.011
Toplam grup sayısı	100
Ortalama grup sapma oranı	0.040

Çizelge 7.16 EETT2 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	449930
İterasyon süresi	4.59 dk
Hatalı örnek sayısı	2
Toplam örnek sayısı	300
Hatalı örnek oranı	0.007
Hatalı grup sayısı	2
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.03
Sapma toplamı	2.45
Toplam örnek sayısı	300
Ortalama sapma	0.008
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.033

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	449930
İterasyon süresi	4.59 dk
Hatalı örnek sayısı	6
Toplam örnek sayısı	400
Hatalı örnek oranı	0.02
Hatalı grup sayısı	6
Toplam grup sayısı	100
Hatalı grup sayısı oranı	0.06
Sapma toplamı	4.12
Toplam örnek sayısı	400
Ortalama sapma	0.010
Toplam grup sayısı	100
Ortalama grup sapma oranı	0.041

Çizelge 7.17 EETT3 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	449925
İterasyon süresi	5.21 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	300
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.00
Sapma toplamı	1.85
Toplam örnek sayısı	300
Ortalama sapma	0.006
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.013

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	449925
İterasyon süresi	5.21 dk
Hatalı örnek sayısı	2
Toplam örnek sayısı	400
Hatalı örnek oranı	0.005
Hatalı grup sayısı	2
Toplam grup sayısı	100
Hatalı grup sayısı oranı	0.02
Sapma toplamı	3.56
Toplam örnek sayısı	400
Ortalama sapma	0.009
Toplam grup sayısı	100
Ortalama grup sapma oranı	0.036

Çizelge 7.18 EKİZ1 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	999875
İterasyon süresi	12.51 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	300
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.00
Sapma toplamı	1.65
Toplam örnek sayısı	300
Ortalama sapma	0.006
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.022

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	999875
İterasyon süresi	12.51 dk
Hatalı örnek sayısı	3
Toplam örnek sayısı	400
Hatalı örnek oranı	0.008
Hatalı grup sayısı	3
Toplam grup sayısı	100
Hatalı grup sayısı oranı	0.03
Sapma toplamı	2.23
Toplam örnek sayısı	400
Ortalama sapma	0.006
Toplam grup sayısı	100
Ortalama grup sapma oranı	0.0022

Çizelge 7.19 EKİZ2 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	1249800
İterasyon süresi	14.05 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	300
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.00
Sapma toplamı	1.08
Toplam örnek sayısı	300
Ortalama sapma	0.004
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.014

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	1.249800
İterasyon süresi	14.05 dk
Hatalı örnek sayısı	3
Toplam örnek sayısı	400
Hatalı örnek oranı	0.008
Hatalı grup sayısı	2
Toplam grup sayısı	100
Hatalı grup sayısı oranı	0.02
Sapma toplamı	2.45
Toplam örnek sayısı	400
Ortalama sapma	0.006
Toplam grup sayısı	100
Ortalama grup sapma oranı	0.025

Çizelge 7.20 EKİZ3 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	999850
İterasyon süresi	12.25 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	300
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.00
Sapma toplamı	1.40
Toplam örnek sayısı	300
Ortalama sapma	0.005
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.019

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	999850
İterasyon süresi	12.25 dk
Hatalı örnek sayısı	2
Toplam örnek sayısı	400
Hatalı örnek oranı	0.005
Hatalı grup sayısı	2
Toplam grup sayısı	100
Hatalı grup sayısı oranı	0.02
Sapma toplamı	2.07
Toplam örnek sayısı	400
Ortalama sapma	0.005
Toplam grup sayısı	100
Ortalama grup sapma oranı	0.020

Çizelge 7.21 İGİG1 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	999845
İterasyon süresi	12.10
Hatalı örnek sayısı	0
Toplam örnek sayısı	300
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.00
Sapma toplamı	1.25
Toplam örnek sayısı	300
Ortalama sapma	0.004
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.017

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	999845
İterasyon süresi	12.10
Hatalı örnek sayısı	3
Toplam örnek sayısı	400
Hatalı örnek oranı	0.008
Hatalı grup sayısı	2
Toplam grup sayısı	100
Hatalı grup sayısı oranı	0.02
Sapma toplamı	1.95
Toplam örnek sayısı	400
Ortalama sapma	0.005
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.026

Çizelge 7.22 İGİG2 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	699925
İterasyon süresi	7.75 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	300
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.00
Sapma toplamı	1.35
Toplam örnek sayısı	300
Ortalama sapma	0.005
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.018

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	699925
İterasyon süresi	7.75 dk
Hatalı örnek sayısı	2
Toplam örnek sayısı	400
Hatalı örnek oranı	0.005
Hatalı grup sayısı	1
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.013
Sapma toplamı	2.15
Toplam örnek sayısı	400
Ortalama sapma	0.005
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.029

Çizelge 7.23 İGİG3 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	949955
İterasyon süresi	10.55 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	300
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.00
Sapma toplamı	1.95
Toplam örnek sayısı	300
Ortalama sapma	0.007
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.026

Girdi sayısı	7
Çıktı sayısı	4
Gizli birim sayısı	35
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	949955
İterasyon süresi	10.55 dk
Hatalı örnek sayısı	3
Toplam örnek sayısı	400
Hatalı örnek oranı	0.008
Hatalı grup sayısı	3
Toplam grup sayısı	75
Hatalı grup sayısı oranı	0.04
Sapma toplamı	2.78
Toplam örnek sayısı	400
Ortalama sapma	0.007
Toplam grup sayısı	75
Ortalama grup sapma oranı	0.037

7.2.2.2 İş Merkezine Dayalı Yaklaşım İçin YSA Eğitim ve Test Sonuçları

Bir iş merkezinde bulunan tezgah sayılarının sistemin bütün faktörlerinin bir arada temsil edildiği bir yapay sinir ağı tarafından belirlenmesi, bütün alternatifleri göz önüne alarak bir sonuç üretmesi ile, sonucun doğruluğunun artmasını sağlaması yanında, iş merkezine dayalı çıkarımlar yapılabilmesine imkan sağlayacaktır. Bu şekilde güvenli ve farklı yorumlar elde edilebilecek bir karar verme sistemi oluşturma amaçlanmıştır. Ayrıca ilk yaklaşımda olduğu gibi sahasında uzman olan bir kişinin büyük miktarda yardımına ihtiyaç duyularak bir sonuç elde etmekten uzak bir yaklaşımdır.

Aşağıdaki çizelgeler bu yaklaşım için oluşturulan örnek setleri ile elde edilen kendi kendini öğrenme ve seçilen test örnekleri ile test edilme işlemlerinin sonuçlarını gösterir. Bu sonuçlardan öğrenmede %100 lük bir başarı oranının gerçekleştiği görülmektedir. Böylece, birinci yaklaşıma göre daha iyi bir sonuç elde edilmiştir.

İş merkezine dayalı olan bu ikinci yaklaşımda dört adet yapay sinir ağı oluşturulmuştur. Bu yaklaşımda iş merkezlerinde bulunan tezgah sayıları ayrı ayrı belirlendiğinden, ayrı ayrı oluşturulan yapay sinir ağlarının kısaltılmış olarak kullanılan isimleri temsil ettiği iş merkezini ifade eder.

M1: Birinci İş Merkezi

M2: İkinci İş Merkezi,

M3: Üçüncü İş Merkezi

M4: Dördüncü İş Merkezi.

Bu dört yapay sinir ağının her biri farklı ya da aynı sonucu üretebilir. Sonuç, istenen performans kriterlerini gerçeklemek için gerekli olan tezgah sayılarının bütün iş merkezleri için ayrı ayrı belirlenmesidir. Ancak bu dört ayrı sonuç, tek bir sonuç gibi değerlendirilir. Yani bu yaklaşım yalnız bir sonuç üretir, o da, istenen performans kriterlerini sağlayabilmek için gerekli olan, her iş merkezinde kullanılacak tezgah sayılarıdır.

Her bir eğitime seti için, deneme-yanılma ile belirlenen farklı iterasyon sayısı ve öğrenme oranları uygulanarak eğitilip test edilen sonuçlar aşağıdaki çizelgelerde sunulmuştur. Bu çizelgelerde yer alan ifadeler daha önce açıklandığı gibidir. Ancak burada çıktı sayısı "bir" adet olduğundan örnek sayısı ile grup sayısı aynıdır. Böyle olmakla birlikte diğer çizelgeler ile benzerlik bozulmaması için çizelge yapısı değiştirilmemiştir (Çizelge 7.24 – 7.27).

Çizelge 7.24 M1 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	1
Gizli birim sayısı	10
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	449950
İterasyon süresi	3.10 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	550
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	550
Hatalı grup sayısı oranı	0.00
Sapma toplamı	3.78
Toplam örnek sayısı	550
Ortalama sapma	0.007
Toplam grup sayısı	550
Ortalama grup sapma oranı	0.007

Girdi sayısı	7
Çıktı sayısı	1
Gizli birim sayısı	10
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	449950
İterasyon süresi	3.10 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	750
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	750
Hatalı grup sayısı oranı	0.00
Sapma toplamı	4.23
Toplam örnek sayısı	750
Ortalama sapma	0.006
Toplam grup sayısı	750
Ortalama grup sapma oranı	0.006

Çizelge 7.25 M2 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	1
Gizli birim sayısı	10
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	249950
İterasyon süresi	0.95 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	550
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	550
Hatalı grup sayısı oranı	0.00
Sapma toplamı	3.55
Toplam örnek sayısı	550
Ortalama sapma	0.006
Toplam grup sayısı	550
Ortalama grup sapma oranı	0.006

Girdi sayısı	7
Çıktı sayısı	1
Gizli birim sayısı	10
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	249950
İterasyon süresi	0.95 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	750
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	750
Hatalı grup sayısı oranı	0.00
Sapma toplamı	4.95
Toplam örnek sayısı	750
Ortalama sapma	0.007
Toplam grup sayısı	750
Ortalama grup sapma oranı	0.007

Çizelge 7.26 M3 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	1
Gizli birim sayısı	15
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	449550
İterasyon süresi	2.23 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	550
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	550
Hatalı grup sayısı oranı	0.00
Sapma toplamı	4.23
Toplam örnek sayısı	550
Ortalama sapma	0.008
Toplam grup sayısı	550
Ortalama grup sapma oranı	0.008

Girdi sayısı	7
Çıktı sayısı	1
Gizli birim sayısı	15
Öğrenme oranı	0.30
Momentum katsayısı	0.70
İterasyon sayısı	449550
İterasyon süresi	2.23 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	750
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	750
Hatalı grup sayısı oranı	0.00
Sapma toplamı	4.98
Toplam örnek sayısı	750
Ortalama sapma	0.007
Toplam grup sayısı	750
Ortalama grup sapma oranı	0.007

değil, sadece istenen belli bir performans kriterinin beklenen değerinden mutlak minimum sapmaya sahip olan seçenek tercih edilmesi durumunda kullanılabilir. Örneğin ikinci yaklaşımın birinci örneğinde minimum ortalama mutlak sapma kriterine göre tercih edilen seçenek "EETT3" iken, yalnız ortalama akış zamanı kriteri göz önüne alındığında seçim kararı mutlak olarak minimum sapma gösteren "EKİZ2" olmaktadır.

MaxFTC satırı, ilgili sütundaki F, T ve C_{max} sapma değerlerinin toplamını gösterir. |min| sütununda ise bu satırdaki değerler arasında maksimum değere sahip olan değer (mutlak değil) bulunmaktadır. Çünkü genellikle bu değerlerin arzulanan değerlerden düşük olması istenir. Çizelge değerleri istenen değerlerden gerçekleşen değerlerin çıkarılması suretiyle elde edildiğinden, maksimum sapma değerine sahip seçeneğin bu kriterlere ait gerçekleşen değerlerinin, diğer seçeneklere göre daha düşük (minimum) olduğunu gösterecektir. Bu kriter de seçim kararı verme esnasında kullanılabilir.

Min4mu satırı, ilgili sütundaki mul, mu2, mu3 ve mu4 sapma değerleri toplamını gösterir. |min| sütununda ise bu satırdaki değerler arasında minimum değere sahip olan değer (mutlak değil) bulunmaktadır. Kullanım oranını maksimize etmek amaçlandığında bu kriter seçim kararı için kullanılabilir. Çizelge değerleri istenen değerlerden gerçekleşen değerlerin çıkarılması suretiyle elde edildiğinden, bu satırdaki minimum değer, ilgili sütunun en yüksek gerçekleşen kullanım oranlarına sahip olduğunu gösterir.

7.2.3.1 Öncelik Kuralına Dayalı Yaklaşım İçin Deneysel Sonuç ve Karar Çizelgeleri

Aşağıdaki çizelgelerde performans ölçülerinin istenen değerlerini sağlamak üzere yapay sinir ağının ürettiği iş merkezleri tezgah sayıları sonuçları kullanılarak yapılan benzetim çalışması sonucunda performans ölçülerinin gerçekleşen değerleri ve aralarındaki fark yüzde olarak verilmiştir. Bu değerlerin düzenlenmesi ile karar çizelgeleri oluşturulmuştur. Karar verme işleminde kullanılacak sonuçlar için ayrıntılı bir örnek verilmiş (Çizelge 7.28), diğer örnekler için sadece hata analizi ile karar çizelgesi gösterilmiştir. Öncelik kuralına dayalı yaklaşım için deneysel sonuç ve karar çizelgeleri aşağıda verilmiştir (Çizelge 7.29 – 7.34)

Çizelge 7.28 Örnek değerlendirme çizelgesi

Kural: EETT OGA: 1 Tezgah: 3 2 2 3			
Değerler	İstenen	Gerçekleşen	Hata(%)
F	5000	6378	-0.28
T	500	2825	-4.65
C _{max}	15000	12726	0.15
Mu1	0.90	0.91	-0.01
Mu2	0.90	0.84	0.07
Mu3	0.90	0.63	0.03
Mu4	0.90	0.82	0.09
Kural: EETT OGA: 2 Tezgah: 3 2 2 3			
F	5000	6289	-0.26
T	500	2793	-4.59
C _{max}	15000	13042	0.13
Mu1	0.90	0.94	-0.04
Mu2	0.90	0.78	0.13
Mu3	0.90	0.89	0.01
Mu4	0.90	0.77	0.14
Kural: EETT OGA: 3 Tezgah: 3 2 2 3			
F	5000	6387	-0.28
T	500	2802	-4.61
C _{max}	15000	13343	0.11
Mu1	0.90	0.91	-0.01
Mu2	0.90	0.81	0.1
Mu3	0.90	0.89	0.01
Mu4	0.90	0.80	0.11
Kural: EKİZ OGA: 1 Tezgah: 3 2 2 2			
F	5000	7942	-0.59
T	500	4393	-7.79
C _{max}	15000	16032	-0.07
Mu1	0.90	0.76	0.16
Mu2	0.90	0.69	0.23
Mu3	0.90	0.73	0.19
Mu4	0.90	0.92	-0.02
Kural: EKİZ OGA: 2 Tezgah: 3 2 2 2			
F	5000	7956	-0.59
T	500	4562	-8.12
C _{max}	15000	16195	-0.08
Mu1	0.90	0.77	0.15
Mu2	0.90	0.70	0.23
Mu3	0.90	0.77	0.14
Mu4	0.90	0.92	-0.02

Kural: EKİZ OGA: 3 Tezgah: 3 2 2 3			
F	5000	6391	-0.28
T	500	3367	-5.73
C _{max}	15000	15896	0.06
mu1	0.90	0.88	0.02
mu2	0.90	0.79	0.12
mu3	0.90	0.86	0.05
mu4	0.90	0.77	0.15
Kural: İGİG OGA: 1 Tezgah: 3 1 2 2			
F	5000	9294	-0.86
T	500	7283	-13.56
C _{max}	15000	21289	-0.42
mu1	0.90	0.56	0.38
mu2	0.90	0.93	-0.03
mu3	0.90	0.57	0.37
mu4	0.90	0.67	0.26
Kural: İGİG OGA: 2 Tezgah: 3 1 2 2			
F	5000	9339	-0.87
T	500	7633	-14.26
C _{max}	15000	22194	-0.48
mu1	0.90	0.50	0.44
mu2	0.90	0.97	-0.08
mu3	0.90	0.61	0.39
mu4	0.90	0.68	0.25
Kural: İGİG OGA: 3 Tezgah: 3 2 2 2			
F	5000	8038	-0.61
T	500	4634	-8.26
C _{max}	15000	16178	-0.08
mu1	0.90	0.71	0.21
mu2	0.90	0.73	0.19
mu3	0.90	0.72	0.20
mu4	0.90	0.94	-0.04

Çizelge 7.27 M4 YSA eğitim ve test sonuçları

Girdi sayısı	7
Çıktı sayısı	1
Gizli birim sayısı	15
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	449550
İterasyon süresi	2.32 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	550
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	550
Hatalı grup sayısı oranı	0.00
Sapma toplamı	3.18
Toplam örnek sayısı	550
Ortalama sapma	0.006
Toplam grup sayısı	550
Ortalama grup sapma oranı	0.006

Girdi sayısı	7
Çıktı sayısı	1
Gizli birim sayısı	15
Öğrenme oranı	0.25
Momentum katsayısı	0.75
İterasyon sayısı	449550
İterasyon süresi	2.32 dk
Hatalı örnek sayısı	0
Toplam örnek sayısı	750
Hatalı örnek oranı	0.00
Hatalı grup sayısı	0
Toplam grup sayısı	750
Hatalı grup sayısı oranı	0.00
Sapma toplamı	4.16
Toplam örnek sayısı	750
Ortalama sapma	0.006
Toplam grup sayısı	750
Ortalama grup sapma oranı	0.006

7.2.3 Deneysel Sonuçlar ve Karar Çizelgeleri

Eğitilmiş olan her ağ üzerine aynı girdi değerleri (istenilen performans ölçütleri) girilerek, bu performans ölçütlerini sağlayabilecek olan iş merkezleri tezgah sayıları elde edilir. Bu iş merkezleri tezgah sayılarına göre OGA Sayısı-Öncelik Kuralı ikilisinin kombinasyonu ile yapılan benzetim sonucunda istenilen performans ölçütlerinden sapma miktarları belirlenebilecektir.

