

T.C.  
SELÇUK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

96287

HÜCRESEL ÜRETİM SİSTEMİ TASARIMINDA KULLANILAN  
YAPAY ZEKA TEKNİKLERİ İLE SEZGİSEL YÖNTEMLERİN  
KARŞILAŞTIRILMASI VE UYGULAMALI ANALİZLERİ



T.C. YÜKSEKÖĞRETİM KURULU  
DOKÜMANTASYON MERKEZİ

Yakup KARA

Yüksek Lisans Tezi

Endüstri Mühendisliği Anabilim Dalı

96287

KONYA, 2000

## ÖZET

Yüksek Lisans Tezi

# HÜCRESEL ÜRETİM SİSTEMİ TASARIMINDA KULLANILAN YAPAY ZEKA TEKNİKLERİ İLE SEZGİSEL YÖNTEMLERİN KARŞILAŞTIRILMASI VE UYGULAMALI ANALİZLERİ

Yakup KARA

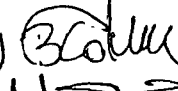
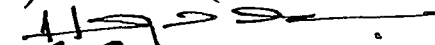
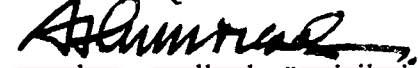
Selçuk Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman : Doç. Dr. Bilal TOKLU

2000,

Jüri : Doç. Dr. Bilal TOKLU (Bşk)   
Doç. Dr. Hadi GÖKGEN   
Doç. Dr. Ahmet PEKER 

Üretim kaynaklarının etkili kullanımını sağlamak amacıyla son yıllarda üreticilerin ilgi odağı haline gelen ve araştırmacılar tarafından üzerinde yoğun araştırmalar yapılan Hücreyel Üretim Sistemlerinin uygulamaya geçirilmesi için ilk adım olan tasarım aşaması bu çalışmanın kapsamını oluşturmaktadır.

Bu çalışmada öncelikle, temelini Hücreyel Üretim Sistemlerinin oluşturduğu bazı modern üretim sistemlerinden bahsedilmiş olup daha sonra H.Ü.S. tasarımında kullanılan 4 yapay zeka tekniği ve 3 sezgisel yöntem anlatılmıştır. Anlatılan yapay zeka tekniklerinden üçü ve sezgisel yöntemlerden ikisi olmak üzere toplam 5 tasarım yöntemi PASCAL programlama dilinde kodlanarak literatürden alınan toplam 10 adet test problemi üzerinde test edilmiş ve elde edilen sonuçlar 4 farklı değerlendirme kriterine göre yorumlanmıştır.

**Anahtar Kelimeler** : hücreyel üretim sistemleri, yapay sinir ağları, tabu araştırma algoritması, çok boyutlu ölçeklendirme, tavlama benzetimi, bulanık mantık,yapay zeka.

## ABSTRACT

Master's Thesis

### COMPARISON OF ARTIFICIAL INTELLIGENCE TECHNIQUES AND HEURISTIC METHODS USED IN CELLULAR MANUFACTURING DESIGN AND THEIR APPLIED ANALYSIS

Yakup KARA

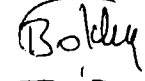
Selçuk University


Graduate School of Natural and Applied Sciences


Department of Industrial Engineering

Supervisor : Assoc. Prof. Dr. Bilal TOKLU

2000,

Jury : Doç.Dr. Bilal TOKLU (Bşk) 

Doç.Dr. Hadi GÖKÇEN 

Doç.Dr. Ahmet PEKER 

In this study, the design of cellular manufacturing systems which has taken immens

attention by researchers has been studied.

At the first step, modern manufacturing systems that are the basis for cellular manufacturing systems are studied. These are four artificial intelligence techniques and three heuristic methods. Three artificial intelligence techniques and two heuristic methods are chosen. These algorithms are coded in PASCAL. Ten test problems are retrieved from the literature and applied to the algorithms. Results are analyzed under four different criteria.

**Keywords** : cellular manufacturing systems, artificial neural networks, tabu search, multidimensional scaling, simulated annealing, fuzzy logic, artificial intelligence.

## ÖNSÖZ

Son yıllarda, dünyada insanlık tarihinin belki de bir benzerini daha önce yaşamadığı sosyal, kültürel, ekonomik ve teknolojik birtakım değişikliklerin yaşandığını, bu değişikliklere paralel olarak dünya ülkeleri ve firmalar arasındaki büyük karmaşayı ve bu karmaşanın sebebinin ise ülkeler ve firmalar arasındaki ekonomik üstünlük sağlama ve ayakta kalabilme çabası olduğunu hepimiz bilmekteyiz.

Bu değişimi firmalar bazında incelediğimizde, firmaların acımasız rekabet koşullarında ayakta kalabilmesi için tek yolun değişime ayak uydurmak olduğu aksi halde küçük beyinli, hantal ve hareket yeteneği az dinozorlar gibi yeryüzünden silinecekleri acı gerçektir.

Yukarıda bahsetmiş olduğumuz değişime ayak uydurabilmek ancak tüm firma fonksiyonlarının esnekleştirilmesi ile mümkün olacaktır. Üretim sisteminin esnekleştirilebilmesi, en etkili ve en verimli biçimde kullanılabilmesi için etkili bir araç ise Hücresel Üretim Sistemleri' dir.

İşte bu çalışmanın amacı Hücresel Üretime geçişin ilk aşaması olan tasarım aşamasında gerçekleştirilen makine grupları ve parça ailelerinin belirlenmesi işlemini ayrıntılı olarak analiz etmek ve bu iş için kullanılan yöntemlerin performansını test etmektir.

Hazırlamış olduğum bu çalışmanın, bu konuyla ilgilenen diğer araştırmacılara ve öğrencilere faydalı olmasını temenni eder, çalışmanın hazırlanması sırasında yardımlarını esirgemeyen başta danışman hocam sayın Doç. Dr. Bilal TOKLU' ya , Arş. Gör. Kaan BAYKAN' a ve Arş. Gör. Halife KODAZ' a teşekkürlerimi sunarım.

Yakup KARA

16 Ağustos 2000

# İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
SİMGELER.....	vii
ŞEKİL LİSTESİ.....	viii
TABLO LİSTESİ.....	ix
1. GİRİŞ.....	1
2. KAYNAK ARAŞTIRMASI.....	3
3. MATERYAL VE METOT.....	10
4. MODERN ÜRETİM SİSTEMLERİ.....	13
4.1. Giriş.....	13
4.2. Bilgisayarla Bütünleşik Üretim.....	14
4.3. Esnek Üretim Sistemleri.....	16
4.3.1. Üretim Sistemlerinde Esneklik.....	16
4.3.2. Esnek Üretim Sistemlerinin Tanımı.....	17
4.3.3. Esnek Üretim Sistemlerinin Avantajları.....	18
4.3.4. Esnek Üretim Sistemi Teknolojisi.....	20
4.3.4.1. Nümerik kontrollü tezgahlar.....	20
4.3.4.2. Otomatik malzeme taşıma sistemi.....	21
4.3.4.2.1. Otomatik güdümlü araçlar.....	21
4.3.4.3. Hiyerarşik bilgisayar ağı ve kontrol.....	22
4.3.5. Esnek Üretim Sistemlerinin Sınıflandırılması.....	23
4.3.6. Klasik Üretim Sistemleri ile Esnek Üretim Sistemlerinin Karşılaştırılması.....	25
4.3.7. Esnek Üretim Sistemleri ile İlgili Problemler.....	25
4.4. Grup Teknolojisi ve Hücreli Üretim Sistemleri.....	26
4.4.1. Giriş.....	26
4.4.2. Grup Teknolojisinin Sistem Performansı Üzerindeki Etkileri.....	27
4.4.3. Grup Teknolojisinin Diğer Fonksiyonel Alanlara Etkisi.....	28

4.4.4. Grup Teknolojisinin Diğer Teknolojilere Etkisi.....	28
4.4.5. Hücresel Üretimde Yerleşim Şekilleri.....	29
4.4.6. Grup Teknolojisinde Kodlama Sistemleri.....	30
5. HÜCRESEL ÜRETİMDE HÜCRE FORMASYONU PROBLEMİ.....	33
5.1. Yapay Zeka Yöntemleri.....	33
5.1.1. Yapay zeka.....	33
5.1.1.1. Uzman sistemler.....	35
5.1.1.1.1. Tanım.....	35
5.1.1.1.2. Uzman sistemlerin avantaj, dezavantaj ve sınırları.....	36
5.1.1.2. Genetik algoritmalar.....	37
5.1.1.3. Yapay sinir ağları.....	39
5.1.1.3.1. Bir sinir ağı nedir.....	39
5.1.1.3.2. Nöronun matematik modeli.....	40
5.1.1.3.3. Aktifleşme fonksiyonları.....	41
5.1.1.3.4. Yapay sinir ağlarının uygulama alanları.....	42
5.1.1.3.5. Öğrenme kuralları.....	43
5.1.1.3.6. Hücresel üretim sistemi tasarımında yapay sinir ağları uygulamaları.....	44
5.1.1.4. Bulanık mantık.....	49
5.1.1.4.1. Bulanık küme teorisi.....	49
5.1.1.4.2. Bulanık küme tabanlı hücre formasyonu.....	50
5.2. Sezgisel Yöntemler.....	52
5.2.1. Tavlama benzetimi.....	52
5.2.1.1. Tavlama benzetimi teorisi.....	52
5.2.1.2. Tavlama benzetimi yardımıyla makine-parça hücre formasyonu probleminin çözümü.....	54
5.2.2. Tabu araştırması.....	56
5.2.2.1. Tabu araştırma algoritması.....	56
5.2.2.2. Hücre formasyonu problemine tabu araştırması yaklaşımı.....	59
5.2.3. Çok boyutlu ölçeklendirme (Multidimensional scaling) yardımıyla hücresel üretim sistemi tasarımı.....	64
6. ARAŞTIRMA SONUÇLARI.....	68

6.1. Yöntemlerin Karşılaştırılması.....	68
6.2. Test Problemleri.....	68
6.3. Test Sonuçları.....	69
6.3.1. Program sonuçları.....	69
6.3.2. Kümelendirme sonuçları.....	72
7. TARTIŞMA.....	75
8. SONUÇ VE ÖNERİLER.....	76
9. KAYNAKLAR.....	77
EKLER.....	79



## SİMGELER

H.Ü.S.	: Hücresel Üretim Sistemleri
BDPP	: Bilgisayar Destekli Proses Planlama
EÜS	: Esnek Üretim Sistemi
BDP	: Bilgisayar Destekli Planlama
EÜH	: Esnek Üretim Hücresi
BDKK	: Bilgisayar Destekli Kalite Kontrol
EMS	: Esnek Montaj Sistemi
BDT	: Bilgisayar Destekli Tasarım
DNK	: Doğrudan Nümerik Kontrol
BDTÇ	: Bilgisayar Destekli Tasarım ve Çizim
VTS	: Veri Toplama Sistemi
MİP	: Malzeme İhtiyaç Planlaması
ÜV	: Üretim Verisi
İÜT	: İleri Üretim Teknolojisi
MV	: Makine Verisi
BBÜ	: Bilgisayarla Bütünleşik Üretim
AK	: Alan Kontrolcü
BDM	: Bilgisayar Destekli Mühendislik
HK	: Hücre Kontrolcü
BDÜ	: Bilgisayar Destekli Üretim



## ŞEKİLLERİN LİSTESİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 1. Makine-Parça İlişki Matrisi.....	10
Şekil 2. Bilgisayarla Bütünleşik Üretim Teknolojisi.....	14
Şekil 3. Bilgisayarla Bütünleşik Üretim Sistemi Kategorileri.....	15
Şekil 4. Esnek Üretim Sistemi Konfigürasyonları.....	24
Şekil 5. Esnek Üretim Sistemi Örneği.....	24
Şekil 6. Hücresel Üretimde Yerleşim Şekilleri.....	30
Şekil 7. Adamlı Bir İmalat Hücresi.....	31
Şekil 8. Uzman Sistemlerin Temel Bileşenleri.....	35
Şekil 9. Biyolojik Sinir Ağı ve Yapay Sinir Ağı.....	39
Şekil 10. Nöronun Matematiksel Modeli.....	40
Şekil 11. Yarışmalı Eğitimin Genel Algoritması.....	48
Şekil 12. Başlangıç Makine-Parça İlişki Matrisi.....	50
Şekil 13. Tavlama Benzetimi Algoritması.....	53
Şekil 14. Makine-Parça Hücre Formasyonu Probleminin Çözümü İçin Tavlama Benzetimi Prosedürü.....	55
Şekil 15. Test Problemi 1.....	
Şekil 16. Test Problemi 2.....	
Şekil 17. Test Problemi 3.....	
Şekil 18. Test Problemi 4.....	
Şekil 19. Test Problemi 5.....	
Şekil 20. Test Problemi 6.....	
Şekil 21. Test Problemi 7.....	
Şekil 22. Test Problemi 8.....	
Şekil 23. Test Problemi 9.....	
Şekil 24. Test Problemi 10.....	

## TABLULARIN LİSTESİ

<u>Tablo</u>	<u>Sayfa</u>
Tablo 1. Yapay Sinir Ağları Yardımıyla H.Ü.S. Tasarımı Uygulamaları.....	6
Tablo 2. Analiz Edilecek Olan Yapay Zeka Tabanlı Prosedürler.....	34
Tablo 3. Karşılaştırmada Kullanılan Test Problemleri.....	69
Tablo 4. ART 1 Nöron Modeli İçin Program Sonuçları.....	70
Tablo 5. Bulanık ART Nöron Modeli İçin Program Sonuçları.....	70
Tablo 6. Yarışmalı Eğitim Algoritması İçin Program Sonuçları.....	71
Tablo 7. Çok Boyutlu Ölçeklendirme Algoritması İçin Program Sonuçları.....	71
Tablo 8. Tabu Araştırma Algoritması İçin Program Sonuçları.....	72
Tablo 9. Test Problemi 1 İçin Çözüm Sonuçları.....	72
Tablo 10. Test Problemi 2 İçin Çözüm Sonuçları.....	72
Tablo 11. Test Problemi 3 İçin Çözüm Sonuçları.....	73
Tablo 12. Test Problemi 4 İçin Çözüm Sonuçları.....	73
Tablo 13. Test Problemi 5 İçin Çözüm Sonuçları.....	73
Tablo 14. Test Problemi 6 İçin Çözüm Sonuçları.....	73
Tablo 15. Test Problemi 7 İçin Çözüm Sonuçları.....	73
Tablo 16. Test Problemi 8 İçin Çözüm Sonuçları.....	74
Tablo 17. Test Problemi 9 İçin Çözüm Sonuçları.....	74
Tablo 18. Test Problemi 10 İçin Çözüm Sonuçları.....	74

## 1. GİRİŞ

Günümüzde özellikle bilgi teknolojilerindeki hızlı gelişmeler, zor müşteri kavramını da beraberinde getirdiği gibi mevcut birtakım optimizasyon tekniklerinin kullanımını da kolaylaştırmış ve yapay zeka tekniklerinin hayata geçirilmesini hızlandırmıştır. Mevcut müşteri ihtiyaçlarına ve ortaya çıkabilecek müşteri beklentilerine hızlı bir şekilde cevap vermek zorunda kalan günümüz işletmeleri, üretim sistemlerinde esneklik kavramının çeşitli boyutlarının, kalite boyutu ile bütünleştirilmesi gerektiği bilincine son yıllarda varmıştır.

Günümüz işletmelerini bir buz pisti üzerinde hareket eden çok sayıda insan olarak düşünürsek, bu buz pisti üzerinde ayakta kalabilmek için sürekli hareket etmek yetmemekte aynı zamanda akıllı hareket etmek gerekmektedir. Bir üretim sisteminin akıllı hareket etmesi insan kaynaklarına bağlı olmakla beraber elindeki diğer imkanlara ve faaliyetlere zeka kavramını kazandırmasıyla daha da genişletilebilir. Bu noktada karşımıza “Yapay Zeka” kavramı çıkmaktadır. Üretim sistemlerinde karşılaşılan çeşitli problemlerin çözümünde kullanılan teknikler kervanına son yıllarda katılan yapay zeka tekniklerinin çeşitli problemlere uygulanmaları sınır tanımadan hızla genişlemektedir.

Müşteri isteklerindeki sürekli değişime paralel olarak ortaya çıkan küçük hacimli, çok ürünlü ve sürekli ürün değişikliğine maruz kalan üretim sistemlerine esneklik ve verimlilik kazandırmak amacıyla son zamanlarda ortaya çıkan yenilikçi bir yaklaşım da “Hücresele Üretim Sistemleri”dir. Hücresele Üretim Sistemleri(H.Ü.S.), ikinci kuşak grup teknolojisi uygulaması olarak ortaya çıkmıştır. Birinci kuşak grup teknolojisi uygulamaları benzer operasyon sıralarına ve benzer geometrik şekillere sahip parçaların parça ailelerinde bir araya getirilmesi faaliyetlerini kapsamakta olup tesis yerleşiminde herhangi bir değişiklik söz konusu değildir. İkinci kuşak grup teknolojisi uygulamaların da ise, parça aileleri ve bu parça ailelerini işleyecek makine hücreleri eşzamanlı olarak şekillendirilmektedir.

Birçok işletme tarafından faydalı bir üretim biçimi olarak görülen H.Ü.S.’nin asıl amacı; parçaların tasarımları ve üretimlerindeki benzerliklerinden faydalanarak, üretim sistemini, kontrolü kolay bazı alt sistemlere(Üretim Hücreleri) bölmek ve bu sayede hazırlık sürelerini, atölye içi envanterleri, işlem sürelerini ve malzeme taşıma

maliyetlerini düşürmek suretiyle üretim sisteminin esnekliğini ve verimliliğini artırmaktır. Vakharia (1986), H.Ü.S.' ne karşı artan ilginin üç önemli sebebinin olduğunu belirtmiştir. Bunlar:

1. Esnek Üretim Sistemleri gibi modern üretim sistemlerinin başarısının üretim hücresi şekline bağlı olması,
2. Geçmişe nazaran daha küçük parti büyüklüklerinde üretimi gerçekleştirilen tüketim mallarına olan talebin atışı,
3. Üretim yöneticilerinin sıkıcılığın ve işçi görevlerinin monotonluğunun azaltılmasının önemine varmış olmaları.

Yukarıda belirtilen sebeplerden dolayı üreticilerin ilgisini çeken H.Ü.S.'nin tasarımı üç önemli aşamadan meydana gelmektedir. Bunlar:

1. Ekonomik ve teknik fizibilite etüdü
2. Üretim Sistemi Tasarımı
3. Uygulama (Wemmerlöv ve Hyer 1986)

Parti tipi üretimi akış tipi üretime dönüştüren hücresele üretim sistemi tasarımı aşamasında öncelikle çözülmesi gereken problemlerin başında, hücre formasyonu problemi yani; benzer tasarım özelliklerine ve operasyon gerekliliklerine sahip parçaların aynı parça ailelerinde, bu parçaları işleyecek uygun makinelerin de aynı makine hücrelerinde toplanmasını sağlamaktır.

Hücre formasyonu problemi uzun yıllar boyunca araştırmacıların ilgi alanına girmiş ve bu problemin çözümüne yönelik olarak çok sayıda yöntem geliştirilmiştir. Bu çalışmada öncelikle, geliştirilmiş hücre formasyonu veya makine-parça grup analizi yöntemleri genel olarak incelenecek daha sonra H.Ü.S. anlatılarak mevcut tekniklerden bazılarının bilgisayar programları desteğinde detaylı analizleri yapılacaktır.

## 2. KAYNAK ARAŞTIRMASI

H.Ü.S. tasarımına yönelik olarak geliştirilmiş olan yöntemler genel olarak aşağıdaki üç yaklaşımdan birini benimsemektedirler. Kullandıkları yaklaşıma göre H.Ü.S. tasarımı yöntemleri üçe ayrılmaktadır:

**1.Yaklaşım:** Öncelikle parçaların geometrik benzerliklerine ve üretim benzerliklerine göre parça aileleri oluşturulur, daha sonra oluşturulan bu parça ailelerinin ihtiyaçlarına göre makineler gruplandırılır.

**2.Yaklaşım:** Önce makine hücreleri belirlenir, daha sonra parçalar operasyon sıralarına göre bu hücrelerden uygun olanlarına atanır.

**3.Yaklaşım:** Makine hücreleri ve parça aileleri eşzamanlı olarak oluşturulur. Bu yaklaşım daha çok yeni bir H.Ü.S. tasarımı söz konusu olduğunda kullanılmaktadır.

Yukarıdaki yaklaşımlardan herhangi biri doğrultusunda çözüm arayan hücre tasarımı teknikleri, aşağıdaki iki tip bilgi kümesinden faydalanmaktadır. Bunlar:

1. Parça Karakteristikleri Bilgileri(Tasarım özellikleri)
2. Üretim Prosesi Bilgileri(İşlem sıraları, rotalama bilgileri)

Literatürde önerilen yöntemlerin büyük bir çoğunluğu çözüm aşamalarında 2. Tip bilgileri kullanmaktadır. Fakat geliştirilen yöntemlere bakıldığında, özellikle sezgisel yöntemlerde yukarıda belirtilen iki tip bilgi kümesine ilaveten aşağıdaki bilgilerin de H.Ü.S. tasarımında kullanıldığını görmek mümkündür. Bu bilgiler:

- Üretilen parçalara ait talep miktarları
- Ürünlerin satış fiyatları
- Kullanılan makinelerin kapasiteleri
- İşlem Süreleri
- Hazırlık süreleri
- Mevcut makine sayıları
- Hücre içi ve hücrelerarası malzeme taşıma maliyeti
- Makine yatırım maliyetleri

H.Ü.S. tasarımında kullanılan prosedürleri genel olarak yedi kategoriye ayırmak mümkündür. Bunlar:

1. Dizi Tabanlı Kümeleme Yöntemleri
2. Hiyerarşik Kümeleme Yöntemleri
3. Nonhiyerarşik Kümeleme Yöntemleri
4. Matematik Programlama Yöntemleri
5. Grafik Tabanlı Yaklaşımlar
6. Yapay Zeka Yöntemleri
7. Sezgisel Yöntemler

**1. Dizi Tabanlı Kümeleme Yöntemleri:** Bu yöntemler, parça ailelerini ve makine hücrelerini eşzamanlı olarak oluşturmak amacıyla gerçekleştirilen bir dizi satır-sütun işlemlerinden meydana gelmektedir. Bu yöntemlerden en eski ve en yaygın olarak kullanılanları, BEA(Bond Energy Algorithm)(McCormick, vd 1972) ve ROC(Rank Order Clustering)(King 1980)' dir. Dizi Tabanlı Kümeleme Yöntemleri sadece makine-parça ilişki matrisi üzerinde işlemler gerçekleştirmekte ve hücre büyüklüğü, işlem zamanı, vb diğer üretim bilgilerine ihtiyaç duymamaktadır.

**2. Hiyerarşik Kümeleme Yöntemleri:** Bu yöntemler yardımıyla birtakım çözümler hiyerarşisi elde edilmekte ve bunların içerisinde en iyi alternatif seçilebilmektedir. Ayrıca bu yöntemler bazı üretim bilgilerinin kümeleme mantığı içerisinde kullanılmasına olanak sağlamakla birlikte Dizi Tabanlı Kümeleme Yöntemlerine nazaran daha avantajlıdır. Hiyerarşik Kümeleme Yöntemleri, parça ailelerini ve makine hücrelerini eşzamanlı olarak oluşturamadığı için sistem tasarımının tamamlanabilmesi için ilave bazı metotlar kullanılmak zorundadır.

**3. Nonhiyerarşik Kümeleme Yöntemleri:** Bu yöntemlere örnek olarak GRAPHICS (Srinivasan ve Narendran 1991) ve ZODIAC (Chandrasekharan ve Rajagopalan 1987) gösterilebilir. Yapılan çalışmalar bu yöntemlerin Dizi Tabanlı Yöntemlere ve Hiyerarşik Kümeleme Yöntemlerine göre daha iyi sonuçlar verdiğini göstermiştir. Bu yöntemlerin uygulamalarında istenen hücre sayısı başlangıçta belirlenmektedir. Girdi olarak ise sadece makine-parça ilişki matrisini kullanmaktadırlar.

**4. Grafik Tabanlı Yaklaşımlar:** Kümeleme problemleri , Şebeke Akış (Network Flow) ve Minimum Yayılan Ağaç ( Minimum Spanning Tree) yaklaşımları ile fömülize edilerek etkili çözümler üretilebilmektedir.

**5. Matematik Programlama Yöntemleri:** Kümeleme problemi aynı zamanda bir optimizasyon problemi olarak da düşünülebilir. Üretimi gerçekleştirilecek parçaların alternatif proses planlarına sahip olduğu durumlarda, hücre sayısının ve hücre büyüklüğünün kısıtlı olduğu durumlarda H.Ü.S. tasarımına matematiksel programlama teknikleri uygulanarak problem modellenir. Bu yöntemlerin büyük bir kısmı Lineer ve Nonlieer Tamsayılı Programlama modellerini kapsamakta olup makine ve parça sayısının çok fazla olduğu problemlerde çözüm üretmek zor olmaktadır. Ayrıca bu yöntemlerin büyük çoğunluğu makine hücrelerini ve parça ailelerini eşzamanlı olarak oluşturamamaktadır.

**6. Yapay Zeka Yöntemleri:** Dört yapay zeka tekniği olarak belirtilen Yapay Sinir Ağları, Uzman Sistemler, Bulanık Mantık ve Genetik Algoritmalar H.Ü.S. tasarımında, tasarımı etkileyen çeşitli faktörleri dikkate alarak büyük boyutlu problemlere etkili çözümler üretmek amacıyla kullanılmaktadır. Bu tekniklerden H.Ü.S. tasarımında en çok kullanılanı Yapay Sinir Ağlarıdır. Ayrıca son zamanlarda bu tekniklerin kombinasyonları sonucu meydana gelen melez sistemlerin de zeki tasarım teknikleri olarak kullanıldığı göze çarpmaktadır.

**7. Sezgisel Yöntemler:** Sadece makine-parça ilişki matrisi üzerinde veya diğer üretim verilerini de kullanarak hücre formasyonu problemine çözüm arayan çok sayıda ve çeşitlilikte sezgisel yöntem mevcuttur. Özellikle büyük boyutlu problemlerin çözümünde etkili sonuçlar vermektelerir.

Literatürde yukarıdaki yedi kategorinin her birine ait çok sayıda kümeleme yöntemiyle karşılaşmak mümkündür. Bu çalışma kapsamında, bu yedi kategori içerisinde Sezgisel Yöntemler ve Yapay Zeka Teknikleri üzerinde durulacaktır. Çalışma kapsamında üzerinde durulacak olan teknikler ise şunlardır:

#### **Yapay Zeka Teknikleri**

- Yapay Sinir Ağları (Artificial Neural Networks)

- Bulanık Mantık ( Fuzzy Logic)

### Sezgisel Yöntemler

- Tavlama Benzetimi (Simulated Annealing)

- Tabu Araştırması (Tabu Search)

- Çok Boyutlu Tabakalandırma(Multidimensional Scaling)

Bu dört tekniği son zamanlarda Hücre Formasyonu problemini çözmek amacıyla araştırmacıların en fazla ilgisini çeken teknikler olarak belirtmek mümkündür. Aşağıda konuyla ilgili çalışmalar kullandıkları tekniklere göre sınıflandırılarak özetlenmiştir.

### 1. Yapay Sinir Ağları (Artificial Neural Networks)

Çizelgeleme, tahmin, desen tanıma, sınıflandırma ve kodlama. vb birçok alanda geniş bir uygulama alanına sahip olan bu teknik, bir kümeleme tekniği olarak da literatürde geniş yer tutmaktadır. Tablo 1, bu çalışmaları yapay sinir ağlarının bazı karakteristiklerini göz önünde bulundurarak vermektedir.

**Tablo 1.** Yapay Sinir Ağları Yardımıyla H.Ü.S. Tasarımı Uygulamaları

Kullanılan Model	Veri Tipi	Yaklaşım Türü	Kaynak
Hatayı geriye yayma	Tasarım	1	Charkraborty,1993
Hatayı geriye yayma	Tasarım	1	Kaparthi, 1991
Hatayı geriye yayma	Tasarım	1	Kao, 1991
Hatayı Geriye Yayma	Rotalama vd.	6	Chen, 1995
Etkileşimli aktivasyon	Rotalama	2	Moon, 1990
Etkileşimli aktivasyon	Rotalama vd.	2	Moon, 1993
Etkileşimli aktivasyon	Rotalama vd.	2	Chu, 1997
Yarışmalı eğitim	Rotalama	3	Chu, 1993
Yarışmalı eğitim	Rotalama	3	Venugupal, 1993
ART I	Rotalama	4	Chen, 1995
ART I	Rotalama	4	Dağlı, 1995
ART I	Rotalama	4	Kusiak, 1991
ART I	Rotalama	4	Liao, 1993
ART I	Rotalama	4	Venugupal, 1993
ART I	Rotalama	5	Kaparthi,1992 ve 1994
Bulanık ART	Rotalama	5	Kaparthi, 1994
Self-organizing map	Tasarım	1	Chakraborty, 1993
Self-organizing map	Rotalama	6	Ramachandran, 1991



### Yaklaşım Türü :

- (1) Sadece parça ailelerini oluşturur. Gerekli hücre sayısı belirtilir.
- (2) Parça ailelerini ve makine hücrelerini eşzamanlı oluşturur. Benzerlik katsayısı hesaplanmaktadır.
- (3) Önce parça aileleri sonra makine hücrelerini oluşturur. Gerekli hücre sayısı belirtilir.
- (4) Önce parça aileleri sonra makine hücrelerini oluşturur. Maksimum hücre sayısı belirtilir.
- (5) Önce makine hücrelerini sonra parça ailelerini oluşturur.
- (6) Parça ailelerini ve makine hücrelerini eşzamanlı oluşturur. Gerekli hücre sayısı belirtilir.

## 2. Bulanık Mantık (Fuzzy Logic)

Hücre formasyonu probleminin çözümüne yönelik olarak geliştirilmiş olan yöntemlerin büyük çoğunluğu, yalnızca sayısal birtakım verileri kullanarak çözüm aramaktadırlar. Bulanık kümeler tasarım aşamasında dilsel değişkenlerin kullanımına olanak sağladığı gibi, yukarıda belirtilen altı kategoriye dahil olan tekniklerden bazılarında bulanık parametreler yardımıyla farklı bir boyut kazandırılmıştır.

Suresh ve Kaparthi (1994), Carpenter ve arkadaşları tarafından 1991 yılında geliştirilen ve hem analog hem de ikili girdilerin kullanılabilirdiği Bulanık ART (Fuzzy Adaptive Resonance Theory) sinir ağlarını hücre formasyonu problemini çözmek amacıyla kullanmışlardır. Bu çalışmada sadece 0 ve 1'lerden oluşan makine-parça ilişki matrisi kullanılmış ve elde edilen sonuçların ART1, DCA ve ROC algoritmalarına karşı üstünlüğü kanıtlanmıştır.

Chun ve arkadaşları (1997), üretim hücrelerini oluşturmak amacıyla bulanıklaştırılmış karışık tamsayı programlama tekniğini kullanmıştır. Geliştirilen yöntem üretim hücrelerini eşzamanlı olarak oluşturmaktadır.

Leem ve Chen (1996), bir operasyonun alternatif makinelerde gerçekleştirilebileceğini düşünerek aynı türden birden fazla sayıda makinelerin bulunduğu ve eleman değerleri (0,1) aralığında değişen makine-parça ilişki matrisini kullanarak üretim hücrelerinin eşzamanlı olarak oluşturulmasını sağlamışlardır. Makine-parça ilişki matrisindeki değerler makinelerin operasyonlar için uygunluk derecesine göre değişmektedir.

Güngör (1998), bulanık kümeler yaklaşımıyla, tasarım ve üretim benzerliklerini, işlem zamanı benzerliklerini ve işlem sırası benzerliklerini dikkate alan bir bulanık karar verme yaklaşımı geliştirmiştir.

### 3. Tavlama Benzetimi (Simulated Annealing)

Su ve Hsu (1998), hücre formasyonu probleminin çözümüne yönelik olarak, genetik algoritmaların ve tavlama benzetiminin kombinasyonu olan Paralel Tavlama Benzetimini kullanmışlardır. Çalışmanın amaçları: hücre içi ve hücrelerarası malzeme taşıma maliyeti, makine yatırım maliyetlerini , hücre içi ve hücrelerarası makine yükü dengesizliğini minimize etmektir. Prosedür, H.Ü.S. tasarımını etkileyen işlem sıraları, hazırlık süreleri, işlem süreleri, ve taşıma maliyetleri gibi faktörleri dikkate alarak toplam maliyeti eşzamanlı olarak minimize etmektedir. Önerilen prosedürün esnekliği, etkililiği ve gerçek üretim ortamında uygulanabilirliği yapılan analizlerle belirtilmiştir.

Boctor (1991), harici eleman sayısını minimize etme kriterine göre, tavlama benzetimini 0 ve 1'lerden oluşan makine-parça ilişki matrisi üzerinde uygulamışlardır. Elde edilen sonuçlar 0-1 tamsayılı lineer formülasyonla elde edilen sonuçlarla kıyaslandığında, tavlama benzetiminin daha etkili sonuçlar verdiği gözlenmiştir.

Abdelmola ve Taboun (1999), yıllık parça taleplerini, parçaların satış fiyatlarını, gerekli makine sayılarını ve malzeme taşıma maliyetlerini dikkate alarak, sistemin toplam verimliliğini eniyileme amacıyla tavlama benzetimini kullanarak bir verimlilik modeli ortaya koymuştur. Elde edilen sonuçlar 0-1 tamsayılı programlama modeli ile kıyaslandığında algoritmanın üstünlüğü kanıtlanmıştır.

Zolfaghari ve Liang (1998), işlem zamanlarını ve parti büyüklüklerini dikkate alarak darboğaz makine sayısı minimize etmeyi amaçlayan bir prosedür geliştirmişlerdir.

Boctor (1996), Talep, satış fiyatı, malzeme taşıma maliyeti ve işlem zamanı gibi faktörleri dikkate alarak ve toplam maliyeti minimize etmek suretiyle, başlangıçta belirlenen hücre sayısına ve hücrelerdeki maksimum ve minimum makine sayısına göre optimum kümeleme işlemini gerçekleştirecek bir tavlama benzetimi prosedürü önermiştir.

### 4. Tabu Araştırması (Tabu Search)

Aljaber, vd. (1996), hücre formasyonu problemini hem makineler hem de parçalar için "En Kısa Yol" problemi olarak modelleyerek bir tabu araştırması tabanlı yaklaşım geliştirmişlerdir. Tabu araştırması kullanılarak makine ve parçalar

için en kısa yol sıraları elde edildikten sonra oluşturulan çözüm matrisinin sınırlarını belirlemek için ayrı bir metot önerilmiştir.

Lozano, vd. (1999), bir lineer şebeke akış modelini kullanarak mümkün makine hücrelerini ve parça ailelerini oluşturmaya çalışan tabu araştırma algoritması geliştirmişlerdir. Geliştirilen yöntem iki tavlama benzetimi yaklaşımı ve üç sezgisel yöneme göre daha etkili sonuçlar vermiştir.



### 3. MATERYAL VE METOT

Bir önceki bölümde de belirtildiği gibi bu çalışmada üzerinde durulacak olan konu H.Ü.S. tasarımında kullanılan sezgisel yöntemler ve yapay zeka yöntemleridir. Literatür araştırması kısmında gösterilen çalışmalardan üç sezgisel prosedür ve üç yapay zeka prosedürü ele alınarak, yine literatürdeki çalışmalardan seçilen çeşitli boyutlarda problemler üzerinde uygulanacaktır. Uygulamanın yapılabilmesi için gerekli olan bilgisayar programları Pascal programlama dilinde kodlanmıştır. Üzerinde optimum çözümlerin aranacağı makine-parça ilişki matrisi 0 ve 1'lerden oluşmaktadır. Şekil 1, örnek bir makine-parça ilişki matrisini temsil etmektedir.

	Parçalar					
Makineler	0	1	0	0	1	1
	1	0	1	1	0	0
	0	0	0	1	1	1
	1	1	0	0	1	1
	1	1	0	0	0	0

$a_{ij}$

Şekil 1. Makine-Parça İlişki Matrisi

$a_{ij} = 1$ , eğer j parçası i makinesinde işlem görüyorsa

$a_{ij} = 0$ , eğer j parçası i makinesinde işlem görmüyorsa

Makine hücrelerinin ve parça ailelerinin oluşturulabilmesi için yukarıda verilen makine-parça ilişki matrisindeki "1" değerlerinin bir araya getirilmesi gerekmektedir. Bu ise ancak birtakım satır-sütun işlemleri ile gerçekleştirilebilir. Kümeleme işlemi tamamlandıktan sonra çeşitli yöntemlerle elde edilen sonuçlar aşağıdaki performans ölçülerine göre karşılaştırılacaktır. Bunlar:

1. Makine Kullanımı (Machine Utilization)
2. Hücrelerarası Hareket Sayısı (Number of Intercellular Elements)
3. Bağ Enerjisi (Bond Energy)
4. Grup Yeteneği (Group Efficiency)

**1. Makine Kullanımı:** Makine kullanımı, her bir hücrede kullanılmakta olan makine yoğunluğu şeklinde tanımlanarak, aşağıdaki gibi formülize edilebilir. Makine kullanımının büyük değere sahip olması tercih edilir.

MK: Makine kullanımı

MS: Hücrelerdeki toplam makine sayısı

$M_k$ : k Hücresindeki makine sayısı

$P_k$ : k hücresindeki parça sayısı

c : Toplam hücre sayısı

$$MK = \frac{MS}{\sum_{k=1}^c M_k.P_k} \quad (1)$$

**2. Hücrelerarası Hareket Sayısı:** Hücrelerarası hareket sayısı, çözüm matrisi elde edildikten sonra kümelenmeden kalan yani herhangi bir hücreye atanmadan harici olarak kalan "1" değerlerinin toplam sayısıdır. Kümelenmeden kalan "1" değerleri, parçaların tüm operasyonlarının tek hücrede tamamlanamadığı anlamına gelmekte olup hücrelerarası parça akışı söz konusudur. Bu durum doğal olarak malzeme taşıma maliyetini artıracığı için harici elemanların minimum seviyede tutulması gerekmektedir.

**3. Bağ Enerjisi:** Bu performans ölçüsü birbirine komşu olan "1" değerlerinin sayısını en yüksek değerde tutmayı amaçlamaktadır. Bu performans ölçüsü ise aşağıdaki gibi formülize edilir.

BE: Bağ enerjisi

$$BE = \frac{1}{2} \sum \sum a_{ij} (a_{ij+1} + a_{ij-1} + a_{i+1j} + a_{i-1j}) \quad (2)$$

**4. Grup Yeteneği:** Bu performans ölçüsü harici parça sayısı ile makine kullanımının ağırlıklı ortalamasıdır. Şu şekilde hesaplanır:

GY: Grup yeteneği

MK: Makine kullanımı ( $m_a$ )

m: Toplam makine sayısı

n: Toplam parça sayısı

c: Toplam hücre sayısı

$\alpha$ : Ağırlık faktörü

$H_p$  = Harici parça sayısı

$$GY = \alpha.m_a + (1 - \alpha).m_b \quad (3)$$

$m_a = MK$

$$m_b = 1 - \frac{H_p}{m.p - \sum_{k=1}^c m_k.p_k} \quad (4)$$



## 4. MODERN ÜRETİM SİSTEMLERİ

### 4.1. GİRİŞ

Modern endüstriyel üretim sistemleri farklı ekipman kullanımına, kullanılan bu ekipmanların yapısına ve teknolojisine, üretilen ürünlere ve faaliyet gösterilen sektöre göre farklılıklar göstermektedir. Genel olarak üretim prosesleri kesikli, sürekli ve tekrarlı olmak üzere üç sınıfa ayrılabilir.

İşlenecek parçaların çok çeşitli ve talebin süresiz olduğu durumlarda, ortaya çıkabilecek bütün değişiklikleri göz önünde bulundurarak bu değişikliklere anında cevap verebilmek amacıyla standart bir makine yerleşimine sahip olmak uygun olmaktadır. En uygun tesis yerleşimi düşük hacme ve yüksek ürün çeşitliliğine sahip olan esnek bir yerleşim tipidir. Tipik bir üretim tesisinde spesifik işlemler için işlem imkanları sağlayan, delme ve tornalama gibi bir çok birim mevcuttur. Parçalar çeşitli işlemler için bu birimler arasında hareket ederek çeşitli işlemlerden geçmektedirler. Ürün tasarımlarındaki farklılıklar çok fonksiyonlu ekipman gerektirmekte ve universal tezgahları verimsiz hale getirmektedir. Böyle bir yerleşimle parçalar işlem öncesi ve sonrası bekleme, bölümler arası hareket ve hazırlıkta ciddi miktarda zaman harcanmaktadır. Kaybedilen bu zaman üretim hazırlık zamanını artırmakta ve düşük verimliliğe sebep olmaktadır.

Sadece birkaç parça tipinin büyük miktarda üretiminin söz konusu olduğu bir tesiste makineler baştan sona sürekli bir akışın sürdürülebilmesi için sıralanabilmekte ve üretim hacminin yüksek olmasından dolayı universal tezgahların kullanımı da bu durumda uygun olmaktadır. Tesis düzenlemesi yapıp iş istasyonları dengelendikten sonra rotalama ve çizelgeleme problemi kolayca çözülebilir. Üretim maliyetleri bu yerleşim tipinde minimum düzeyde olup üretimdeki aksamalar ise son derece pahalıya mal olmakta ve farklı bir kalemin üretimi için yerleşimin değiştirilmesi de çok yüksek maliyetlere yol açmaktadır.

Bu iki uç durum arasında çok sayıda yatırım, tekrarlı talep durumuna göre ele alınır. Bu ise sık sık “Kitle Üretimi” olarak adlandırılır. Çok sayıda proses endüstrisi bir ürün çeşitliğinde karışım için bu üretim hattını kullanabilmektedirler.

Hızla gelişen teknoloji imkanlarının birtakım üretim felsefeleri ile bütünleştirilerek üretim sistemlerinde kullanılmaya başlanması ile birlikte, üretim sistemleri esnek bir hal almış ve ortaya çıkan değişikliklere uyum sağlamak için üretim sistemlerinde çok ciddi değişikliklere gidilmesine gerek kalmamıştır.

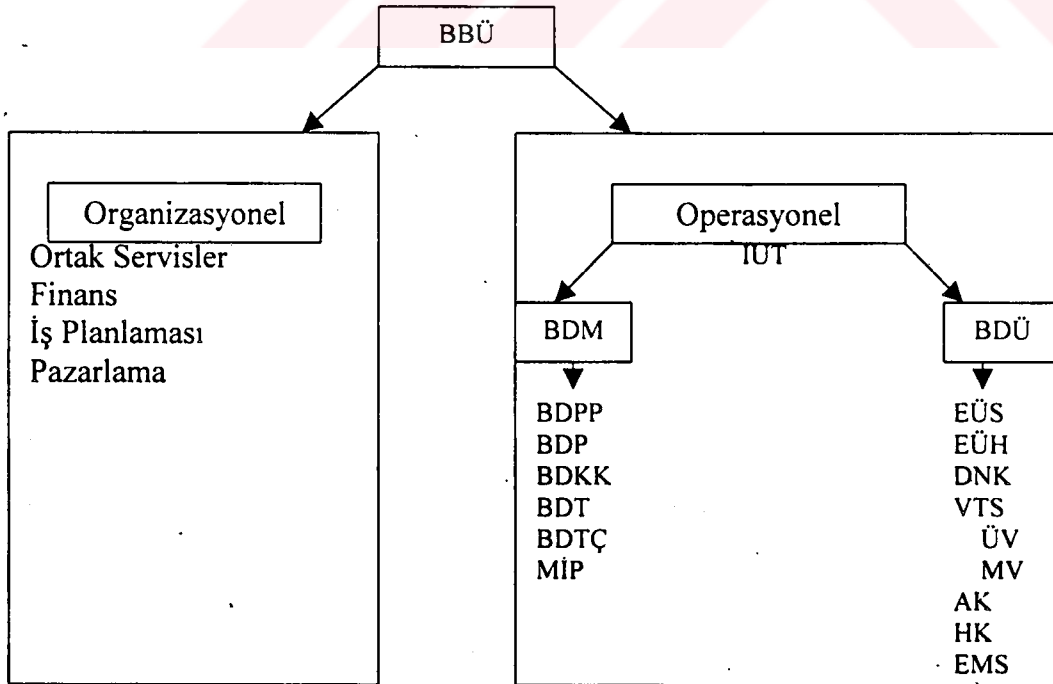
#### 4.2. BİLGİSAYARLA BÜTÜNLEŞİK ÜRETİM

Bilgisayarla Bütünleşik Üretim genel olarak; optimal üretim verimliliğini elde etmek için insan tarafından işletilen bir bilgisayarlar ve makineler ağı olarak tanımlanabilir(Mabert 1991).

Bilgisayarla Bütünleşik Üretim Sistemi genel olarak aşağıdaki prosesleri entegre eden bir yapıya sahiptir.

- Ürün ve parça tasarımı.
- Proses planlama.
- Programlama (makine, robot, AGV).
- Üretim planlama.
- İmalat.
- Alım, depolama, taşıma.

Bilgisayarla Bütünleşik Üretim teknolojisini Şekil 1'deki gibi göstermek mümkündür.

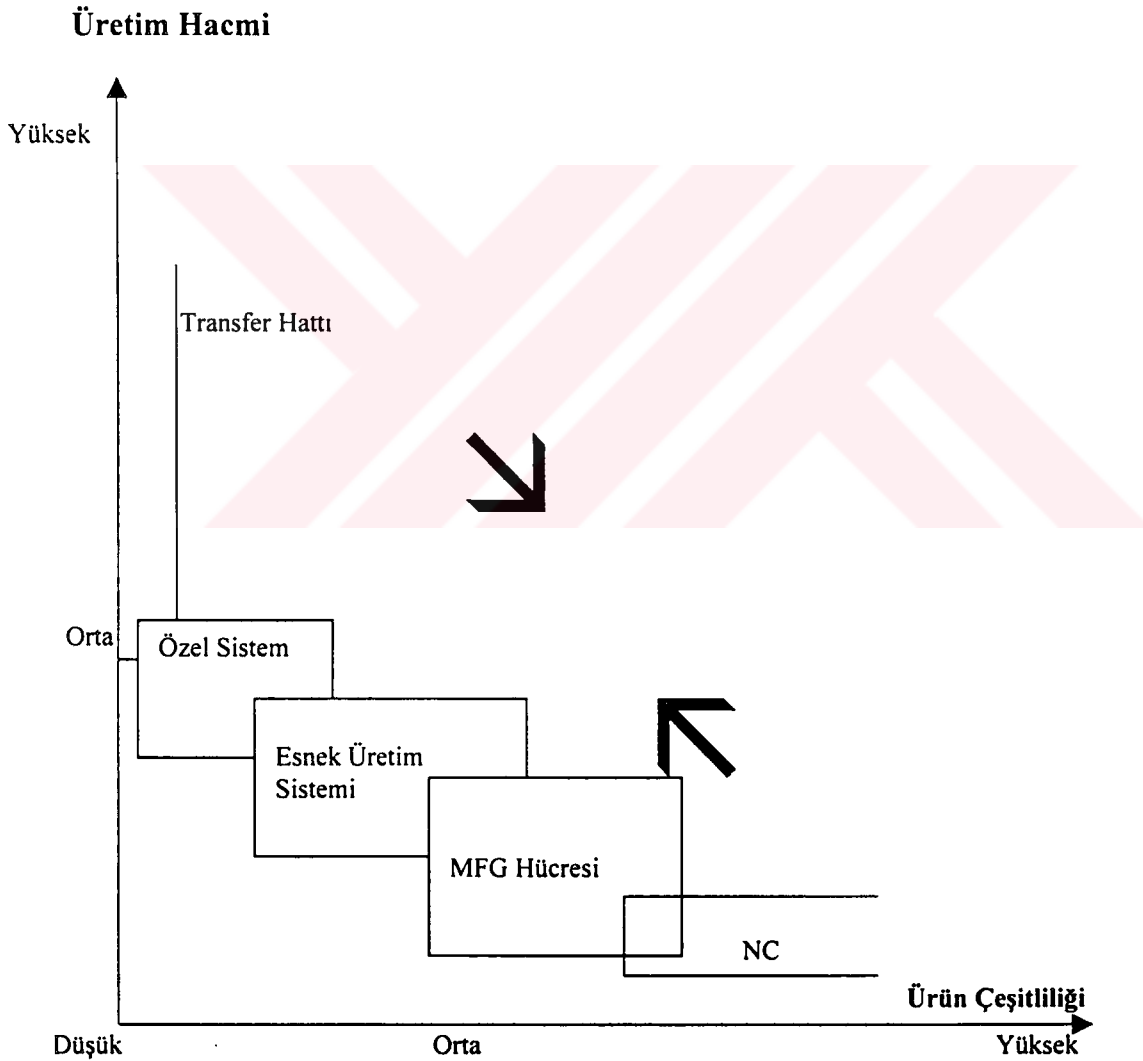


Şekil 2. Bilgisayarla Bütünleşik Üretim Teknolojisi (Parrish 1990)



Bilgisayarla Bütünleşik Üretim Sistemleri bilgisayarları, malzeme taşıma araçlarını ve nümerik kontrollü (NC) makinelerin esnekliklerini bütünleştiren bir sistemdir. Transfer hatları sınırlı parça çeşitliğinde üretime sahiptir. Bilgisayarla bütünleşik imalat sistemi yüksek üretim hacmi hedefine ulaşırken NC makinelerin esnekliğini ve esnekliğin farklı derecelerini sağlayabilmektedir. Bu esneklik derecelerini görebilmek amacıyla BBÜ sistemi Şekil 3’de görüldüğü gibi üç kategoriye ayrılabilir (Maleki 1991).

Geleceğin fabrikaları olarak adlandırılan ileri imalat sistemlerine ulaşma yolundaki en ileri evrimsel aşama ve Bilgisayarla Bütünleşik İmalat Sistemlerinin son noktası Esnek Üretim Sistemleridir (EÜS). EÜS, BBÜ ortamının bir parçasıdır.



Şekil 3. Bilgisayarla Bütünleşik Üretim Sistemi Kategorileri ( Maleki 1991)

### 4.3. ESNEK ÜRETİM SİSTEMLERİ

#### 4.3.1. Üretim Sistemlerinde Esneklik

Bir üretim sisteminin esnekliğini değişik açılardan incelemek mümkün olup bir sistemin değişikliklere reaksiyon gösterebilme ve uyum sağlayabilme hızı olarak tanımlanabilir. Esneklik tasarımdan üretime, üretimden dağıtıma kadar bir ürünün hayat çevrimi süresince bulunmak zorundadır. Esnek Üretim Sistemleri değişen Pazar şartlarını karşılama amacına ulaşmak için gerekli esneklikte üreticiler sağlayabilmektedirler. Bir üretim sisteminin esnekliğinin derecesi ve seviyesi ise teknolojik kısıtlar ile sınırlandırılmaktadır (Maleki 1991).

Bir üretim sistemi için genel olarak aşağıdaki esneklik tiplerinden bahsetmek mümkündür (Hunt 1989).

- **Proses Esnekliği:** Makine ve teçhizatla önemli yatırımlara gidilmeden farklı ürünler üretebilecek proseslere adapte olabilme yeteneğidir. Bu esneklik tipi çeşitli malzeme kullanımına ve çeşitli ürün dizaynlarının üretimine olanak sağlayan esnek otomasyon teknolojilerinin kilit taşıdır.
- **Program Esnekliği:** Prosesin bazı kısımlarının başarısız olduğu durumlarda bile üretimi devam ettirmek amacıyla işlem rotası değişikliklerine, sistemin uyarlanabilir kontrolüne, beklenmedik operasyonlara ve destek faaliyetlerine olanak sağlayarak ekonomik üretimi devam ettirebilme yeteneğidir.
- **Yenilik Esnekliği:** Bu esneklik türü ise mevcut teknolojileri uygulayabilme yeteneği olarak tanımlanabilir.

Literatürde yer alan esneklik tanımlarından da anlaşılacağı gibi, bir üretim sisteminin esnekliği esneklik tiplerine ulaşılma düzeyine göre göreceli olarak değişmektedir. Bu tanımlardan bir kısmı aşağıda verilmiştir.

- **Tezgaah Esnekliği:** Çeşitli tipte parça işlenebilmesi için tezgahta yapılması gereken ayarların ve değişikliklerin ne kadar kolay yapılabildiğini gösteren bir ölçüdür.
- **Ürün Esnekliği:** Üretim sisteminin yeni bir ürün dizisine geçiş yapabilme kabiliyetinin ölçüsüdür. Gelecekteki ürün tasarımları ile ilgili belirsizlik sebebiyle ürün esnekliğine hangi boyutta gerek duyulacağına karar vermek zordur.