Aşağıdaki çizelgelerde istenen performans kriteri değerlerine göre belirlenen konfigürasyon ve bu konfigürasyona göre benzetim işlemi sonucunda beklenen değerlerden sapmalar yüzde olarak gösterilmektedir. Karar çizelgelerinde belirtilen hata yüzdesi şu şekilde hesaplanmıştır

$$\text{Hata Yüzdesi} = (\text{Beklenen Değer} - \text{Gerçekleşen Değer}) / \text{Beklenen Değer}$$

Karar verilirken genellikle ortalama minimum mutlak sapma dikkate alındığından seçim kararı olabilecek olan minimum mutlak sapma bu çalışmada da dikkate alınmıştır. Bununla birlikte farklı değerlendirme kriterleri ile kararlar alma konusunda yardımcı olabilecek kriterlerin değerleri de çizelgeye dahil edilmiştir.

[Min] ile gösterilen sütun, ilgili performans kriterinin sapma değerlerinden o satırdaki mutlak minimum olan değeri gösterir. Performans kriterlerinin mutlak sapma ortalamaları ile

her öncelik kuralı-OGA sayısı ikilisi için bütün iş merkezlerinde kullanılması gerekli olan tezgah sayıları teklif edilmiştir. Bu tezgah sayıları aynı ya da farklı olabilir. Elde edilen bu değerlerin benzetimi ile performans kriterlerinin gerçekleşen değerleri ve bununla birlikte hata oranları belirlenir.

Çizelge 7.29 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 1)

İstenen Performans Değerleri										
F = 5000 T = 500 C _{max} = 15000 mu1=0.90 mu2= 0.90 mu3= 0.90 mu4= 0.90										
Kural	OGA			Tezgah Sayısı						
EETT	1			3	2	2	3			
EETT	2			3	2	2	3			
EETT	3			3	2	2	3			
EKİZ	1			3	2	2	2			
EKİZ	2			3	2	2	2			
EKİZ	3			3	2	2	3			
İĞİG	1			3	1	2	2			
İĞİG	2			3	1	2	2			
İĞİG	3			3	2	2	2			
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	-0.28	-0.26	-0.28	-0.59	-0.59	-0.28	-0.86	-0.87	-0.61	0.26
T	-4.65	-4.59	-4.61	-7.79	-8.12	-5.73	-13.56	-14.26	-8.26	4.59
C _{max}	0.15	0.13	0.11	-0.07	-0.08	0.06	-0.42	-0.48	-0.08	0.06
mu1	-0.01	-0.04	-0.01	0.16	0.15	0.02	0.38	0.44	0.21	0.01
mu2	0.07	0.13	0.10	0.23	0.23	0.12	-0.03	-0.08	0.19	0.03
mu3	0.03	0.01	0.01	0.19	0.14	0.05	0.37	0.39	0.20	0.01
mu4	0.09	0.14	0.11	-0.02	-0.02	0.15	0.26	0.25	-0.04	0.02
Ort.	0.75	0.76	0.75	1.29	1.33	0.92	2.27	2.40	1.37	0.75
MaxFTC	-4.78	-4.72	-4.78	-8.45	-8.79	-5.95	-14.84	-15.61	-8.95	-4.72
Min4mu	0.18	0.24	0.21	0.56	0.50	0.34	0.98	1.00	0.56	0.18

Yukarıdaki çizelge, performans kriterlerinin istenen değerlerine en yakın değerleri elde edebilmek için oluşturulan karar verme çizelgesidir. Seçilen bir karar verme kriterine göre, bu çizelge kullanılarak karar verme işlemi gerçekleştirilebilir.

Yukarıdaki çizelgede, ortalama mutlak sapma kriterine göre seçim kararı için iki alternatif vardır. Birincisi için iş merkezi tezgah sayıları: 3, 2, 2, 3, öncelik kuralı: EETT ve OGA sayısı:1'dir. İkincisi için iş merkezi tezgah sayıları: 3, 2, 2, 3, öncelik kuralı: EETT ve OGA, sayısı:3'dür. Tabi burada OGA sayısı 1 adet olan birinci seçenek aynı sonucu daha az maliyet ile sağlayacağından tercih edilmesi daha uygun olabilecek bir seçenektir. Yalnız ortalama mutlak minimum sapma ile karar verilmeyip başka kriterler de kullanılabilir. Ne ifade ettiği daha önce açıklanan bu kriterlerden bazılarına göre seçim kararı aşağıda gösterilmiştir.

MaxFTC kriterine göre seçim kararı: İş merkezi tezgah sayıları: 3, 2, 2, 3, öncelik kuralı: EETT ve OGA sayısı:2 dir.

Min4mu kriterine göre seçim kararı: İş merkezi tezgah sayıları: 3, 2, 2, 3,

Öncelik kuralı: EETT ve OGA sayısı: 1' dir.

Yine bu kriterlerden başka kriterler de kullanılabilir. Örneğin; istenen ortalama akış zamanından minimum mutlak sapma gösteren seçenek için, iş merkezi tezgah sayıları: 3, 2, 2, 3, öncelik kuralı: EETT ve OGA sayısı : 2' dir.

Bu şekilde alternatif seçim kararları üretilebilir. Hatta hepsinin bir arada bulunduğu bir çizelge düzenlenerek amaca en uygun seçeneğin belirlenmesi yolu tercih edilebilir. Ancak bu çalışmada seçim kararı verilirken Minimum ortalama mutlak sapma (ORT), MaxFTC ve Min4mu kriterleri göz önüne alınmıştır. Ayrıca bu kriterlerden ayrı olarak karar verilme sırasında finansal analiz yapılması bazı alternatif sonuçlar hakkında daha iyi kararlar alınmasını sağlayacaktır.

Çizelge 7.30 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 2)

İstenen Performans Değerleri										
F = 5000 T = 500 C _{max} = 15000 mu1=0.80 mu2= 0.80 mu3= 0.80 mu4= 0.80										
Kural	OGA			Tezgah Sayısı						
EETT	1			4	2	2	3			
EETT	2			4	2	2	3			
EETT	3			3	2	2	3			
EKİZ	1			3	2	2	3			
EKİZ	2			3	2	2	3			
EKİZ	3			3	2	2	3			
İĞİĞ	1			4	2	2	3			
İĞİĞ	2			3	2	2	3			
İĞİĞ	3			4	2	2	3			
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİĞ1	İĞİĞ2	İĞİĞ3	Min
F	-0.24	-0.21	-0.35	-0.37	-0.31	-0.29	-0.36	-0.39	-0.41	0.21
T	-4.58	-4.74	-4.71	-6.79	-5.95	-5.73	-6.26	-4.95	-5.05	4.58
C _{max}	0.09	0.08	0.11	0.04	0.08	0.06	0.12	0.09	0.08	0.04
mu1	0.19	0.12	-0.09	-0.08	-0.15	-0.06	0.19	-0.09	0.21	0.06
mu2	-0.07	0.06	0.03	0.05	0.01	0.08	-0.03	-0.04	0.06	0.01
mu3	-0.08	-0.12	-0.07	-0.03	-0.02	-0.05	-0.12	-0.07	-0.11	0.02
mu4	0.09	0.06	0.03	0.08	0.04	0.05	0.09	0.12	0.04	0.03
Ort.	0.76	0.77	0.77	1.06	0.94	0.90	1.02	0.82	0.85	0.76
MaxFTC	-4.73	-4.87	-4.95	-7.12	-6.18	-5.96	-6.50	-5.22	-5.38	-4.73
Min4mu	0.13	0.12	0.10	0.02	0.12	0.02	0.13	-0.08	0.20	-0.08

Minimum ortalama mutlak sapma kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 4, 2, 2, 3

Öncelik kuralı: EETT, OGA sayısı: 1.

MaxFTC kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 4, 2, 2, 3

Öncelik kuralı: EETT, OGA sayısı: 1.

Min4mu kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 3, 2, 2, 3

Öncelik kuralı: İĞİG, OGA sayısı: 2.

Çizelge 7.31 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 3)

İstenen Performans Değerleri										
F = 8000 T = 7000 C _{max} = 19000 mu1=0.70 mu2= 0.70 mu3= 0.70 mu4= 0.70										
Kural	OGA			Tezgah Sayısı						
EETT	1			3	2	2	2			
EETT	2			3	2	2	2			
EETT	3			3	2	2	2			
EKİZ	1			3	2	2	2			
EKİZ	2			3	2	2	3			
EKİZ	3			3	2	2	2			
İĞİG	1			3	2	2	2			
İĞİG	2			3	2	2	3			
İĞİG	3			3	2	2	2			
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	-0.08	-0.06	-0.02	-0.06	-0.19	-0.08	-0.06	-0.09	-0.04	0.02
T	0.31	0.29	0.31	0.41	0.39	0.31	0.29	0.45	0.28	0.28
C _{max}	0.11	0.15	0.12	0.09	0.21	0.12	0.15	0.28	0.09	0.09
mu1	-0.07	-0.08	-0.09	-0.05	-0.17	-0.06	-0.09	-0.27	-0.04	0.04
mu2	0.09	0.12	0.08	0.06	-0.12	0.09	0.11	-0.14	0.06	0.06
mu3	-0.04	-0.07	-0.06	-0.04	-0.17	-0.04	-0.09	-0.19	0.05	0.04
mu4	-0.29	-0.31	-0.27	-0.32	-0.09	-0.26	-0.30	-0.12	-0.29	0.09
Ort.	0.14	0.15	0.14	0.15	0.19	0.14	0.16	0.22	0.12	0.12
MaxFTC	0.34	0.38	0.41	0.44	0.41	0.35	0.38	0.64	0.33	0.64
Min4mu	-0.31	-0.34	-0.34	-0.35	-0.55	-0.27	-0.37	-0.72	-0.22	-0.72

Minimum ortalama mutlak sapma kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 3, 2, 2, 2

Öncelik kuralı: İGİG.

OGA sayısı: 3.

MaxFTC kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 3, 2, 2, 3

Öncelik kuralı: İGİG,

OGA sayısı: 2.

Min4mu kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 3, 2, 2, 3

Öncelik kuralı: İGİG,

OGA sayısı: 2.

Burada bir uzman karar vericiye ihtiyaç vardır. Çünkü hem farklı sonuçlar hem de birbirine çok yakın performans değerleri elde edilmiştir. Öncelik kuralına dayalı olan bu yaklaşımın bu gibi problemleri, iş merkezine dayalı ikinci yaklaşım ile ortadan kaldırılmıştır.

Çizelge 7.32 Hata analizi ve karar çizelgesi (1 .yaklaşım örnek 4)

İstenen Performans Değerleri										
F = 15000 T = 12000 C _{max} = 28000 mu1=0.80 mu2= 0.70 mu3= 0.70 mu4= 0.80										
Kural	OGA			Tezgah Sayısı						
EETT	1			2	1	1	1			
EETT	2			2	1	1	1			
EETT	3			1	1	1	1			
EKİZ	1			2	1	1	1			
EKİZ	2			2	1	1	2			
EKİZ	3			2	1	1	1			
İGİG	1			2	1	1	1			
İGİG	2			2	1	1	2			
İGİG	3			2	1	1	1			
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İGİG1	İGİG2	İGİG3	Min
F	-0.08	-0.12	-0.16	-0.07	0.14	-0.05	-0.09	0.16	-0.08	0.05
T	-0.08	-0.09	-0.23	0.14	0.16	-0.08	-0.06	0.17	-0.07	0.06
C _{max}	-0.13	-0.14	-0.31	-0.13	0.07	-0.14	-0.13	0.05	-0.16	0.05
mu1	0.28	0.27	-0.14	0.36	0.12	0.26	0.29	0.09	0.24	0.09
mu2	0.12	0.06	0.15	0.03	-0.14	0.05	0.09	-0.08	0.07	0.03
mu3	-0.04	-0.03	0.08	0.28	-0.18	-0.04	-0.05	-0.21	-0.07	0.03
mu4	-0.12	-0.16	0.00	-0.19	0.19	-0.14	-0.16	0.21	-0.14	0.00
Ort.	0.12	0.12	0.15	0.17	0.14	0.11	0.12	0.14	0.12	0.11
MaxFTC	-0.29	0.35	-0.70	-0.06	0.37	-0.27	-0.28	0.38	-0.31	0.38
Min4mu	0.24	0.14	0.09	0.48	-0.01	0.13	0.17	0.01	0.10	-0.01

Minimum ortalama mutlak sapma kriterine göre seçim kararı:

İş merkezi tezgah sayısı: 2, 1, 1, 1,

Öncelik kuralı: EKİZ

OGA sayısı: 3.

MaxFTC kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 2, 1, 1, 2

Öncelik kuralı: İGİG,

OGA sayısı: 2.

Min4mu kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 2, 1, 1, 2

Öncelik kuralı: EKİZ,

OGA sayısı: 2.

Yine burada bir uzman karar vericiye ihtiyaç vardır. Çünkü hem farklı sonuçlar hem de birbirine çok yakın performans değerleri elde edilmiştir. Öncelik kuralına dayalı olan bu yaklaşımın bu gibi problemleri, iş merkezine dayalı ikinci yaklaşım ile ortadan kaldırılmıştır.

Çizelge 7.33 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 5)

İstenen Performans Değerleri										
F = 17500 T = 11500 C _{max} = 37500 mu1=0.70 mu2= 0.60 mu3= 0.60 mu4= 0.70										
Kural	OGA			Tezgah Sayısı						
EETT	1			1	1	1	1			
EETT	2			2	1	1	1			
EETT	3			1	1	1	1			
EKİZ	1			1	1	1	1			
EKİZ	2			2	1	1	2			
EKİZ	3			2	1	1	1			
İĞİG	1			1	1	1	1			
İĞİG	2			2	1	1	2			
İĞİG	3			2	1	1	1			
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	0.05	0.07	0.02	0.09	0.07	0.04	0.02	0.04	0.06	0.02
T	-0.28	-0.08	-0.30	-0.04	-0.09	-0.07	-0.27	-0.15	-0.06	0.04
C _{max}	-0.06	0.09	-0.04	-0.07	0.11	0.18	-0.07	0.09	0.10	0.04
mu1	-0.29	0.18	-0.31	-0.27	0.17	0.14	-0.38	0.21	0.18	0.14
mu2	0.08	-0.09	0.05	0.03	-0.11	-0.17	0.04	-0.03	-0.12	0.03
mu3	-0.05	-0.18	0.03	-0.31	-0.15	-0.17	0.04	-0.18	-0.25	0.03
mu4	-0.11	-0.28	-0.14	-0.19	-0.31	-0.24	-0.09	-0.26	-0.28	0.09
Ort.	0.13	0.14	0.13	0.14	0.14	0.14	0.13	0.14	0.15	0.13
MaxFTC	-0.29	0.08	-0.32	-0.02	0.09	0.15	-0.32	-0.02	0.10	0.15
Min4mu	-0.37	-0.37	-0.37	-0.74	-0.40	-0.44	-0.39	-0.26	-0.47	-0.74

Minimum ortalama mutlak sapma kriterine göre seçim kararı için 3 alternatif vardır.

Bunlar için sırasıyla:

İş merkezi tezgah sayıları: 1, 1, 1, 1; 1, 1, 1, 1 ve 1, 1, 1, 1

Öncelik kuralı: EETT, EETT ve İĞİG,

OGA sayısı: 1, 3 ve 1.

MaxFTC kriterine göre de seçim kararı:

İş merkezi tezgah sayıları: 2, 1, 1, 1

Öncelik kuralı: EKİZ,

OGA sayısı: 3.

Min4mu kriterine göre seçim kararı:

İş merkezi tezgah sayıları: 1, 1, 1, 1

Öncelik kuralı: EKİZ, OGA sayısı: 1.

Çizelge 7.34 Hata analizi ve karar çizelgesi (1.yaklaşım örnek 6)

İstenen Performans Değerleri										
F = 18000 T = 13000 C _{max} = 36000 mu1=0.60 mu2= 0.50 mu3= 0.60 mu4= 0.50										
Kural	OGA			Tezgah Sayısı						
EETT	1			2	1	1	2			
EETT	2			2	2	1	2			
EETT	3			2	1	1	2			
EKİZ	1			2	1	1	2			
EKİZ	2			2	1	1	2			
EKİZ	3			2	1	1	2			
İĞİG	1			2	2	1	2			
İĞİG	2			2	2	1	2			
İĞİG	3			2	1	1	2			
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	0.28	0.30	0.24	0.29	0.33	0.26	0.37	0.28	0.32	0.24
T	0.21	0.29	0.20	0.18	0.12	0.31	0.38	0.24	0.28	0.12
C _{max}	0.19	0.24	0.26	0.17	0.28	0.16	0.26	0.28	0.19	0.16
mu1	-0.09	-0.14	-0.07	-0.12	-0.08	-0.06	-0.18	-0.20	-0.13	0.06
mu2	-0.45	0.09	-0.48	-0.55	-0.56	-0.42	0.11	0.08	-0.43	0.08
mu3	-0.36	-0.45	-0.39	-0.09	-0.34	-0.31	-0.45	-0.43	-0.37	0.09
mu4	-0.10	-0.19	-0.12	-0.11	-0.09	-0.06	-0.20	-0.21	-0.14	0.06
Ort.	0.24	0.24	0.25	0.22	0.26	0.23	0.28	0.25	0.27	0.22
MaxFTC	0.68	0.83	0.70	0.64	0.73	0.73	1.01	0.80	0.79	1.01
Min4mu	-1.00	-0.69	-1.06	-0.87	-1.07	-0.85	-0.72	-0.76	-1.07	-1.07

Minimum ortalama mutlak sapma kriterine göre seçilen karar:

İş merkezi tezgah sayısı: 2, 1, 1, 2,

Öncelik kuralı: EKİZ

OGA sayısı: 1.

MaxFTC kriterine göre seçim kararı için karar:

İş merkezi tezgah sayısı: 2, 2, 1, 2,

Öncelik kuralı: İĞİG

OGA sayısı: 1.

Min4mu kriterine göre seçimde 2 alternatif vardır:

İş merkezi tezgah sayısı: 2, 1, 1, 2 ; 2, 1, 1, 2

Öncelik kuralları: EKİZ, İĞİG

OGA sayısı: 2, 3

Bütün bu örneklerde görüldüğü gibi, öncelik kuralına dayalı yaklaşım olarak isimlendirdiğimiz bu yaklaşım, karar verme sırasında bir seçeneğin diğer seçenek ile tam olarak karşılaştırılamayacağı çözümler de üretebilmektedir. Dolayısıyla karar verme işlemi uzman bir karar vericiye ve ek işlemlere gerek duymaktadır.