- **İş Akışı Esnekliği:** Sistemde meydana gelebilecek bir aksaklığını üretim miktarını ne ölçüde etkileyeceğinin bir göstergesidir.
- **Hacim Esnekliği:** Esnek Üretim Sistemlerinin değişik üretim hacimlerinde verimli şekilde üretim yapabilmesiyle ilgilidir.
- **Genişleyebilme Esnekliği:** Üretim tesisinin kapasitesinin artırılabilmesiyle ilgili bir ölçüdür.
- **İşlemsel Esneklik:** Her bir parça tipiyle ilgili işlemlerin sıralamasının değiştirilebilirliğinin bir göstergesidir.
- **Malzeme Taşıma Esnekliği:** İmalat Sistemi içerisinde çeşitli parçaların etkili bir biçimde hareket ettirilebilme yeteneğinin bir göstergesidir.

Yukarıda bahsedilen esneklik tiplerinin bir üretim sistemi içerisinde değişik düzeylerde bir arada bulunması sonucu E.Ü.S. ortaya çıkar. Yani bir üretim sisteminin genel anlamda esnekliği bu esneklik tiplerinin elde edilebilme yeteneğinin bir sonucudur.

#### 4.3.2. Esnek Üretim Sistemlerinin Tanımı

Günümüzde sadece kısmi entegrasyonlarının bulunduğu B.B.Ü. teknolojisinin tam manasıyla uygulanabilmesi için E.Ü.S. temel taşlardan birini teşkil etmektedir. Literatürde çok sayıda tanımını bulmak mümkün olan E.Ü.S.'nin genel anlamda kesin bir tanımını yapmak mümkün değildir.

E.Ü.S.'nin tanımı 1967 yılında Dalezek ve Rudolf tarafından şu şekilde ortaya konmuştur.

“Birbirlerine ortak bir kontrol ve taşıma sistemi ile bağlanmış ve bir tarafta otomatik üretimi gerçekleştiren, diğer tarafta farklı iş parçalarına farklı üretimi sağlayabilen üretim istasyonları kurulmasıdır.

Literatürde bulunan diğer tanımlardan bazıları ise şöyledir.

- Bir Esnek Üretim Sistemi merkezi bir bilgisayar altında mantıksal olarak organize edilmiş ve merkezi bir taşıma sistemi tarafından fiziksel olarak birbirine bağlanmış üretim ekipmanları topluluğudur (Parrish 1990).

Genelde bir E.Ü.S., bilgisayarlı nümerik kontrollü makinelerin, otomatik malzeme taşıma sistemi ile bağlantılı ve bilgisayar kontrolü altında çalışmasıdır (Browne 1984).

Bu tanımlamalardan yola çıkılarak E.Ü.S.' nin şöyle bir tanımını yapmak mümkündür.

*“Bir E.Ü.S., tasarımdan üretime, üretimden dağıtıma kadar her aşamada kullanıcılara çeşitli esneklik dereceleri ve verimlilik sağlayan merkezi bir bilgisayar altında organize edilmiş nümerik kontrollü makineler ve otomatik malzeme taşıma sistemlerinin oluşturduğu bir bütündür.”*

Bu tanımlamadan anlaşılacağı gibi E.Ü.S. etkenlik, esneklik, kalite ve verimlilik gibi birçok unsuru beraberinde getirmekte olup kullanıcılara üstün bir rekabet avantajı sağlamaktadır.

Önceki konularda bahsetmiş olduğumuz çeşitli esneklik derecelerine rağmen bütün E.Ü.S. aşağıdaki ortak karakteristiklere sahiptir (Maleki 1991).

- . **Bütünleştirme:** Sistem entegrasyonu tarafından belirlenen bir kurallar kümesi üzerine kurulu sistem bileşenlerinin birbirine bağlılığıdır.
- . **Zeka:** Verilen girdileri yorumlama ve kullanıcıların beklentilerine uygun çıktı üretebilme yeteneğidir.
- . **Çabukluk:** Sistemin değişikliklere tepki gösterebilme hızı olarak tanımlanabilir.

#### 4.3.3. Esnek Üretim Sistemlerinin Avantajları

Esnek Sistemlerinin etkili kullanımı ile elde edilebilecek çok sayıda avantajı mevcuttur. E.Ü.S.' nin uygulanması sonucu elde edilen faydayla bütünleştirilmiş iki grup karakteristik söz konusudur.

1. Karşılaşılan Günlük Problemlere Yanıt Verme
  - Mühendislik değişiklikleri
  - İşlem değişiklikleri
  - Makine elverişsizliği
  - Kesme aleti başarısızlığı
2. Uzun Dönemli Sistem Değişiklikleri
  - Değişen Ürün Hacimleri

- Farklı Parça Karışımları
- Yeni Ürün İlaveleri

Yukarıdaki karakteristiklerden yola çıkarak ulaşılmak istenen amaçlar ve bu amaçlara ulaşmak için kullanılan yöntemler şunlardır.

- Makinelerdeki operatörleri çıkartmak,
  - Adamsız işlemleri desteklemek için “Çekirdek Araç” sağlamak
- koşuluyla DÜŞÜK İŞGÜCÜ sağlamak,
- Kontrol dışı değişkenlerin sayısını azaltmak,
  - Plandan sapmaların çabuk farkına varacak ve tepki gösterecek araçlar tedarik etmek,
  - İletişimi güçlendirmek koşuluyla ETKİLİ İŞLEMSEL KONTROL sağlama,
  - Hazırlık zamanlarını ortadan kaldırmak,
  - İnsan müdahalesini otomasyona bırakmak,
  - Makineleri kesme çevrimleri içinde tutmak için hızlı aktarım araçları sağlamak koşuluyla ETKİLİ MAKİNE KULLANIMI sağlamak,
  - Parti büyüklüklerini düşürmek,
  - İş hacmini geliştirmek,
  - Tam zamanında üretim için planlama araçları geliştirmek koşuluyla DÜŞÜK ENVANTER sağlama (Maleki 1991).

Parrish (1990) ise E.Ü.S.’nin amaçlarını aşağıdaki gibi sıralamıştır.

1. Maliyetleri Düşürmek
  - Üretim araçlarının daha etkin kullanımı
  - Stokların düşürülmesi
  - Daha kısa işlem zamanları
  - Yarı mamul maliyetlerinin düşürülmesi
2. Teknik Performansı Arttırmak
  - Artmış üretim seviyeleri
  - Daha geniş ürün karışımı
  - Eş zamanlı ürün karışımı üretimi
  - Üretim sisteminin lojistik sistemle bütünleştirilmesi

- Küçük parti büyüklükleri
- Daha kısa veya sıfır deęişiklik ve hazırlık zamanları
- 3. Sipariř Hazırlıklarını İyileřtirmek
  - Daha kısa hazırlık zamanları/daęıtım zamanları
  - Üretim kapasitelerinin belirlenmesi
- 4. Gelecekteki Güvenlik İçin Yardım Sağlamak
  - Artan rekabet
  - Artan kalite
  - Daha iyi firma imajı

#### 4.3.4. Esnek Üretim Sistemi Teknolojisi

##### 4.3.4.1. Nümerik kontrollü tezgahlar

Nümerik kontrollü (NC) tezgahlar E.Ü.S.' nin temel yapı taşları olup, E.Ü.S.' nin esneklięi büyük ölçüde bu tezgahların esneklięine baęlıdır. Nümerik kontrollü tezgahlar, yapılacak işlemleri otomatik olarak gerçekleştirme temeline dayanan otomasyon türü olup ilk olarak 1952'de Amerika'da Massachusetts Institute of Technology tarafından icat edilmiştir. Nümerik kontrol kavramı ürünlerin daha etkili ve ekonomik üretimi ihtiyacı sonucu ortaya çıkmıştır. Nümerik kontrol teknolojisi temelde makinelerde kullanılmasına rağmen, ilerlemeler NC teknolojisini robotlar gibi otomatik sistemlerde kullanılan dięer ürünlerde kullanılabilir bir hale getirmiştir. Esnek üretim sistemlerinde kullanılan NC tezgahlar ise Bilgisayarlı Nümerik Kontrollü (CNC) tezgahlardır. Çok sayıda nümerik kontrollü (NC) tezgahın bir bilgisayar aracılıęıyla birbirine baęlanması sonucu ise Doğrudan Nümerik Kontrollü (DNC) tezgahlar şekillenmektedir (Maleki 1991).

Bilgisayarlı Nümerik kontrollü tezgahların bazı özellikleri ise şunlardır:

- Uyarlanabilir kontrol,
- Arıza tanıma,
- Otomatik veri toplama ve aktarma,
- Hata aktarma,
- Zaman verisi toplama ve aktarma,
- Paket kodu okuma,

#### 4.3.4.2.Otomatik malzeme taşıma sistemi

Malzeme taşıma sistemi hammaddenin üretim sistemine girişi ile mamulün üretim sisteminden çıkışı arasındaki tüm taşıma faaliyetlerini kapsamaktadır. Esnek üretim sistemlerinde malzeme taşıma sistemi şu elemanlardan oluşmaktadır:

1. Otomatik Güdümlü Araçlar
2. Robotlar
3. Konveyörler

Otomatik malzeme taşıma sistemlerinin gerçekleştirdiği faaliyetler ise şunlardır:

1. Farklı sistem ve alt sistemler arası taşıma
2. İş istasyonları arası taşıma
3. İş istasyonları içinde taşıma

##### 4.3.4.2.1. Otomatik güdümlü araçlar (OGA)

OGA' lar fabrika ortamında birim yüklerin bir yerden diğerine otomatik olarak taşınmasında kullanılan insansız araçlardır (Koff 1987).

Bir OGA Sisteminin elemanları şunlardır:

- Gelen işler
- Birim yükler
- OGA' lar
- Paletler: OGA üzerinde birim yüklerin konduğu taşıyıcılar
- Yollar: Atölye üzerinde malzeme akışını sağlayan yapılar
- Düğümler: Yolların kesişmesi ve yükleme, boşaltma işlemlerinin yapılması ile kritik bir önem kazanan noktalar
- İş istasyonları
- Kuyruklar
- Kontrol elemanları
- Performans elemanları (Bilge vd. 1996).

Bu sistem elemanları dahilinde bir OGA sistemi için genel olarak rotalama, çizelgeleme ve atama problemleri mevcuttur.

#### 4.3.4.3.Hiyerarşik bilgisayar ağı ve kontrol

Esnek üretim sistemleri farklı görevleri uyum içerisinde gerçekleştiren çok sayıda modülden meydana gelmiş bilgisayarla bütünleşik sistemlerdir. Her modüler sistem bir modüler kontrol sistemine ihtiyaç duyar ve farklı bileşenler bireysel kontrolcü üniteler tarafından kontrol edilir. Bu kontrolcüler görevlerini daha yüksek seviyede bir kontrolcünün denetimi altında gerçekleştirirler. Bu ise hiyerarşik kontrol sistemini şekillendirmektedir (Maleki 1991).

Bu hiyerarşik yapı içerisindeki bileşenlerin fonksiyonları ise şunlardır:

1. CNC donanımının kontrolü
2. Malzeme taşıma donanımının kontrolü
3. Sistemdeki parça akışlarının kontrolü
4. Sistem performansının ölçülmesinde kullanılan bilgilerin kontrolü.

Esnek üretim sistemi ana bilgisayarı ise aşağıdaki işlemleri gerçekleştirmektedir (Soon 1997).

1. Üretim emirleri kontrolü
2. Üretim ekipmanlarının hazırlanması
3. EÜS' nin üretim kontrolü
4. Sistem ve ekipmanın izlenmesi

Sonuç olarak bir EÜS' nde bulunan ekipmanları birincil ve ikincil olmak üzere iki grupta toplayarak aşağıdaki gibi özetlemek mümkündür (Parrish 1990).

- **Birincil Ekipman**

İş merkezleri

- Üniversal işleme merkezi
- Torna merkezleri
- Taşlama makineleri
- Delme makineleri

İşlem Merkezleri

- Yıkama makineleri
- Koordinat ölçüm makineleri
- Robotlu iş istasyonları



- Elle işlem istasyonları

• **İkincil Ekipman**

Destek İstasyonları

- Palet yükleme/boşaltma istasyonları

- Takım ayar alanı

Destek Ekipmanı

- Robotlar

- Palet tampon istasyonları

- Palet depoları

- Takım depoları

- Hammadde depoları

- Taşıma sistemi

- Taşıma üniteleri

#### 4.3.5. Esnek Üretim Sistemlerinin Sınıflandırılması

Literatürde E.Ü.S.' nin sınıflandırılmasına ilişkin bir takım yaklaşımlar söz konusudur. Bu yaklaşımlardan biri Kusiak (1985) tarafından yapılmıştır. Kusiak E.Ü.S.' ni beş sınıfa ayırmıştır.

**Esnek Üretim Modülü:** Bir CNC tezgahın, palet değiştirici ve parça güven stok alanının oluşturduğu bütün E.Ü.M.' dür.

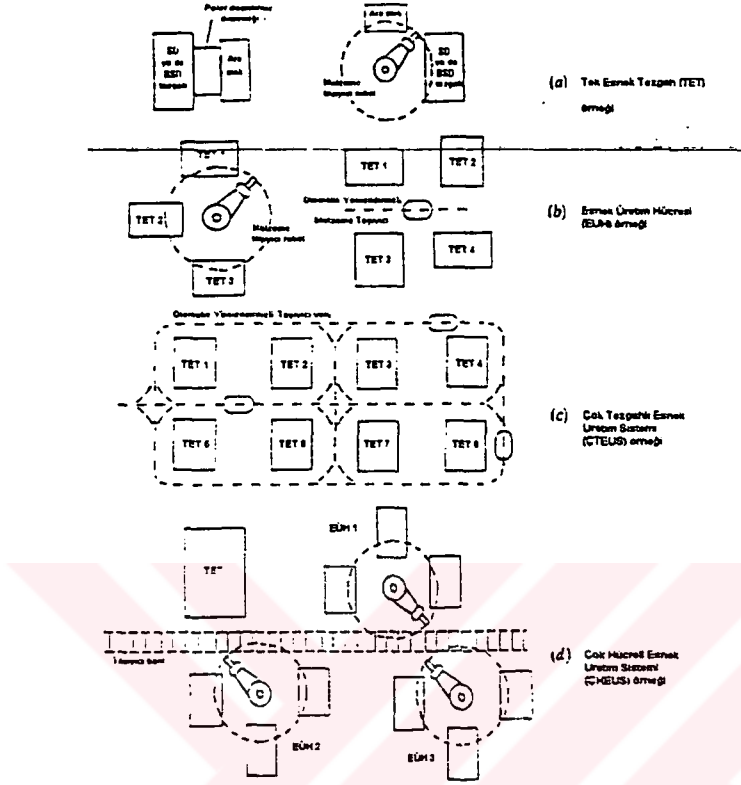
**Esnek Üretim Hücresi:** E.Ü.M.' lerinin bir araya gelmesi ve bazen bir otomatik malzeme taşıma sistemi ile birbirlerine bağlanması sonucu meydana gelen, üretilecek parça ailelerine uygun olarak tasarımı yapılan bir sistemdir.

**Esnek Üretim Grubu:** Aynı üretim ortamında merkezi bir bilgisayar kontrolü altında ve bir malzeme taşıma sistemi ile birleştirilmiş E.Ü.M ve E.Ü.H.' lerinin bir araya getirdiği üretim topluluğudur.

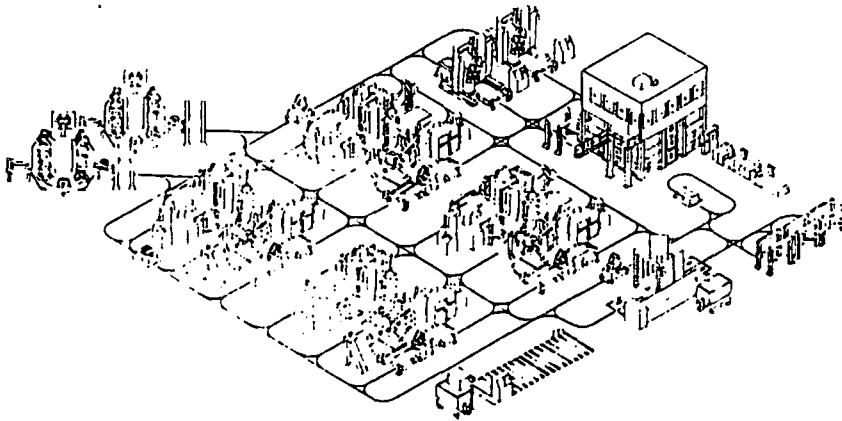
**Esnek Üretim Sistemi:** Fabrikasyon, imalat ve montaja yönelik farklı üretim gruplarının bir araya gelerek oluşturdukları sistemdir.

**Esnek Üretim Hattı:** İstasyon adı verilen tahsisli takım tezgahları topluluğudur (Kusiak 1985).

Esnek üretim sistemlerinin konfigürasyonları şekil 4 (Maccarthy ve B.L.J. Liu 1993)'de ve bir E.Ü.S. örneğinde şekil 5'de gösterilmektedir.



Şekil 4. Esnek Üretim Sistemi Konfigürasyonları (Maccarthy 1993)



Şekil 5. Esnek Üretim Sistemi Örneği (Maleki 1991)

#### 4.3.6. Klasik Üretim Sistemleri İle Esnek Üretim Sistemlerinin Karşılaştırılması

Mevcut Esnek Üretim Sistemleri ve eşdeğer klasik üretim sistemleri üzerinde yapılan gözlemler sonucunda iki sistem arasında Sekiz temel farklılık ortaya konmuştur (Kusiak 1988).

Bunlar:

1. Bir esnek üretim sistemlerindeki makinelerin ve malzeme taşıma sisteminin otomasyon derecesi eşdeğer klasik bir sisteme nazaran çok daha yüksektir.
2. Bir esnek üretim sistemi, klasik üretim sisteminden daha az sayıda makinelerden meydana gelmektedir.
3. Bir esnek üretim sistemindeki makinelerin yerleşimi kullanılan malzeme taşıma aracı tiplerine bağlıdır.
4. Bir işlem planındaki hazırlıkların sayısı esnek üretim sistemlerinde, klasik sistemlere göre önemli ölçüde küçüktür.
5. E.Ü.S.' nde her makinedeki işlem zamanı daha uzundur.
6. E.Ü.S.' ndeki bilgi hacmi ve akışı daha yüksektir.
7. E.Ü.S.' nde parti büyüklükleri, sipariş miktarı, kapasite ve takım ömrü gibi faktörler büyük ölçüde etkilidir.
8. E.Ü.S. tasarımı kendi operasyonları üzerinde bir etkiye sahiptir.

#### 4.3.7. Esnek Üretim Sistemleri ile İlgili Problemler

E.Ü.S. ile ilgili problemleri, tasarım ve işletim problemleri olmak üzere iki grupta toplanmaktadır.

1. Tasarım Problemleri
  - Parça ailelerinin seçilmesi
  - E.Ü.S üretim sisteminin seçilmesi
  - Malzeme taşıma sisteminin seçilmesi
  - Takım-tertibat ve paletlerin seçilmesi
  - Uygun bilgisayar sisteminin seçilmesi
  - Bu sistemlerin düzenlenmesi ve bütünleştirilmesi

2. İşletim Problemleri
  - Planlama
  - Gruplandırma
  - Çizelgeleme

#### 4.4. GRUP TEKNOLOJİSİ VE HÜCRESEL ÜRETİM SİSTEMLERİ

##### 4.4.1. Giriş

Grup teknolojisi (GT) fikri hazırlıkları, parti büyüklüklerini ve taşıma mesafelerini azaltmak amacıyla ilk olarak Rusya'da Mitrofanov tarafından tek makine fikri olarak ortaya çıkarılmıştır. GT, esas olarak yüksek verimlilikte bir sistem esnekliğini sağlamaya çalışmaktadır. Birçok benzer parça bazı tasarım özelliklerine göre gruplandırılarak bir makinede başarılı olarak işlenmiş ve böylece makine kullanım oranı %40'ın üzerine çıkarılabilmektedir. Daha sonraları bu fikir genişletilerek, benzer işlem sırasına sahip parçalar ve bu parçaları işleyecek makine grupları bir araya toplanarak Üretim Hücreleri oluşturulmuştur. Üretim hücreleri adamlı veya adamsız olabilmektedir. İnsan faaliyetlerini robotların gerçekleştirdiği adamsız üretim hücreleri "Hücreyel Esnek Üretim Sistemi" olarak adlandırılır.

Grup teknolojisinin temel fikri, üretimi gerçekleştirilecek olan ürünlerin tasarımlarındaki veya üretim aşamalarındaki birtakım benzerlerinden faydalanarak prosesteki benzerlikleri bir araya toplamak ve bu durumdan fayda sağlamaktır.

Grup teknolojisi fikri üretim sistemlerine uygulandığında Hücreyel Üretim Sistemleri ortaya çıkmaktadır. Çok büyük çeşitlilikte parçaların az miktarlarda üretiminin söz konusu olduğu üretim sistemlerinde, oluşturulan parça ailelerini işleyecek makineler de bir araya getirilerek üretim sistemi birkaç alt sisteme (Hücre) bölünmesi sonucu, seri üretimle elde edilen verimlilik seviyelerine ulaşılabilir.

H.Ü.S. sayesinde çok çeşitli ürünler yüksek verimlilik seviyelerinde üretilebilmektedir. Ancak bir parçanın üretim işlemlerinin bir hücrede tamamlanması gerekir.

Parçalar arasında çeşitlilik ve miktar farklılıklarının yanı sıra Hücresel Üretim Sisteminin (H.Ü.S.) önemli diğer bazı yönleri de vardır. Bunlar hücrenin yapısına bağlı olarak farklılık gösterecek olan proses planlama, üretim planlama ve makine yükleme faaliyetleridir. Greene ve Sadowski (1984) H.Ü.S.'nin karmaşıklığını "makine yoğunluğu" ve "iş yoğunluğu" terimleri ile tanımlamışlardır.

Makine yoğunluğu hücreler arasındaki makine tipi ortaklığı olarak ifade edilerek aşağıdaki hücre karakteristiklerine bağlıdır.

- Hücre sayısı
- Her hücredeki makine tipi sayısı
- Toplam farklı makine tipi sayısı
- Artık hücre

İş yoğunluğu ise işlerin uygun olarak atanabildiği hücrelerin oranı olarak tanımlanır ve her iş için gerekli operasyon sayısı ve iş tipi sayısı gibi karakteristikleri kapsamaktadır. Aynı zamanda hücre karakteristiklerini de kapsamaktadır.

#### 4.4.2. Grup Teknolojisinin Sistem Performansı Üzerindeki Etkileri

Grup teknolojisi ve hücresel üretim sisteminin sistem performansı üzerindeki etkilerini şöyle sıralamak mümkündür.

1. Bir parçanın bir hücre içerisinde bütün işlerinin tamamlanması nedeniyle parça taşıma süresi ve mesafenin minimum olması.
2. Hücre içindeki makinelerin yakınlığı ve akış sonucu çevrim zamanın azalması.
3. Hücrelerde benzer parçaların benzer makinelerde işlenmesi nedeniyle değişiklikler küçük olmakta, bu da hazırlık zamanlarını düşürmektedir.
4. Hazırlık zamanlarının düşmesi sonucu küçük parti büyüklüklerinin ekonomik olması ve üretim akışının düzgünleşmesi.
5. G.T. işlenmekte olan işleri ve stokları düşük seviyede tutmaktadır.
6. Hücrenin bir parça tipini tahmin edilen oranda üretme kapasitesi sonucu elde edilen daha doğru ve güvenilir teslim zamanları.
7. H.Ü.S. optimum makine kullanımını sağlamaktadır.

8. Parçaların bir hücre içerisinde tamamen işlenmesini sağlamak için ek makine yatırımları gerekebilmektedir.

9. Hücredeki düşük makine kullanım oranı çok fonksiyonlu işgücü kullanımını mümkün kılmaktadır.

10. İşlemlerin küçük bir alanda gerçekleşmesi geri beslemeyi kolaylaştırmakta ve bu da kaliteyi artırmaktadır.

11. İşlenmekte olan işlerdeki azalma nedeniyle makine ilaveleri ve genişleme için dikkate değer büyüklükte boş alan mevcut bulunabilmektedir (Singh ve Rajamani 1996).

#### **4.4.3. Grup Teknolojisinin Diğer Fonksiyonel Alanlara Etkisi**

1. G.T. uygulaması, benzer yeni parçaların belirlenmesine, çeşitliliğin düşürülmesine, standardizasyonun yükseltilmesine ve yeni parça tasarımı sayısının azalmasına yardımcı olmaktadır.

2. G.T. akış kontrolü için kendini uygun kılan bir takım karakteristiklere sahiptir. Böylece parçaların üretimi kısa dönemli talebe göre yapılmakta ve sabit sıklıklarla değişken parti büyüklükleri üretilmektedir. Fakat talebin belirsiz olduğu durumlarda stok kontrol politikası benimsenebilir.

3. G.T. ile üretim planları standartlaştırılabilmektedir.

4. G.T. ile önleyici bakım programı gerekli hale gelmektedir. uygun eğitim sayesinde düzenli bakım faaliyetleri geliştirilebilmekte ve tezgah ömürleri artırılabilir.

5. Zaman, performans ve yatırım açısından daha doğru maliyetlendirme bilgileri elde edilebilmektedir, çünkü her hücre bir maliyet merkezidir.

6. G.T. uygulaması sayesinde satın alma ve satış faaliyetlerinde de birtakım iyileşmeler meydana gelmektedir (Singh ve Rajamani 1996).

#### **4.4.4. Grup Teknolojisinin Diğer Teknolojilere Etkisi**

1. G.T., bir tesisteki pahalı NC makinelerin ekonomik ayarlanmalarını destekler.

2. Esnek Üretim Sistemleri G.T. temeli üzerine inşa edilmektedir.

3. G.T. aynı zamanda Bilgisayar Destekli Üretim, Malzeme İhtiyaç Planlaması, Tam zamanında Üretim ve Eş Zamanlı Mühendislik Uygulamalarının gerçekleştirilmesi için ideal bir teknolojidir.

Grup teknolojisi bir işletmedeki bütün alanları etkileyen bir yöntem stratejisidir. Başarılı bir şekilde uygulanabilmesi için sistem performansı, departmanlar ve diğer teknolojiler üzerindeki etkileri anlaşılmalıdır (Singh ve Rajamani 1996).

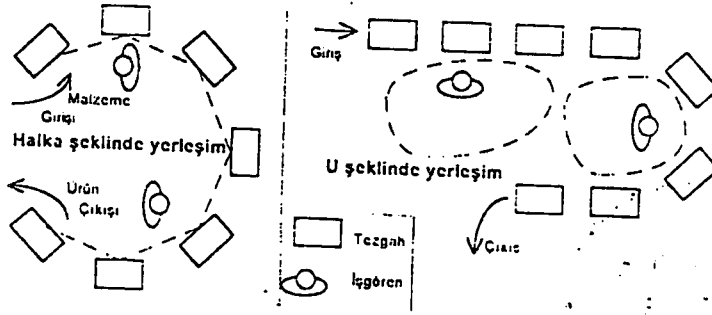
Atölye tipi üretimde parçaların iş istasyonları arasında defalarca hareket etmesi gerekmektedir. Bu da beraberinde yüksek malzeme aktarımını, yüksek süreç içi stokları, yüksek hazırlık zamanlarını ve sonuç olarak da yüksek üretim maliyetlerini beraberinde getirmektedir. Ayrıca atölye tipi üretimde geçerli olan bir makineye bir işçi kuralı da düşük tezgah kullanımı, kısa taşıma mesafeleri ve hücre içi kontrolün kolaylığı gibi sebeplerden dolayı H.Ü.S. 'nde geçerli olmayıp, bir işçi birden çok tezgahı kullanabilmektedir.

H.Ü.S. uygulamalarının temel amaçlarını ise şu şekilde sıralayabiliriz.

- Parça akışlarının basitleştirilmesi
- Tedarik, malzeme taşıma ve hazırlık sürelerinin düşürülmesi
- Proses içi stokların düşürülmesi
- Verimliliğin artırılması
- Üretim planlama ve kontrolün kolaylaştırılması
- Kalitenin artırılması

#### 4.4.5. Hücresel Üretimde Yerleşim Şekilleri

Hücresel üretimde tezgahlar parçaların işlem sıralarına göre yerleştirilerek, işgörenlerin hücre içerisinde tezgahlara parça yükleyerek ve parça boşaltarak hareket edebilmektedirler. H.Ü.S.' nde tezgahlar genellikle U biçiminde yerleştirilmektedir. Şekil 6 hücresel üretimde yerleşim şekillerini, Şekil 7 de adamlı bir üretim hücrelerini göstermektedir.



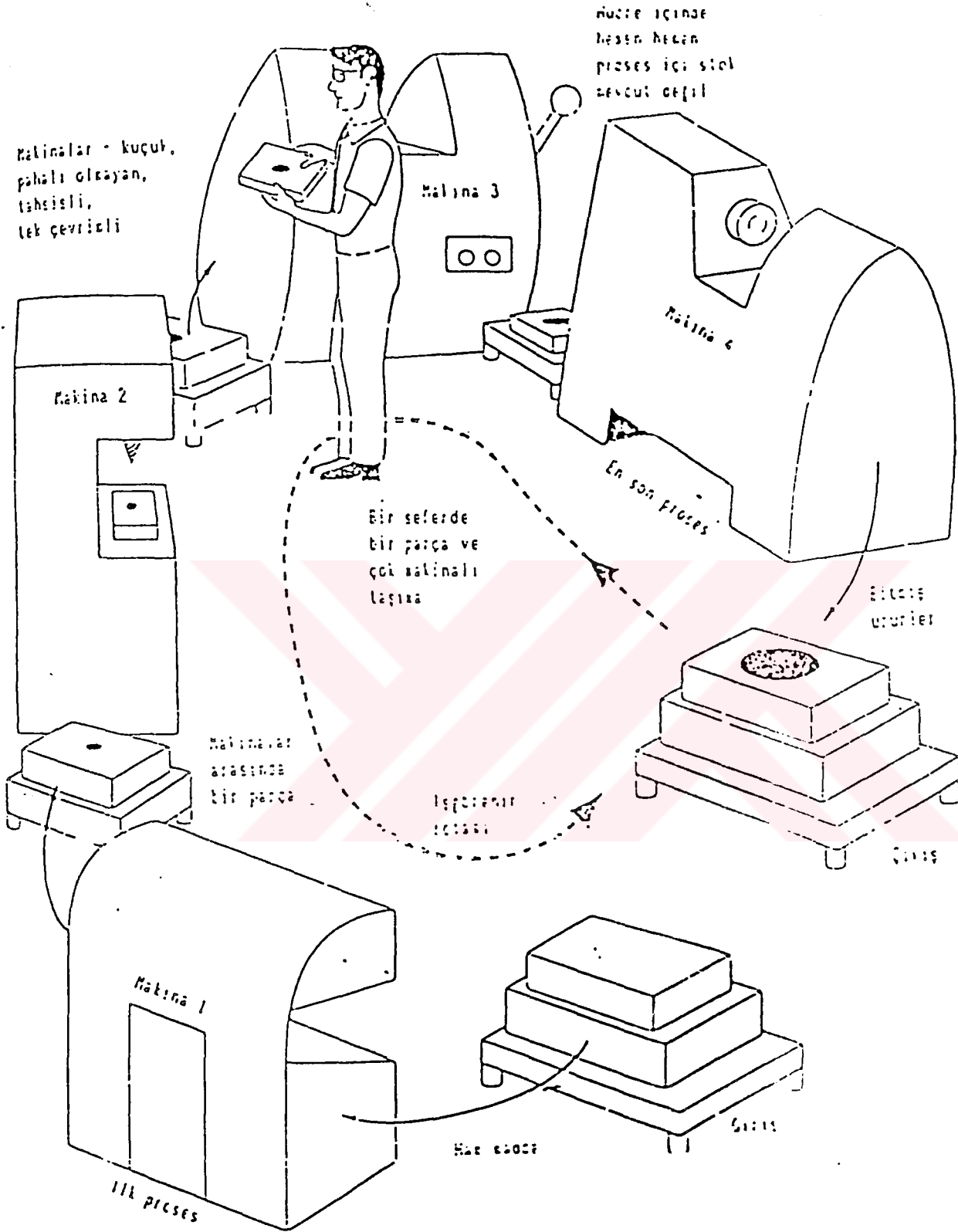
Şekil 6. Hücresel üretimde yerleşim şekilleri

#### 4.4.6. Grup Teknolojisinde Kodlama Sistemleri

Parça aileleri, üretilen parçaların geometrik şekli, boyutları ve malzeme tipi gibi tasarım özellikleri veya operasyon sıraları, parti büyüklüğü, işlem süreleri ve üretim hacimleri gibi üretim özellikleri dikkate alınarak oluşturulmaktadır. Bir üretim sisteminin etkinliğinin ve verimliliğinin artırılabilmesi için üretim hazırlık sürelerinin düşürülmesi, etkili proses kontrolünün sağlanması, üretim akışının geliştirilmesi vb. iyileşmelerin sisteme kazandırılması gerekmektedir. Üretilen parçaların gruplandırılarak benzer parçaların ortak parça ailelerinde toplanması sayesinde, benzer faaliyetler birlikte gerçekleştirilebilmekte, benzer görevler standartlaştırılabilmekte, tekrarlı problemler hakkındaki bilgiler toplanabilmekte ve bunların sonucunda sistemin etkinliği artırılabilir.

Parçalara ait yukarıda belirtilen tasarım ve üretim özellikleri hakkındaki geniş bilgiler parça kodlarında saklanmaktadır. Bir parçaya ait kod, o parçaya ait





Şekil 7. Adamlı bir imalat hücresi

bilgilerin saklandığı karakterler sırası olarak tanımlanabilir. Bu parça kodlarını oluşturan rakamların birbirleri ile bağlanma türüne göre parça kodları üçe ayrılır.

**a) Monokod:** Bu kodlarda her bir basamağın anlamı bir önceki basamağa bağlıdır. Koddaki basamaklar birbirinden bağımsız olarak düşünülemez. Bu sayede monokodlarda kısa yapılar içinde büyük bilgiler saklanabilmektedir. Kısa olmalarına rağmen oluşturulmaları ve anlaşılması zordur.

**b) Polikod:** Bu kodlarda basamaklar birbirlerinden bağımsızdırlar. Her basamak ürüne ait ayrı bir özelliği temsil etmektedir. Bu kodların oluşturulması ve anlaşılması kolaydır fakat uzun kodlardır.

**c) Karışık Kodlar:** Polikodların ve monokodların avantajlı yönlerinin bir arada bulunduğu kodlardır. Çok sık kullanılan ve çok karmaşık olmayan özellikler polikodla, diğerleri ise monokodla verilerek iki sistem birleştirilir.



## 5. HÜCRESEL ÜRETİMDE HÜCRE FORMASYONU PROBLEMİ

Literatürde hücre formasyonu problemi veya makine-parça grup analizi olarak adlandırılan üretim sistemlerinde grup teknolojisi uygulamaları, parça ailelerinin ve makine gruplarının oluşturulması ile başlamaktadır. Parça aileleri ve bu parça ailelerini işleyecek makine grupları oluşturulurken dikkat edilmesi gereken en önemli husus, parça ailelerinin sadece bir makine grubu içinde işlem görmesini sağlamak yani hücreler arası etkileşimi en aza indirmektir.

Çok sayıda araştırmacı hücre formasyonu problemini ele alarak çok çeşitli prosedürler geliştirmişlerdir. Bu prosedürlerin büyük bir kısmı "0" ve "1" lerden oluşan makine-parça ilişki matrisini kullanmaktadırlar. Eğer bir parça bir makinede birden fazla sayıda operasyon görüyor ise bu durum göz ardı edilir.

Literatür araştırması kısmında bahsedildiği gibi, hücre formasyonu probleminin çözümüne yönelik olarak geliştirilen yöntemleri 7 ana başlık altında toplamaktaydık. Bu çalışma kapsamında bu yöntemlerden "Yapay Zeka Yöntemleri" ve "Sezgisel Yöntemleri" detaylı olarak inceleyeceğiz. Şimdi sırasıyla bu yöntemleri ele alalım.

### 5.1. Yapay Zeka Yöntemleri

#### 5.1.1. Yapay zeka

Yapay zeka insanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak tanımlanır. Daha da genişletecek olursak yapay zeka, bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekasına özgü kapasitelerle programlanmış bilgisayarın düşünme girişimidir. Yapay zeka konusunda ilk çalışma, herhangi bir tanımlı fonksiyonun sinir hücrelerinden oluşan ağlarla hesaplanabileceği ve mantıksal "ve" ve "veya" işlemlerinin gerçekleştirilebileceği fikrinin McCullah ve Pitts tarafından ortaya atılmasıdır. Daha sonra yapılan çalışmalar beraberinde ticari uygulamaları da getirmiş ve yapay zeka bir endüstri haline gelmiştir. Bu çalışmalar sonucunda şu yaklaşımlar ortaya çıkmıştır (Brown 1995).

Yapay zekayı oluşturan bileşenleri ise aşağıdaki gibi sıralamak mümkündür.

1. Uzman Sistemler
2. Bilginin Sunulması
3. Problemlerin Çözülmesi
4. Bilgisayarlarla Tabii Dilde İlişki Kurma
5. Öğretim
6. Kognitiv Modelleme
7. Görünen Bilginin İşlenmesi ve Robot Tekniği
8. Yapay Sinir Ağları

Yapay zeka bilgisayarları daha kullanışlı yapmakta ve endüstrilerde çalışanlar için bir çok avantaj arz etmektedir. Yapay zeka tekniklerinin birçok farklı alanda uygulamaları mevcut olduğu gibi, bir üretim sisteminde karşılaşılan birtakım problemlere de uygulandığı da görülmektedir. Bu problemlerden biri de hücreyel üretim sistemlerinde makine-parça grup analizleridir.

Yapay zekayı günlük yaşantıda ve üretim sistemlerinde karşılaşılan problemlerin çözümünde etkili olarak kullanılacak bir araçlar kutusu olarak düşünebiliriz. Bu araçlar kutusunda bulunan yapay zeka teknikleri ise şunlardır:

1. Uzman Sistemler
2. Genetik Algoritmalar
3. Yapay Sinir Ağları
4. Bulanık Mantık

Aşağıdaki bölümlerde sırasıyla bu teknikler incelenerek, bu tekniklere dayalı olarak geliştirilmiş ve makine-parça grup analizi probleminin çözümünde kullanılan prosedürler aynı bölüm içerisinde verilecektir.

Bu çalışma kapsamında incelenen ve yapay zeka tekniklerine dayalı olarak geliştirilen prosedürler şunlardır:

Tablo 2. Analiz edilecek olan YZ tabanlı prosedürler

Yapay Zeka Tekniği	Kullanılan Model	Kaynak
Yapay Sinir Ağları	Adaptive Resonance Theory (ART1)	Liao ve Chen 1993
Yapay Sinir Ağları	Bulanık ART (Fuzzy ART)	Suresh veKaparathi 1994
Yapay Sinir Ağları	Yarışmalı Eğitim (Competitive Learning)	Chu 1993
Bulanık Mantık	Bulanık Küme Tabanlı Yaklaşım	Leem ve Chen 1996

Tablodan da anlaşılacağı gibi genetik algoritmalar ve uzman sistemler çalışmamız kapsamında yer almamaktadır. Yukarıda verilen dört prosedür ait oldukları YZ tekniği ile birlikte detaylı olarak anlatılacaktır.

### 5.1.1.1. Uzman sistemler

#### 5.1.1.1.1. Tanım

Yapay zekanın bir alt dalı olan uzman sistemlerin çeşitli tanımlarını yapmak mümkündür. Bu tanımlardan bazıları uzman sistemlerin fonksiyonları , bazıları yapısı , bazıları da hem fonksiyonları hem de yapısı üzerine kuruludur.

\*Bir uzman sistem , bazı faydalı yönlerden bir insan uzman gibi davranan bir bilgisayar programıdır (Winston & Prendergast 1984 ).

\*Uzman sistemler , dar bir problem alanında problemleri etkili ve verimli bir biçimde çözerler (Waterman 1986).

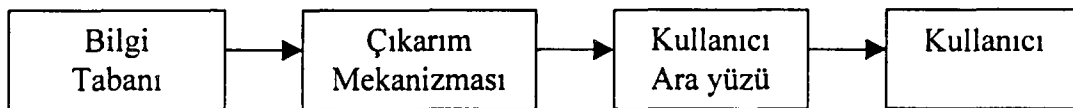
\*Uzman sistemler , sembolik olarak ifade edilebilen problemler ile ilgilidir (Liebowitz 1988).

\*Uzman sistemler , uzman bilgisini zor gerçek dünya problemlerine uygularlar (Waterman 1986).

\*Uzman sistemler ,çözümleri için önemli uzman bilgisi gerektiren problemleri çözerler (Feigenbaum & King 1985).

\*Uzman sistemler ,zeki tavsiye verebilir veya bir işleme fonksiyonu hakkında zeki karar verebilirler (BCSSGF 1984).

Bir uzman sistemi oluşturan bileşenler Şekil 8' de verilmiştir.



Şekil 8. Uzman sistemlerin temel bileşenleri

Uzman sistemlerin, üretim, tasarım, tıp, eğitim, vb. birçok alanda uygulamaları ile karşılaşmamız mümkündür. Bir üretim ortamında uzman sistemlerden aşağıdaki konularda faydalanabiliriz.

- arıza tespiti
- proses izleme

- planlama ve çizelgeleme
- kontrol ve benzetim

#### 5.1.1.1.2 Uzman sistemlerin avantaj , dezavantaj ve sınırları

##### a) Avantajları :

- Maliyet azalması
- Verimlilik artışı
- Kalite iyileştirmesi
- İşleyiş hatalarını azaltma
- Esneklik
- Daha ucuz cihaz kullanımı
- Tehlikeli çevrelerde işlem
- Güvenilirlik
- Cevap verme süresi
- Tam ve kesin olmayan bilgi ile çalışma
- Eğitim
- Problem çözme kabiliyeti
- Sınırlı bir sahada karışık problemlerin çözümü

##### b) Dezavantajları :

- Alışılmamış durumlara cevap verememe
- Değişen duruma kolay adapte olamama
- Sadece sembolik girdilere bağlı olma
- Uzmanlık alanının dışında başarısızlık

##### c) Sınırları :

Uzman sistemlerin ticari olarak yayılamamasının önündeki bazı problemler şunlardır:

- Bilgi her zaman okunabilir nitelikte değildir.
- İnsanlardan bilgi almak zordur.
- Uzman sistemler ancak sınırlı sahalarda , bazı durumlarda ise çok sınırlı sahalarda iyi çalışabilirler.
- Yardım için bilgi mühendisine ihtiyaç gösterir. Bilgi mühendisi az bulunur ve pahalı olduğu için sistemin maliyetini yükseltir.

- Sistemin maliyeti ve geliştirme süreci engelleyici bir faktördür.
- Herhangi bir uzmanın durum değerlendirmesi için yaklaşımı farklı bile olsa doğru olmalı.
- Çok tecrübeli bir uzman bile olsa , zaman baskısı altında olduğu zaman iyi bir durumsal değerlendirme yapması zordur.

Belirli bir konuda uzman sistem geliştirmeden önce dikkat edilmesi gereken bazı hususlar vardır. Bunları şöyle sıralamak mümkündür (Basu vd 1995):

1. Bilgi tabanının oluşturulabilmesi için, ilgilenilen alana ait yeterli bilginin mevcut olması.
2. Problem alanını inceleyen uzmanların büyük bir kısmının fikir birliği içinde olmaları.
3. İncelenen alanla ilgili bilgilerin istikrarlı olması.
4. Sistemin incelenen kısmının, meydana gelecek gelişmelerin maliyetini karşılayacak ölçüde geniş olması.

H.Ü.S. tasarımında da uzman sistemlerden yararlandığını görmekteyiz. Kusiak(1988), Maus ve Keyes (1991) ve Basu vd (1995), üretim hücrelerinin oluşturulması aşamasında sistem tasarımını etkileyen çeşitli faktörleri de dikkate alarak uzman sistem yaklaşımını H.Ü.S. tasarımına uygulamışlardır.

#### 5.1.1.2. Genetik algoritmalar

Genetik Algoritmalar isimli çözüm arama tekniği 1960'lı yıllarda John Holland tarafından ortaya atılmış fakat 1989 yılında David E. Goldberg' in bu tekniği gaz boru hatlarının denetiminde kullanmasına kadar pek faydalı bir teknik olduğu düşünülmemişti. Bu çalışmayı takiben genetik algoritmalar (GA) çok çeşitli alanlarda özellikle bir optimizasyon tekniği olarak sıkça kullanılmaya başlamıştır.

GA canlılardaki evrim sürecini taklit etmekte olup temeli rasgeleliğe dayanmaktadır. GA' da söz konusu problemin çözümüne ulaşabilmek için, her evrim süreci sonunda çözüm için en iyi olan birey topluluklarının bir sonraki nesle

aktarılması gerekmektedir. Söz konusu problem öncelikle maksimizasyon veya minimizasyon tipinde bir ifadeye indirgenerek genetik prosese tabi tutulur. Aktarımların doğru olarak yapılabilmesi için aşağıdaki operatörlere ihtiyaç vardır.

\* **Üreme** : GA' ın temel yapısını teşkil eden seçici bir operatördür. Bu operatörün kullanılmasındaki ana prensip, problem için tanımlanan kriterlere uygun olarak daha önceden belirlenen kodlama sistemi yardımıyla rasgele değer atamaktır. Burada dikkat edilmesi gereken husus çok az sayıda birey topluluğuyla işlem yapılmamasıdır.

\* **Çaprazlama** : Üreme operatörü ile elde edilen en iyi birey toplulukları. GA' ın yapısı içerisinde çaprazlama işlemine tabi tutulur. Çaprazlama operatörünün yardımıyla bireyler arasında bir gen aktarımı yapılacak ve böylece bir önceki nesle nazaran çok daha iyi sonuç vermesi beklenen yeni birey yapıları elde edilmiş olacaktır.

\* **Mutasyon** : Üreme ve çaprazlama operatörleriyle işlem yapıldığında, daha önceden tespit edilen hata değerine ulaşamama durumunda veya tavsiye edilen evrim süresinin bittiği durumlarda başvurulmuş ve birey elemanlarının gen yapılarını değiştirerek, çözümsüzlük durumlarının atlatılmasında yardımcı olan bir GA operatörüdür.

Bir problemin çözümü için muhtemel aday çözümler dizi yapısında ifade edildikten ve hedeflenen çözüm için bir değerlendirme fonksiyonu belirlendikten sonra aşağıdaki adımlar izlenerek sonuç elde edilmeye çalışılır.

**Adım 1.** Rasgele veya bilinen çözümleri içeren ve uygunluk değerleri hesaplanmış bir başlangıç kümesi (popülasyon) oluşturulur.

**Adım 2.** Algoritmanın durdurma kriteri veya iterasyon sayısı belirlenir.

**Adım 3.** Başlangıç çözümleri, probleme bağlı olarak belirlenen temel genetik operatörler yardımıyla evrimden geçirilerek yeni çözümler oluşturulur.

**Adım 4.** Ortaya çıkan aday çözümlerin iyilik ve kullanılabilirlik derecesi, önceden belirlenen değerlendirme fonksiyonu kullanılarak hesaplanır.

**Adım 5.** Hedeflenen çözüme en uzak olan aday çözümler, yeni bireylere yer açmak amacıyla popülasyondan çıkartılır.

**Adım 6.** Durdurma kriteri sağlanıyor ise adım 7'ye sağlanmıyor ise adım 3'e git.



## Adım 7. Bitir.

Moon ve Gen (1999), Rao vd. (1999), Chan vd.(1998), Gupta vd.(1996), Su ve Hsu (1996). genetik algoritmaları hücre formasyonu problemine uygulamışlardır.

Moon ve Gen (1999) geliştirdikleri prosedürde. alternatif proses planları. makine duplikasyonları. işlem zamanları, makine kapasiteleri. üretim hacmi. hücre büyüklüğü ve hücre sayısı gibi parametreleri dikkate almışlardır. Hücre formasyonu problemi 0-1 tamsayılı programlama modeli olarak formüle edilmiş ve genetik algoritma tabanlı bir yaklaşımla çözülmüştür.

### 5.1.1.3. Yapay sinir ağları

#### 5.1.1.3.1. Bir sinir ağı nedir ?

Canlılarda en gelişmiş sinir sistemi insanda bulunmaktadır. Sinir sistemi endoktrin sistemle birlikte çalışarak insan organizmasının bütünlüğünü sağlar. Sinir sistemi sinir hücrelerinden oluşmakta olup, insan beynindeki sinir hücrelerinin sayısı 14 milyar civarındadır. Bir sinir hücresinde akson. dendritlerle gelen uyarıyı hücre gövdesinden başka nöronlara ve efektör organlara iletimde görev yapar.

Bir sinirsel ağ tecrübesel bilgiyi saklayabilen ve onu kullanabilir yapan paralel dağılımlı bir işlemcidir. Bir sinir ağı iki yönden beyne benzer:

1. Bilgi ağ tarafından bir öğrenme prosesi yoluyla elde edilir.
2. Sinaptik ağırlıklar olarak bilinen ara nöronun bağlantı güçleri bilgi depolamak için kullanılır.

Yapay sinir ağlarında sinaptik ağırlıkları arzu edilen tasarım düzeyine ulaştırmak için değiştiren prosedüre ise "Öğrenme Algoritması" adı verilir.

Şekil 9, biyolojik bir sinir ağını ve bir yapay sinir ağını göstermektedir.



Şekil 9. (a)Biyolojik sinir ağı ve (b)yapay sinir ağı (Vemuri 1992)

### 5.1.1.3.2. Nöronun matematik modeli

Bir nöron ağını matematiksel olarak modellerken;

- Bir nöronun girişini etkileyen potansiyel  $X_1, X_2, \dots, X_n$  ( $X$  vektörü) olarak alınır.
- Sinapsın efektifliğine benzer  $W_1, W_2, \dots, W_n$  ağırlıkları alınır.
- Nöronun girişleri ve sinaps ağırlıkları çarpılarak toplanır. Böylece nörondaki toplam potansiyel seviyesi;

$$U_K = \sum_{i=1} W_i X_i$$

$$Y_K = \delta (U_K - \theta_K)$$

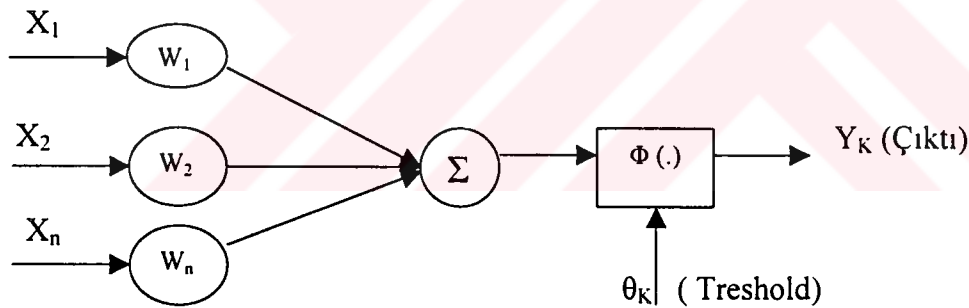
$X_i$  = Girdi Sinyalleri

$W_i$  = Sinaptik Ağırlıklar

$\theta_K$  = Treshold

$Y_K$  = Çıktı

Şekil 10, nöronun matematiksel modelini göstermektedir.



Şekil 10. Nöronun Matematiksel Modeli (Haykın 1994)

Threshold ayrı bir dışsal parametre olarak düşünülür. Burada Thresholdun sinaptik ağırlığı  $W_0$  ve girdi değeri de  $X_0$  olarak alınır.

$$X_0 = -1 \text{ ise } W_0 = \theta_K \text{ (Treshold)}$$

$$X_0 = +1 \text{ ise } W_0 = b_K \text{ (Bias)}$$

### 5.1.1.3.3. Aktifleşme fonksiyonları

#### a) Eşik (Threshold) Fonksiyonu

Aktifleşme fonksiyonumuzu  $\Phi(v)$  olarak tanımlarsak;

$$\Phi(v) = \begin{cases} 1 & \text{eğer } v \geq 0 \\ 0 & \text{eğer } v < 0 \end{cases}$$

Eşik fonksiyonunu kullanırken k. Nöronun çıktısı ise aşağıdaki gibi ifade edilir.

$$Y_k = \begin{cases} 1 & \text{eğer } v \geq 0 \\ 0 & \text{eğer } v < 0 \end{cases}$$

Burada

$$V_k = \sum_{i=1}^p W_{kj} X_j - \theta_k$$

#### b) Parçalı Doğrusal (Piecewise- Linear) Fonksiyon

$$\Phi(v) = \begin{cases} 1 & \text{eğer } v \geq 1/2 \\ v & \text{eğer } -1/2 > v > 1/2 \\ 0 & \text{eğer } v \leq -1/2 \end{cases}$$

#### c) Sigmoid Fonksiyon

Sigmoid fonksiyon YSA oluşturulmasında en yaygın kullanılan fonksiyondur.

$$\Phi(v) = \frac{1}{1 + \exp(-av)}$$

a: Sigmoid fonksiyonun eğim parametresi

Bir sigmoid için hiperbolik tanjant fonksiyonu kullanabiliriz.

$$\Phi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)}$$

#### 5.1.1.3.4. Yapay sinir ağlarının uygulama alanları

Yapay sinir ağlarının en yaygın kullanım alanlarını şu şekilde sıralamak mümkündür (Patterson 1996).