7.2.3.2 İş Merkezine Dayalı Yaklaşım İçin Deneysel Sonuçlar ve Karar Çizelgeleri

İş merkezi temel alınarak oluşturulan yapay sinir ağı, performans ölçülerinin istenen değerlerini sağlamak üzere bir iş merkezinde kullanılması gerekli olan tezgah sayısı sonucunu üretir. Her iş merkezi için ayrı ayrı işlemler yapılarak her bir iş merkezinde kullanılması gerekli olan tezgah sayısı belirlenir. Dolayısıyla sonuç bir tanedir. Ve bu sonuç bütün faktörler göz önüne alınarak elde edilmiştir. Bu yüzden en uygun değer olduğu söylenebilir. Böylece bir önceki yaklaşımda olduğu gibi bir uzman kullanımı ve ek çalışmalara olan ihtiyaç ortadan kaldırılmıştır. Belirlenen sonucun farklı öncelik kuralı-OGA sayısı durumları karşısında performans ölçülerinin hangi değerlerle gerçekleşeceğini belirlemek için benzetim kullanılır. Sonuç olarak, istenen değerler, gerçekleşen değerler ve aralarındaki fark yüzdesi elde edilerek bu değerlerin düzenlenmiş ve aşağıdaki karar çizelgeleri oluşturulmuştur. İş merkezine dayalı yaklaşım için deneysel sonuçlar ve karar çizelgeleri aşağıda verilmiştir (Çizelge 7.35 – 7.40).

Çizelge 7.35 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 1)

İstenen Performans Değerleri										
F = 5000 T = 500 C _{max} = 15000 mu1=0.90 mu2= 0.90 mu3= 0.90 mu4= 0.90										
TEZGAH SAYISI										
	M1 M2 M3 M4									
	3 2 2 2									
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İGİG1	İGİG2	İGİG3	Min
F	-0.54	-0.59	-0.61	-0.58	-0.52	-0.57	-0.59	-0.61	-0.63	0.52
T	-7.85	-7.88	-7.74	-8.08	-8.07	-8.01	-7.98	-7.76	-7.99	7.74
C _{max}	-0.08	-0.07	-0.05	-0.07	-0.12	-0.10	-0.09	-0.06	-0.09	0.05
mu1	0.15	0.14	0.13	0.11	0.12	0.13	0.14	0.13	0.12	0.11
mu2	0.18	0.21	0.14	0.17	0.16	0.22	0.21	0.21	0.20	0.14
mu3	0.18	0.19	0.15	0.14	0.20	0.16	0.17	0.14	0.17	0.14
mu4	-0.09	-0.08	-0.09	-0.04	-0.06	-0.05	-0.04	-0.06	-0.07	0.04
Ort.	1.30	1.31	1.27	1.32	1.32	1.32	1.32	1.28	1.32	1.27
MaxFTC	-8.47	-8.54	-8.40	-8.73	-8.71	-8.68	-8.66	-8.43	-8.71	-8.40
Min4mu	0.42	0.46	0.33	0.38	0.42	0.46	0.48	0.42	0.42	0.33

Yukarıdaki çizelgede ulaşılmak istenen performans değerlerine karşı, yapay sinir ağı iş merkezlerinde kullanılması gerekli olan tezgah sayılarını, bütün faktörleri ve bütün değerlerini

göz önünde tutarak sırasıyla 3, 2, 2, 2 olarak belirlemiştir. Belirlenen bu değer hangi şartlar altında istenen performans değerlerine en yakın performans değerleri sonucunu verecektir? Bu da, belirlenen bu değer farklı durumlarla benzetimi ile elde edilir. Benzetim sonucunda elde edilen değerlerin istenen performans değerlerinden sapma miktarları, yüzde olarak yukarıdaki çizelgede gösterilmiştir.

Bu yaklaşımda da farklı kararlara varılabilecek değişik karar verme kriterleri kullanılabilir. Ancak diğer yaklaşımdan farklı olarak bir tek durum incelendiğinden (burada 3,2,2,2), alternatif kararlar arasında karşılaştırma yapmak çok kolaydır. Karar vermek için iyi bir uzmana gerek yoktur. Sadece hangi kriter veya kriterlere göre karar verilmesi gerektiğini belirlemek yeterli olabilecektir.

Yukarıdaki çizelgede 3, 2, 2, 2 olarak belirlenen iş merkezleri tezgah sayılarına karşı seçilen karar; minimum ortalama mutlak sapma kriterine göre, öncelik kuralı: EETT ve OGA sayısı: 3, maxFTC kriterine göre, öncelik kuralı: EETT ve OGA sayısı: 3, min4mu kriterine göre, öncelik kuralı: EETT ve OGA sayısı: 3.

Görüldüğü gibi burada karara varmak daha kolaydır. Çünkü alternatifler arasındaki fark açıkça ayırt edilebilmektedir. Hatta çeşitli karar verme kriterlerinin yanında seçilen en iyi değere sahip olan seçeneğin diğer seçeneklerle olan farkı ve bu farkın önemli olup olmadığı da rahatlıkla belirlenebilecektir. Böylelikle daha esnek kararların alınabilmesi mümkün olacaktır.

Bunlarla birlikte, çok az bir ihtimal ile de olsa birinci yaklaşım sadece bu örnekte olmak üzere, hedeflenen performans değerlerine daha yakın bir sonuç üretmiştir. Bu da, böyle bir ihtimal az da olsa her iki yaklaşımın bir arada kullanılması, bir karar verme prosedüründe daha uygun, daha faydalı olacaktır.

Çizelge 7.36 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 2)

İstenen Performans Değerleri										
F = 5000 T = 500 C _{max} = 15000 mu1=0.80 mu2= 0.80 mu3= 0.80 mu4= 0.80										
TEZGAH SAYISI										
				M1	M2	M3	M4			
				4	2	2	3			
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	-0.18	-0.17	-0.19	-0.16	-0.25	-0.21	-0.28	-0.23	-0.24	0.16
T	-4.25	-4.52	-4.78	-7.05	-6.95	-7.78	-5.19	-4.85	-4.78	4.25
C _{max}	0.09	0.13	0.12	0.07	0.06	0.09	0.04	0.07	0.05	0.04
mu1	0.12	0.16	0.14	0.17	0.09	0.11	0.10	0.21	0.13	0.09
mu2	0.04	0.06	0.05	0.07	0.09	0.13	0.11	0.10	0.08	0.04
mu3	-0.04	-0.07	0.00	0.03	0.05	0.07	0.13	0.11	0.05	0.00
mu4	0.02	0.04	0.13	0.08	0.06	0.09	0.13	0.10	0.11	0.02
Ort.	0.70	0.72	0.77	1.09	1.08	1.21	0.85	0.81	0.78	0.70
MaxFTC	-4.34	-4.56	-4.85	-7.14	-7.14	-7.90	-5.43	-5.01	-4.97	-4.34
Min4mu	0.14	0.19	0.32	0.35	0.29	0.40	0.47	0.52	0.37	0.14

Yukarıdaki çizelgede, hedeflenen performans kriterleri değerlerine ulaşabilmek için yapay sinir ağı tarafından 4, 2, 2, 3 olarak belirlenen iş merkezleri tezgah sayıları ile birlikte minimum ortalama mutlak sapma kriterine göre, öncelik kuralı:EETT ve OGA sayısı: 1, maxFTC kriterine göre, öncelik kuralı: EETT ve OGA sayısı: 1, min4mu kriterine göre, öncelik kuralı: EETT ve OGA sayısı: 1'dir . Görüldüğü gibi bu örnekte de karar verme işlemi kolaylıkla gerçekleştirilebilmektedir.

Çizelge 7.37 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 3)

İstenen Performans Değerleri										
F = 8000 T = 7000 C _{max} = 19000 mu1=0.70 mu2= 0.70 mu3= 0.70 mu4= 0.70										
TEZGAH SAYISI										
				M1	M2	M3	M4			
				3	2	2	2			
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	-0.02	-0.05	-0.06	-0.07	-0.03	-0.04	-0.05	-0.04	-0.07	0.02
T	0.28	0.27	0.31	0.29	0.30	0.24	0.31	0.32	0.26	0.24
C _{max}	0.09	0.10	0.13	0.09	0.11	0.12	0.14	0.13	0.10	0.09
mu1	-0.03	-0.06	-0.05	-0.08	-0.04	-0.06	-0.10	-0.07	-0.07	0.03
mu2	0.09	0.06	0.05	0.08	0.07	0.09	0.12	0.13	0.10	0.05
mu3	-0.06	-0.04	-0.02	-0.04	-0.05	-0.06	-0.12	-0.10	-0.06	0.02
mu4	-0.28	-0.29	-0.31	-0.34	-0.29	-0.29	-0.31	-0.32	-0.33	0.28
Ort.	0.12	0.12	0.13	0.14	0.13	0.13	0.16	0.16	0.14	0.12
MaxFTC	0.35	0.32	0.38	0.31	0.38	0.32	0.40	0.41	0.29	0.41
Min4mu	-0.28	-0.33	-0.33	-0.38	-0.31	-0.32	-0.41	-0.36	-0.36	-0.41

Yukarıdaki çizelgede, hedeflenen performans kriterleri değerlerini ulaşabilmek için yapay

sınır ağı tarafından 3, 2, 2, 2 olarak belirlenen iş merkezleri tezgah sayıları ile birlikte minimum ortalama mutlak sapma kriterine göre, öncelik kuralı: EETT ve OGA sayısı: 1 ve EETT ve OGA sayısı: 2; maxFTC kriterine göre, öncelik kuralı: İĞİG ve OGA sayısı: 2, min4mu kriterine göre, öncelik kuralı: İĞİG ve OGA sayısı: 1'dir . Birbirine çok yakın performans kriterleri değerlerine sahip bir karar çizelgesi çıkmasına rağmen, arasındaki en iyi değere sahip olan seçeneği bulmanın yanında, performans kriterlerinin hedeflenen bu değerlerine ulaşmak için öncelik kuralı ve OGA sayısının önemli derecede bir etkisinin olmadığı sonucu da ortaya çıkmaktadır.

Çizelge 7.38 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 4)

İstenen Performans Değerleri										
F = 15000 T = 12000 C _{max} = 28000 mu1=0.80 mu2=0.70 mu3= 0.70 mu4= 0.80										
TEZGAH SAYISI M1 M2 M3 M4 2 1 1 1										
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	-0.07	-0.06	-0.09	-0.10	-0.11	-0.06	-0.07	-0.10	-0.08	0.06
T	-0.03	-0.03	-0.06	-0.07	-0.04	-0.05	-0.06	-0.07	-0.08	0.03
C _{max}	-0.11	-0.11	-0.15	-0.13	-0.10	-0.12	-0.13	-0.11	-0.16	0.10
mu1	0.30	0.23	0.27	0.28	0.31	0.25	0.24	0.28	0.30	0.23
mu2	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.08	0.07	0.03
mu3	-0.05	-0.03	-0.04	-0.06	0.00	-0.02	-0.05	-0.04	-0.06	0.00
mu4	-0.16	-0.14	-0.17	-0.13	-0.12	-0.15	-0.14	-0.13	-0.13	0.12
Ort.	0.11	0.09	0.12	0.12	0.11	0.10	0.11	0.12	0.13	0.09
MaxFTC	-0.21	-0.20	-0.30	-0.30	-0.25	-0.23	-0.26	-0.28	-0.32	-0.20
Min4mu	0.12	0.10	0.11	0.15	0.26	0.16	0.14	0.19	0.18	0.10

Bu örnekte de bütün karar alternatiflerin yaklaşık aynı değerlere sahip oldukları söylenebilir. Ancak bizim kullandığımız üç kritere göre de en uygun seçim kararının 2,1,1,1 tezgah sayıları ile, öncelik kuralının EETT ve OGA sayısının ise 2 adet olduğu görülmektedir.

Çizelge 7.39 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 5)

İstenen Performans Değerleri										
F = 17500 T = 11500 C _{max} = 37500 mu1=0.70 mu2= 0.60 mu3= 0.60 mu4= 0.70										
TEZGAH SAYISI										
	M1	M2	M3	M4						
	1	1	1	1						
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	-0.02	-0.03	0.03	0.03	0.02	-0.03	0.04	0.02	0.01	0.01
T	-0.26	-0.29	-0.24	-0.26	-0.27	-0.29	-0.26	-0.24	-0.27	0.24
C _{max}	-0.01	-0.03	-0.05	-0.02	-0.06	-0.04	-0.02	-0.02	-0.01	0.01
mu1	-0.29	-0.30	-0.29	-0.30	-0.30	-0.29	-0.30	-0.31	-0.29	0.29
mu2	0.02	0.05	0.04	0.06	0.07	0.02	0.03	0.04	0.05	0.02
mu3	-0.05	-0.04	-0.01	-0.02	-0.03	0.01	-0.02	-0.05	-0.04	0.01
mu4	-0.11	-0.09	-0.10	-0.11	-0.09	-0.11	-0.12	-0.11	-0.09	0.09
Ort.	0.11	0.12	0.11	0.11	0.12	0.11	0.11	0.11	0.11	0.11
MaxFTC	-0.29	-0.35	-0.26	-0.25	-0.31	-0.36	-0.24	-0.24	-0.27	0.24
Min4mu	-0.43	-0.38	-0.36	-0.39	-0.35	-0.37	-0.41	-0.43	-0.37	-0.43

Yapay sinir ağı tarafından iş merkezleri tezgah sayıları 1, 1, 1, 1 olarak belirlenmiştir. Öncelik kuralı ve OGA sayısı kararları için: Minimum ortalama mutlak sapma kriterine göre EETT1, EETT3, EKİZ1, EKİZ3, İĞİG1, İĞİG2 ve İĞİG3 olmak üzere yedi farklı alternatif, maxFTC kriterine göre İĞİG1 ve İĞİG2 olmak üzere iki farklı alternatif ve min4mu kriterine göre EETT1 ve İĞİG2 olmak üzere iki alternatif bulunmaktadır. Birbirine çok yakın performans kriterleri değerlerine sahip bir karar çizelgesi çıkmasına rağmen, arasındaki en iyi değere sahip olan seçeneği bulmanın yanında, performans kriterlerinin hedeflenen bu değerlerine ulaşmak için öncelik kuralı ve OGA sayısının önemli derecede bir etkisinin olmadığı sonucu da ortaya çıkmaktadır.

Çizelge 7.40 Hata analizi ve karar çizelgesi (2.yaklaşım örnek 6)

İstenen Performans Değerleri										
F = 18000 T = 13000 C _{max} = 36000 mu1=0.60 mu2= 0.50 mu3= 0.60 mu4= 0.50										
TEZGAH SAYISI										
	M1	M2	M3	M4						
	2	1	1	2						
Hata (%)	EETT1	EETT2	EETT3	EKİZ1	EKİZ2	EKİZ3	İĞİG1	İĞİG2	İĞİG3	Min
F	0.29	0.26	0.27	0.29	0.28	0.30	0.31	0.27	0.28	0.26
T	0.21	0.24	0.19	0.20	0.23	0.16	0.21	0.20	0.23	0.16
C _{max}	0.21	0.20	0.18	0.16	0.17	0.14	0.19	0.20	0.21	0.14
mu1	-0.10	-0.08	-0.07	-0.06	-0.05	-0.02	-0.08	-0.07	-0.06	0.02
mu2	-0.52	-0.56	-0.60	-0.58	-0.51	-0.43	-0.49	-0.48	-0.52	0.43
mu3	-0.38	-0.40	-0.39	-0.36	-0.35	-0.31	-0.45	-0.41	-0.40	0.31
mu4	-0.12	-0.14	-0.13	-0.11	-0.14	-0.06	-0.10	-0.09	-0.11	0.06
Ort.	0.26	0.27	0.26	0.25	0.25	0.20	0.26	0.25	0.26	0.20
MaxFTC	0.71	0.70	0.64	0.65	0.68	0.60	0.71	0.67	0.72	0.72
Min4mu	-1.12	-1.18	-1.19	-1.11	-1.05	-0.83	-1.12	-1.05	-1.09	-1.19

Bu örnekte iş merkezi tezgah sayıları 2, 1, 1,2 olarak belirlenmiştir. Diğer kararlar ise, minimum ortalama mutlak sapma kriteri esas alındığı zaman EKİZ3, maxFTC kriteri İGİG3 ve min4mu kriterine göre zaman EETT3 seçeneği tercih edilecektir.

Bütün bu örneklerde verilen kararlarda birinci derecede önemli olan, iş merkezlerinde kullanılması gerekli tezgah sayısının belirlenmesidir. Bu kararın verilmesinde yapılacak bir hata diğer kararlar üzerinde ve gerçekleştirilmek istenen hedeflerde olumsuz sonuçlara yol açabilecektir. Bu noktada sistemi bir bütün olarak inceleyip, bir açıdan verilen kararların daha objektif olmasını sağlayarak ve bütün alternatifleri göz önüne alarak bir karar verme sistemi oluşturmak son derece önemli olacaktır.

İşte iş merkezine dayalı yaklaşım olarak isimlendirilen ikinci yaklaşım, bu özelliklere sahip bir karar verme sistemidir. Aynı zamanda, aşağıdaki etki analizi kısmında da anlatılacağı gibi, sistemin genel davranışı hakkında bilgi verebilecek bir modelleme aracı olarak da kullanılabilir.

7.2.4 Etki Analizi

Benzetim-yapay sinir ağları ve tekrar benzetimden sonra istenilen performans değerlerine uygun bir konfigürasyonunun belirlenmiş olması yeterli olamayacaktır. Çünkü, bir sonuç elde edilmesine ve bu sonucun optimum olması ihtimali çok fazla olmasına rağmen, belirlenmiş olan bu üçlü (daha fazla da olabilir) konfigürasyonun istenilen performans kriterlerinin hangi değer aralığında optimum kalacağı bilinmemektedir. Ayrıca hangi iş merkezi hangi performans kriterine karşı duyarlıdır? Yani, performans kriterlerinin hangi değer aralığında iş merkezi tezgah sayıları değişmeyerek kalabilir?

Seçilen konfigürasyon uygulandığı takdirde performans ölçülerinin sabit bir değer olmayacağı açıktır. O halde bu değerlerin değişme aralığı ne olacaktır? Hangi performans kriteri hangi değerden daha düşük veya yüksek olursa konfigürasyonda ne gibi değişiklikler yapılması gerekecektir? Bu sorulara cevap verilmesi ile hem karar verme işlemi daha sağlam bir temele oturtulacak, hem de darboğaz noktaların neresi olduğu tespit edilebilecektir. Kısaca ifade etmek gerekirse sistemin bir duyarlılık haritası çıkarılmış olacaktır. Etki analizi olarak isimlendirdiğimiz bu işlemler yine eğitilmiş ağlar kullanılarak yapılabilir.

Burada F, T, CM, mul, mu2, mu3 ve mu4 performans kriterlerinin çeşitli değerleri, eğitilmiş olan yapay sinir ağına girilerek öncelik kuralı ve OGA sayısı dikkate alınarak (1.yaklaşım) ve dikkate alınmadan (2. yaklaşım) iş merkezlerinde kullanılacak olan tezgah sayıları üzerindeki etkisi araştırılmıştır. Bu performans kriterlerinden birinin tek başına etkisini bulmak için, seçilen performans kriterinin değeri istenen belirli bir değer aralığında kademeli olarak azaltılarak diğer performans kriterlerinin sabit tutulan değerleri ile birlikte eğitilmiş olan yapay sinir ağına sunulmuş ve tezgah sayılarındaki değişim gözlenmiştir. Böylelikle sistem konfigürasyonuna etki eden performans kriterleri ve bunların etki alanı değer aralığının bulunması amaçlanmıştır.

En düşük sayıdaki konfigürasyon ile en yüksek sayıdaki konfigürasyon değerlerinin benzetim sonuçlarına göre performans kriterlerinin değerleri elde edilmiştir. Bulunmak istenen performans kriterinin aldığı değerlerin en düşük ile en yüksek arasındaki fark değeri 100 eşit parçaya bölünmüştür. Elde edilen değerler en büyük değerden başlayıp her adımda bir önceki değerden çıkarılmak suretiyle daha önce eğitilmiş olan yapay sinir ağına verilmiş ve sonuçlar elde edilmiştir.

Bu şekilde yapay sinir ağından elde edilen sonuçlarda, girilen değer azalarak değiştiği halde, belli aralıklarda değişmediği görülmüştür. Bu alanın belirlenmesi ile performans kriterlerinin arzulanan değerinin hangi konfigürasyon tasarımı ile sağlandığını belirlemenin yanında, bu konfigürasyonun performans kriterinin hangisine ve hangi değer aralığında duyarlı olduğu da belirlenmiş olacaktır.

Daha önce yapılan yapay sinir ağı çalışmalarında üretilen çözümler noktasal bir değer ifade etmektedir. Oysa üretilen çözüm değeri ancak çözüm kümesinin bir elemanı olmaktadır. Burada o çözüm değerinin de içinde bulunduğu bir çözüm kümesi üretilmektedir. Burada kullanılan maksimum ve minimum değerler benzetim sonucuna göre belirlenmiştir. Yani bir nevi sistemin duyarlılık ya da darboğaz haritası çıkarılmıştır. Sistemin genel etki alanı çizelgesi bu bölümün sonuna kadar yer alan çizelgelerde gösterildiği gibidir (Çizelge 7.41 – 7.54).

Çizelge 7.41 Ortalama akış süresi (F) etki alanı (1.yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	mu4	İş Mer. Tez. Say.
EETT1	17989	12487	37456	0.85	0.51	0.58	0.78	1 1 1 1
	5287	12487	37456	0.85	0.51	0.58	0.78	1 1 1 1
EETT2	17887	12524	37768	0.83	0.48	0.58	0.78	1 1 1 1
	5268	12524	37768	0.83	0.48	0.58	0.78	1 1 1 1
EETT3	17789	12573	37689	0.83	0.49	0.59	0.79	1 1 1 1
	5245	12573	37689	0.83	0.49	0.59	0.79	1 1 1 1
İĞİG1	17778	12453	37675	0.82	0.47	0.57	0.77	1 1 1 1
	5298	12453	37675	0.82	0.47	0.57	0.77	1 1 1 1
İĞİG2	17689	12678	37564	0.85	0.49	0.60	0.80	1 1 1 1
	5223	12678	37564	0.85	0.49	0.60	0.80	1 1 1 1
İĞİG3	17745	12596	37452	0.84	0.48	0.59	0.79	1 1 1 1
	5265	12596	37452	0.84	0.48	0.59	0.79	1 1 1 1
EKİZ1	17895	12564	37485	0.83	0.48	0.58	0.78	1 1 1 1
	5278	12564	37485	0.83	0.48	0.58	0.78	1 1 1 1
EKİZ2	17945	12687	37569	0.82	0.49	0.60	0.79	1 1 1 1
	5278	12687	37569	0.82	0.49	0.60	0.79	1 1 1 1
EKİZ3	17987	12756	37658	0.83	0.50	0.59	0.80	1 1 1 1
	5256	12756	37658	0.83	0.50	0.59	0.80	1 1 1 1

Çizelge 7.42 Ortalama akış süresi (F) etki alanı (2. yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	mu4	Tez. Say.
M1	17989	12487	37456	0.85	0.51	0.58	0.78	1
	5287	12487	37456	0.85	0.51	0.58	0.78	1
M2	17887	12524	37768	0.83	0.48	0.58	0.78	1
	5268	12524	37768	0.83	0.48	0.58	0.78	1
M3	17789	12573	37689	0.83	0.49	0.59	0.79	1
	5245	12573	37689	0.83	0.49	0.59	0.79	1
M4	17778	12453	37675	0.82	0.47	0.57	0.77	1
	5298	12453	37675	0.82	0.47	0.57	0.77	1

Ortalama akış süresi etki alanı analizi, ortalama akış süresinin yaklaşık 5000 ile 18000 değerleri arasında yapılmıştır. Diğer performans kriterleri değerleri sabit tutulup, ortalama akış süresi 18000 değerinden başlayıp 5000 değerine kadar eşit aralıklarla düşürülerek (100 eşit aralık - $(18000-5000) / 100$), istenen performans değerlerini başarabilmek için iş merkezlerinde kullanılması gerekli olan tezgah sayısının değişimi gözlenmiştir. Çizelgelerde yer alan değerler değişim noktalarıdır. Ayrıca değişim olup olmadığına

bakılmaksızın son değerler de çizelgede yer almaktadır. Yukarıdaki her iki yaklaşıma ait çizelgelerden de görüldüğü gibi, ortalama akış süresi 5000 ile 18000 değerleri arasında değiştirildiği halde iş merkezleri tezgah sayılarında herhangi bir değişim olmamıştır. Bu sonuç, performans kriterlerinden ortalama akış süresi kriterinin konfigürasyon tasarımı üzerinde tek başına etkili bir kriter olmadığını göstermektedir.

Çizelge 7.43 Ortalama gecikme süresi (T) etki alanı (1.yaklaşım)

	F	T	C _{max}	mu1	mu2	Mu3	mu4	İş Mer. Tez. Say.
EETT1	17989	12895	37456	0.85	0.51	0.58	0.78	1 1 1 1
	17989	595	37456	0.85	0.51	0.58	0.78	1 1 1 1
EETT2	17887	12945	37768	0.83	0.48	0.58	0.78	1 1 1 1
	17887	586	37768	0.83	0.48	0.58	0.78	1 1 1 1
EETT3	17789	12968	37689	0.83	0.49	0.59	0.79	1 1 1 1
	17789	581	37689	0.83	0.49	0.59	0.79	1 1 1 1
İGİG1	17778	12989	37675	0.82	0.47	0.57	0.77	1 1 1 1
	17778	578	37675	0.82	0.47	0.57	0.77	1 1 1 1
İGİG2	17689	12898	37564	0.85	0.49	0.60	0.80	1 1 1 1
	17689	589	37564	0.85	0.49	0.60	0.80	1 1 1 1
İGİG3	17745	12951	37452	0.84	0.48	0.59	0.79	1 1 1 1
	17745	789	37452	0.84	0.48	0.59	0.79	1 1 1 1
EKİZ1	17895	12894	37485	0.83	0.48	0.58	0.78	1 1 1 1
	17895	795	37485	0.83	0.48	0.58	0.78	1 1 1 1
EKİZ2	17945	12952	37569	0.82	0.49	0.60	0.79	1 1 1 1
	17945	806	37569	0.82	0.49	0.60	0.79	1 1 1 1
EKİZ3	17987	12874	37658	0.83	0.50	0.59	0.80	1 1 1 1
	17987	812	37658	0.83	0.50	0.59	0.80	1 1 1 1

Çizelge 7.44 Ortalama gecikme süresi (T) etki alanı (2. yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	mu4	Tez. Say.
M1	17989	12895	37456	0.85	0.51	0.58	0.78	1
	17989	595	37456	0.85	0.51	0.58	0.78	1
M2	17887	12945	37768	0.83	0.48	0.58	0.78	1
	17887	586	37768	0.83	0.48	0.58	0.78	1
M3	17789	12968	37689	0.83	0.49	0.59	0.79	1
	17789	581	37689	0.83	0.49	0.59	0.79	1
M4	17778	12989	37675	0.82	0.47	0.57	0.77	1
	17778	578	37675	0.82	0.47	0.57	0.77	1

Ortalama gecikme süresi etki alanı analizi, yaklaşık 500 ile 13000 değerleri arasında yapılmıştır. Diğer performans kriterleri değerleri sabit iken, ortalama gecikme süresi 13000 değerinden başlayıp 500 değerine kadar eşit aralıklarla düşürülerek (100 eşit aralık) istenen performans değerlerini başarabilmek için iş merkezlerinde kullanılması gerekli olan tezgah sayısının değişimi gözlenmiştir.

Yukarıdaki her iki yaklaşıma ait çizelgelerden de görüldüğü gibi, ortalama gecikme süresi değeri 500 seviyesine kadar düşürüldüğü halde iş merkezleri tezgah sayılarında herhangi bir değişim gözlenmemiştir. Bu sonuca göre, performans kriterlerinden ortalama gecikme süresi kriterinin de, tasarım sürecinde tek başına etkili bir kriter olmadığı görülmektedir.

Çizelge 7.45 Maksimum tamamlanma süresi (C_{max}) etki alanı (1.yaklaşım)

	F	T	C_{max}	mu1	mu2	Mu3	mu4	İş Mer. Tez. Say.
EETT1	17989	12895	37456	0.85	0.51	0.58	0.78	1 1 1 1
	17989	12895	22345	0.85	0.51	0.58	0.78	1 1 1 2
	17989	12895	18458	0.85	0.51	0.58	0.78	2 1 1 2
	17989	12895	16521	0.85	0.51	0.58	0.78	2 2 1 2
	17989	12895	13259	0.85	0.51	0.58	0.78	2 2 2 2
	17989	12895	10982	0.85	0.51	0.58	0.78	2 2 2 2
EETT2	17887	12945	37768	0.83	0.48	0.58	0.78	1 1 1 1
	17887	12945	23872	0.83	0.48	0.58	0.78	1 2 1 1
	17887	12945	21954	0.83	0.48	0.58	0.78	2 2 1 1
	17887	12945	19256	0.83	0.48	0.58	0.78	2 2 2 1
	17887	12945	16545	0.83	0.48	0.58	0.78	2 2 2 2
	17887	12945	12154	0.83	0.48	0.58	0.78	2 2 2 2
EETT3	17789	12968	37689	0.83	0.49	0.59	0.79	1 1 1 1
	17789	12968	23145	0.83	0.49	0.59	0.79	1 2 1 1
	17789	12968	17456	0.83	0.49	0.59	0.79	2 2 1 2
	17789	12968	16545	0.83	0.49	0.59	0.79	2 2 2 2
	17789	12968	13254	0.83	0.49	0.59	0.79	3 3 2 2
	17789	12968	10897	0.83	0.49	0.59	0.79	3 3 2 2

Çizelge 7.45 Maksimum tamamlanma süresi (C_{max}) etki alanı (1.yaklaşım) (devam)

İGİG1	17778	12989	37675	0.82	0.47	0.57	0.77	1	1	1	1
	17778	12989	22136	0.82	0.47	0.57	0.77	1	1	2	1
	17778	12989	19893	0.82	0.47	0.57	0.77	2	1	2	1
	17778	12989	18452	0.82	0.47	0.57	0.77	2	2	2	1
	17778	12989	12156	0.82	0.47	0.57	0.77	2	2	2	2
	17778	12989	10897	0.82	0.47	0.57	0.77	3	2	3	2
	17778	12989	10235	0.82	0.47	0.57	0.77	3	3	3	2
İGİG2	17689	12898	37564	0.85	0.49	0.60	0.80	1	1	1	1
	17689	12898	24897	0.85	0.49	0.60	0.80	1	1	1	2
	17689	12898	22136	0.85	0.49	0.60	0.80	1	2	1	2
	17689	12898	18145	0.85	0.49	0.60	0.80	2	2	1	2
	17689	12898	13156	0.85	0.49	0.60	0.80	2	2	2	2
	17689	12898	11236	0.85	0.49	0.60	0.80	3	3	2	3
	17689	12898	10258	0.85	0.49	0.60	0.80	3	3	3	3
İGİG3	17745	12951	37452	0.84	0.48	0.59	0.79	1	1	1	1
	17745	12951	25456	0.84	0.48	0.59	0.79	2	1	1	1
	17745	12951	21783	0.84	0.48	0.59	0.79	2	1	2	2
	17745	12951	17125	0.84	0.48	0.59	0.79	2	2	2	2
	17745	12951	14236	0.84	0.48	0.59	0.79	3	3	2	2
	17745	12951	11245	0.84	0.48	0.59	0.79	3	3	3	2
	17745	12951	10784	0.84	0.48	0.59	0.79	3	3	3	3
EKİZ1	17895	12894	37485	0.83	0.48	0.58	0.78	1	1	1	1
	17895	12894	24896	0.83	0.48	0.58	0.78	2	2	1	1
	17895	12894	20458	0.83	0.48	0.58	0.78	2	2	1	2
	17895	12894	16256	0.83	0.48	0.58	0.78	2	2	2	2
	17895	12894	12782	0.83	0.48	0.58	0.78	2	2	3	3
	17895	12894	10987	0.83	0.48	0.58	0.78	3	2	3	3
	17895	12894	10235	0.83	0.48	0.58	0.78	3	3	3	3
EKİZ2	17945	12952	37569	0.82	0.49	0.60	0.79	1	1	1	1
	17945	12952	21458	0.82	0.49	0.60	0.79	2	1	1	1
	17945	12952	17421	0.82	0.49	0.60	0.79	2	2	2	1
	17945	12952	12365	0.82	0.49	0.60	0.79	2	2	2	2
	17945	12952	11897	0.82	0.49	0.60	0.79	2	3	3	2
	17945	12952	10245	0.82	0.49	0.60	0.79	3	3	3	3
EKİZ3	17987	12874	37658	0.83	0.50	0.59	0.80	1	1	1	1
	17987	12874	24782	0.83	0.50	0.59	0.80	1	2	1	1
	17987	12874	20496	0.83	0.50	0.59	0.80	2	2	1	2
	17987	12874	15783	0.83	0.50	0.59	0.80	2	2	2	2
	17987	12874	12736	0.83	0.50	0.59	0.80	2	3	3	2
	17987	12874	10781	0.83	0.50	0.59	0.80	3	3	3	2
	17987	12874	10156	0.83	0.50	0.59	0.80	3	3	3	3

Maksimum tamamlanma süresi kriteri için etki alanı analizi, yaklaşık 37500 ile 10000 değerleri arasında yapılmıştır. İstenen performans değerlerini başarabilmek için iş merkezlerinde kullanılması gerekli olan tezgah sayısının belirlenmesi üzerinde maksimum tamamlanma süresi kriterinin etkili olduğu gözlenmiştir.

Yukarıdaki birinci yaklaşıma ait çizelgeden de görüldüğü gibi, maksimum tamamlanma süresi kriterinin konfigürasyon üzerindeki etkisi seçilen öncelik kuralı ve OGA sayısına göre değişiklik arz etmektedir. Örnek olarak, maksimum tamamlanma süresi değeri 37456 ile 10982 arasında değiştirilen EETT1 yapay sinir ağı sonucu incelendiğinde aşağıdaki sonuçlar elde edilir.

- İş merkezi tezgah sayıları en az "bir" en fazla "iki" adet olarak gerçekleşmiştir.
- Maksimum tamamlanma süresindeki değişime ilk duyarlılık gösteren iş merkezi dördüncü iş merkezi, en son duyarlılık gösteren iş merkezi ise, üçüncü iş merkezidir. Maksimum tamamlanma süresindeki değişimin tezgah sayılarında değişimi gerektirdiği iş merkezleri, öncelik itibarıyla 4, 1, 2 ve 3 olarak sıralanabilir.
- Bu 4, 1, 2 ve 3 sırası, iş merkezinde kullanılan tezgah sayıları aynı kalmak üzere, maksimum tamamlanma süresi düşürülmek istendiğinde karşılaşılabilecek darboğaz noktaların sırasını da göstermektedir.
- Maksimum tamamlanma süresi kriterinin etkili olduğu iş merkezleri ve etki alanı;

İş merkezi: 4

Etki alanı : 37456 - 22344 -> Tezgah sayısı = 1, 22345- 10982 -> Tezgah sayısı = 2

İş merkezi : 1

Etki alanı : 37456 - 18457 -> Tezgah sayısı = 1, 18458- 10982 -> Tezgah sayısı = 2

İş merkezi : 2

Etki alanı : 37456 - 16520 -> Tezgah sayısı = 1, 16521- 10982 -> Tezgah sayısı = 2

İş merkezi : 3

Etki alanı : 37456 - 13258 -> Tezgah sayısı = 1, 13259- 10982 -> Tezgah sayısı = 2.

Maksimum tamamlanma süresinin değeri yaklaşık 13000 değerinin altında olması istenildiği zaman bütün iş merkezlerinde "iki" adet tezgah kullanılmalıdır. Bu sonuçlara göre, performans kriterlerinden maksimum tamamlanma süresi kriterinin, tasarım sırasında tek başına etkisinden bahsedilebilir.