1) *Kısıt Tatmini*: Problemlerin birçok tipi için çözümler, sistemi tanımlayan kısıtlı değişkenlerin sadece belirli değişkenler olmasını gerektirir. Örneğin “gezgin satıcı problemi” hem optimizasyon hem de kısıt tatmini problemidir. Bu tür problemlere çözümler bazı YSA mimarileri kullanılarak bulunmuştur.

2) *İçerik Adreslenebilir Hafıza*: YSA mimarilerini bazıları hafıza gibi hareket etmeyi ve bileşik desenlerle sunulduğunda tekrar elde edilen desenleri depolamayı öğrenebilirler. Geri çağrılan desen depolanan desenin aynısı, biraz değişikliği veya tamamen farklı bir desen olabilir.

3) *Kontrol*: Yapay sinir ağları, sürücüsüz görevler de dahil olmak üzere mobil robotların kontrolü için öğrenmede etkili bir biçimde kullanılmıştır. Kimya, elektrik vb alanlarda birçok kontrol uygulamaları olup. Bu alandaki uygulamalar sınırsız görünmektedir.

4) *Veri Sıkıştırma*: Büyük miktarda veri toplanırken ve işlenirken yapay sinir ağları kullanılarak desenler kendinden daha küçük bir dizine aktarılabilir.

5) *Teşhis*: Teşhis, tıp, mühendislik ve üretim gibi birçok alanda uygulaması olan ortak bir yapay sinir ağı uygulamasıdır. Sınıflandırma problemlerinden biridir.

6) *Tahmin*: Kabul edilebilir doğruluk düzeyinde bir tahmin yapabilme için bir YSA “eski desen/gelecek çıktı” çiftlerin makul örnekleri kümesiyle eğitilmesidir.

7) *Genel Eşleme*: YSA’ nın en göze çarpan karakteristiklerinden biri de, öğrenme örneklerinin bir kümesinden keyfi fonksiyonları öğrenme kabiliyetidir. Bir YSA n dizinlik girdi vektörünü bazı kriterlere göre m dizinlik çıktı vektörüne aktarmayı öğrenebilir.

8) *Multisensör Veri Birleştirme*: Bu işlemde tek kaynaktan bilgi elde etmekten kaçınmak için duyma, görme, dokunma gibi hisleri beraber kullanması gibi.

### 5.1.1.3.5. Öğrenme kuralları

Yapay sinir ağlarının temel yapısı, beyne sıradan bilgisayardan daha çok benzemektedir. Yinede birimleri gerçek nöronlar kadar karmaşık değil ve ağların çoğunun yapısı, beyin kabuğundaki bağlantılarla karşılaştırıldığında büyük ölçüde basit kalmaktadır.

YSA' daki her işlem birimi basit anahtar görevi yapar ve şiddetine göre gelen sinyalleri söndürür yada iletir. Böylece sistem içindeki her birim belli bir yüke sahip olur. Yapay sinir ağları beynin bazı fonksiyonları ve özellikle öğrenme yöntemlerinin benzetim yolu ile gerçekleştirmek için tasarlanır. Ağların eğitimi için çeşitli algoritmalar geliştirilmiştir.

Bir nöronu tasarlamak için önce katmanlarının sayısını, sonra gizli katmandaki nöronların sayısını, daha sonra da ilişkilerin ağırlıklarını belirlemek gerekir. Bir nöron ağı ilk verildiğinde ağırlık katsayıları rasgele verilir ve bu yüzden ağ henüz doğru çalışmaya hazır değildir. Ağın doğru olarak çalışması için eğitilmesi yeni ağırlık değerlerinin uygun bir değere getirilmesi gerekmektedir. Bu işi ise eğitme algoritması gerçekleştirir. Eğitim süreci yüksek makine süresi talep etmektedir.

Eğitme yöntemlerini dört sınıfa ayırabiliriz.

1) **Danışmansız Eğitim:** Eğitimin amacı ağın uygun parametrelerini değiştirmekle giriş vektörü kümesine özgü olabilecek konuma uygunlukları belirlemektir. Bu sınıf eğitimde en çok kullanılan algoritma "Yarışmalı Eğitim Algoritması" dır.

2) **Danışmalı Eğitim:** Bu yöntemde giriş vektörleri kümesi ve onlara uygun çıkış vektörleri önceden bellidir.

3) **Topolojik Değişiklik Eğitim:** Bu yönde ağda ilişki ve ilişkisiz (serbest) nöronlar mevcuttur. Eğitim sürecinde serbest nöronlardan bazıları ağa eklenerek yeni ilişkiler oluşturulabilir.

4) **Stokastik Eğitim:** Eğitmenin amacı elemanların çalışmasının cari ihtimaller paylar paylanması arzu edilene en çok yakınlaştırılmasıdır.

### 5.1.1.3.6. Hücresel üretim sistemi tasarımında yapay sinir ağları uygulamaları

Yapay sinir ağlarını genel olarak tanıttıktan sonra, bu çalışma kapsamında incelenecek olan yapay sinir ağları tabanlı hücre formasyonu prosedürlerini sırasıyla inceleyebiliriz. Önceki kısımlarda da belirtildiği gibi bu prosedürler:

- 1) ART1 Nöron Modeli (Adaptive Resonance Theory) (Liao ve Chen 1993)
- 2) Bulanık ART (Fuzzy ART) (Suresh ve Kaparathi 1994)
- 3) Yarışmalı Eğitim (Competitive Learning) (Chu 1993)

#### 1. ART1 nöron modeli ( Liao ve Chen 1993)

Bu modelde makine-parça ilişki matrisindeki her bir satır sırasıyla ikili girdi vektörü olarak ele alınır. Modele girilen her vektör, daha önce model tarafından algılanmış olan diğer vektörler ile arasındaki benzerliğe göre sınıflandırılmaktadır. Belirli bir ihtiyat değeri sınırlarında, okunan vektör diğer herhangi biri ile benzer olarak saptanırsa bu örneğin ait olduğu sınıfa dahil edilir, aksi halde yeni bir sınıf oluşturulur. Başlangıç makine-parça ilişki matrisi öncelikle satırlar halinde sisteme girilerek makine grupları elde edilir. Daha sonra bu matris sütunlar halinde sisteme girilerek parça aileleri bulunur.

Bu modelin uygulanması için gerekli olan algoritma aşağıdaki gibidir.

**Adım 1.** Aşağıdaki başlangıç değerleri ile nöral sistemi başlat.

- a) girdi indeksi  $k=1$
- b) ihtiyat değeri  $\gamma_0$ ,  $0 \leq \gamma_0 \leq 1$
- c) üst-alt ağırlıklar,  $t_{ij}(k) = 1$
- d) alt-üst ağırlıklar,  $b_{ij}(k) = 1/(1+n)$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq m$

**Adım 2.** Yeni bir girdi vektörü ( $a^k$ ) gir.

$$a^k = \{ a_1^k, \dots, a_i^k, \dots, a_n^k \}, \quad a_i^k \in (0,1)$$

**Adım 3.** Her makine için ayrı ayrı eşleme puanlarını ( $u_j$ ) hesapla.

$$u_j = \sum_{i=1}^n b_{ij}(k) a_i^k, \quad 1 \leq j \leq m, \quad \text{Flag}[j] = 0$$

**Adım 4.** En iyi makine grubunu ( $u_j^*$ ) seç.

$$\text{Flag}[j^*] = 0 \text{ olan } j^* \text{ ler için } u_j^* = \max(u_j) \text{ seç}$$

**Adım 5.** İhtiyat testini yap.

$$\|a^k\| = \sum_{i=1}^n a_i^k$$

$$\|T_j \cdot a^k\| = \sum_{i=1}^n t_{ij}(k) \cdot a_i^k$$

eğer  $\|T_j \cdot a^k\| / \|a^k\| > \gamma_0$ , ise adım 7' ye git,

eğer  $\|T_j \cdot a^k\| / \|a^k\| \leq \gamma_0$ , ise adım 6' ya git.

**Adım 6.**  $u_j = 0$  ve geçici olarak  $\text{Flag}[j^*] = 1$  yap ve adım 3' e git.

**Adım 7.**  $j^*$ 'a ait ağırlıkları güncelleştir.

$$t_{ij}(k+1) = t_{ij}(k) \cdot a_i^k$$

$$b_{ij}(k+1) = t_{ij}(k) \cdot a_i^k / 0.5 + \sum_{i=1}^n t_{ij}(k) \cdot a_i^k$$

**Adım 8.** Başka girdi vektörü varsa,  $\text{Flag}[j] = 0$  ve  $k = k+1$  yap, adım 2'ye git. Aksi halde bitir.

Algoritmadaki  $\gamma$  parametresi, bir girdi vektörünün daha önce depolanmış olan girdi vektörlerine karşı yakınlık derecesini ifade etmektedir.  $\gamma$  ' nun farklı değerlerine karşılık makine ve parça gruplarının sayıları da farklılık göstermektedir.

## 2. Bulanık ART nöron modeli (Suresh ve Kaparathi 1994)

Bulanık ART sinir ağı ilk olarak 1991 yılında Carpenter, vd. tarafından geliştirilmiş olup ART1 nöron modelinden birtakım farklılıklar göstermektedir.

Bunlar:

1. Girdi vektörleri ikili veya analog olabilir.
2. Sadece bir çeşit ağırlık vektörü( $w_{ij}$ ) vardır.
3. İhtiyat değerine( $\rho$ ) ek olarak(ART1'de  $\gamma$  idi) , seçim parametresi( $\alpha$ ) ve öğrenme oranı parametresi( $\beta$ ) vardır.
4. Kategorilere ayırmada etkin bir araç olan genel güncelleştirme kuralı uygulanır.

Algoritmanın işleyişi aşağıdaki gibidir.

**Adım 1.** Aşağıdaki başlangıç değerleri ile algoritmaya başla.

$$w_{ij}(0) = 1, \quad 0 \leq i \leq N-1, \quad 0 \leq j \leq (M-1)$$

$$\alpha > 0, \quad \beta \in [0,1], \quad \rho \in [0,1]$$

**Adım 2.** Yeni bir  $I$  girdi vektörü oku.

$I$  : parça-makine ilişki matrisinden alınan satır

**Adım 3.** Her bir çıktı nodu için seçim fonksiyonu değerlerini ( $T_j$ ) hesapla.

$$T_j = \| I \wedge w_j \| / \alpha + \| w_j \| , \quad 0 \leq j \leq (M-1)$$

Burada  $\wedge$ , bulanık VE operatörüdür,  $(x \wedge y) = \min(x,y)$

**Adım 4.** En iyi çıktı değerini seç.

$$T_0 = \max \{ T_j \}$$

**Adım 5.** Rezonans testini yap.

Eğer  $\| I \wedge w_0 \| / \| I \| \geq \rho$  ise adım 7'ye git. Aksi halde adım 6'ya git.

**Adım 6.**  $T_0 = -1$  yap ve adım 4'e git.

**Adım 7.** En iyi örneğin ağırlık değerlerini güncelleştir.

$$w_0^{\text{yeni}} = \beta(I \wedge w_0^{\text{eski}}) + (1 - \beta)w_0^{\text{eski}}$$

**Adım 8.** Başka girdi vektörü varsa adım 2'ye git, aksi halde bitir.

Algoritmada girdi vektörü olarak ( $I$ ) parça-makine ilişki matrisindeki satırlar kullanılmaktadır. Yani her bir parçaya ait makine sırası algoritmaya girilerek, parçaların ait oldukları makine grupları belirlenmektedir.

### 3. Yarışmalı eğitim algoritması (Chu 1993)

Yarışmalı eğitim danışmasız eğitim algoritmalarının en basit olanlarından biridir. Chu (1993)'nun hücre tasarımında kullanmış olduğu eğitim algoritmasının yapısı iki katmandan meydana gelmektedir. Birincisi girdi katmanı ve ikincisi yarışma katmanıdır. Bu iki katmanda bulunan birimler birbirleri ile tam ilişkilidirler. Parça-makine ilişki matrisinin satırları girdi katmanına verilir ve bu vektörler yarışma katmanında sınıflandırılır.

Algoritmada kullanılan notasyonlar aşağıda verilmiştir.

İndisler:

$i$ : parça indeksi  $i = 1, \dots, N$

$j$ : makine indeksi  $j = 1, \dots, M$

$k$ : hücre indeksi  $k = 1, \dots, C$

Parametreler:

$M$  : makine sayısı

$N$  : parça sayısı



$C$  : hücre sayısı

$l$  : öğrenme oranı ( $0 < l < 1$ )

nepoch : ağın eğitilmesi için gerekli olan döngü sayısı

$P_i$  : girdi katmanında  $l$  aktivasyon değerine sahip ünitelerin sayısı

$X_{ji}$  : eğer  $i$  parçası  $j$  makinesinde işlem görüyorsa 1, aksi halde 0.

$W_{kj}$  :  $k$  birimi ile  $j$  birimi arasındaki bağlantı ağırlığı

$S_k$  :  $k$  biriminin net girdisi

$a_k$  : eğer  $k$  birimi kazanan ise 1, aksi halde 0.

Hücre tasarımı amacıyla geliştirilmiş olan prosedür genel olarak aşağıdaki beş fazdan meydana gelmektedir.

1. Sistem parametrelerinin seçimi
2. Ağın eğitilmesi
3. Sınıflandırmanın test edilmesi
4. Kümeleme sonuçlarının gösterilmesi ve performansın ölçülmesi
5. Gerekli ise başa dön ve sistem parametrelerini değiştir.

Yarışmalı eğitimin genel algoritması şekil 11' de verilmiştir.

Yukarıdaki algoritmaya göre, parça-makine ilişki matrisinin satırları girdi katmanına verilerek, yarışma katmanındaki hücreler birbirleri ile çekişmektedirler. Bu çekişme sonucunda en yüksek aktivasyon değerine sahip olan hücre yarışmayı kazanmakta ve girilen satır hangi parçaya ait ise o parça kazanan hücreye atanmaktadır.

Bu algorithmada, hücrelerdeki makine sayısına ilişkin bir üst sınır konulmuş ise, yarışmayı kazanan hücreyi takip eden ikinci hücreye atama yapılabilir.

Yukarıda kısaca açıklanmaya çalışılan yapay sinir ağları yöntemlerinin algoritmaları Pascal programlama dilinde kodlanmıştır. Programlar ve program sonuçları ileriki bölümlerde sunulmuştur.

**Sistem parametrelerini seç (l ve nepoch)**

**Başlangıç ağırlıklarını belirle ( $W_{kj}$ )**

$$\sum_j W_{kj} = 1$$

**Parça-makine matrisindeki bütün satırlar tamamlanuncaya kadar aşağıdakileri yap**

**Rasgele bir satır seç**

**Bu satırı matristen sil**

**Aşağıdaki döngüyü her k için yap**

**$a_k = 0$  ve  $S_k = 0$  ile başla**

**Rasgele seçilen satıra göre net çıktıyı ( $S_k$ ) hesapla**

$$S_k = \sum_j W_{kj} X_{ji}$$

**Kazanan c' yi bul , burada:**

$$S_c = \text{Max}_{(k)} S_k$$

**Aşağıdaki aktivasyon değerlerini ata:**

**Eğer  $k = c$  ise  $a_k = 1$**

**Aksi halde  $a_k = 0$**

**$P_i$  değerini hesapla**

$$P_i = \sum_j X_{ji}$$

**Ağırlıkları güncelleştir:**

**Eğer  $k = c$  ise  $W'_{kj} = l.X_{ji}/P_i + (1-l).W_{kj}$**

**Aksi halde  $W'_{kj} = W_{kj}$**

**Return**

**Return**

**Return**

Şekil 11. Yarışmalı eğitimin genel algoritması (Chu 1993)

Şimdi yapay zeka yöntemlerinden bulanık mantık ve bulanık küme tabanlı hücre formasyonu tekniği anlatılacaktır.

#### 5.1.1.4. Bulanık mantık

##### 5.1.1.4.1. Bulanık küme teorisi

Bulanık küme teorisi (Fuzzy Set Theory) ilk olarak 1965 yılında Zadeh tarafından ortaya atılmıştır. Bulanık mantıkta bir değişkene ait kesin değerler dilsel (linguistic) ifadelere dönüştürülerek bulanık küme ile temsil edilmektedir. Klasik küme teorisinde herhangi bir nesne veya olay bir kümeye ya aittir, ya da ait değildir. Halbuki bulanık küme teorisinde bu olay ya da nesnelere o kümeye belirli bir üyelik derecesinde kısmen de olsa ait olabilmektedir. Karşılaşılan problemlerde ortaya çıkabilecek birtakım belirsizliklerin ana sebebi bu problemlere ilişkin verilerdeki eksikliklerdir. İşte bulanık küme teorisi karmaşık problemlerin modellenmesinde ve çözümünde etkili bir araç olarak kullanılmaktadır.

Klasik mantık ile karşılaştırıldığında bulanık küme teorisinin bazı avantaj ve dezavantajlarından söz etmek mümkündür. Bunlar:

##### **Avantajları**

1. İnsanın düşünme tarzına yakın olması
2. Uygulamasının matematiksel modele ihtiyaç duymaması
3. Yazılımın basit olması nedeniyle maliyetinin az olması

##### **Dezavantajları**

1. Uygulamada kullanılan kuralların oluşturulmasının uzmana bağlı olması
2. Üyelik fonksiyonlarının deneme-yanılma yoluyla bulunması sebebiyle uzun zaman alabilmesi
3. Kararlılık analizinin yapılışındaki zorluklar (Allahverdi 1999).

1975 yılında Mamdani ve Assilian öncülüğünde başlayan bulanık mantık uygulamaları günümüzde özellikle kontrol sistemleri gibi birçok alanda karşımıza çıkmaktadır. Karmaşık bir problem olarak karşımıza çıkan hücre formasyonu problemi de bunlardan birisidir.

Leem ve Chen (1996), parçaların alternatif makinelerde değişik etkinliklerde ve değişik işlem zamanlarında işlenebileceğini dikkate alarak, farklı üyelik derecelerine sahip bu alternatifler arasında bulanık karar verme yoluna gitmişlerdir.

Bu yöntemde karşılaşılabilecek en önemli problem üyelik derecelerinin belirlenmesidir.

#### 5.1.1.4.2. Bulanık küme tabanlı hücre formasyonu (Leem ve Chen 1996)

Bu prosedür genel olarak üç ana kuraldan meydana gelmektedir. Bunlar:

1. Makineler arasındaki benzerlik değerlerinin hesaplanması
2. En yüksek benzerlik değerlerine sahip makinelerin bir araya getirilerek hücrelerin oluşturulması
3. En yüksek benzerliğe sahip olan iki makinenin tek bir makine olarak düşünülmesi

Yukarıdaki üç kuralın kombine edilmesi sonucu makine hücrelerinin ve parça ailelerinin oluşturulması mümkün olmaktadır.

Geliştirilmiş olan prosedürde kullanılan kavramlar şunlardır:

**-Başlangıç makine-parça ilişki matrisi :** Bu matris sadece 0 ve 1'lerden oluşmamaktadır. Parçaların, aynı türden bir veya birden fazla sayıda makinelerde işlenme durumlarına ilişkin üyelik dereceleri bu matriste yer almaktadır. Bu üyelik dereceleri 0 ve 1 arasında değişmekte olup, misal olarak P2 parçasının bir nolu torna tezgahında işlenmesi durumunda üyelik derecesi 0.4 iken, aynı parça için iki nolu torna tezgahının üyelik derecesi 1 olabilmektedir. Şekil 12, örnek bir makine-parça ilişki matrisini göstermektedir.

<b>Makine</b>	<b>Parçalar</b>				
	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>
<b>T1</b>	1	0.4	0.5	0.1	0.4
<b>T2</b>	0	1	0.5	0	0
<b>M1</b>	0.3	1	1	0.9	1
<b>M2</b>	0.8	0.5	0.2	0.8	0.6
<b>F</b>	1	0	0	0	1

Şekil 12. Başlangıç makine-parça ilişki matrisi

**-Benzerlik katsayısı ve benzerlik katsayısı matrisi:** Başlangıç makine-parça ilişki matrisinde bulunan, parçaların alternatif makinelerde işlenme durumunda almış farklı üyelik derecelerine göre makineler arasındaki benzerlik katsayıları

hesaplanmaktadır. Hücre formasyonu problemlerinde sık sık karşımıza çıkan bu yöntemin literatürde birçok örneğine ve çeşitli benzerlik katsayısı hesaplama tekniğine rastlamak mümkündür. Makineler arasındaki benzerlik katsayıları, üyelik dereceleri kullanılarak aşağıdaki eşitlik yardımıyla hesaplanmakta ve tüm katsayılar bir matriste gösterilmektedir.

$S_{ij}$  : i ve j makineleri arasındaki benzerlik katsayısı

$\mu_{ik}$  : k parçası için i makinesinin üyelik derecesi

$\mu_{jk}$  : k parçası için j makinesinin üyelik derecesi

$$S_{ij} = \left[ \sum_{k=1}^p (\mu_{ik} \wedge \mu_{jk}) \right] / \left[ \sum_{k=1}^p (\mu_{ik} \vee \mu_{jk}) \right]$$

Her makinenin birbiriyle olan benzerlik katsayıları yukarıdaki formül yardımıyla hesaplanarak benzerlik katsayısı matrisinde gösterilmektedir.

Geliştirilen prosedürün işleyişi ise adım adım aşağıdaki gibidir.

**Adım 0: Başlama** :Mevcut hücre sayısını(CNC) m ve arzu edilen hücre sayısını c olarak al ve başlangıç makine-parça ilişki matrisine göre benzerlik katsayılarını hesapla.

**Adım 1: Hücre Formasyonu** : En yüksek benzerlik katsayısı değerine sahip  $M_i^*$  ve  $M_j^*$  makinelerini bul ve bu iki makineyi aynı hücreye al.

**Adım 2: Dönüşüm**:  $M_i^*$  ve  $M_j^*$  makinelerinin elemanlarını dönüştürerek, makine-parça ilişki matrisini yeniden düzenle.

**Adım 3: Benzerlik Değerlerinin Hesaplanması**: Yeni ilişki matrisine göre benzerlik katsayılarını yeniden hesaplayarak mevcut hücre sayısını bir azalt(CNC= CNC-1).

**Adım 4: Değerleme** : Mevcut hücre sayısını(CNC) kontrol et. Eğer CNC > c ise adım 1'e git, aksi halde CNC = c oluncaya kadar devam et.

En yüksek benzerliğe sahip iki makine şu şekilde bir araya getirilir:

$$T1 = (0.0, 1.0, 0.3, 0.0, 0.7, 0.8, 0.0)$$

$$G2 = (0.0, 1.0, 0.5, 0.0, 0.8, 0.9, 0.0)$$

$$\begin{aligned} M(T1,G2) &= \sum_{k=1}^7 (\mu_{T1} \vee \mu_{G2}) \\ &= (0.0, 1.0, 0.5, 0.0, 0.8, 0.9, 0.0) \end{aligned}$$

Yukarıdaki adımların , gerekli şart sağlanıncaya kadar devam ettirilmesi sonucu sonuç makine-parça ilişki matrisi ortaya çıkmakta , parça aileleri ve makine hücreleri şekillenmektedir.

## 5.2. Sezgisel Yöntemler

Matematiksel yöntemlerin optimum sonucu bulmada başarısız oldukları büyük çaplı problemlerde, optimuma yakın sonuçları garanti edebilen birtakım sezgisel yöntemler kullanılmaktadır. Şimdi bu çalışmada incelenen sezgisel yöntemler ve H.Ü.S. tasarımı uygulamaları sırasıyla anlatılacaktır.

### 5.2.1. Tavlama benzetimi (Simulated annealing )

#### 5.2.1.1. Tavlama benzetimi teorisi

Optimal çözümü elde etmenin mümkün olmadığı büyük boyutlu optimizasyon problemleri için çözümü bulmak ancak birtakım sezgisel yöntemler kullanılarak mümkün olmaktadır. Bu yöntemlerden biri de Tavlama Benzetimi algoritmasıdır.

Bu algoritma, global minimumu ararken lokallikten kurtulmak amacıyla kullanılan stokastik bir metottur. Bu ise fonksiyon değerindeki bir azalmaya geçiş ek olarak, fonksiyon değerindeki bir artışa geçişi de kabul ederek yapılmaktadır. Tavlama benzetimi algoritması, fiziksel tavlama prosesi ile ayrık minimizasyon problemleri için yaklaşık minimum çözümü bulma problemi arasındaki benzeşmeden ortaya çıkmıştır. Fiziksel tavlama prosesi, bir ısı banyosu içerisindeki katı bir cismin düşük enerji durumlarını elde etmek için termal bir proses olarak bilinir.

Metropolis vd (1953) yılında bir ısı banyosu içindeki taneler kümesinin denge dağılımını hesaplamak amacıyla, bilgisayar simülasyonu metodu önermiştir. Bu metodda; ilk durum enerjisi  $E_1$ 'dir. Tanelerden birinin küçük bir yer değiştirmesi sonucunda oluşan yeni durumun enerjisi  $E_2$ 'dir. Eğer  $E_2 - E_1 \leq 0$  ise yeni durum reddedilmez. Yeni durum;  $\exp(-(E_2 - E_1) / (kT))$  olasılığıyla kabul edilir.

k: Boltzman sabiti

T: Isı banyosunun sıcaklığı

Tavlama benzetimini kombinasyonel optimizasyon problemlerine ilk uygulayan kişi Kirkpatrick vd (1983) dir. Kirkpatrick enerjiyi maliyet fonksiyonuyla, fiziksel sistemin durumlarını da minimizasyon probleminin çözümleriyle değiştirmiştir. Fiziksel sistemdeki tanelerin rahatsızlığı, kombinasyonel minimizasyon problemlerindeki bir denemeye karşılık gelmektedir. Minimizasyon işlemi öncelikle çözüm uzayını etkin olarak yüksek bir sıcaklıkta eritmekle ve daha sonra sistem sabit bir çözümde donuncaya kadar yavaş yavaş soğutmakla gerçekleşir (Dekkers ve Aarts 1991).

Algoritmanın ana adımlarını aşağıdaki gibi özetlemek mümkündür.

- Bir başlangıç çözümü ile başla ve komşu çözümler üret
- Yeni çözüm eskisinden daha iyi ise kabul et, aksi halde belirli bir kabul edilme olasılığına göre yine kabul et.
- Kabul edilme olasılığını düşürmek için sıcaklığı da düşür. Her bir sıcaklık değerinde çok sayıda hareket yapılabilir.
- Durdurma kriteri sağlandığında algoritmayı durdur.