Çizelge 7.46 Maksimum tamamlanma süresi (C_{max}) etki alanı (2.yaklaşım)

	F	T	C_{max}	mu1	mu2	mu3	mu4	Tez. Say.
M1	17989	12895	37456	0.85	0.51	0.58	0.78	1
	17989	12895	19856	0.85	0.51	0.58	0.78	2
	17989	12895	10982	0.85	0.51	0.58	0.78	2
M2	17887	12945	37768	0.83	0.48	0.58	0.78	1
	17887	12945	19126	0.83	0.48	0.58	0.78	2
	17887	12945	12154	0.83	0.48	0.58	0.78	2
M3	17789	12968	37689	0.83	0.49	0.59	0.79	1
	17789	12968	16354	0.83	0.49	0.59	0.79	2
	17789	12968	10897	0.83	0.49	0.59	0.79	2
M4	17778	12989	37675	0.82	0.47	0.57	0.77	1
	17778	12989	20196	0.82	0.47	0.57	0.77	2
	17778	12989	10235	0.82	0.47	0.57	0.77	2

Daha önce de bahsedildiği gibi iş merkezine dayalı ikinci yaklaşım, bütün öncelik kuralı ve OGA sayısı örneklerini içinde barındırır. Dolayısı ile sonuç çok daha geneldir. Burada elde edilen sonuçlara göre;

İş merkezi tezgah sayıları en az "bir" en fazla "iki" adet olarak gerçekleşmiştir. Maksimum tamamlanma süresindeki değişime ilk duyarlılık gösteren iş merkezi dördüncü iş merkezi, en son duyarlılık gösteren iş merkezi ise, üçüncü iş merkezidir. Maksimum tamamlanma süresindeki değişimin tezgah sayılarında değişimi gerektirdiği iş merkezleri, öncelik itibarıyla 4, 1, 2 ve 3 olarak sıralanabilir.

Bu 4, 1, 2 ve 3 sırası, iş merkezinde kullanılan tezgah sayıları aynı kalmak üzere, maksimum tamamlanma süresi düşürülmek istendiğinde karşılaşılabilecek darboğaz noktaların sırasını, tersine düşünüldüğünde darboğaz olmayan noktaların sırasını da göstermektedir.

Maksimum tamamlanma süresi kriterinin etkili olduğu iş merkezleri ve etki alanı;

İş merkezi : 4

Etki alanı : 37675 - 20195 -> Tezgah sayısı = 1, 20196 - 10235 -> Tezgah sayısı = 2.

İş merkezi: 1

Etki alanı : 37456 - 19855 -> Tezgah sayısı = 1, 19856- 10982 -> Tezgah sayısı = 2.

İş merkezi: 2

Etki alanı : 37768 - 19125 -> Tezgah sayısı = 1, 19126- 12154 -> Tezgah sayısı = 2.

İş merkezi : 3

Etki alanı : 37675 - 20195 -> Tezgah sayısı = 1, 20196- 10235 -> Tezgah sayısı - 2.

Maksimum tamamlanma süresi değeri yaklaşık 16000 değerinin altında olmasının istenmesi halinde, bütün iş merkezlerinde "iki" adet tezgah kullanılması gereklidir.

Bu sonuçlara göre de, performans kriterlerinden maksimum tamamlanma süresi kriterinin, tasarım sırasında tek başına etkisinden bahsedilebilir. Her iki yaklaşımda da ele aldığımız örneklerde sonuç 4, 1, 2, 3 olmuştur. Yani sistemi genel olarak incelediğimiz zaman da ikinci yaklaşımın sonuçları dikkate alınmalıdır. Burada birinci yaklaşımda ele aldığımız örnek de olduğu gibi OGA sayısının "bir" ve öncelik kuralının "EETT" olarak tercih edilmesi gibi bir tasarım süreci sonunda da sıralama yine 4, 1, 2, 3 olmuştur. Ancak, OGA sayısının ve öncelik kurallarının değişimi durumunda sıralama da değişirse o zaman sistemi incelemek için birinci yaklaşım kullanılmalıdır.

Çizelge 7.47 İş merkezi 1 kullanım oranı (mu1) etki alanı (1.yaklaşımı)

	F	T	C _{max}	mu1	mu2	mu3	mu4	İş Mer. Tez. Say.
EETT1	17989	12895	37456	1.00	0.51	0.58	0.78	1 1 1 1
	17989	12895	37456	0.64	0.51	0.58	0.78	2 1 1 1
	17989	12895	37456	0.41	0.51	0.58	0.78	3 1 1 1
	17989	12895	37456	0.31	0.51	0.58	0.78	4 1 1 1
	17989	12895	37456	0.20	0.51	0.58	0.78	5 1 1 1
	17989	12895	37456	0.11	0.51	0.58	0.78	6 1 1 1
	17989	12895	37456	0.01	0.51	0.58	0.78	7 1 1 1
EETT2	17887	12945	37768	1.00	0.48	0.58	0.78	1 1 1 1
	17887	12945	37768	0.65	0.48	0.58	0.78	2 1 1 1
	17887	12945	37768	0.39	0.48	0.58	0.78	3 1 1 1
	17887	12945	37768	0.24	0.48	0.58	0.78	4 1 1 1
	17887	12945	37768	0.16	0.48	0.58	0.78	5 1 1 1
	17887	12945	37768	0.09	0.48	0.58	0.78	6 1 1 1
	17887	12945	37768	0.01	0.48	0.58	0.78	7 1 1 1
EETT3	17789	12968	37689	1.00	0.49	0.59	0.79	1 1 1 1
	17789	12968	23145	0.49	0.49	0.59	0.79	2 1 1 1
	17789	12968	17456	0.30	0.49	0.59	0.79	3 1 1 1
	17789	12968	16545	0.21	0.49	0.59	0.79	4 1 1 1
	17789	12968	13254	0.14	0.49	0.59	0.79	5 1 1 1
	17789	12968	10897	0.10	0.49	0.59	0.79	6 1 1 1
	17789	12968	10897	0.01	0.49	0.59	0.79	7 1 1 1

Çizelge 7.47 İş merkezi1 kullanım oranı (mu1) etki alanı (1.yaklaşım) (devamı)

İĞİG1	17778	12989	37675	1.00	0.47	0.57	0.77	1	1	1	1
	17778	12989	22136	0.58	0.47	0.57	0.77	2	1	1	1
	17778	12989	19893	0.32	0.47	0.57	0.77	3	1	1	1
	17778	12989	18452	0.21	0.47	0.57	0.77	4	1	1	1
	17778	12989	12156	0.13	0.47	0.57	0.77	5	1	1	1
	17778	12989	10897	0.10	0.47	0.57	0.77	6	1	1	1
	17778	12989	10235	0.01	0.47	0.57	0.77	7	1	1	1
İĞİG2	17689	12898	37564	1.00	0.49	0.60	0.80	1	1	1	1
	17689	12898	24897	0.51	0.49	0.60	0.80	2	1	1	1
	17689	12898	22136	0.36	0.49	0.60	0.80	3	1	1	1
	17689	12898	18145	0.27	0.49	0.60	0.80	4	1	1	1
	17689	12898	13156	0.16	0.49	0.60	0.80	5	1	1	1
	17689	12898	11236	0.12	0.49	0.60	0.80	6	1	1	1
	17689	12898	10258	0.01	0.49	0.60	0.80	7	1	1	1
İĞİG3	17745	12951	37452	1.00	0.48	0.59	0.79	1	1	1	1
	17745	12951	25456	0.56	0.48	0.59	0.79	2	1	1	1
	17745	12951	21783	0.38	0.48	0.59	0.79	3	1	1	1
	17745	12951	17125	0.27	0.48	0.59	0.79	4	1	1	1
	17745	12951	14236	0.15	0.48	0.59	0.79	5	1	1	1
	17745	12951	11245	0.09	0.48	0.59	0.79	6	1	1	1
	17745	12951	10784	0.01	0.48	0.59	0.79	7	1	1	1
EKİZ1	17895	12894	37485	1.00	0.48	0.58	0.78	1	1	1	1
	17895	12894	24896	0.61	0.48	0.58	0.78	2	1	1	1
	17895	12894	20458	0.39	0.48	0.58	0.78	3	1	1	1
	17895	12894	16256	0.27	0.48	0.58	0.78	4	1	1	1
	17895	12894	12782	0.18	0.48	0.58	0.78	5	1	1	1
	17895	12894	10987	0.10	0.48	0.58	0.78	6	1	1	1
	17895	12894	10235	0.01	0.48	0.58	0.78	7	1	1	1
EKİZ2	17945	12952	37569	1.00	0.49	0.60	0.79	1	1	1	1
	17945	12952	21458	0.62	0.49	0.60	0.79	2	1	1	1
	17945	12952	17421	0.41	0.49	0.60	0.79	3	1	1	1
	17945	12952	12365	0.30	0.49	0.60	0.79	4	1	1	1
	17945	12952	11897	0.19	0.49	0.60	0.79	5	1	1	1
	17945	12952	10245	0.12	0.49	0.60	0.79	6	1	1	1
	17945	12952	10245	0.01	0.49	0.60	0.79	7	1	1	1
EKİZ3	17987	12874	37658	1.00	0.50	0.59	0.80	1	1	1	1
	17987	12874	24782	0.60	0.50	0.59	0.80	2	1	1	1
	17987	12874	20496	0.38	0.50	0.59	0.80	3	1	1	1
	17987	12874	15783	0.24	0.50	0.59	0.80	4	1	1	1
	17987	12874	12736	0.16	0.50	0.59	0.80	5	1	1	1
	17987	12874	10781	0.09	0.50	0.59	0.80	6	1	1	1
	17987	12874	10156	0.01	0.50	0.59	0.80	7	1	1	1

Çizelge 7.48 İş merkezi 1 kullanım oranı (μ_1) etki alanı (2.yaklaşım)

	F	T	C _{max}	μ_1	μ_2	μ_3	μ_4	Tez. Say.
M1	17989	12895	37456	1.00	0.51	0.58	0.78	1
	17989	12895	37456	0.59	0.51	0.58	0.78	2
	17989	12895	37456	0.31	0.51	0.58	0.78	3
	17989	12895	37456	0.20	0.51	0.58	0.78	4
	17989	12895	37456	0.16	0.51	0.58	0.78	5
	17989	12895	37456	0.11	0.51	0.58	0.78	6
	17989	12895	37456	0.06	0.51	0.58	0.78	7
	17989	12895	37456	0.01	0.51	0.58	0.78	7
M2	17887	12945	37768	1.00	0.48	0.58	0.78	1
	17887	12945	37768	0.01	0.48	0.58	0.78	1
M3	17789	12968	37689	1.00	0.49	0.59	0.79	1
	17789	12968	23145	0.01	0.49	0.59	0.79	1
	17778	12989	37675	1.00	0.47	0.57	0.77	1
	17778	12989	22136	0.01	0.47	0.57	0.77	1

İncelemeler, iş merkezi kullanım oranları için %100 değerinden %1 değerine kadar 100 adımda yapılmıştır. İş merkezi kullanım oranları etki alanı analizi çizelgelerinden de görüleceği üzere, iş merkezi kullanım oranlarında yapılan değişiklikler, değişiklik yapılan iş merkezinde kullanılması gerekli olan tezgah sayısı üzerinde büyük oranda etkili olmakta, diğer iş merkezleri üzerinde ise etkili olmamaktadır.

Yukarıdaki çizelgelerden ikinci yaklaşımın sonuçları hem daha genel hem de daha kısa olması dolayısıyla aşağıda açıklanmıştır. Birinci yaklaşımın sonuçları daha önce de söylenildiği gibi, belirli bir öncelik kuralı ve OGA sayısı kullanıldığı durumlarda daha faydalı olabilecektir. Burada elde edilen sonuçlara göre

- İş merkezi tezgah sayıları en az "bir", en fazla "yedi" adet olarak gerçekleşmiştir.
- Birinci iş merkezi kullanım oranındaki değişime sadece birinci iş merkezi duyarlılık göstermiştir.
- Bu duyarlılık, iş merkezinde kullanılan tezgah sayıları aynı kalmak üzere, birinci iş merkezi kullanım oranı düşürülmek istendiğinde karşılaşılabilecek darboğaz noktanın yine birinci iş merkezinde yaşanacağını göstermektedir.
- Birinci iş merkezi kullanım oranı kriterinin etki alanı;

İş merkezi: 1

Etki alanı : % 100 - %60 -> Tezgah sayısı = 1,

%59 - %32 -> Tezgah sayısı = 2,

%31 - %21 -> Tezgah sayısı = 3,

%20 - %17 -> Tezgah sayısı = 4,

%16 - %12 -> Tezgah sayısı = 5,

%11 - % 7 -> Tezgah sayısı = 6,

% 6 - % 2 -> Tezgah sayısı = 7,

% 1 - -> Tezgah sayısı = 7.

İş merkezi: 2

Etki alanı : % 100 - %01 -> Tezgah sayısı = 1.

İş merkezi : 3

Etki alanı : %100 - %01 -> Tezgah sayısı = 1.

İş merkezi : 4

Etki alanı : % 100 - %01 -> Tezgah sayısı = 1.

- Birinci iş merkezinde "bir" ya da "iki" tezgah kullanılması ile uygun bir performans seviyesi yakalanabilecektir.

Bu sonuçlara göre, performans kriterlerinden birinci iş merkezi kullanım oranı kriterinin tasarım sırasında göz önüne alınması gerekli bir kriterdir.

Çizelge 7.49 İş merkezi 2 kullanım oranı (mu2) etki alanı (1.yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	Mu4	İş Mer. Tez. Say.
EETT1	17989	12895	37456	0.85	1.00	0.58	0.78	1 0 0 1
	17989	12895	37456	0.85	0.78	0.58	0.78	1 1 1 1
	17989	12895	37456	0.85	0.28	0.58	0.78	1 2 1 1
	17989	12895	37456	0.85	0.16	0.58	0.78	1 3 1 1
	17989	12895	37456	0.85	0.11	0.58	0.78	1 4 1 1
	17989	12895	37456	0.85	0.07	0.58	0.78	1 5 1 1
	17989	12895	37456	0.85	0.04	0.58	0.78	1 6 1 1
	17989	12895	37456	0.85	0.01	0.58	0.78	1 7 1 1
EETT2	17887	12945	37768	0.83	1.00	0.58	0.78	1 0 1 1
	17887	12945	37768	0.83	0.71	0.58	0.78	1 1 1 1
	17887	12945	37768	0.83	0.31	0.58	0.78	1 2 1 1
	17887	12945	37768	0.83	0.26	0.58	0.78	1 3 1 1
	17887	12945	37768	0.83	0.16	0.58	0.78	1 4 1 1
	17887	12945	37768	0.83	0.11	0.58	0.78	1 5 1 1
	17887	12945	37768	0.83	0.05	0.58	0.78	1 6 1 1
	17887	12945	37768	0.83	0.01	0.58	0.78	1 7 1 1
EETT3	17789	12968	37689	0.83	1.00	0.59	0.79	1 0 1 1
	17789	12968	23145	0.83	0.71	0.59	0.79	1 1 1 1
	17789	12968	17456	0.83	0.29	0.59	0.79	1 2 1 1
	17789	12968	16545	0.83	0.21	0.59	0.79	1 3 1 1
	17789	12968	13254	0.83	0.14	0.59	0.79	1 4 1 1
	17789	12968	10897	0.83	0.07	0.59	0.79	1 5 1 1
	17789	12968	10897	0.83	0.04	0.59	0.79	1 6 1 1
	17789	12968	10897	0.83	0.01	0.59	0.79	1 7 1 1
İGİG1	17778	12989	37675	0.82	1.00	0.57	0.77	0 0 1 1
	17778	12989	22136	0.82	0.78	0.57	0.77	1 1 1 1
	17778	12989	19893	0.82	0.34	0.57	0.77	1 2 1 1
	17778	12989	18452	0.82	0.25	0.57	0.77	1 3 1 1
	17778	12989	12156	0.82	0.14	0.57	0.77	1 4 1 1
	17778	12989	10897	0.82	0.10	0.57	0.77	1 5 1 1
	17778	12989	10235	0.82	0.02	0.57	0.77	1 6 1 1
	17778	12989	10235	0.82	0.01	0.57	0.77	1 7 1 1
İGİG2	17689	12898	37564	0.85	1.00	0.60	0.80	1 0 1 1
	17689	12898	24897	0.85	0.81	0.60	0.80	1 1 1 1
	17689	12898	22136	0.85	0.36	0.60	0.80	1 2 1 1
	17689	12898	18145	0.85	0.24	0.60	0.80	1 3 1 1
	17689	12898	13156	0.85	0.13	0.60	0.80	1 4 1 1
	17689	12898	11236	0.85	0.08	0.60	0.80	1 5 1 1
	17689	12898	10258	0.85	0.02	0.60	0.80	1 6 1 1
	17689	12898	10258	0.85	0.01	0.60	0.80	1 7 1 1
İGİG3	17745	12951	37452	0.84	1.00	0.59	0.79	1 0 1 1
	17745	12951	25456	0.84	0.73	0.59	0.79	1 1 1 1
	17745	12951	21783	0.84	0.31	0.59	0.79	1 2 1 1
	17745	12951	17125	0.84	0.24	0.59	0.79	1 3 1 1
	17745	12951	14236	0.84	0.14	0.59	0.79	1 4 1 1
	17745	12951	11245	0.84	0.09	0.59	0.79	1 5 1 1
	17745	12951	10784	0.84	0.02	0.59	0.79	1 6 1 1
	17745	12951	10784	0.84	0.01	0.59	0.79	1 7 1 1

Çizelge 7.49 İş merkezi 2 kullanım oranı (mu2) etki alanı (1.yaklaşım) (devamı)

	F	T	C _{max}	mu1	mu2	mu3	Mu4	İş Mer. Tez. Say.
EKİZ1	17895	12894	37485	0.83	1.00	0.58	0.78	0 0 1 1
	17895	12894	24896	0.83	0.82	0.58	0.78	1 0 1 1
	17895	12894	20458	0.83	0.34	0.58	0.78	1 1 1 1
	17895	12894	16256	0.83	0.26	0.58	0.78	1 2 1 1
	17895	12894	12782	0.83	0.17	0.58	0.78	1 3 1 1
	17895	12894	10987	0.83	0.10	0.58	0.78	1 4 1 1
	17895	12894	10235	0.83	0.04	0.58	0.78	1 5 1 1
	17895	12894	10235	0.83	0.02	0.58	0.78	1 6 1 1
	17895	12894	10235	0.83	0.01	0.58	0.78	1 7 1 1
EKİZ2	17945	12952	37569	0.82	1.00	0.60	0.79	0 0 1 0
	17945	12952	21458	0.82	0.87	0.60	0.79	0 0 1 1
	17945	12952	17421	0.82	0.38	0.60	0.79	1 1 1 1
	17945	12952	12365	0.82	0.24	0.60	0.79	1 2 1 1
	17945	12952	11897	0.82	0.19	0.60	0.79	1 3 1 1
	17945	12952	10245	0.82	0.11	0.60	0.79	1 4 1 1
	17945	12952	10245	0.82	0.07	0.60	0.79	1 5 1 1
	17945	12952	10245	0.82	0.04	0.60	0.79	1 6 1 1
	17945	12952	10245	0.82	0.01	0.60	0.79	1 7 1 1
EKİZ3	17987	12874	37658	0.83	1.00	0.59	0.80	0 0 1 0
	17987	12874	24782	0.83	0.81	0.59	0.80	1 0 1 0
	17987	12874	20496	0.83	0.34	0.59	0.80	1 1 1 1
	17987	12874	15783	0.83	0.27	0.59	0.80	1 2 1 1
	17987	12874	12736	0.83	0.16	0.59	0.80	1 3 1 1
	17987	12874	10781	0.83	0.10	0.59	0.80	1 4 1 1
	17987	12874	10156	0.83	0.07	0.59	0.80	1 5 1 1
	17987	12874	10156	0.83	0.03	0.59	0.80	1 6 1 1
	17987	12874	10156	0.83	0.01	0.59	0.80	1 7 1 1

Çizelge 7.50 İş merkezi 2 kullanım oranı (mu2) etki alanı (2. yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	Mu4	Tez. Say.
M1	17989	12895	37456	0.85	1.00	0.58	0.78	1
	17989	12895	37456	0.85	0.01	0.58	0.78	1
M2	17887	12945	37768	0.83	1.00	0.58	0.78	1
	17887	12945	37768	0.83	0.32	0.58	0.78	2
	17887	12945	37768	0.83	0.24	0.58	0.78	3
	17887	12945	37768	0.83	0.16	0.58	0.78	4
	17887	12945	37768	0.83	0.11	0.58	0.78	5
	17887	12945	37768	0.83	0.07	0.58	0.78	6
	17887	12945	37768	0.83	0.04	0.58	0.78	7
	17887	12945	37768	0.83	0.01	0.58	0.78	8
M3	17789	12968	37689	0.83	1.00	0.59	0.79	1
	17789	12968	23145	0.83	0.01	0.59	0.79	1
M4	17778	12989	37675	0.82	1.00	0.57	0.77	1
	17778	12989	22136	0.82	0.01	0.57	0.77	1

İkinci yaklaşımdan elde edilen sonuçlara göre;

- İş merkezi tezgah sayıları en az "bir" en fazla "sekiz" adet olarak gerçekleşmiştir.
- İkinci iş merkezi kullanım oranındaki değişime sadece ikinci iş merkezi duyarlılık göstermiştir.
- Bu duyarlılık, iş merkezinde kullanılan tezgah sayıları aynı kalmak üzere, ikinci iş merkezi kullanım oranı düşürülmek istendiğinde karşılaşılabilecek darboğaz noktanın yine ikinci iş merkezinde yaşanacağını göstermektedir.
- İkinci iş merkezi kullanım oranı kriterinin etki alanı;
İş merkezi : 2

Etki alanı : %100 - %33 -> Tezgah sayısı = 1,

%32 - %25 -> Tezgah sayısı = 2,

%24 - %17 -> Tezgah sayısı = 3,

%16 - %12 -> Tezgah sayısı = 4,

%11 - % 8 -> Tezgah sayısı = 5,

% 7 - % 5 -> Tezgah sayısı = 6,

% 4 - % 2 -> Tezgah sayısı = 7,

%01 - -> Tezgah sayısı = 8.

İş merkezi : 1

Etki alanı : %100 - %01 -> Tezgah sayısı = 1.

İş merkezi : 3

Etki alanı : % 100 - %01 -> Tezgah sayısı = 1.

İş merkezi : 4

Etki alanı : % 100 - %01 -> Tezgah sayısı = 1.

- İkinci iş merkezinde sadece "bir" tezgah kullanılması ile uygun bir performans seviyesi yakalanabilecektir.

Bu sonuçlara göre, performans kriterlerinden ikinci iş merkezi kullanım oranı kriterinin tasarım sırasında göz önüne alınması gerekli bir kriterdir.

Çizelge 7.51 İş merkezi 3 kullanım oranı (mu3) etki alanı (1.yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	mu4	İş Mer. Tez. Say.
EETT1	17989	12895	37456	0.85	0.51	1.00	0.78	1 1 1 1
	17989	12895	37456	0.85	0.51	0.37	0.78	1 1 2 1
	17989	12895	37456	0.85	0.51	0.24	0.78	1 1 3 1
	17989	12895	37456	0.85	0.51	0.18	0.78	1 1 4 1
	17989	12895	37456	0.85	0.51	0.12	0.78	1 1 5 1
	17989	12895	37456	0.85	0.51	0.09	0.78	1 1 6 1
	17989	12895	37456	0.85	0.51	0.05	0.78	1 1 7 1
	17989	12895	37456	0.85	0.51	0.01	0.78	1 1 8 1
EETT2	17887	12945	37768	0.83	0.51	1.00	0.78	1 0 0 1
	17887	12945	37768	0.83	0.51	0.84	0.78	1 1 1 1
	17887	12945	37768	0.83	0.51	0.32	0.78	1 1 2 1
	17887	12945	37768	0.83	0.51	0.19	0.78	1 1 3 1
	17887	12945	37768	0.83	0.51	0.13	0.78	1 1 4 1
	17887	12945	37768	0.83	0.51	0.07	0.78	1 1 5 1
	17887	12945	37768	0.83	0.51	0.04	0.78	1 1 6 1
	17887	12945	37768	0.83	0.51	0.01	0.78	1 1 7 1
EETT3	17789	12968	37689	0.83	0.51	1.00	0.79	1 1 0 1
	17789	12968	23145	0.83	0.51	0.89	0.79	1 1 1 1
	17789	12968	17456	0.83	0.51	0.37	0.79	1 1 2 1
	17789	12968	16545	0.83	0.51	0.26	0.79	1 1 3 1
	17789	12968	13254	0.83	0.51	0.18	0.79	1 1 4 1
	17789	12968	10897	0.83	0.51	0.11	0.79	1 1 5 1
	17789	12968	10897	0.83	0.51	0.05	0.79	1 1 6 1
	17789	12968	10897	0.83	0.51	0.01	0.79	1 1 7 1
İGİG1	17778	12989	37675	0.82	0.51	1.00	0.77	1 1 0 1
	17778	12989	22136	0.82	0.51	0.87	0.77	1 1 1 1
	17778	12989	19893	0.82	0.51	0.42	0.77	1 1 2 1
	17778	12989	18452	0.82	0.51	0.29	0.77	1 1 3 1
	17778	12989	12156	0.82	0.51	0.17	0.77	1 1 4 1
	17778	12989	10897	0.82	0.51	0.09	0.77	1 1 5 1
	17778	12989	10235	0.82	0.51	0.06	0.77	1 1 6 1
	17778	12989	10235	0.82	0.51	0.03	0.77	1 1 7 1
İGİG2	17689	12898	37564	0.85	0.51	1.00	0.80	1 1 0 1
	17689	12898	24897	0.85	0.51	0.84	0.80	1 1 1 1
	17689	12898	22136	0.85	0.51	0.34	0.80	1 1 2 1
	17689	12898	18145	0.85	0.51	0.21	0.80	1 1 3 1
	17689	12898	13156	0.85	0.51	0.11	0.80	1 1 4 1
	17689	12898	11236	0.85	0.51	0.07	0.80	1 1 5 1
	17689	12898	10258	0.85	0.51	0.03	0.80	1 1 6 1
	17689	12898	10258	0.85	0.51	0.01	0.80	1 1 7 1

Çizelge 7.51 İş merkezi 3 kullanım oranı (mu3) etki alanı (1.yaklaşım) (devamı)

	F	T	C _{max}	mu1	mu2	mu3	mu4	İş Mer. Tez. Say.
İGİG3	17745	12951	37452	0.84	0.51	1.00	0.79	1 1 0 1
	17745	12951	25456	0.84	0.51	0.89	0.79	1 1 1 1
	17745	12951	21783	0.84	0.51	0.39	0.79	1 1 2 1
	17745	12951	17125	0.84	0.51	0.26	0.79	1 1 3 1
	17745	12951	14236	0.84	0.51	0.14	0.79	1 1 4 1
	17745	12951	11245	0.84	0.51	0.09	0.79	1 1 5 1
	17745	12951	10784	0.84	0.51	0.05	0.79	1 1 6 1
	17745	12951	10784	0.84	0.51	0.01	0.79	1 1 7 1
EKİZ1	17895	12894	37485	0.83	0.51	1.00	0.78	1 1 1 1
	17895	12894	24896	0.83	0.51	0.48	0.78	1 1 2 1
	17895	12894	20458	0.83	0.51	0.26	0.78	1 1 3 1
	17895	12894	16256	0.83	0.51	0.18	0.78	1 1 4 1
	17895	12894	12782	0.83	0.51	0.12	0.78	1 1 5 1
	17895	12894	10987	0.83	0.51	0.09	0.78	1 1 6 1
	17895	12894	10235	0.83	0.51	0.06	0.78	1 1 7 1
	17895	12894	10235	0.83	0.51	0.03	0.78	1 1 8 1
EKİZ2	17945	12952	37569	0.82	0.51	1.00	0.79	1 1 1 1
	17945	12952	21458	0.82	0.51	0.46	0.79	1 1 2 1
	17945	12952	17421	0.82	0.51	0.32	0.79	1 1 3 1
	17945	12952	12365	0.82	0.51	0.24	0.79	1 1 4 1
	17945	12952	11897	0.82	0.51	0.16	0.79	1 1 5 1
	17945	12952	10245	0.82	0.51	0.08	0.79	1 1 6 1
	17945	12952	10245	0.82	0.51	0.03	0.79	1 1 7 1
	17945	12952	10245	0.82	0.51	0.01	0.79	1 1 8 1
EKİZ3	17987	12874	37658	0.83	0.51	1.00	0.80	1 1 1 1
	17987	12874	24782	0.83	0.51	0.39	0.80	1 1 2 1
	17987	12874	20496	0.83	0.51	0.26	0.80	1 1 3 1
	17987	12874	15783	0.83	0.51	0.19	0.80	1 1 4 1
	17987	12874	12736	0.83	0.51	0.14	0.80	1 1 5 1
	17987	12874	10781	0.83	0.51	0.09	0.80	1 1 6 1
	17987	12874	10156	0.83	0.51	0.07	0.80	1 1 7 1
	17987	12874	10156	0.83	0.51	0.04	0.80	1 1 8 1

Çizelge 7.52 İş merkezi3 kullanım oranı (mu3) etki alanı (2.yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	mu4	Tez. Say.
M1	17989	12895	37456	0.85	0.51	1.00	0.78	1
	17989	12895	37456	0.85	0.51	0.01	0.78	1
M2	17887	12945	37768	0.83	0.51	1.00	0.78	1
	17887	12945	37768	0.83	0.51	0.01	0.78	1
M3	17789	12968	37689	0.83	0.51	1.00	0.79	1
	17789	12968	23145	0.83	0.51	0.38	0.79	2
	17789	12968	17456	0.83	0.51	0.26	0.79	3
	17789	12968	16545	0.83	0.51	0.20	0.79	4
	17789	12968	13254	0.83	0.51	0.14	0.79	5
	17789	12968	10897	0.83	0.51	0.09	0.79	6
	17789	12968	10897	0.83	0.51	0.04	0.79	7
	17789	12968	10897	0.83	0.51	0.01	0.79	8
M4	17778	12989	37675	0.82	0.51	1.00	0.77	1
	17778	12989	22136	0.82	0.51	0.01	0.77	1

Üçüncü iş merkezi kullanım oranı etki alanı için, 2. yaklaşım ile elde edilen sonuçlara göre;

- İş merkezi tezgah sayıları en az "bir", en fazla "sekiz" adet olarak gerçekleşmiştir.
- Üçüncü iş merkezi kullanım oranındaki değişime sadece üçüncü iş merkezi duyarlılık göstermiştir.
- Bu duyarlılık, iş merkezinde kullanılan tezgah sayıları aynı kalmak üzere, üçüncü iş merkezi kullanım oranı düşürülmek istendiğinde karşılaşılabilecek darboğaz noktanın yine üçüncü iş merkezinde yaşanacağını göstermektedir.

- Üçüncü iş merkezi kullanım oranı kriterinin etki alanı;

İş merkezi : 3

Etki alanı : %100 - %39 -> Tezgah sayısı = 1,

%38 - %27 -> Tezgah sayısı = 2,

%26 - %21 -> Tezgah sayısı = 3,

%20 - %15 -> Tezgah sayısı = 4,

%14 - %10 -> Tezgah sayısı = 5,

% 9 - % 5 -> Tezgah sayısı = 6,

%4 - %2 -> Tezgah sayısı = 7,

%1 - -> Tezgah sayısı = 8,

İş merkezi : 1

Etki alanı : %100 - %01 -> Tezgah sayısı - 1.

İş merkezi : 2

Etki alanı : % 100 - %01 -> Tezgah sayısı = 1.

İş merkezi : 4

Etki alanı : %100 - %01 -> Tezgah sayısı - 1.

- Üçüncü iş merkezinde sadece "bir" tezgah kullanılması ile uygun bir performans seviyesi yakalanabilecektir.

Bu sonuçlara göre performans kriterlerinden üçüncü iş merkezi kullanım oranı kriterinin tasarım sırasında göz önüne alınması gerekli bir kriterdir.

Çizelge 7.53 İş merkezi 4 kullanım oranı (mu4) etki alanı (1.yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	mu4	İş Mer. Tez. Say.
EETT1	17989	12895	37456	0.85	0.51	0.58	1.00	1 1 1 1
	17989	12895	37456	0.85	0.51	0.58	0.45	1 1 1 2
	17989	12895	37456	0.85	0.51	0.58	0.29	1 1 1 3
	17989	12895	37456	0.85	0.51	0.58	0.16	1 1 1 4
	17989	12895	37456	0.85	0.51	0.58	0.11	1 1 1 5
	17989	12895	37456	0.85	0.51	0.58	0.09	1 1 1 6
	17989	12895	37456	0.85	0.51	0.58	0.05	1 1 1 7
	17989	12895	37456	0.85	0.51	0.58	0.01	1 1 1 8
EETT2	17887	12945	37768	0.83	0.51	0.58	1.00	1 1 1 1
	17887	12945	37768	0.83	0.51	0.58	0.48	1 1 1 2
	17887	12945	37768	0.83	0.51	0.58	0.31	1 1 1 3
	17887	12945	37768	0.83	0.51	0.58	0.20	1 1 1 4
	17887	12945	37768	0.83	0.51	0.58	0.12	1 1 1 5
	17887	12945	37768	0.83	0.51	0.58	0.08	1 1 1 6
	17887	12945	37768	0.83	0.51	0.58	0.04	1 1 1 7
	17887	12945	37768	0.83	0.51	0.58	0.01	1 1 1 8
EETT3	17789	12968	37689	0.83	0.51	0.58	1.00	1 1 1 1
	17789	12968	23145	0.83	0.51	0.58	0.42	1 1 1 2
	17789	12968	17456	0.83	0.51	0.58	0.27	1 1 1 3
	17789	12968	16545	0.83	0.51	0.58	0.19	1 1 1 4
	17789	12968	13254	0.83	0.51	0.58	0.11	1 1 1 5
	17789	12968	10897	0.83	0.51	0.58	0.08	1 1 1 6
	17789	12968	10897	0.83	0.51	0.58	0.05	1 1 1 7
	17789	12968	10897	0.83	0.51	0.58	0.01	1 1 1 8
İĞİG1	17778	12989	37675	0.82	0.51	0.58	1.00	1 1 1 1
	17778	12989	22136	0.82	0.51	0.58	0.49	1 1 1 2
	17778	12989	19893	0.82	0.51	0.58	0.32	1 1 1 3
	17778	12989	18452	0.82	0.51	0.58	0.21	1 1 1 4
	17778	12989	12156	0.82	0.51	0.58	0.14	1 1 1 5
	17778	12989	10897	0.82	0.51	0.58	0.09	1 1 1 6
	17778	12989	10235	0.82	0.51	0.58	0.04	1 1 1 7
	17778	12989	10235	0.82	0.51	0.58	0.01	1 1 1 8
İĞİG2	17689	12898	37564	0.85	0.51	0.58	1.00	1 1 1 1
	17689	12898	24897	0.85	0.51	0.58	0.47	1 1 1 2
	17689	12898	22136	0.85	0.51	0.58	0.31	1 1 1 3
	17689	12898	18145	0.85	0.51	0.58	0.24	1 1 1 4
	17689	12898	13156	0.85	0.51	0.58	0.16	1 1 1 5
	17689	12898	11236	0.85	0.51	0.58	0.09	1 1 1 6
	17689	12898	10258	0.85	0.51	0.58	0.04	1 1 1 7
	17689	12898	10258	0.85	0.51	0.58	0.01	1 1 1 8
İĞİG3	17745	12951	37452	0.84	0.51	0.58	1.00	1 1 1 1
	17745	12951	25456	0.84	0.51	0.58	0.51	1 1 1 2
	17745	12951	21783	0.84	0.51	0.58	0.34	1 1 1 3
	17745	12951	17125	0.84	0.51	0.58	0.22	1 1 1 4
	17745	12951	14236	0.84	0.51	0.58	0.13	1 1 1 5
	17745	12951	11245	0.84	0.51	0.58	0.09	1 1 1 6
	17745	12951	10784	0.84	0.51	0.58	0.06	1 1 1 7
	17745	12951	10784	0.84	0.51	0.58	0.01	1 1 1 8