Şekil 13 tavlama benzetiminin algoritmasını göstermektedir.

#### **PROCEDURE SIMULATED ANNEALING**

**Begin**

*"initialize (c,x)";*

*stopcriterion := false;*

*while stopcriterion = false*

**begin**

*for i := 1 to L do*

**begin**

*"generate y from x";*

*if  $f(y) - f(x) \leq 0$  then accept;*

*else if  $\exp(-(f(y)-f(x)) / c) > \text{random} [0,1)$  then accept;*

*if accept then  $x := y$*

**end;**

*"lower c"*

**end**

**end.**

Şekil 13. Tavlama benzetimi algoritması (Dekkers ve Aarts 1991)

Yukarıda verilen algoritmanın etkili sonuçlar vermesi, algoritmada kullanılan parametrelerin seçimine bağlıdır. Algoritmada kullanılan parametreler ise tavlama benzetiminde tavlama çizelgesini meydana getirmektedir. Tavlama çizelgesi şunlardan meydana gelmektedir:

- başlangıç sıcaklığı
- soğutma fonksiyonu
- her bir sıcaklık değerinde gerçekleşecek iterasyon sayısı
- durdurma kriteri

Başlangıçtaki sıcaklık değerimiz hemen hemen tüm komşu çözümlerin kabul edilmesine izin verecek kadar büyük iken, kontrol parametresindeki azalma sayesinde komşu çözümlerin kabul edilme olasılığı da düşmektedir.

### **5.2.1.2. Tavlama benzetimi yardımıyla makine- parça hücre formasyonu probleminin çözümü (Boctor 1991)**

Boctor (1991) tarafından geliştirilen algoritmaya geçmeden önce, algoritma için gerekli olan başlangıç çözümünün ve komşu çözümlerin nasıl elde edildiğini açıklamamız gerekmektedir. Algoritmada başlangıç çözümü aşağıdaki adımlar izlenerek elde edilir:

1. M adet makine N adet hücreye atanır.  $h$ ,  $(M / N)$ 'den büyük en küçük tamsayı olmak üzere, ilk  $h$  makine ilk hücreye, sonraki  $h$  makine ikinci hücreye olmak üzere tüm makineler atanır.
2. Her bir makinenin tekrar atanması durumunun avantajlı olup olmadığına bakılır. Avantajlı olması halinde bu makine, harici eleman sayısında en büyük azalmayı sağlayacak şekilde başka bir hücreye atanır. Daha sonra parçaların yerlerinin optimal olup olmadığına bakılarak gerekli değişiklikler yapılır. Bu adımdan sonra dikkat edilmesi gereken nokta, bazı hücrelerin maksimum makine sayısını aşmış olabileceğidir.
3. Maksimum makine sayısını aşmış hücredeki, doymamış bir hücreye atandığında harici eleman sayısında minimum artışa sebep olacak makine seçilerek doymamış olan hücreye atanır. Daha sonra parça atamalarındaki gerekli düzenlemeler yapılır.



Komşu çözümü üretmek için ise, başlangıç çözümü üzerindeki makine atamasından rasgele bir makine seçilir ve rasgele seçilen bir hücreye atanır. Eğer seçilen hücre maksimum makine sayısını(m) aşmış ise tekrar rasgele bir hücre seçilir ve makine bu hücreye atanır. Tüm seçimler rassal olarak yapılmaktadır.

Şekil 14, hücre formasyonu için önerilen tavlama benzetimi prosedürünün işleyişini göstermektedir.

**Procedure**

*call initial (başlangıç çözümünü bul)*

*Bu çözümü mevcut çözüm olarak sakla*

*Bu çözümü geçici olarak en iyi çözüm olarak sakla*

*c:=0 (durdurma sayacını başlat)*

**Repeat until**  $c = C_{max}$

*R:=R<sub>max</sub>*

*r:=0 (tekrar sayacını başlat)*

*T:=  $\alpha * T$*

**Repeat until**  $r = R$

*call neighbour (komşu çözüm üret)*

*$\delta := Z(\text{komşu}) - Z(\text{mevcut})$*

**if**  $\delta \leq 0$  veya  $\text{random}(0,1) \leq \exp(-\delta / T)$  **then**

*komşu çözümü mevcut çözüm olarak sakla*

**if**  $Z(\text{mevcut}) < Z(\text{en iyi})$  **then**

*mevcut çözümü en iyi çözüm olarak sakla*

*c:= 0 (durdurma sayacını yeniden başlat)*

*r:= 0 (tekrar sayacını yeniden başlat)*

*R:=  $\beta * R$  (tekrar sayısını düşür)*

**endif**

**endif**

**endrepeat**

**endrepeat**

**end**

Şekil 14. Makine-parça hücre formasyonu probleminin çözümü için tavlama benzetimi prosedürü (Boctor 1991).

Yukarıdaki algoritmanın etkili sonuç verebilmesi için, algoritmada geçen  $C_{max}$ ,  $R_{max}$ ,  $T$ ,  $\alpha$  ve  $\beta$  parametrelerinin tasarımcı tarafından doğru olarak tespit edilip prosedüre verilmesi gerekmektedir. Bu parametrelerin değerlerinin tasarımcı tarafından deneme yanılma yoluyla elde edilmesi mümkündür. Boctor(1991)'un çalışmasında bu değerler sırasıyla 320, 64, 32, 0.5 ve 0.8 olarak alınmıştır.

Yukarıdaki prosedürde amaç fonksiyonumuz  $Z$ , harici eleman sayısını belirtmekte olup her parçanın ve her makinenin sadece ve sadece bir üretim hücresine atandığı hücre bölme durumunda aşağıdaki eşitlik yardımıyla hesaplanmaktadır.

$$Z = \sum_{i=1}^M \sum_{j=1}^P a_{ij} \left( \sum_{k=1}^N |x_{ik} - y_{jk}| \right) / 2$$

Burada:

k: hücre indeksi

N: üretim hücrelerinin sayısı

M: makinelerin sayısı

P: toplam parça sayısı

$x_{ik}$ : eğer i makinesi k hücresine atandıysa 1, aksi halde 0

$y_{jk}$ : eğer j parçası k hücresine atandıysa 1, aksi halde 0

## 5.2.2. Tabu araştırması (Tabu search)

### 5.2.2.1. Tabu araştırma algoritması

Tabu araştırma algoritması, ilk olarak Glover (1986) tarafından çeşitli optimizasyon problemlerinin çözümü amacıyla geliştirilmiş yüksek seviyeli metaheuristic bir algoritmadır. Bu algoritma, bir başlangıç çözümüyle başlayarak, bir adım önce geliştirilen çözüme tekrar bakmamak şartıyla en iyi çözümü araştırır (İçmeli 1994). Genel olarak tabu araştırması, aşağıda verilen tipteki problemler için optimal veya optimale yakın sonuçları garanti etmektedir.

minimize  $c(x)$

subject to  $x \in X$

Burada  $c(x)$ ,  $x$  değişkenine bağlı bir fonksiyon ve  $X$ , mümkün çözümlerin kümesidir. Tabu araştırması bir  $x \in X$  mümkün çözümüyle başlanır. Daha sonra bu mümkün çözüm  $x \in X$  'i yeni bir çözüm olan  $x'$  ( $x' = m(x)$ )'ye dönüştüren  $m \in M(x)$  fonksiyonuna uygulanır. Bu işlem bir "hareket" veya  $x'$  in komşuluğu olarak adlandırılır. Bu algoritma, lokal opimallikten kaçınmak amacıyla daha önce elde edilen çözümlere geri dönmeyi yasaklayan bir hafıza yapısına sahiptir. Bu yapı **tabu listesi** olarak adlandırılır. Tabu listesinin büyüklüğü üzerinde çalışılan problemin tipine göre farklılık göstermektedir.

Bir tabu araştırma uygulaması genel olarak şunlar tarafından karakterize edilmektedir (Aljaber 1997):

1. Başlangıç çözümü
2. Mümkün çözüm  $x'$  den yapılabilecek hareket tipi
3. Komşuluk büyüklüğü
4. Tabu listesinin büyüklüğü
5. Durdurma kriteri

Tabu araştırma tekniğinin Taillard (1990) tarafından geliştirilen genel prosedürü ise aşağıdaki gibidir.

1. Herhangi bir  $x_0$  başlangıç çözümü ve boş tabu listesi  $T$  ile başla.  $x^* = x_0$ ,  $c^* = c(x_0)$  ve  $k = 0$  olarak al. Burada:

$x^*$  : geçici en iyi çözüm

$c^*$  : bu geçici çözüme karşılık gelen amaç fonksiyonu değeri

2.  $M(x_k)$  içerisinde tam veya kısmi tarama yaparak, amaç fonksiyonu değerini düşüren ve tabu listesi tarafından yasaklanmayan bir hareket seç.  $x_{k+1} = m(x_k)$  yap.
3. Eğer  $c(x_{k+1}) < c^*$  ise  $c^* = c(x_{k+1})$  ve  $x^* = x_{k+1}$  yap.
4. Tabu listesini güncelleştir.
5. Eğer durdurma kriteri sağlanıyorsa optimuma ulaşılmış demektir ve dur. Aksi halde 2'ye git.

En iyi kabul edilebilir hareket tabu sınırlamaları ve amaç değerleri açısından mevcut çözümün komşuluğunda en yüksek değerlendirmeye sahip harekettir. Değerlendirme fonksiyonu, amaç fonksiyonunda en az kötüleşmeyi üreten hareketi veya diğer bir deyişle en fazla gelişmeyi sağlayan hareketi seçer. Tabu listesi, kabul edilen hareketlerin karakteristiklerini depo etmek için kullanılmaktadır. Bu

karakteristikler daha sonraki iterasyonlarda belirli hareketleri tabu olarak sınıflandırmak amacıyla kullanılır. Kötüleşme gösteren hareketleri kabul etmeme, hali hazırda karşılaşılmış olan çözümlere tekrar dönülmesini ve kör döngü olayının meydana gelmesine sebep olur. Tabu listesini kullanmanın amacı bu istenilmeyen durumu önlemektir. Böylece, görevi sadece tabu listesini kontrol etmek ve değiştirmek olan bir “yasaklama” (forbidding) stratejisi ile araştırmayı sınırlamak zaruridir. Yasaklama stratejisini kullanarak daha önceden izlenilmiş olan tekrar izlenilmesinden sakınılır ve yeni bölgelerin araştırılması sağlanmış olur. *Basit bir tabu araştırma aşağıdaki stratejilere ve elemanlara sahiptir* (Karaboğa ve Kalınlı 1996).

#### **i) Yasaklama stratejisi:**

Tabu listesine hangi hareketin girip girmeyeceğine karar veren stratejidir. Gayesi, döngüyü önlemek ve belirli hareketleri yasaklamak için bir mekanizmanın kurulmasıdır. Döngü olayının meydana gelme ihtimali mevcut çözümün bir önceki çözümden uzaklığı ile ilgilidir. Son Ts iterasyon boyunca alınan herhangi bir kararın tersini temsil eden hareketlerin seçilmesini önlemek suretiyle tabu araştırmasını geçmişteki Ts iterasyon boyunca karşılaşılan çözümlerden uzaklaşmaya doğru meyleder. Burada Ts tabu listesinin büyüklüğü olarak adlandırılır.

#### **ii) Aspirasyon kriteri:**

Bu kriter, temel olarak bir hareketin yeteri kadar iyi olması durumunda ve döngüyü önlemek için yeterli şartların sağlanması halinde tabu durumundan çıkmasını sağlar. Bu sayede, tabu sınırlamaları aşılmadığı müddetçe herhangi bir hareketin kabul edilmesine imkan sağlar.

#### **iii) Kısa dönem stratejisi:**

Bu strateji, geçmişteki araştırma davranışı hakkında geniş bir bilgi depo eder.

1. Klasik seçme stratejisi: Bu stratejide, tabu sınırlamaları ve aspirasyon kriteri dikkate alınarak mevcut çözümden, amaç fonksiyonunda en az kötüleşme veya en büyük gelişmeyi üreten hareket seçilir.

2. İlk ve en iyiyi seçme stratejisi: Bu stratejide, amaç değeri açısından mevcut çözümden daha iyi bir gelişme kaydeden ilk çözüm seçilir. Gerekli değişiklikler yapılır ve çözümden diğer bir hareketin üretilmesi yapılarak işleme

devam edilir. Aday listesindeki (yani komşuluktaki) tüm hareketler bir gelişme kaydetmezse bu strateji en iyi gelişmemiş olan hareketi seçer.

#### iv) Öğrenme stratejisi:

Bu strateji sayesinde tabu araştırmanın icrası anında bilgi toplanır ve bu bilgi algoritmanın diğer çalışmaları esnasında araştırmayı yönlendirmede kullanılır.

Literatüre bakıldığında tabu araştırmasının birçok uygulamasıyla karşılaşmak mümkündür. Çizelgeleme, planlama, desen tanıma, iş sıralama gibi birçok alanda başarılı tabu araştırma uygulamalarına rastlamak mümkündür.

#### 5.2.2.2. Hücre formasyonu problemine tabu araştırması yaklaşımı (Aljaber ve arkadaşları 1997)

Tabu araştırma algoritmasını kümeleme problemlerine uygulamak için, hücre formasyonu problemi "En Kısa Yol" problemi olarak düşünülmektedir. Tabu araştırmasına, en kısa yolu bulma amacına sahip bir permütasyon problemi olarak yaklaşılarak En Kısa Yol problemi olarak kabul edilebilir. En kısa yol problemi için verilen bir grafta toplam mesafe aşağıdaki formülle hesaplanmaktadır.

$$\text{DIST}(\Omega) = \sum_{i=1}^n \sum_{j=1}^n D_{ij} X_{ij} \quad i \neq j$$

Burada:

n: bir graftaki nokta sayısı

$X_{ij} = 1$ , eğer i ve j arasında bir bağlantı varsa, aksi halde  $X_{ij} = 0$

$D_{ij}$ : i ve j noktaları arasındaki mesafe

Parçalar veya makineler arasındaki mesafe, literatürdeki mesafe ölçülerinden biri olan Jaccard Benzerlik Ölçüsü kullanılmaktadır. Bunun için iki parça veya makine arasında hesaplanan Jaccard benzerlik katsayısı üst limitinden çıkartılır.

Yani,

$$D_{ij} = 1 - S_{ij} = 1 - [C_{ij} / (T_i + T_j - C_{ij})]$$

Burada:

$S_{ij}$ : i ve j parça ve makineleri arasında hesaplanan Jaccard benzerlik ölçüsü

$C_{ij}$ : i ve j parçalarına ait ortak makine sayısı (i ve j makinelerine ait ortak parça sayısı)

$T_1$  : 1 makinesinde işlenen toplam parça sayısı (veya 1 parçasını işleyen toplam makine sayısı)

Daha önce de belirtilmiş olduğu gibi tabu araştırmasının performansını etkileyen birtakım parametreler mevcuttur. Bu parametrelere ait en iyi değerler problemlerin tipine göre farklılıklar göstermektedir. Hücre formasyonu problemi için önerilen tabu araştırma algoritmasında kullanılan parametreler şöyledir:

**1. Başlangıç çözümü:** Tabu araştırması uygulamalarında başlangıç çözümü rassal veya sezgisel olarak alınmaktadır. Hücre formasyonu problemi için önerilen algoritma başlangıç çözümü olarak başlangıç makine-parça ilişki matrisinde verilen sırayı(permütasyonu) kullanmaktadır.

**2. Komşu çözümler (hareketler) :** Permütasyon problemleri için komşuluklar birtakım yollardan belirlenmektedir. Bunların içerisinde en yaygın olarak kullanılanları şunlardır:

-API(Adjent Pairwise Interchange)

-Insert

-Swap

Önerilen algoritmada kullanılan hareket (move) tipi Insert tipidir.  $\sigma'$  verilen bir  $\sigma$  permütasyonunun komşusu olmak üzere  $\sigma'$  şu şekilde elde edilir:

$$\sigma = \langle x_1, \dots, x_i, x_{i+1}, \dots, x_n \rangle$$

$$\sigma' = \langle x_1, \dots, x_{i-1}, x_j, x_i, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots, x_n \rangle$$

**3. Tabu listesinin büyüklüğü :** Tabu araştırması üzerinde yapılan çeşitli araştırmalar tabu listesinin büyüklüğünü  $7(\pm 2)$  olarak saptamıştır.

**4. Komşu büyüklüğü :** Komşu büyüklüğü ifadesi, bir hareket gerçekleştirilmeden önce komşu çözümlerin nasıl inceleneceğini ifade etmektedir. Komşu çözüm seçilmeden önce ya tüm komşular incelenir ya da komşular kısmen incelenir. Tüm komşu çözümlerin incelenmesi durumunda en başarılı hareket seçilmektedir fakat uzun zaman aldığı için bu yöntem genellikle tercih edilmez. Önerilen hücre formasyonu algoritmasında benimsenen ikinci alternatifte ise, komşu çözümler incelenir ve çözümü iyileştiren ve tabu olmayan ilk hareket kabul edilir.

**5. Tabu hareketlerini tanımlama ve saklama :** Önerilen algoritmada tabu hareketleri kabul edilen son  $S$  hareket olarak belirtilmektedir. Yani kabul edilen her

hareket tabu listesine girerken listenin sonunda bulunan tabu hareketleri listeden çıkartılır (FIFO).

**6. Durdurma kriteri :** Durdurma kriteri, algoritma işleme zamanı ve elde edilen sonucu açısından önemli bir rol oynamaktadır. En sık kullanılan iki durdurma kuralı şöyledir (Taillard 1990):

1. Eğer mevcut çözüm üzerinde herhangi bir iyileşme olmadan  $k_1$  adet iterasyon gerçekleşti ise dur.
2. Eğer toplam iterasyon sayısı  $k_2$  sabitine ulaştı ise dur.

İyi bir çözüme kaç iterasyon sonunda ulaşılacağına önceden karar vermek zor olacağından, önerilen algoritmada birinci kural kullanılmıştır.

Aljaber ve arkadaşları (1997) yukarıda belirtmiş olduğumuz altı tabu araştırma parametresinin uygun kombinasyonlarını bulmaya yönelik olarak yaptıkları testlere göre, kısmi komşu büyüklüğünü, Insert tipi hareketi benimsemişler ve tabu listesi büyüklüğünü de 7 olarak belirlemişlerdir.

Önerilen hücre formasyonu prosedürü üç fazdan meydana gelmektedir. Bunlar:

1. Makine ve parçaların sıralaması içi tabu araştırması
2. Çözüm matrisinin oluşturulması
3. Makine-parça kümelerinin sınırlarının belirlenmesi

#### **1.FAZ: Makine ve parçaların sıralaması içi tabu araştırması**

Makine ve parça sıraları birbirinden bağımsız olarak yapılmaktadır ve bunun için kullanılan tabu araştırma algoritmasında kullanılan notasyonlar şunlardır:

$n$ : parça-makine ilişki matrisindeki parça sayısı

$m$ : parça-makine ilişki matrisindeki makine sayısı

NIWI : mevcut çözümü iyileştirmeden yapılan iterasyon sayısı

$N_i(x)$  : Insert tipi hareketle belirlenen komşu çözüm

$S$  : tabu listesinin büyüklüğü

$T$  : tabu hareketlerinin kümesi

$x$ : makine veya parçalar için sıra

$DIST(x)$  :  $x$  sırasına karşılık gelen mesafe

$k_1$  : durdurma kuralına karşılık gelen iterasyon sayısı

$\langle x,y \rangle$  :  $x$  çözümünü  $y$  çözümüne dönüştüren bir hareket

1. fazın işleyişi ise aşağıdaki gibidir.

(i) *Makineler için en kısa yol sıralarının oluşturulması*

**Adım 0:** parça-makine ilişki matrisini gir

$$k_1 = m$$

$$NIWI = 0$$

$$S = 7$$

$$T = \{ \phi \}$$

$$x = x_0 \text{ (} x_0 : \text{başlangıç parça-makine ilişki matrisinde verilen makine}$$

sırası) olarak al.

**Adım 1 :** makineler için mesafe matrisini oluştur.

**Adım 2 :** 'DIST(x)'i hesapla ve incumbent = DIST(x) yap

**Adım 3 :** NIWI = NIWI +1

**Adım 4 :**  $\langle x, x' \rangle \in N_1(x)$  ve  $\langle x, x' \rangle \notin \{T\}$  olmak üzere bir hareket gerçekleştir

**Adım 5 :** DIST(x')yü hesapla

**Adım 6 :** **If** (DIST(x') < incumbent) **then**

- tabu listesi T'yi ( $\langle x, x' \rangle$ )yü içine alacak şekilde güncelleştir

- incumbent = DIST(x') yap

-  $x = x'$  yap

- NIWI = 0 yap

- adım 3'e git

**else**

- **if** (NIWI <  $k_1$ ) **then**

- adım 3'e git

**else**

- stop

(ii) *Parçalar için en kısa yol sıralarının oluşturulması*

Yukarıda makine sıralarının oluşturulması için verilen algoritmadaki tüm adımlar parçalar için yapılarak parça sıraları oluşturulur.

**2. FAZ : Çözüm matrisinin oluşturulması**

Bu fazda, 1. fazda elde edilen makine ve parça sıralarına göre, başlangıç parça-makine ilişki matrisi yeniden düzenlenir.



### 3. FAZ : Makine-parça kümelerinin sınırlarının belirlenmesi

Birinci ve ikinci faz yardımıyla elde edilen çözüm matrisi bu fazda yatay ve dikey çizgiler yardımıyla alt matrislere veya parça-makine gruplarına ayrılır. Aljebra ve arkadaşları bu fazda yeni bir sezgisel prosedür geliştirmişlerdir. Aşağıdaki notasyonlar tanımlandıktan sonra önerilen prosedürün algoritmik şekil verilecektir.

UL = parça-makine gruplarının sayısı için kullanıcı tarafından belirlenen üst limit

ng = parça-makine gruplarının mevcut sayısı

PF(J) = parça ailesi J

MG(I) = makine grubu I

H(I) = MG(I) ve MG(I+1) makine gruplarını ayıran yatay çizgi

V(J) = PF(J) ve PF(J+1) parça ailelerini ayıran dikey çizgi

S(I,J) = I satır kümesi ve J sütun kümesi tarafından oluşturulan alt matris

NS(I,J) = S(I,J) alt matrisinde bulunan 1'lerin sayısı

X(I,J) = 1 eğer MG(I) ve PF(J) birbirine atandıysa, aksi halde X(I,J) = 0

B = { S(I,J)/ X(I,J) = 1 } B'nin elemanları "blok" olarak ifade edilir

$\Delta(I,J) = S(I,J) \in B$  bloğunun bozulması sonucu, hücre içi hareket sayısında meydana gelen değişiklik.

#### Önerilen Prosedür :

**Adım 0 :** ng = UL ve B = { $\phi$ } olarak başla.

**Adım 1 :** Çözüm matrisindeki makine ve parçalar arasındaki mesafelerde en uzun ng-1 tanesini seç. Bu elemanlar arasından yatay ve dikey çizgiler çizerek ng adet makine grubu ve ng adet parça ailesinin sınırlarını belirle.

**Adım 2 :** Aşağıdaki lineer atama modelini çözerek her parça ailesini bir makine grubuna ata. Bu model Hungarian (Macar) algoritması yardımıyla çözülebilir.

$$\text{Maximize} : \sum_{I=1}^{UL} \sum_{J=1}^{UL} NS(I,J)X(I,J)$$

$$\text{Subject to} : \sum_{J=1}^{UL} X(I,J) = 1 \quad I=1,2,\dots,UL$$

$$: \sum_{I=1}^{UL} X(I,J) = 1 \quad J=1,2,\dots,UL$$

$$X(I,J) = 0 \text{ veya } 1 \quad I,J = 1,2,\dots,UL$$

**Adım 3** : Durdurma kriteri sağlanıncaya kadar iteratif olarak grupların sayısı düşürülür. Hücre içi hareketlerin daha fazla düşürülemediği durumda durdurma kriteri sağlanmış olur. Burada izlenecek adımlar şunlardır:

1. Her bir  $S(I,J) \in B$  olmak üzere  $\Delta(I,J) = \Delta H(I) + \Delta V(J) - NS(I,J)$  'yi hesapla

Burada:

$$\Delta H(I) = \max \{NS(I,J^+(I)), NS(I,J^-(I))\}$$

$$\Delta V(J) = \max \{NS(I^+(J),J), NS(I^-(J),J)\}$$

Burada:

$$I^+(J) = I \mid S(I,J+1) \in B; \text{ ve } I^-(J) = I \mid S(I,J-1) \in B$$

$$J^+(I) = J \mid S(I+1,J) \in B; \text{ ve } J^-(I) = J \mid S(I-1,J) \in B$$

2.  $\Delta(I^*,J^*) = \max \{ \Delta(I,J) \mid S(I,J) \in B \}$  'yi belirle

3. Eğer  $\Delta(I^*,J^*) \leq 0$  ise  $\delta$ 'ya git, aksi halde hücre içi hareketlerdeki azalmayı maksimum yapacak  $H^* - V^*$  çizgi kombinasyonunu elimine ederek  $S(I^*,J^*)$  bloğunu ayır.  $H^*$  ve  $V^*$  aşağıdaki gibi tanımlanır.

Eğer  $NS(I^*,J^+(I^*)) > NS(I^*,J^-(I^*))$  ise  $H^* = H(I^*)$ , aksi halde  $H^* = H(I^*-1)$

Eğer  $NS(I^+(J^*),J^*) > NS(I^-(J^*),J^*)$  ise  $V^* = V(J^*)$ , aksi halde  $V^* = V(J^*-1)$

4.  $MG(I^*)$  ve  $PF(J^*)$  ile yapılan atamayı boz.  $MG(I^*)$ 'ı kendisinden  $H^*$  yatay çizgisiyle ayrılan komşu makine grubuyla birleştir. Aynı şekilde  $PF(J^*)$ 'i kendisinden  $V^*$  dikey çizgisiyle ayrılan komşu parça ailesiyle birleştir

5. Matrisi güncelleştir,  $ng = ng-1$  yap ve 1'e git

6. Hücre içi hareketlerin daha fazla düşürülemediği durumda dur.

### 5.2.3. Çok boyutlu ölçeklendirme ( Multidimensional scaling) yardımıyla H.Ü.S. tasarımı ( Mukhopadhyay ve ark. 1994)

Çok boyutlu tartılandırma ilk olarak Shepart (1962) tarafından ortaya atılmış bir yöntem olup özellikle pazarlama alanında büyük ilgi görmüştür. Buradaki temel mantık, nesnelere arasındaki benzerlik yargılarından yola çıkarak alternatiflerin grafiksel modelinin çıkarılmasıdır. İki ürün arasındaki benzerliğin(mesafe) bu ürünlerin kümelendirilmesi üzerinde büyük etkisi vardır.