Çizelge 7.53 İş merkezi 4 kullanım oranı (mu4) etki alanı (1.yaklaşım) (devamı)

	F	T	C _{max}	mu1	mu2	mu3	mu4	İş Mer. Tez. Say.
EKİZ1	17895	12894	37485	0.83	0.51	0.58	1.00	1 1 1 1
	17895	12894	24896	0.83	0.51	0.58	0.49	1 1 1 2
	17895	12894	20458	0.83	0.51	0.58	0.29	1 1 1 3
	17895	12894	16256	0.83	0.51	0.58	0.16	1 1 1 4
	17895	12894	12782	0.83	0.51	0.58	0.11	1 1 1 5
	17895	12894	10987	0.83	0.51	0.58	0.08	1 1 1 6
	17895	12894	10235	0.83	0.51	0.58	0.06	1 1 1 7
	17895	12894	10235	0.83	0.51	0.58	0.03	1 1 1 8
	17895	12894	10235	0.83	0.51	0.58	0.01	1 1 1 8
EKİZ2	17945	12952	37569	0.82	0.51	0.58	1.00	1 1 1 1
	17945	12952	21458	0.82	0.51	0.58	0.44	1 1 1 2
	17945	12952	17421	0.82	0.51	0.58	0.29	1 1 1 3
	17945	12952	12365	0.82	0.51	0.58	0.21	1 1 1 4
	17945	12952	11897	0.82	0.51	0.58	0.14	1 1 1 5
	17945	12952	10245	0.82	0.51	0.58	0.09	1 1 1 6
	17945	12952	10245	0.82	0.51	0.58	0.04	1 1 1 7
	17945	12952	10245	0.82	0.51	0.58	0.01	1 1 1 8
EKİZ3	17987	12874	37658	0.83	0.51	0.58	1.00	1 1 1 1
	17987	12874	24782	0.83	0.51	0.58	0.50	1 1 1 2
	17987	12874	20496	0.83	0.51	0.58	0.32	1 1 1 3
	17987	12874	15783	0.83	0.51	0.58	0.21	1 1 1 4
	17987	12874	12736	0.83	0.51	0.58	0.14	1 1 1 5
	17987	12874	10781	0.83	0.51	0.58	0.10	1 1 1 6
	17987	12874	10156	0.83	0.51	0.58	0.07	1 1 1 7
	17987	12874	10156	0.83	0.51	0.58	0.04	1 1 1 8
	17987	12874	10156	0.83	0.51	0.58	0.01	1 1 1 8

Çizelge 7.54 İş merkezi 4 kullanım oranı (mu4) etki alanı (2.yaklaşım)

	F	T	C _{max}	mu1	mu2	mu3	mu4	Tez. Say.
M1	17989	12895	37456	0.85	0.51	0.58	1.00	1
	17989	12895	37456	0.85	0.51	0.58	0.01	1
M2	17887	12945	37768	0.83	0.51	0.58	1.00	1
	17887	12945	37768	0.83	0.51	0.58	0.01	1
M3	17789	12968	37689	0.83	0.51	0.58	1.00	1
	17789	12968	23145	0.83	0.51	0.58	0.01	1
M4	17778	12989	37675	0.82	0.51	0.58	1.00	1
	17778	12989	22136	0.82	0.51	0.58	0.49	2
	17778	12989	19893	0.82	0.51	0.58	0.32	3
	17778	12989	18452	0.82	0.51	0.58	0.21	4
	17778	12989	12156	0.82	0.51	0.58	0.14	5
	17778	12989	10897	0.82	0.51	0.58	0.09	6
	17778	12989	10235	0.82	0.51	0.58	0.04	7
	17778	12989	10235	0.82	0.51	0.58	0.01	8

Burada elde edilen sonuçlara göre:

- İş merkezi tezgah sayıları en az "bir", en fazla "sekiz" adet olarak gerçekleşmiştir.
- Dördüncü iş merkezi kullanım oranındaki değişime sadece dördüncü iş merkezi duyarlılık göstermiştir.
- Bu duyarlılık, iş merkezinde kullanılan tezgah sayıları aynı kalmak üzere, dördüncü iş merkezi kullanım oranı düşürülmek istendiğinde karşılaşılabilecek darboğaz noktanın yine dördüncü iş merkezinde yaşanacağını göstermektedir.
- Dördüncü iş merkezi kullanım oranı kriterinin etki alanı;
İş merkezi : 4

Etki alanı : %100 - %50 -> Tezgah sayısı = 1,

%49 - %33 -> Tezgah sayısı = 2,

%32 - %22 -> Tezgah sayısı = 3,

%21 - %15 -> Tezgah sayısı = 4,

%14 - %10 -> Tezgah sayısı = 5,

% 9 - % 5 -> Tezgah sayısı = 6,

% 4 - % 2 -> Tezgah sayısı = 7,

%01 - -> Tezgah sayısı = 8.

İş merkezi: 1

Etki alanı : % 100 - %01 -> Tezgah sayısı = 1.

İş merkezi: 2

Etki alanı : % 100 - %01 -> Tezgah sayısı = 1.

İş merkezi: 3

Etki alanı : %100 - %01 -> Tezgah sayısı = 1.

- Dördüncü iş merkezinde sadece "bir" tezgah kullanılması ile uygun bir performans seviyesi yakalanabilecektir.

Bu sonuçlara göre, performans kriterlerinden dördüncü iş merkezi kullanım oranı kriterinin tasarım sırasında göz önüne alınması gerekli bir kriterdir.

7.2.5 Deneyisel Çalışmanın Sonuçları

Bütün bu elde edilen sonuçlara göre, incelenen sistem için söylenebilecek olan, çok genel olarak:

1. Sistemin performans kriterlerine karşı duyarlı olduğu (hangi performans kriteri hangi değer aralığında kalırsa sistem konfigürasyonu değişmeden kalabilir) alanlar yukarıdaki gibidir.
2. Performans kriterinin istenen değerlerini karşılamada sırası ile sistem konfigürasyonunun hangi elemanı etkilidir, darboğaz olabilecek noktalar neresidir? Etki analizi ile belirlenebilir.
3. İncelenen model için "Ortalama Akış Süresi" kriteri sistem tasarımında tek başına etkili bir kriter değildir.
4. Aynı bu kriter gibi "Ortalama Gecikme Süresi" kriteri de tek başına etkili bir kriter değildir.
5. "Maksimum Tamamlanma Süresi" kriterinin sistem tasarımında tek başına etkisinden söz edilebilir. Ama bu çok fazla bir etki değildir.
6. Sistem tasarımında tek başına en etkili faktör "İş Merkezi Kullanım Oranı"dır.
7. Bu model için, en uygun performans değerleri sonuçlarını elde edebilecek konfigürasyon tasarımı, uygulanacak çizelgeleme stratejisi ve kullanılacak OGA sayısından bağımsız olarak, iş merkezlerinde kullanılacak tezgah sayısının en az "bir" en çok "iki" adet olduğu durumlarda aranmalıdır. İş merkezlerinde ikiden fazla tezgah kullanılması halinde, belki istenen ama optimum olmayan sonuçlar elde edilecektir.

8. SONUÇ VE DEĞERLENDİRME

Bu tezde, işletme hedeflerine ulaşmak için, bir atölye çizelgeleme sisteminin iş merkezlerinde kullanılması gerekli olan tezgah sayısının yine sistemde kullanılacak çizelgeleme stratejisi ile birlikte belirlenmesi hedeflenmiştir. Bunun için önce bütün çizelgeleme stratejisi alternatiflerinin içerisinde bir grup belirlenerek bilgisayar benzetimi ile istenen performans kriterlerinin gerçekleşen değerleri elde edilmiştir. Benzetim sonuçları yapay sinir ağına kullanılmak üzere eğitime ve test örnek setleri oluşturulmuştur. Burada iki ayrı yaklaşım ele alınmıştır. Birinci yaklaşımda, öncelik kuralı – OGA sayısı kombinasyonlarına göre oluşturulmuştur. Performans kriterlerinin hedeflenen değerlerini temsil eden yedi adet girdi ile her iş merkezinde kullanılması gerekli olan tezgah sayısını temsil eden dört adet çıktı yapısına sahip olan dokuz ayrı yapay sinir ağı kullanılmıştır. İkinci yaklaşım ise, iş merkezine dayalı yaklaşım olarak isimlendirilmiştir. Bu yaklaşımda ise, yine performans kriterlerinin hedeflenen değerlerini temsil eden yedi adet girdi ile bir adet çıktı yapısına sahip olan dört ayrı yapay sinir ağı kullanılmıştır. Bu yaklaşımın istenen performans kriterleri değerlerine göre, bütün alternatif konfigürasyon örneklerini içeren örnek seti ile, en uygun sonucu tek bir ifade ile vereceğinden, kullanılması, anlaşılması, değerlendirilmesi ve en uygun sonucu bulabilmesi açılarından daha iyi bir çözüm sunmaktadır.

Yapay sinir ağı, istenen performans kriterlerini gerçekleştirebilmek için gerekli olan tezgah sayılarını üretir. Bu değerler ile öncelik kuralı – OGA sayısı kombinasyonlarına göre bilgisayar benzetimi yapılarak sonuçlar karar tabloları halinde sunulmuştur. Karar, istenen bir performans kriteri değerinin hedeflenen değerden sapma değeri minimum olan seçenek olabileceği gibi, istenen bütün performans kriterlerinin hedeflenen değerlerinden sapma ortalaması minimum olan seçenek de olabilir. Karar verici amacını gerçekleştirebilmek için gerekli olan konfigürasyonu bulmakta zorlanmayacaktır.

Sistemin genel davranışı hakkında bilgi edinmek için bir etki analizi sunulmuştur. Yapay sinir ağı kullanılarak yapılan bu analiz ile sistemin davranışını genel olarak incelemek ya da verilen çizelgeleme stratejisi kararının hangi performans kriterine ve hangi değer aralığında duyarlı olduğunu belirlemek mümkündür.

Benzetim veya uzman sistem – benzetim ya da diğer teknikler kullanılarak yapılan tasarım proseslerinde, amaçlara ulaşılıncaya kadar işlemler tekrar edilir. Bu işlemler sırasında ise, iyi bir uzmana veya uzmanlık bilgisine gereksinim duyulmaktadır. Bu tezde sunulan atölye

çizelgeleme tasarımında, karar vermek için defalarca işlem yapmaya veya konunun iyi bir uzmanı olmaya gerek yoktur. Tüm işlemler kolaylıkla uygulanabilir ve sonuçları açıklanabilir.

Bu tezde “Etki Analizi” yaklaşımı kullanılmıştır. Diğer yapay sinir ağı çalışmalarında üretilen sonuç tek bir değer olup, noktasal bir ifadedir. Ama yapay sinir ağı, aynı çözümü başka değerler için de vermektedir. Dolayısıyla diğer çalışmalarda üretilen sonuçlar, çözüm kümesi içindeki bir noktayı çözüm değeri olarak ortaya koymaktadır. Burada ise, üretilen sonucun hangi değerler arasında kalabileceği etki analizi ile ortaya koyulmuştur. Bu analiz ile sadece çözüm kümesine ait değerler bulunmayıp, darboğaz noktaların neresi olduğunun ya da olabileceğinin belirlenmesi sonucu da üretilmektedir.

Bu çalışma, atölye çizelgeleme tasarımının nasıl olması gerektiği konusunda karar verme sırasında gerekli olan verileri ortaya koyarak, adeta sistemin bir haritasını çıkararak karar vericiye büyük ölçüde yardım eder. Karar vericinin çok defa tekrarlarla veya farklı işlemlere, başka bilgilere olan ihtiyacını en aza indirerek en uygun kararı alabilmesini sağlayacaktır.

Bu çalışma bir atölye çizelgelemede, planlama, çizelgeleme ve analiz etme olarak kullanılan, benzetim – yapay sinir ağı temelli bir çalışmadır. Bu uygulama, kullanıcıya iş akış zamanı, gecikmeler, maksimum tamamlanma zamanı ve kaynak kullanımının hedeflenen değerlerine göre iş merkezlerinde bulunan tezgah sayıları ve sistemdeki taşıyıcı sayısı konfigürasyonunu yine sistemde kullanılacak öncelik kuralı ile birlikte değerlendirerek en uygun konfigürasyon tasarımını bulabilmesini sağlar. Bir benzetim – yapay sinir ağı yoluyla gözlem ve değerlendirmeyi, gecikmeler, darboğazlar, yüksek akış zamanına götüren durumları görebilmeyi mümkün kılar. Kullanıcı, fabrika modellerinin ve çizelgeleme politikalarının farklı durumlarını oluşturma ve analiz etme kabiliyetine sahip olur.

Gelecekte yapılacak çalışmalar için; değişik örnek setleri ve farklı ağ yapısı düzenleme, karar verme işlemini daha kolay ve daha uygun kararlar alınabilir hale getirebilecektir. Bundan başka karar verme ve etki analizi sırasında, değerlendirme kriterlerinin tekli etkisi dışında, çoklu etkileri de araştırılarak daha farklı bir karar verme tablosu ortaya koyulabilir. Bununla birlikte, yapay sinir ağı üzerine yapılacak başka çalışmalarda noktasal değer yerine, bir değer aralığının bulunması veya bazı eski çalışmaların yeniden değerlendirilmesi yapılabilir. Burada sonuç bir değer aralığı şeklinde olacağından, etki analizi sonuçları başka teknikler için (örneğin; bulanık mantık, bulanık mantık – yapay sinir ağı teknikleri gibi) girdi kabul edilerek farklı çıkarımlar elde edilebilir. Bu şekilde sonuca ulaşmak karar verme işleminin daha kolay, daha etkili ve daha uygun olmasını sağlayabilecektir.

KAYNAKLAR

- Arizono, I., Yamamoto, A. ve Ohta, H., (1992), "Scheduling for Minimizing Total Actual Flow Time by Neural Networks", *International Journal of Production Research*, 30: 503-511.
- Askin, R. G. ve Standridge, C. R., (1993), *Modelling and Analysis Manufacturing Systems*, John Wiley & Sons, New York.
- Baker, K. R., (1974), *Introduction to Sequencing and Scheduling*, John Wiley&Sons, New York
- Baker, K. R., (1984), "Sequencing Rules and Due Date Assignments in a Job shop", *Management Science*, 30:1093 – 1104.
- Baker, K. R., (1994), *Elements of Sequencing and Scheduling*, Amos Tuck School of Business, D. Hannover, NH.
- Barrow, H., (1989), *AI, Neural Networks and Early Vision*, AISB Newsletter.
- Başıgil, H., (1988), "Sıralama ve Çizelgeleme", Yıldız Teknik Üniversitesi, İstanbul.
- Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G. ve Weglarz, J., (1996), *Scheduling Computer and Manufacturing Processes*, Springer, Berlin.
- Carbonel, J., (1987), *Machine Learning*, Encyclopedia of Artificial Intelligence, Wiley Interscience.
- Chang, Y. C., and Jeng, M. D., (1995), "A Neural Network Model for the Job-Shop Scheduling Problem with the Consideration of Lot sizes", *IEEE International Conference on Robotics and Automation*.
- Cheung, J. Y., (1994), *Scheduling, Artificial Neural Networks for Intelligent Manufacturing*, edited by Cihan Dagli, Chapman&Hall, London.
- Crooks, T., (1992), "Care and Fedinng of Neural Networks", *AI Expert*, 7 (7):37-41.
- Çakar, T., (1997), *İmalat Sistemlerinin Tasarlaması Ve Öncelik Kurallarının Belirlenmesinde Yapay Sinir Ağlarının Kullanılması*, Doktora Tezi, İstanbul.
- Dağlı, C., (1994), *Artificial Neural Networks for Intelligence Manufacturing*, Chapman&Hall, London.
- Davis, L., (1994), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.
- Dinçmen, M., Demirci, E., (1990), "Atölye Tipi Üretimin Etkileşimli Benzetimi", *Endüstri Mühendisliği Dergisi*, 2(6): 3-10.
- Feng, S., Li, L., Ling Cen L. ve Huang, J., (2003), "Using MLP networks to design a production scheduling system", *Computers & Operations Research*, 30 (6): 821-832.
- Fonseca, D. J. ve Navarrese, D., (2002), "Artificial neural networks for job shop simulation", *Advanced Engineering Informatics*, 16 (2): 241-246.
- Foo, S. Y., Takefuji, Y., Szu, H., (1995), "Scaling Properties of Neural Networks for Job Shop Scheduling", *Neurocomputing*, 8: 79 - 91.
- Forsyth, R. ve Naylor, C., (1985), *Guide to Artificial Intelligence*, Chapman&Hall, New York

- French, S., (1982), Sequencing and Scheduling - An Intriduction to the Mathematics of the Job-Shop University of Manchester, Manchester.
- Haddock, J. ve O'Keefe, R. M., (1990), "Using Artificial Intelligence to Facilitate Manufacturing System Simulation", Computers and Industrial Engineering, 8 (3):275-283.
- Halaç, O., (1998), İşletmelerde simülasyon Teknikleri, Alfa basım dağıtım, İstanbul.
- Haykin, S., (1994), Neural Networks, Maxwell-McMillan, Ontario-Canada.
- Holland, J. H., (1975), Adaptation in Natural and Artificial Systems, An Abour: Unv. Mich. Press.
- Hopfield, J. J. ve Tank, D. W., (1985), "Neural Computation of Decision in Optimization Problems", Biological Cybernetics, 52:141-152.
- Kocabaş, Ş., (1991), "A Review of Learning, the knowledge Engineering Review", 6(3):195-122.
- Lakhmi, C. J. ve Vemuri, V. R., (1999), Industrial Applications of Neural Networks, CRC press, Boca Raton, Florida.
- Langley, P. ve Simon, H. A., (1995), "Applications of Machine Learning and Rule Induction", Communications of the ACM, 38(11):55-64.
- Law, A. M., Kelton, W. D., (1992), Simulation Modeling and Analysis, İkinci Basım, McGraw-Hill, New York.
- Law, M. A. ve Kelton, W. D., (1991), Simulation Modelling and Analysis, Mc Graw-Hill, London.
- Lenat, D., (1983), "Eurisko: A Program that Learns New Heuristics and Domain Concepts", Artificial Intelligence, 21 (2): 61-98.
- Lin, C. L. ve Lee, C. S. G., (1995), Neural Fuzzy Systems, Prentice Hall, New Jersey.
- Michalski, R. S., (1986), "Understanding the Nature of Learning", Machine Learning, 1(1):20-29.
- Mitchell, T. ve Keller, R., (1986), "Explanation Based generalizatio: A Unify View", Machine Learning, 1(1): 8-15.
- Morton, T. E. ve Pentico, D. W., (1993), Heuristic Scheduling Systems: With Applications to Production Systems and Project Mnagement, Wiley, New York.
- Ovacık, I. M. ve Uzsoy, R., (1997), Decomposition Methods Forcomplex Factory Scheduling Problems, Kluwer Academic Publishers, Norwell, Massachusetts.
- Özkazanç, A. Ü., (1999), Atölye Tipi Üretim Ortamında İşlerin Çizelgelenmesi İçin Yapay Sinirsel Ağ Yaklaşımı, Yüksek Lisans Tezi, Osmangazi Üniversitesi, Eskişehir.
- Öztemel, E., (1996), "Yapay Sinir Ağlarına Dayalı Deneysel Tasarım Metodu ve Endüstriyel Bir Uygulama", Doğa, Çevre ve Mühendislik Dergisi, 20(2):73-78.
- Pham, D. T. ve Aksoy, M. S., (1994), "Rules: Simple Rule Extraction System", Expert System with Applications, 8 (1): 59-65.
- Pinedo, M., (1995), Scheduling: Theory, Algorithms and Systems, Prentice Hall, New York.

- Porter, B. ve Kibler, D., (1986), "A Method for Learning Problem Solving Heuristics", *Machine Learning*, 1 (3): 10-16.
- Rich, E. ve Knight, K., (1991), *Artificial Intelligence*, McGraw-Hill, Singapur
- Sabuncuoğlu, I., (1998), "Scheduling with Neural Networks: A Review of the Literature and New Research Directions", *Production Planning and Control*, 9 (1): 2-12.
- Sabuncuoğlu, İ., Gurgun, B., (1996), "A Neural Network Model for Scheduling Problems", *European Journal of Operational Research*, 93 (2): 288 - 299.
- Schalkoff, R. J., (1997), *Artificial Neural Networks*, McGraw-Hill, Singapur.
- Sim, S. K., Yeo, K. T. ve Lee, W. H., (1994), "An Expert Neural Network System for Dynamic Job Shop Scheduling", *International Journal of Production Research*, 32 (8):1759-1773.
- Simon, H., (1983), "Search and Reasoning in Problem Solving", *Artificial Intelligence*, 21(2):7 – 30.
- Taşgetiren, M. F., Cedimoğlu, İ. H. ve Öztemel, E., "(1995), "Sinir Ağı Çizelgeleme Sistemi. Uzman Çizelgeleme Sisteminden esinlenen Bir Yaklaşım", *YA/EM'95*, ODTÜ, Ankara.
- Thompson, M. B., (1994), "Expanding Simulation Beyond Planning and Design", *Industrial Engineering*, 26 (8), 43-54.
- Tsoukalas, L. H. ve Uhrig, R. E., (1997), *Fuzzy and Neural Approaches in Engineering*, John Wiley & Sons, New York.
- Wu, B., (1992), *Manufacturing System Design and Analysis*, Chapman & Hall, New York.
- Yang, S. ve Wang D., (2000), "Constraint Satisfaction Adaptive Neural Network and Heuristics Combined Approaches for Generalized Job-Shop Scheduling", *IEEE Transactions on Neural Networks*, 11 (2): 474 – 486.
- Yu, H. ve Liang, W., (2001), "Neural Network and Genetic Algorithm-Based Hybrid Approach To Expanded Job-Shop Scheduling", *Computers & Industrial Engineering*, 39 (3-4): 337-356.
- Zurada, J. M., (1992), *Introduction Artificial Neural Systems*, West Publishing Company, St. Paul.

EKLER

- Ek 1 Siman Model Programı
Ek 2 Siman deneysel programı



Ek 1 Siman model programı

```

BEGIN;
;! *****
;! *****   GEZEN BIRIM CEKME MODULU   *****
;! *****
H1   QUEUE,21;
      SCAN:x(11).LT.6.AND.NQ(31).GT.0.and.nq(34).eq.0;
      SEARCH,31,1,NQ:MIN(A(9));
      REMOVE:J,31,st5;
      QUEUE,22;
      WAIT:10:NEXT(H1);
;
H2   QUEUE,23;
      SCAN:x(12).LT.6.AND.NQ(32).GT.0.and.nq(35).eq.0;
      SEARCH,32,1,NQ:MIN(A(9));
      REMOVE:J,32,st5;
      QUEUE,24;
      WAIT:20:NEXT(H2);
;
H3   QUEUE,25;
      SCAN:x(13).LT.6.AND.NQ(33).GT.0.and.nq(36).eq.0;
      SEARCH,33,1,NQ:MIN(A(9));
      REMOVE:J,33,st5;
      queue,26;
      wait:30:NEXT(H3);
;*****
ST5   STATION,5;
      QUEUE,13;
      SEIZE:ROBOT;
      DELAY:1;
      RELEASE:ROBOT;
      BRANCH,1:
        IF,A(5).EQ.1,RAPOR:
        ELSE,DEVAM;
DEVAM  tally:7,bet(49);
      queue,27;
      request:AVG(sds,6);
      ASSIGN:M=A(4);
      TRANSPORT:AVG(A(6)),M;
;
;! *****
;! *****   ATELYE MODULU   *****
;! *****
;
MC1   STATION,1;
      FREE:AVG(A(6));
      assign:x(11)=x(11)+1;
      SIGNAL:10;
      QUEUE,1;
      SCAN:X(1).LT.3;

```

```

    ASSIGN:X(1)=X(1)+1;
    DELAY:A(2);
    QUEUE,5;
    SEIZE:MAK1;
    DELAY:A(3);
    RELEASE:MAK1;
    ASSIGN:X(1)=X(1)-1;
    ASSIGN:X(11)=X(11)-1;
    BRANCH,1:
        IF,A(1).eq.1,At1:
        IF,A(1).eq.2,At2:
        IF,A(1).eq.5,At5;
at1  ASSIGN:A(2)=rn(7,1);
    ASSIGN:A(3)=rn(20,1)*10;
    BRANCH,1:
        IF,X(12).LT.6,ATT1:
        ELSE,ATT11;
ATT1  ASSIGN:A(4)=3:next(gmak1);
ATT11 assign:a(8)=3;
    ASSIGN:A(4)=6:next(gmak1);
at2  ASSIGN:A(2)=rn(4,1);
    ASSIGN:A(3)=rn(17,1)*10;
    BRANCH,1:
        IF,X(13).LT.6,ATT2:
        ELSE,ATT21;
ATT2  ASSIGN:A(4)=2:next(gmak1);
ATT21 assign:a(8)=2;
    ASSIGN:A(4)=6:next(gmak1);
at5  ASSIGN:A(2)=rn(13,1);
    ASSIGN:A(3)=rn(26,1)*10;
    BRANCH,1:
        IF,X(13).LT.6,ATT5:
        ELSE,ATT51;
ATT5  ASSIGN:A(4)=4:next(gmak1);
ATT51 assign:a(8)=4;
    ASSIGN:A(4)=6:next(gmak1);
;
GMAK1 QUEUE,9;
    REQUEST:AVG(SDS,6);
    ASSIGN:M=A(4);
    TRANSPORT:AVG(A(6)),M;
;-----
MC2  STATION,2;
    FREE:AVG(A(6));
    assign:x(12)=x(12)+1;
    SIGNAL:20;
    assign:x(5)=0;
    QUEUE,2;
    SCAN:X(2).LT.5;
    ASSIGN:X(2)=X(2)+1;
    DELAY:A(2);

```

```

    QUEUE,6;
    SEIZE:MAK2;
    DELAY:A(3);
    RELEASE:MAK2;
    ASSIGN:X(2)=X(2)-1;
    ASSIGN:X(12)=X(12)-1;
    BRANCH,1:
        IF,A(1).eq.2,AAt2:
        IF,A(1).eq.3,AAt3:
        IF,A(1).eq.4,AAt4;
Aat2  ASSIGN:A(2)=rn(10,1);
      ASSIGN:A(3)=rn(23,1)*10;
      BRANCH,1:
          IF,X(14).LT.6,AATT2:
          ELSE,AATT21;
AATT2 ASSIGN:A(4)=4:next(gmak2);
AATT21 assign:a(8)=4;
      ASSIGN:A(4)=6:next(gmak2);
Aat3  ASSIGN:A(2)=rn(11,1);
      ASSIGN:A(3)=rn(24,1)*10;
      BRANCH,1:
          IF,X(14).LT.6,AATT3:
          ELSE,AATT31;
AATT3 ASSIGN:A(4)=4:next(gmak2);
AATT31 assign:a(8)=4;
      ASSIGN:A(4)=6:next(gmak2);
Aat4  ASSIGN:A(2)=rn(8,1);
      ASSIGN:A(3)=rn(21,1)*10;
      BRANCH,1:
          IF,X(13).LT.6,AATT4:
          ELSE,AATT41;
AATT4 ASSIGN:A(4)=3:next(gmak2);
AATT41 assign:a(8)=3;
      ASSIGN:A(4)=6:next(gmak2);
GMAK2 QUEUE,10;
      REQUEST:AVG(SDS,6);
      ASSIGN:M=A(4);
      TRANSPORT:AVG(A(6)),M;
;-----
MC3  STATION,3;
      FREE:AVG(A(6));
      assign:x(13)=x(13)+1;
      SIGNAL:30;
      assign:x(6)=0;
      QUEUE,3;
      SCAN:X(3).LT.5;
      ASSIGN:X(3)=X(3)+1;
      DELAY:A(2);
      QUEUE,7;
      SEIZE:MAK3;
      DELAY:A(3);

```

```

RELEASE:MAK3;
ASSIGN:X(3)=X(3)-1;
ASSIGN:X(13)=X(13)-1;
BRANCH,1:
  IF,A(1).eq.1,AAAt1:
  IF,A(1).eq.4,AAAt4;
AAAt1  ASSIGN:A(2)=rn(9,1);
        ASSIGN:A(3)=rn(22,1)*10;
        BRANCH,1:
          IF,X(14).LT.6,AAATT1:
          ELSE,AAATT11;
AAATT1  ASSIGN:A(4)=4:next(gmak3);
AAATT11 assign:a(8)=4;
        ASSIGN:A(4)=6:next(gmak3);
AaAt4  ASSIGN:A(2)=rn(12,1);
        ASSIGN:A(3)=rn(25,1)*10;
        BRANCH,1:
          IF,X(14).LT.6,AAATT4:
          ELSE,AAATT41;
AAATT4  ASSIGN:A(4)=4:next(gmak3);
AAATT41 assign:a(8)=4;
        ASSIGN:A(4)=6:next(gmak3);
GMAK3  QUEUE,11;
        REQUEST:AVG(SDS,6);
        ASSIGN:M=A(4);
        TRANSPORT:AVG(A(6)),M;
;-----
MC4    STATION,4;
        FREE:AVG(A(6));
        assign:x(14)=x(14)+1;
        assign:x(7)=0;
        QUEUE,4;
        SCAN:X(4).LT.4;
        ASSIGN:X(4)=X(4)+1;
        DELAY:A(2);
        QUEUE,8;
        SEIZE:MAK4;
        DELAY:A(3);
        RELEASE:MAK4;
        ASSIGN:X(4)=X(4)-1;
        ASSIGN:X(14)=X(14)-1;
        ASSIGN:A(5)=1;
        QUEUE,12;
        REQUEST:AVG(SDS,6);
        TRANSPORT:AVG(A(6)),5;
;-----
ST6    STATION,6;
        FREE:AVG(A(6));
        BRANCH,1:
          IF,A(8).EQ.2,B2:
          IF,A(8).EQ.3,B3:

```

```

      IF,A(8).EQ.4,B4;
B2   QUEUE,34;
      SCAN:X(12).LT.6.and.x(5).eq.0;
      assign:x(5)=1;
      ASSIGN:A(4)=2:NEXT(GG);
B3   QUEUE,35;
      SCAN:X(13).LT.6.and.x(6).eq.0;
      assign:x(6)=1;
      ASSIGN:A(4)=3:NEXT(GG);
B4   QUEUE,36;
      SCAN:X(14).LT.6.and.x(7).eq.0;
      assign:x(7)=1;
      ASSIGN:A(4)=4:NEXT(GG);
GG   QUEUE,37;
      REQUEST:AVG(SDS,6);
      ASSIGN:M=A(4);
      TRANSPORT:AVG(A(6)),M;
;-----
rapor  FREE:AVG(A(6));
SS    assign:x(20)=tnow-a(9);
      branch,1:
      if,x(20).le.0,ss1:
      else,ss2;
ss1   assign:x(21)=0:next(ss3);
ss2   count:2,1;
      assign:x(21)=x(20);
ss3   TALLY:A(1),INT(7);
      TALLY:6,INT(7);
      TALLY:8,BET(50);
      COUNT:1,1:dispose;
;
;.! *****
;.! *** GEZEN B~R~M OLUSTURMA VE ATAMA MODULU ***
;.! *****
      CREATE:EX(27,3);
      ASSIGN:A(1)=DP(28,2);
      assign:a(9)=tnow+5*rn(29,4);
      assign:x(25)=x(25)+1;
      branch,1:
      if,x(25).gt.500,ipt:
      else,devam1;
ipt   assign:a(7)=0:dispose;
devam1  BRANCH, 1:
      IF,A(1).EQ.1,A1:
      IF,A(1).EQ.2,A2:
      IF,A(1).EQ.3,A3:
      IF,A(1).EQ.4,A4:
      IF,A(1).EQ.5,A5;
A1    ASSIGN:A(2)=RN(1,1);
      ASSIGN:A(3)=RN(14,1)*10;
      ASSIGN:A(4)=1:NEXT(TOP1);

```

```
A2  ASSIGN:A(2)=RN(2,1);
    ASSIGN:A(3)=RN(15,1)*10;
    ASSIGN:A(4)=1:NEXT(TOP1);
A3  ASSIGN:A(2)=RN(5,1);
    ASSIGN:A(3)=RN(18,1)*10;
    ASSIGN:A(4)=2:NEXT(TOP2);
A4  ASSIGN:A(2)=RN(6,1);
    ASSIGN:A(3)=RN(19,1)*10;
    ASSIGN:A(4)=2:NEXT(TOP2);
A5  ASSIGN:A(2)=RN(3,1);
    ASSIGN:A(3)=RN(16,1)*10;
    ASSIGN:A(4)=1:NEXT(TOP1);
;
TOP1 QUEUE,31:DETACH;
TOP2 QUEUE,32:DETACH;
TOP3 QUEUE,33:DETACH;
;
END;
```



Ek 2 Siman deneysel programı

```

BEGIN;
;
PROJECT, mak, Gökay Akkaya, 05/ 08/ 2003;
;
DISCRETE,800,10,100,10;
;
REPLICATE,1,0,40320;
;
ARRIVALS:1,QUEUE(21),0,1,:
      2,QUEUE(23),0,1,:
      3,QUEUE(25),0,1,;
;
RESOURCES:1,MAK1,3:
      2,MAK2,5:
      3,MAK3,5:
      4,MAK4,4:
      5,ROBOT,1;
;
COUNTERS:1,T.P.SAYISI,500:
      2,gec is say;
;
PARAMETERS:1,8,2:
      2,7,1:
      3,6,2:
      4,4,2:
      5,4,2:
      6,3,2:
      7,7,2:
      8,6,2:
      9,3,1:
      10,3,1:
      11,4,2:
      12,3,1:
      13,3,1:
      14,12,3:
      15,11,2:
      16,12,2:
      17,7,2:
      18,8,1:
      19,7,3:
      20,13,3:
      21,10,2:
      22,7,2:
      23,5,2:
      24,7,1:
      25,6,2:
      26,6,1:
      27,10:
      28,.2,1,.4,2,.6,3,.8,4,1,5:

```

```

29,320,50;
;
DSTAT:1,NR(1),MC1 KUL:
2,NR(2),MC2 KUL:
3,NR(3),MC3 KUL:
4,NR(4),MC4 KUL:
5,NR(5),ROBOT KUL:
6,NT(1),AVG KUL:
7,NQ(1),MC1 KUY:
8,NQ(2),MC2 KUY:
9,NQ(3),MC3 KUY:
10,NQ(4),MC4 KUY:
11,NQ(34),M2 ARA BEK:
12,NQ(35),M3 ARA BEK:
13,NQ(36),M4 ARA BEK:
14,NQ(31),M1 BEK:
15,NQ(32),M2 BEK:
16,NQ(33),M3 BEK:
17,x(21),ort gec;
;
TALLIES:1,PARCA1 AKIŞ Z.:
2,PARCA2 AKIŞ Z.:
3,PARCA3 AKIŞ Z.:
4,PARCA4 AKIŞ Z.:
5,PARCA5 AKIŞ Z.:
6,TOPLAM AKIŞ Z.:
7,GİRİŞ AR SURE:
8,ÇIKIŞ AR SURE;
;
TRANSPORTERS:1,AVG,3,1,10,1-a;
;
DISTANCES:1,1-6,10,20,30,10,20/
10,20,20,15/
10,30,15/
40,15/
10;
END;

```

ÖZGEÇMİŞ

Doğum tarihi 06.01.1968

Doğum yeri Samsun

Lise 1982-1985 İzmir Balçova Lisesi

Lisans 1986-1991 İstanbul Teknik Üniversitesi
Sakarya Mühendislik Fakültesi.
Endüstri Mühendisliği Bölümü

Yüksek Lisans 1995-1997 Sakarya Üniversitesi Fen Bilimleri Enstitüsü
Endüstri Müh. Anabilim Dalı

Doktora 1999- 2004 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Endüstri Müh. Anabilim Dalı

Çalıştığı kurum(lar)

1993-1995 Er-Pa Şeker Fabrikası - İzmir

1995-1999 Atatürk Üniversitesi Mühendislik Fak
Endüstri Müh. Bölümü Araştırma Görevlisi
YTÜ Makine Fakültesi

1999- Endüstri Müh. Bölümü, Araştırma Görevlisi