Önerilen yöntemde makineler ve parçalar arasındaki benzerliklerin yanı sıra bunlar arasındaki benzersizlikler de dikkate alınmaktadır. Benzerlik ve

benzersizliklerin hesaplanmasında herhangi bir hücreye atanması gerçekleşmeyen makine ve parçalar kullanılır.

Başlangıç makine-parça ilişki matrisi oluşturulduktan sonra, makine gruplarının ve parça ailelerinin sıralanarak kümelendirilmesi gerekmektedir. Önerilen metotta makineler güç katsayısı( $S_{ij}$ )'na göre sıralanır. Güç katsayısının yüksek değere sahip olması makinenin seçim önceliğini de beraberinde getirmektedir. İki makine arasındaki güç katsayısı, bu iki makine arasındaki mesafeye( $d_{ij}$ ) ve eşleme katsayısına( $M_{ij}$ ) bağlıdır. Buna göre i ve j makineleri arasındaki mesafe aşağıdaki gibi hesaplanır:

$$d_{ij} = [M_{ij}^2 + N_{ij}^2]^{1/2}$$

Burada,  $M_{ij}$  : atanmış bir i makinesi ile atanmamış bir j makinesi arasındaki benzerlik

$N_{ij}$  : atanmış bir i makinesi ile atanmamış bir j makinesi arasındaki benzersizlik

Bu durumda,

$$S_{ij} = M_{ij} / [M_{ij}^2 + N_{ij}^2]^{1/2} \text{ olur.}$$

Sezgisel yöntemde kullanılan notasyonlar ise aşağıdaki gibidir.

**X(P,M)** : makine-parça ilişki matrisi

**P** : ilişki matrisi X(P,M)'deki toplam parça sayısı

**M** : ilişki matrisi X(P,M)'deki toplam makine sayısı

**MD**: atanan makineler

**PD** : atanan parçalar

**MND** : atanmamış makineler

**PND** : atanmamış parçalar

**PMA** : bir önceki iterasyonda atanan makine

**PPA(PMA)** : PMA için atanan parça

**N(J)** : J makinesinde işlem gören parça sayısı

**MT(I,J)** : atanmış bir I makinesi ile atanmamış bir J makinesi arasındaki benzerlik sayısı (her iki makinede işlem gören parça sayısı)

**NT(I,J)** : atanmış bir I makinesi ile atanmamış bir J makinesi arasındaki benzersizlik sayısı

**D(I,J)** : atanmış bir I makinesi ile atanmamış bir J makinesi arasındaki mesafe

$S(I,J)$  : atanmış bir I makinesi ile atanmamış bir J makinesi arasındaki güç katsayısı

$NP(P)$  : atanmamış bir J makinesi tarafından yapılan parça

Prosedürün adım adım işleyişi ise şöyledir:

**Adım 1 :** İlişki matrisi  $X(P,M)$ 'i oku

Başla

$$MD = 0$$

$$PD = 0$$

**Adım 2 :**  $N(J)$ 'yi bul,  $\forall J \in M$  veya  $MND$

**Adım 3 :** minimum  $N(J)$ 'yi seç, eğer bir eşitlik varsa ilk  $J$ 'yi al

**Adım 4 :** atanmış makine =  $J$  yap ve adım 7'ye git

**Adım 5 :**  $MIJ$ ,  $NIJ$  ve  $S(I,J)$ 'yi bul

Burada,  $I=PMA$  ve  $\forall J \in MND$

**Adım 6 :** maksimum  $S(I,J)$  değerini seç, eğer  $\max S(I,J) = 0$  ise adım 2'ye git

Eğer  $S(I,J)$ 'ler arasında bir eşitlik varsa, maksimum  $M(I,J)$ 'yi seç

Eğer  $M(I,J)$ 'ler arasında bir eşitlik varsa en küçük  $J$  sayısını seç

Atanmış makine =  $J$  yap

**Adım 7 :**  $PMA = J$  yap

İlişki matrisinden  $PPA(PMA)$ 'yi bul

Eğer  $PPA(PMA) \leq 1$  ise adım 10'a git, aksi halde devam et

**Adım 8 :**  $\forall p \in PPA(PMA)$  için  $NP(P)$ 'yi bul

$$NP(P) = \sum X(P,J), \quad X(P,J) = 1 \text{ ve } \forall j \in MND \text{ için}$$

**Adım 9 :** minimum  $NP(P)$ 'yi ve böylece minimum  $P$ 'yi seç, eğer bir eşitlik varsa minimum  $P$  sayısını seç

**Adım 10 :** atanmış parçalar  $(PD) = P$  yap,  $\forall p \in PPA(PMA)$  için

$$PD = PD + PPA(PMA)$$

$$PND = PND - PPA(PMA)$$

$$MD = MD + PMA$$

$$MND = MND - PMA$$

**Adım 11 :** eğer  $MND > 1$  ise adım 5'e git

eğer  $MND = 1$  ise adım 7'ye git, aksi halde devam et

**Adım 12 :** çözüm matrisini oluştur ve grup etkinliğini hesapla

**Adım 13 : Bitir.**

Sonuç olarak yazar tarafından belirtilen algoritmanın avantajları ise şunlardır:

1. Algoritmanın basit ve hesap zamanının düşük olması sebebiyle büyük veri setleri bu algoritma yardımıyla çözülebilir.

2. Bu metot block diagonal matris oluşturma yeteneğine sahiptir.

3. Bu algoritma herhangi bir kullanıcı müdahalesine ve kararına ihtiyaç duymamaktadır (Mukhopadhyay ve ark. 1994).

Bu sezgisel prosedür makine ve parça sıraları için optimuma yakın sonuçlar vermesine karşın hücre sınırlarını belirleme yönünde bir çözüm sunmamaktadır. Bu prosedür yardımıyla elde edilen çözüm matrisini üretim hücrelerine bölmek amacıyla, Aljeber ve arkadaşları tarafından önerilen sınır belirleme algoritması kullanılmıştır.



## 6. ARAŞTIRMA SONUÇLARI

### 6.1. Yöntemlerin Karşılaştırılması

Önceki bölümlerde iki farklı kategoriye ayrılmış toplam beş adet prosedür Pascal programlama dilinde kodlanmış olup bu programlar ekte sunulmuştur. Bu algoritmalar literatürden alınan 10 farklı makine-parça ilişki matrisi üzerinde uygulanarak, elde edilen sonuçlar değerlendirilecektir. Hatırlanacağı gibi bu değerlendirmede dikkat edilecek 4 kriter şunlar idi:

1. Makine Kullanımı (Machine Utilization)
2. Hücrelerarası Hareket Sayısı (Number of Intercellular Elements)
3. Bağ Enerjisi (Bond Energy)
4. Grup Yeteneği (Group Efficiency)

### 6.2. Test Problemleri

Bu bölümde, H.Ü.S. tasarımı konusunda yapılan çok sayıda araştırma sırasında kullanılan makine-parça ilişki matrislerinden 10 adeti alınarak analiz çalışmalarında kullanılmıştır. Bu matrisler hakkında özet bilgiler Tablo 3'de verilmiş olup bu matrisler ekte sunulmuştur. Bu matrisler, H.Ü.S. tasarımı için geliştirilen çok sayıda algoritmanın etkinliğinin test edilmesi ve diğer bazı algoritmalara nazaran sonuçlarının kalitesinin değerlendirilmesi amacıyla literatürde karşımıza çıkan matrislerdir. Test problemi tanımlaması yapılırken, problemi ilk olarak kullanan araştırmacının isminin ilk üç harfi ve yayınlanma tarihi birleştirilmiştir.

Tablo 3. Karşılaştırmada kullanılan test problemleri

Sıra No	Test problemi	Boyut	Kaynak
1	BUR75	16×43	Burbidge(1975)
2	MOO93	12×19	Moon(1993)
3	KIN80	14×24	King(1980)
4	MUK94	7×9	Mukhopadhyay(1994)
5	BOE91A	8×20	Boe ve Cheng(1991)
6	MOS85	10×10	Mosier ve Taube(1985)
7	BOC90	7×11	Boctor(1990)
8	BOE91B	20×35	Boe ve Cheng(1991)
9	BOC910	16×30	Boctor(1991)
10	BOC911	"	"

### 6.3. Test Sonuçları

#### 6.3.1. Program sonuçları

Tablo 3'te tanıtılan 10 adet test problemi Pascal programlama dilinde kodlanan üç yapay zeka tekniği ve iki sezgisel yöntem üzerinde aynı parametre değerleri kullanılarak uygulanmış ve yapay zeka teknikleri yardımıyla bazı makine ve parça grupları, sezgisel yöntemler yardımıyla ise bazı yeni makine ve parça sıraları elde edilmiştir. Daha sonra, elde edilen bu sonuçlar kullanılarak test problemleri üzerinde değişiklikler yapılmış ve çözüm matrisleri elde edilmiştir.

Elde edilen makine ve parça grupları ile sıraları Tablo 4,5,6,7 ve 8'de verilmiştir.

Uygulamalarda tüm test problemleri için aynı parametre değerleri kullanılmış olup bu parametre değerleri şunlardır:

1. ART 1 nöron modeli için ihtiyat değeri  $\rho = 0.49$  alınmıştır.
2. Bulanık ART nöron modeli için  $\alpha = 0.5$ ,  $\beta = 0.1$  ve  $\rho = 0.7$  alınmıştır.
3. Yarışmalı eğitim algoritması için öğrenme oranı  $l = 0.05$  ve  $nepoch = 100$  alınmıştır.

4. Grup yeteneğinin hesaplanmasında kullanılan ağırlık faktörü tüm çözüm matrisleri için  $\alpha = 0.5$  olarak alınmıştır.

Tablo 4. ART 1 nöron modeli için program sonuçları

Test Prob.	Makine grubu sayısı	Makine grupları	Parça grubu sayısı	Parça grupları
1	8	(1) (2.9.16) (3.14) (4.5.15) (6.7) (8.11.13) (10) (12)	12	(1.6.39) (2.4.18) (3.9.11.15.20.31) (5.14.16.29) (7.17.34.35.36) (8.12.23.43) (10.28.32.38.40) (13.25.26) (19.21.33.41) (22.30) (24.27) (37.42)
2	6	(1.2) (3) (4.8) (5.6.7) (9.12) (10.11)	7	(1.4.13.14.16) (2.3.5.10) (6.8.18) (7.11.19) (9.17) (12) (15)
3	6	(1.12) (2.3.10) (4.5.7) (6.8.9.14) (11) (13)	7	(1.2.17.19.20) (6.7.8.18) (3.4.21.24) (5.9.10.14.16.22) (11.13) (12.15) (23)
4	3	(1.2.7) (3.4.6) (5)	3	(1.2.6) (3.7.8) (4.5.9)
5	3	(1.3) (2.4.7.8) (5.6)	3	(1.5.10.12.15) (3.4.6.7.18.20) (2.8.9.11.13.14.16.17.19)
6	5	(1) (2.3.4.8) (5.6) (7.9) (10)	4	(1.4) (2.9.10) (3.8) (5.6.7)
7	4	(1.5) (2.3) (4.7) (6)	5	(1.2.6) (3.4.11) (5.8.10) (7) (9)
8	10	(1.6) (2.4.13.18) (3) (5) (7.8.17) (9.10.20) (11.12) (14) (15.16) (19)	11	(1.3.5.15.17.29) (2.7.10) (8.14.16) (4.6.9.11.21.33) (12.13.18.27) (19.22) (20.23.25.34) (24.31) (26) (28) (30.32.35)
9	10	(1.2) (6.9) (3.5.10) (4) (7) (8) (9.11) (12.15) (14) (16)	12	(1.3.4.7.19) (2.5) (6) (8.20.23.29) (9.10) (11.13.15.18) (12.22) (14) (16) (17.25.26.27) (21) (24.28.30)
10	8	(1.2.8) (3.10.16) (4.6.13) (5.11) (7.12) (9) (14) (15)	10	(1.2.14) (3.18.21.22) (4.9.12.13.15.16) (5.6.8.10.26) (7.24.25) (11.19) (17) (20.23.27.29) (28) (30)

Tablo 5. Bulanık ART nöron modeli için program sonuçları

Test Prob.	Makine grubu sayısı	Makine grupları	Parça grubu sayısı	Parça grupları
1	4	(1.2.6.7.9.16) (10) (3.12.14) (4.5.8.11.15)	7	(1.2.3.4.6.8.12.31) (5.9.14.15.16.19.21.23.29.33.41) (7.10.17.18.28.32.38.40) (11.20.22.24.27.30) (13.25.26.34.39) (37.42.43)
2	4	(1.2.3.5.6.7) (4.8) (9.12) (10.11)	4	(1.4.13.14.16) (2.3.5.7.9.10.17) (6.8.12.15.18) (11.19)
3	4	(1.12.13) (2.3.10.11) (4.5.7) (6.8.9.14)	4	(1.2.17.19.20.23) (3.4.21.24) (5.9.10.11.12.13.14.15.16.22) (6.7.8.18)
4	2	(1.2.5.7) (3.4.6)	2	(1.2.4.5.9) (3.6.7.8)
5	3	(1.3) (2.4.7.8) (5.6)	3	(1.5.10.12.15) (2.8.9.11.13.14.16.17.19) (3.4.6.7.18.20)
6	3	(1.7.9) (2.3.4.8) (5.6.10)	3	(1.4.5.6.7) (2.9.10) (3.8)
7	3	(1.5.6) (2.3) (4.7)	3	(1.2.6.9) (3.4.7.11) (5.8.10)
8	4	(1.6.9.10.11.20) (2.4.13.14.18) (3.5.7.8.17) (12.15.16.19)	5	(1.3.5.15.17.20.23.25.29) (2.7.10.12.13.18.24.27.31) (4.6.9.11.21.28.30.33) (8.14.16.19.22.26) (32.34.35)
9	4	(1.2.6.9.11.13.16) (3.4.5.7.8.10) (12.15) (14)	6	(1.3.4.6.7.11.13.14.19) (2.5.9.10.12.17) (8.20.23.24.25.26.27.28.29) (15.16.18) (21) (22.30)
10	3	(1.2.5.8.9.11.14) (3.10.15.16) (4.6.7.12.13)	4	(1.2.4.14.15.17) (3.7.18.20.21.22.23.24.27.30) (5.6.8.9.10.11.12.13.16.19.26) (25.28.29)



Tablo 6. Yarışmalı eğitim algoritması için program sonuçları

Test Prob.	Makine grubu sayısı	Makine grupları	Parça grubu sayısı	Parça grupları
1	4	(1,2,3,6,7,9,10,12,16) (4,5,15) (11,13) (8,14)	4	(1,3,5,8,9,11,12,13,14,15,20,22,24,25,26,27,30,31,39) (16,19,21,23,29,33,41,43) (6,7,17,34,35,36) (2,4,10,18,28,32,37,38,40,42)
2	3	(1,2,4,5,6,8) (3,10,11) (7,9,12)	4	(1,4,13,14,16) (6,8,12,15,18) (3,7,11,19) (2,5,9,10,17)
3	4	(2,3,9,11) (1,12,13) (6,8) (4,5,7,10,14)	4	(1,12,17,19,20,23) (5,9,10,12,14,15,16,22) (11,13) (3,4,6,7,8,18,21,24)
4	2	(1,3,5,6) (2,4,7)	2	(1,4,5,9) (2,3,6,7,8)
5	3	(6) (1,3) (2,4,5,7,8)	3	(3,4,6,7,18,20) (2,8,9,11,13,14,16,17,19) (1,5,10,12,15)
6	3	(1,2,7,9) (3,4,8) (5,6,10)	3	(3,8) (1,4,5,6,7) (2,9,10)
7	3	(1,3,5,6) (2,7) (4)	3	(1,2,6,9) (3,7,11) (4,5,8,10)
8	4	(2,4,6,7,9,10,13,20) (11,12,14,15,16,19) (1,8,17,18) (3,5)	3	(4,6,8,9,11,14,21,28,30,32,33) (16,19,22,23,26) (1,2,3,5,7,10,12,13,15,17,18,20,24,25,27,29,31,34,35)
9	4	(3,5,7,10,13) (4,8,12,15) (1,6,11,14,16) (2,9)	4	(15,19,26,27,29) (8,10,23) (1,6,16,18,25,28,30) (2,5,9,12,17,20,21,22,24)
10	4	(2,4,6,12,13) (1,7,8) (11,14) (3,5,9,10,15,16)	3	(1,2,3,7,14,17) (18,20,21,22,23,24,25,27,28,29,30) (4,5,6,8,9,10,11,12,13,15,16,19,26)

Tablo 7. Çok boyutlu ölçeklendirme algoritması için program sonuçları

Test Prob.	En iyi makine sırası	En iyi parça sırası
1	1,16,9,2,6,8,5,4,15,13,11,12,7,10,14,3	42,37,18,7,10,32,38,2,4,28,40,6,34,39,12,13,17,33,1,14,23,43,19,11,15,20,31,3,8,27,41,9,21,24,16,29,5,30,22,25,26,35,36
2	2,1,5,6,3,4,8,10,11,9,12,7	7,8,6,12,17,18,15,13,1,4,14,16,10,2,9,5,3,19,11
3	1,13,12,7,5,4,2,3,11,10,14,9,8,6	6,7,18,23,8,1,2,17,20,19,4,3,21,24,10,14,12,5,13,9,15,16,22,11
4	2,7,1,5,4,6,3	4,1,5,9,2,6,3,8,7
5	8,2,7,4,5,6,1,3	4,7,20,3,6,18,12,10,1,5,15,14,11,19,17,9,2,8,13,16
6	5,6,10,2,4,3,8,7,9,1	3,8,10,9,2,7,5,6,1,4
7	5,1,6,4,7,2,3	7,3,1,11,4,5,10,8,2,6,9
8	13,4,14,18,2,7,3,17,8,1,11,15,19,12,16,20,9,6,10,5	2,15,12,24,27,7,10,18,31,29,3,17,1,5,15,20,32,30,25,35,9,34,11,6,16,28,4,21,33,19,22,26,23,8,14
9	10,5,3,12,15,7,8,11,9,2,13,16,6,1,14,4	1,12,15,16,4,3,13,11,22,18,21,7,19,14,6,24,8,9,28,30,2,10,25,5,17,20,27,26,29,23
10	9,14,2,8,11,1,5,16,3,10,15,4,6,12,7,13	11,16,7,19,12,5,10,15,4,9,13,18,26,6,8,2,28,17,1,14,25,3,21,23,24,27,20,29,30,3

Tablo 8. Tabu araştırma algoritması için program sonuçları

Test Prob.	En iyi makine sırası	En iyi parça sırası
1	7,10,9,1,15,5,3,4,14,8,6,16,2,11,12,13	7,10,32,4,6,13,33,1,14,42,38,2,18,28,40,34,17,23,39,12,3,20,19,11,15,43,31,27,8,41,9,21,16,5,24,29,30,22,26,35,25,36
2	2,9,12,11,10,1,6,3,8,4,7,5	11,1,4,5,17,19,13,16,14,10,3,7,6,9,2,15,18,8,12
3	9,14,13,5,4,7,11,10,6,8,2,3,1,12	6,18,8,17,19,21,7,23,2,1,20,3,4,10,12,24,14,13,11,15,16,5,9,22
4	2,7,1,5,3,6,4	9,8,3,6,2,7,1,5,4,1
5	3,1,2,4,8,7,5,6	14,2,8,4,6,3,18,9,12,10,20,17,19,11,16,13,15,5,7,1
6	1,9,7,8,10,6,5,3,2,4	3,8,7,5,6,4,1,10,9,2
7	3,2,7,4,6,1,5	1,7,8,9,11,10,5,4,6,2,3
8	5,8,9,2,13,3,1,6,4,18,7,17,16,19,14,20,12,11,10,15	4,11,33,1,5,17,3,15,29,2,7,10,6,14,21,9,19,18,8,16,12,13,27,22,33,20,24,25,34,36,31,35,30,28,32
9	5,12,15,10,3,7,11,9,13,8,2,16,6,1,4,14	6,22,11,17,10,28,1,12,15,16,3,4,13,18,21,7,19,14,26,20,2,23,29,27,5,25,30,24,8,9
10	16,11,2,14,8,9,1,5,3,15,4,10,12,7,13,6	25,3,21,26,18,5,11,7,12,16,19,15,8,28,17,10,9,4,13,6,2,1,14,23,24,27,20,29,30,22

### 6.3.2. Kümelenendirme sonuçları

Literatürden alınan 10 adet test probleminin karşılaştırması yapılacak beş yöntem üzerinde uygulanması sonucu elde edilen hücre sayısı (HS), makine kullanımı (MK), hücrelerarası hareket sayısı (HHS), bağ enerjisi (BE) ve grup yeteneği (GY) değerleri, her bir test problemi için ayrı ayrı tablolar halinde aşağıda verilmiştir.

Tablo 9. Test problemi 1 için çözüm sonuçları

Çözüm Yöntemi	HS	MK	HHS	BE	GY
ART I	4	0,088	74	108	0,471
Bulanık ART	2	0,045	35	95	0,470
Yarışmalı Eğitim	3	0,065	43	85	0,484
Çok Boyutlu Ölçeklendirme	3	0,068	28	122	0,503
Tabu Araştırması	4	0,058	32	98	0,492

Tablo 10. Test problemi 2 için çözüm sonuçları

Çözüm Yöntemi	HS	MK	HHS	BE	GY
ART I	4	0,187	20	48	0,532
Bulanık ART	3	0,150	11	47	0,537
Yarışmalı Eğitim	3	0,130	6	44	0,543
Çok Boyutlu Ölçeklendirme	3	0,154	10	53	0,543
Tabu Araştırması	2	0,104	16	44	0,481

Tablo 11. Test problemi 3 için çözüm sonuçları

<b>Çözüm Yöntemi</b>	<b>HS</b>	<b>MK</b>	<b>HHS</b>	<b>BE</b>	<b>GY</b>
<i>ART 1</i>	5	0,202	18	58	0,567
<i>Bulanık ART</i>	4	0,162	2	62	0,577
<i>Yarışmalı Eğitim</i>	4	0,148	11	53	0,549
<i>Çok Boyutlu Ölçeklendirme</i>	4	0,162	3	70	0,575
<i>Tabu Araştırması</i>	3	0,154	8	60	0,564

Tablo 12. Test problemi 4 için çözüm sonuçları

<b>Çözüm Yöntemi</b>	<b>HS</b>	<b>MK</b>	<b>HHS</b>	<b>BE</b>	<b>GY</b>
<i>ART 1</i>	2	0,212	6	22	0,506
<i>Bulanık ART</i>	2	0,218	3	21	0,560
<i>Yarışmalı Eğitim</i>	2	0,212	5	26	0,521
<i>Çok Boyutlu Ölçeklendirme</i>	2	0,212	3	33	0,556
<i>Tabu Araştırması</i>	2	0,212	4	27	0,539

Tablo 13. Test problemi 5 için çözüm sonuçları

<b>Çözüm Yöntemi</b>	<b>HS</b>	<b>MK</b>	<b>HHS</b>	<b>BE</b>	<b>GY</b>
<i>ART 1</i>	3	0,153	9	79	0,539
<i>Bulanık ART</i>	3	0,153	10	79	0,539
<i>Yarışmalı Eğitim</i>	3	0,150	13	71	0,514
<i>Çok Boyutlu Ölçeklendirme</i>	3	0,153	9	76	0,535
<i>Tabu Araştırması</i>	3	0,153	11	73	0,522

Tablo 14. Test problemi 6 için çözüm sonuçları

<b>Çözüm Yöntemi</b>	<b>HS</b>	<b>MK</b>	<b>HHS</b>	<b>BE</b>	<b>GY</b>
<i>ART 1</i>	2	0,200	4	25	0,560
<i>Bulanık ART</i>	3	0,303	2	29	0,636
<i>Yarışmalı Eğitim</i>	3	0,294	5	26	0,609
<i>Çok Boyutlu Ölçeklendirme</i>	3	0,294	4	33	0,616
<i>Tabu Araştırması</i>	3	0,285	3	28	0,619

Tablo 15. Test problemi 7 için çözüm sonuçları

<b>Çözüm Yöntemi</b>	<b>HS</b>	<b>MK</b>	<b>HHS</b>	<b>BE</b>	<b>GY</b>
<i>ART 1</i>	2	0,189	5	13	0,532
<i>Bulanık ART</i>	3	0,269	2	15	0,624
<i>Yarışmalı Eğitim</i>	2	0,162	1	14	0,566
<i>Çok Boyutlu Ölçeklendirme</i>	3	0,269	2	21	0,615
<i>Tabu Araştırması</i>	3	0,250	6	15	0,563

Tablo 16. Test problemi 8 için çözüm sonuçları

<b>Çözüm Yöntemi</b>	<b>HS</b>	<b>MK</b>	<b>HHS</b>	<b>BE</b>	<b>GY</b>
<i>ART 1</i>	3	0,083	67	129	0,468
<i>Bulanık ART</i>	4	0,113	41	137	0,517
<i>Yarışmalı Eğitim</i>	4	0,098	39	112	0,511
Çok Boyutlu Ölçeklendirme	4	0,110	28	158	0,526
Tabu Araştırması	3	0,114	25	145	0,538

Tablo 17. Test problemi 9 için çözüm sonuçları

<b>Çözüm Yöntemi</b>	<b>HS</b>	<b>MK</b>	<b>HHS</b>	<b>BE</b>	<b>GY</b>
<i>ART 1</i>	4	0,126	59	97	0,480
<i>Bulanık ART</i>	2	0,066	25	97	0,481
<i>Yarışmalı Eğitim</i>	3	0,091	52	90	0,461
Çok Boyutlu Ölçeklendirme	2	0,066	15	120	0,502
Tabu Araştırması	3	0,086	22	116	0,490

Tablo 18. Test problemi 10 için çözüm sonuçları

<b>Çözüm Yöntemi</b>	<b>HS</b>	<b>MK</b>	<b>HHS</b>	<b>BE</b>	<b>GY</b>
<i>ART 1</i>	3	0,100	47	94	0,477
<i>Bulanık ART</i>	3	0,093	14	93	0,523
<i>Yarışmalı Eğitim</i>	3	0,090	30	81	0,495
Çok Boyutlu Ölçeklendirme	3	0,091	12	119	0,526
Tabu Araştırması	3	0,112	24	108	0,518

## 7. TARTIŞMA

Bu çalışma kapsamında incelenen 3 yapay zeka tekniđi kullanıcıya makine gruplarını ve parça ailelerini verirken, incelenen 2 sezgisel yöntem başlangıç makine-parça ilişki matrisindeki makine ve parça sıralarını deđiştirmektedir. Yapay zeka tekniklerinin kullanımı sonucu elde edilen bu gruplar kullanılarak yeni bir makine-parça ilişki matrisi düzenlenir ve üretim sisteminde bulunacak hücre sayısı ise kullanıcı tarafından belirlenir. Sezgisel yöntemler kullanılarak elde edilen yeni makine ve parça sıraları kullanılarak elde edilen yeni makine-parça ilişki matrisinin hücrelere bölünmesi işlemi ise Aljebra ve arkadaşları tarafından geliştirilen algoritmada ayrı bir fazda gerçekleştirilirken, Çok Boyutlu Ölçeklendirme algoritmasında bölme işlemine yönelik herhangi bir prosedür geliştirilmemiştir.

Ayrıca incelenen yapay zeka tekniklerinin kullanımı için birtakım dışsal parametre deđerleri gerekli olup, bu parametre deđerlerinin seçimi kullanıcıya bırakılmıştır. Optimum parametre deđerlerinin problemin niteliğine göre farklılık göstereceđi bilindiđi gibi bu deđerlerin ne olacağı da belirli deđildir.



## 8. SONUÇ VE ÖNERİLER

Yukarıdaki tablolarda sunulan sonuçlar yapay zeka teknikleri ve sezgisel yöntemler için ayrı ayrı incelendiğinde, değerlendirmede gözönüne alınan dört kriter bakımından yapay zeka teknikleri içerisinde genel olarak tüm test problemleri için en iyi sonuçları Bulanık ART modelinin verdiği anlaşılmıştır.

İncelenen iki sezgisel yöntemden ise yine genel olarak tüm test problemleri için en iyi sonuçları Çok Boyutlu Ölçeklendirme algoritmasının verdiği anlaşılmıştır.

Çok Boyutlu Ölçeklendirme algoritması ve Bulanık ART modeli yardımıyla elde edilen sonuçlar karşılaştırıldığında ise büyük boyutlu problemlerde Çok Boyutlu Ölçeklendirme algoritmasının daha etkili sonuçlar verdiği anlaşılmıştır.

Çeşitli boyutlarda problemler için bu beş yöntem birbiriyle karşılaştırıldığında ise büyük boyutlu problemlerde sezgisel yöntemlerin yapay zeka tekniklerine nazaran daha etkili sonuçlar verdiği ve yapay zeka tekniklerinin kullanıcı müdahalesine çok fazla ihtiyaç duyduğu bu çalışmada tespit edilmiştir.

Sonuç olarak bu çalışmada, 4 değerlendirme kriteri ve işletim kolaylığı bakımından en etkili prosedür Çok Boyutlu Ölçeklendirme algoritması bulunmuştur.

İleriki çalışmalarda yapay zeka teknikleri için gerekli olan parametrelerin farklı değerleri kullanılarak değişik sonuçlar elde edilerek yeni bir kıyaslama yoluna ve elde edilen bu sonuçlar için faktör analizi çalışmalarına başvurulabilir.

## 9. KAYNAKLAR

- ABDELMOLA A. I., TABOUN S.M. 1999. Productivity Model for T Cell Formation Problem: A Simulated Annealing Algorithm. *Computers & Industrial Engineering*, 37.
- ALAHVERDİ, N.1999. Yapay Sinir Ağları, Ders Notları.
- ALJABER N.,vd. 1997. A Tabu Search Approach to The Cell Formation Problem. *Computers & Industrial Engineering*, 32, 1.
- BASU A. ,vd. 1995. An Expert System Based Approach to Manufacturing Cell Design. *International Journal of Production Research*,33, 10.
- BİLGE,Ü., vd. 1996. Malzeme Taşıma Sistemlerinde Otomatik Güdümlü Araçlar İçin Zeki Bir Kontrol Mekanizmasının Geliştirilmesi, IMS.
- BOCTOR F. F. 1991. A Linear Formulation of The Machine-Part Cell Formation Problem, *International Journal of Production Research*,29, 2.
- BOCTOR F. F. 1996. The Minimum Cost, Machine-Part Cell Formation Problem, *International Journal of Production Research*, 34, 4.
- BROWN, E.1995. Carol, Introduction To Artificial Intelligence and Expert Systems, [www.bus.orst.edu/faculty](http://www.bus.orst.edu/faculty).
- CHU C.H. 1993. Manufacturing Cell Formation by Competitive Learning, *International Journal of Production Research*, 31, 4.
- CHU C-H. 1997. An Improved neural Network for manufacturing cell formation, *Decision support Systems*, 20.
- ERSÖZ ,S.1998. Üretim Pazarlama Entegrasyonunda Uzman Sistemler, Basılmamış Doktora Tezi, Gazi Üniv.
- GOONATILAKE, S.,KHEBBAL, S. 1995. Intelligent Hybrid System, John Wiley and Sons.
- GU P. 1999. A Genetic Algorithms-Based Approach for Design of Manufacturing Systems: An Industrial Application. *International Journal of Production Research*, 37,3.
- GÜNGÖR, Z. 1998. An Approach to Fuzzy Decision Making in Cell Formation Problem. IMS.
- HAYKIN, S. 1994. *Neural Networks, A Comprehensive Foundaditon* Prentice –Hall.
- HUNT, D.1989. *Computer Integrated Manufacturing Handbook*, Chapman and Hall.
- KAPARTHI S., SURESH N.C. 1994. Performance of Fuzzy ART Neural Network for Group Technology Cell Formation, *International Journal of Production Research*, 32, 7.
- KUSIAK A. 1988. A Knowledge Based System for Group Technology, *International Journal of Production Research*, 26, 5.
- KUSIAK, A.1998. *Artificial Intelligence in Industry*, ISF Publication.
- LEEM C. W., CHEN J. G. 1996. Fuzzy Set Based Machine-Cell Formation in Cellular Manufacturing, *Journal of Intelligent Manufacturing*, 7.
- LIAO T.W., CHEN L. 1993. An Evaluation of ART1 Neural Models for GT Part Family and Machine Cell Forming, *Journal of Manufacturing Systems*, 12, 4.
- LOGENDRAN R., vd. 1994. Tabu Search-Based Heuristics for Cellular Manufacturing Systems in The Presence of Alternative Process Plans. *International Journal of Production Research*, 32, 2.

- LOZANO S. , vd. 1999. A One Step Tabu Search Algorithm for Manufacturing Cell Design, Journal of Operational Research Society, 50.
- MABERT, A.V., JACOBS, R.F. 1991. Integrated Production systems, IIE .
- MALEKİ ,R. 1991. Flexible Manufacturing Systems, The Technology and Management, Printice-Hall.
- MOON C., GEN M. 1999. A Genetic Algorithm-Based Approach for Design of Independent Manufacturing Cells. International Journal of Production Economics, 60-61.
- MOON Y.B. 1990. Forming Part-Machine Families for Cellular Manufacturing : A Neural Network Approach, International Journal of Advanced Manufacturing Technology, 5.
- MOON Y.B., CHI S.C. 1993. Generalized Part Family Formation Using Neural Network Techniques, Journal of Manufacturing Systems, 11, 3.
- MOSCA, R.1996. Study of the behaviour of flexible production systems using neural nets, Production Planning and Control, 7,5 462-470.
- MUKHOPADHYAY S. K., vd.1994. Machine-Compenent Grouping in Cellular Manufacturing by Multidimensional Scaling, International Journal of Production Research, 32, 2.
- PARRISH, D.1990. Flexible Manufacturing, Butterworth Heinemann.
- PATTERSON, D.1996. Artifical Neural Networks, Theory and Applications, Prentice-Hall.
- SINGH ,N., RAJOMANI, D.1996. Celular Manufacturing Systems desing, Planing and control, Champman & Hall.
- SU C.T., HSU C.M. 1998. Multi-Objective Machine-part Cell Formation Through Parallel Simulated Annealing, International Journal of Production Research. 36, 8.
- VELLA, A.1998. The application of Artifical Intelligence to Manufacture IMS.
- VEMURİ, V.R.1992. Artifical Neural Networks concepts and Control Application, IEEE Computer Society Press.





```

10100010001
11000100000
01000100100
00011000010
00100010000
00110000001
00001001010

```

Şekil 21. Test problemi (makine-parça) 7 (Boctor 1990)

```

10000100001001100101001110000101011
01000000010110000110000100100010000
1010100000000010000000000001010000
01000010000110000000000100100000000
00001000000001001000001001000000000
00000001000001000010010010000100000
10101010000100101011011101001111000
10001000100000101001001010000000000
00000001000001000010001000000001001
00000001000001010010010001000000000
00010100100000010000000100010101001
0001010010100000000010000000000100
01000000000110000000000100000000000
01000010010110000100000100100010000
00010100101000000000100000010100000
00010100100100000100000001000101000
10101000000000101001000010000000001
01000010000110000001000100000010000
00010000101000000000100000010101000
00000001000001000010001001000000000

```

Şekil 22. Test problemi (makine-parça) 8 (Boe ve Cheng 1991)

```

000010010101000000010110010010
010010001000000010011100111000
000000000011101001000100000000
001000100000110000000110000001
001100000011101100000100000000
000000000100000000001010011111
000000100010011100000000000000
100001000010000101000000000000
010000001101000000000101100101
100000000001001100000000000000
000000011001000100000001000101
101100100010101101001000000000
000000001100001010000010111010
000000010000000000010010100000
101100100010110100100000000000
000000010000000010010011111111

```

Şekil 23. Test problemi (makine-parça) 9 (Boctor 1991)

```

00001101011110000000000010000
00011000111110110100000010000
11010000000000001000000000000
0010000000000000001001011101100
010011011100000000000000010100
00100000000000000000011011001011
000000100000000000010011101110
000110001101100000100000010000
00000000010000000000000000000
11000000000010010000000000000
000011011110100000100000010000
00100000000000000001011010000111
00100010000000000001001101001001
0000101001110001001000000000000
1000000000000110000000000000000
1100000000000000010000000000000

```

Şekil 24. Test problemi (makine-parça) 10 (Boctor 1991)

## Ek B. Algoritmaların Bilgisayar Programları

### Ek B.1. Procedure ART 1

```

Uses crt;
Var
i, j, k, Pay, Toplam, Sayac, Ara_Toplam, Parca_Sayisi, Makine_Say
isi, Best:Byte;
  Parca_esik, Makine_esik, Max, Sonuc:Real;
  Step3_e_don:Boolean;
  U:Array [1..74] of Real;
  Y_str, B_str:String;
  Hucre: Array [1..74] of String;
  E, Makine, Parca, Flag: Array [1..74] of Byte;
  B: Array [1..74, 1..74] of Real;
  A: Array [1..74, 1..74] of Byte;
  T: Array [1..74, 1..74] of Byte;

Procedure Deger_Al;
Begin
  Repeat
    Clrscr;
    Write('Makine grubu icin esik degerini
giriniz.....(0-1 arasinda):');
    Readln(Makine_Esik);
    Until ((Makine_Esik>0)and (Makine_Esik<1));
    Repeat
      Clrscr;
      Write('Parca grubu icin esik degerini
giriniz.....:');
      Readln(Parca_Esik);
      Until ((Parca_Esik>0) and (Parca_Esik<1));
      Clrscr;
      Write('Parça sayısın
giriniz....:');Readln(Parca_Sayisi);
      Write('Makine sayısın
giriniz....:');Readln(Makine_Sayisi);
    end;

procedure Step_1;
Begin
  For i:=1 to Parca_Sayisi do
    For j:=1 to Makine_Sayisi do
      Begin
        T[i, j]:=1;
        B[i, j]:=1/(Parca_Sayisi+1);
      end;
    end;
  end;

```

```

    End;
    For j:=1 to Makine_Sayisi do Flag[j]:=0;
    K:=1;
End;

Procedure Step_2;
Begin
    For j:=1 to Makine_Sayisi do
        For i:=1 to Parca_Sayisi do
            Begin
                Write(j,'.Makinenin ',i,'.Parca degerini
giriniz...');
                Readln(A[i,j]);
            End;
        End;
    End;

Procedure Step_3;
Begin
    for j:=1 to Makine_Sayisi do
        Begin
            U[j]:=0;
        End;

        For j:=1 to Makine_Sayisi do
            Begin
                For i:=1 to Parca_Sayisi do
                    U[j]:=U[j]+B[i,j]*A[i,k];
                End;
            End;
        End;

Procedure Step_3_Parca;
Begin
    for i:=1 to Parca_Sayisi do
        Begin
            U[i]:=0;
        End;

        For I:=1 to Parca_Sayisi do
            Begin
                For J:=1 to Makine_Sayisi do
                    U[I]:=U[I]+B[I,J]*A[K,J];
                End;
            End;
        End;

Procedure Step_4;
Begin
    Max:=0;
    For J:=1 to Makine_Sayisi do
        Begin

```

```

        If ((U[J]>Max) and (Flag[J]=0)) then
        Begin
            Max:=U[J];
            Best:=J;
        End;
    End;
    Makine[K]:=Best;
End;

```

```

Procedure Step_4_Parca;
Begin
    Max:=0;
    For I:=1 to Parca_Sayisi do
    Begin
        If ((U[I]>Max) and (Flag[I]=0)) then
        Begin
            Max:=U[I];
            Best:=I;
        End;
    End;
    Parca[K]:=Best;
End;

```

```

Procedure Step_7;
Begin
    for I:=1 to parca_Sayisi do E[i]:=T[I,best];
    For I:=1 to Parca_Sayisi do T[I,Best]:=E[I]*A[I,K];
    For I:=1 to Parca_Sayisi do
    Begin
        Ara_Toplam:=0;
        For Sayac:=1 to Parca_Sayisi do
Ara_Toplam:=Ara_Toplam+E[Sayac]*A[Sayac,K];
        B[I,Best]:=E[I]*A[I,K]/(0.5+Ara_Toplam);
    End;
End;

```

```

Procedure Step_7_Parca;
Begin
    for J:=1 to Makine_Sayisi do E[J]:=T[Best,J];
    For J:=1 to Makine_Sayisi do T[Best,J]:=E[J]*A[K,J];
    For J:=1 to Makine_Sayisi do
    Begin
        Ara_Toplam:=0;
        For Sayac:=1 To Makine_Sayisi Do
Ara_Toplam:=Ara_Toplam+E[Sayac]*A[K,Sayac];
        B[Best,J]:=E[J]*A[K,J]/(0.5+Ara_Toplam);
    End;
End;

```

```

Procedure Step_6;
Begin
    Flag[Best]:=1;
    Step3_e_Don:=True;
End;

Procedure Step_6_Parca;
Begin
    Flag[Best]:=1;
    Step3_e_Don:=True;
End;

Procedure Step_5;
Begin
    Toplam:=0;
    Pay:=0;
    For I:=1 To Parca_Sayisi do Toplam:=Toplam+A[I,K];
    For I:=1 To Parca_Sayisi do
Pay:=Pay+T[I,Best]*A[I,K];
    Sonuc:=Pay/Toplam;
    If Sonuc>Makine_Esik then Step_7
    else Step_6;
End;

Procedure Step_5_Parca;
Begin
    Toplam:=0;
    Pay:=0;
    For J:=1 To Makine_Sayisi Do Toplam:=Toplam+A[K,J];
    For J:=1 To Makine_sayisi Do
Pay:=Pay+T[Best,J]*A[K,J];
    Sonuc:=Pay/Toplam;
    If Sonuc>Parca_Esik Then Step_7_Parca
    Else Step_6_Parca;
End;

Procedure Step_8(Var Sinir:Byte);
Begin
    For J:=1 To Sinir Do Flag[J]:=0;
    K:=K+1;
End;

Procedure Makine_Hucresi_Yaz;
Var
    X,Y:Byte;
Begin
    For X:=1 To Makine_Sayisi Do
    For y:=1 To makine_sayisi Do
    Begin

```

```

        If Makine[Y]=X Then
        Begin
            Str(Y,Y_Str);
            if Hucre[X]='' Then
Hucre[X]:=Hucre[X]+Y_Str
            Else
            Hucre[X]:=Hucre[X]+' '+Y_Str;
        End;
    End;
    For X:=1 To Makine_Sayisi Do
    Begin
        If Hucre[X]<>' ' Then
        Begin
            Writeln('(' ,Hucre[X],') Nolu Makineler
',X,'. makine grubuna aittir');
        End;
    End;
End;

Procedure Parca_Hucresi_Yaz;
Var
    A,B:Byte;
Begin
    For A:=1 To Parca_Sayisi Do Hucre[A]:='';
    For A:=1 To parca_Sayisi Do
    For B:=1 To parca_Sayisi Do
    Begin
        If Parca[B]=A Then
        Begin
            Str(B,B_Str);
            If Hucre[A]='' Then
Hucre[A]:=Hucre[A]+B_Str
            Else
            Hucre[A]:=Hucre[A]+' '+B_Str;
        End;
    End;
    For A:=1 To Parca_Sayisi Do
    Begin
        If Hucre[A]<>' ' Then
        Begin
            Writeln('(' ,Hucre[A],') Nolu Parcalar
',A,'. parca grubuna aittir');
        End;
    End;
End;

```

```

{***** ANA PROGRAM
*****}

```

```

Begin
  Textcolor(2);
  Clrscr;
  Deger_A1;
  Step_1;
  Step_2;
  Repeat
    Repeat
      Step3_e_Don:=False;
      Step_3;
      Step_4;
      Step_5;
    Until Step3_e_don=False;
    Step_8(Makine_Sayisi);
  Until K>Makine_Sayisi;
  Clrscr;
  Textcolor(4);
  Writeln('          Makine Gruplar□      ');
  Writeln('          -----          ');
  Textcolor(2);
  Makine_Hucresi_Yaz;
{*****}
*****}
  Step_1;
  Repeat
    Repeat
      Step3_e_Don:=False;
      Step_3_Parca;
      Step_4_Parca;
      Step_5_Parca;
    Until Step3_e_Don=False;
    Step_8(Parca_Sayisi);
  Until K>Parca_Sayisi;
  Textcolor(4);
  Writeln;
  Writeln('          Parca Gruplar□      ');
  writeln('          -----          ');
');Writeln;
  Textcolor(2);
  Parca_Hucresi_Yaz;
  Readkey;
End.

```



**Ek B.2. Procedure Bulanik ART**

```

uses crt;
type

xx=array[1..50,1..50]of real;
var
mat1,mat:xx;
yer:xx;
atama:array[1..40,1..40]of integer;
ata,ter:array[1..40] of real;
wi:array[1..40,1..40]of real;
w,u,s,t,x,y,z:real;
b1,b2,kk,aa,ss,ii,cc,c,rr,tt,k,i,j,n,m:integer;
ttt,enb,enk:real;
label A1,A2,A3;
begin
  clrscr;
  u:=0;s:=0; rr:=1; cc:=0; ii:=0;ss:=0; aa:=1;
  write('ALFA deSerini giriniz.....:');readln(x);
  write('BETA deSerini giriniz.....:');readln(y);
  write('RO deSerini giriniz.....:');readln(z);
  write('makine adetini giriniz.....:');readln(n);
  write('Parça sayısn giriniz.....:');readln(m);
  clrscr;
  for i:=1 to n do
  for j:=1 to m do
    begin
      write(i,'.Makinenin ',j,'.parçasın giriniz..');
      readln(mat[i,j]);
    end;
  to m do
  begin
    wi[i,j]:=1;
    {mat[i,j] :=mat[i,j]/ata[i];}
    yer[i,j]:=mat[i,j];
  end;
  for i:=1 to n do
  begin
    for k:=1 to n do
    begin
      u:=0;w:=0;

      for b1:=1 to n do
      for b2:=1 to m do
      begin
        yer[b1,b2]:=mat[b1,b2];
      end;
      for j:=1 to m do

```

```

begin
  begin
    yer[i,j]:=wi[k,j];
  end;
  u:=u+yer[i,j];
  w:=w+wi[k,j]
end;
ter[k]:=u/(w+x);
end;
A1:
  enb:=ter[1]; {enk:=ter[1];} tt:=1;
for k:=1 to n-1 do
  begin
    if enb<ter[k+1] then
      begin
        enb:=ter[k+1]; tt:=k+1;
      end;
    { if ter[k+1]<enk then
      begin enk:=ter[k+1]; end;}
  end;
{ if enb=enk then begin c:=1 end;}

  c:=tt;
  for j:=1 to m do
    begin
      yer[i,j]:=mat[i,j];
    end;
    u:=0; s:=0;
    for j:=1 to m do
      begin
        if yer[i,j]>wi[c,j] then
          begin
            yer[i,j]:=wi[c,j];
          end;
        end;
        ttt:=(u/s);
        if (u/s)>=z then
          begin
            for j:=1 to m do
              begin
                wi[c,j]:=((y*yer[i,j])+((1-
y)*wi[c,j]));
              end;
            end
          else begin
            ter[c]:=-1; rr:=1; cc:= cc+1;
            {if cc=n then begin goto a2 end else} goto A1
;
          end;
        end;
      end;
    end;
  end;
end;

```

```

cc:=0;
begin
  atama[c,i]:=i;
  {ii:=ii+1;}
  if c>ss then
    begin
      ss:=ss+1;
    end;
  { gotoxy((1*j),3+ii);
  write(i,'.Ma..','c','. Ma Grubuna ayit');readln;}
end;
end;
if aa=1 then
begin
  gotoxy(20,1);writeln('MAK~NA GRUPLARI');
  gotoxy(18,2);writeln('-----');
  for i:=1 to ss do
  begin
    for j:=1 to n do
    begin
      ii:=j*2;
      gotoxy(ii,3+i);
      if atama[i,j]<>0 then
      begin
        write(atama[i,j],',');
      end
    end;
    gotoxy((j*2)+2,3+i);
    writeln('  Nolu Makinalar...','i','  .Makina
Grubuna aittir');

    end;
  end;
if aa=3 then
begin
  gotoxy(20,12);writeln('PAR~A GUPLARI');
  gotoxy(18,13);writeln('-----');
  for i:=1 to ss do
  begin
    for j:=1 to n do
    begin
      ii:=j*2;
      gotoxy(ii,14+i);
      if atama[i,j]<>0 then
      begin
        write(atama[i,j],',');
      end
    end;
  end;
end;

```

```
        gotoxy((j*2)+2,14+i);
        writeln(' ` Nolo Par#alar...:',i,' ` .Par#a
Grubuna aittir');
    end;
    aa:=2;
end;
    if aa=1 then
    begin
        for i:=1 to n do
        for j:=1 to m do
        begin
            mat1[j,i]:=mat[i,j];
        end;
        kk:=n;
        n:=m;
        M:=kk;
        for i:=1 to n do
        for j:=1 to m do
        begin
            mat[i,j]:=mat1[i,j];
        end;
        aa:=3;
        goto A3;
    end;

readkey;
end.
```

### Ek B.3. Procedure Yarışmalı Eğitim

```

uses crt;
var
  ss,n, a, aa, sat,enbn,msay,psay,cell,k,i,j,t:integer;
  y, x,w,temp:array[1..50,1..50]of real;
  s,top,p:array[1..50]of real;
  enb,l:real;
  m:array[1..50]of byte;
  atama:array[1..50,1..50]of byte;
label A1;
begin
  ss:=0;aa:=1 ;
  clrscr;
  write('makina sayı[s]n giriniz...:');readln(msay);
  write('parça sayı[s]n giriniz...:');readln(psay);
  write('hücre sayı[s]n giriniz...:');readln(cell);
  write('l("$renme oran) deşerini
giriniz...:');readln(l);
  write('n(nepoch) deşerini giriniz...:');readln(n);
  clrscr;
  for j:=1 to psay do
    for i:=1 to msay do
      begin
        gotoxy(i*3,j);
        read(x[j,i]);

        end;
  clrscr;
  repeat
  ss:=ss+1;
  A1:
  for i:=1 to cell do
    top[i]:=0;
  if ss=1 then
  begin
    randomize;
    for k:=1 to cell do
    for i:=1 to msay-1 do
    begin
      for i:=1 to cell do
        w[i,msay]:=1-top[i];
    end;
    for i:=1 to cell do
    for j:=1 to psay do
    begin
      atama[i,j]:=0;
    end;
    for sat:=1 to psay do

```

```

begin
  for k:=1 to cell do
    begin
      s[k]:=0;
    end;
  for k:=1 to cell do
    for t:=1 to msay do
      begin
        temp[k,t]:=w[k,t]*x[sat,t];
        s[k]:=s[k]+temp[k,t];
      end;

      enb:=s[1];enbn:=1;
      for i:=2 to cell do
        begin
          if s[i]>enb then
            begin
              enb:=s[i];enbn:=i;
            end;
          end;
        atama[enbn,sat]:=sat;
        p[sat]:=0;
        for k:=1 to msay do
          begin
            p[sat]:=p[sat]+x[sat,k];
          end;
        for i:=1 to msay do
          begin
            end;
          end;
        end;
      if ss=1+n then
        begin
          if aa=1 then
            begin
              for i:=1 to cell do
                for j:=1 to psay do
                  begin
                    if atama[i,j]>0 then
                      begin
                        writeln(' ',atama[i,j],' Nolu Par#a ',i, '. Par#a
grubuna aittir ');
                      end;
                    end;
                  aa:=2;
                end;
              if aa=3 then
                begin
                  readkey;
                  readkey;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```
for i:=1 to cell do
for j:=1 to psay do
begin
if atama[i,j]>0 then
begin
writeLn(' ',atama[i,j],' Nolu Makine ',i,'.
Makine grubuna aittir' );
end;
end;
end;

if aa=2 then
begin
for i:=1 to psay do
for j:=1 to msay do
begin
y[j,i]:=x[i,j];
end;
a:=psay;
psay:=msay;
msay:=a;
for i:=1 to psay do
for j:=1 to msay do
begin
x[i,j]:=y[i,j];
end;
aa:=3;
goto A1;

end;
end;
until(ss=n+1);
readkey;
end.
```

#### Ek B.4. Procedure Tabu Araştırması

```

uses crt;
var
mat2,mat3,mat1,mat:array[1..50,1..50] of integer;
tablol,tablo:array[1..7,1..50]of integer;
ata:array[1..50]of integer;
d:array[1..50,1..50]of real;
TTT,k,i,j,n,m,t1,t2,a:integer;
rr,dd,kk,AA,t,b,yy,w,z,wi,x,y,par,mak:integer;
k1,yenidx,dx:real;
label A1,A3;
begin
  randomize;
  clrscr; wi:=0; yy:=1; b:=0; t:=0; aa:=1;dd:=1;
  write('makine sayıların giriniz...:');readln(mak);
  write('Parçaların giriniz...:');readln(par);
  for i:=1 to mak do
    for j:=1 to par do
      begin
        write(i, '.Makinenin',j, '.parçasının giriniz..');
        readln(mat[i,j]);
      end;
  clrscr;
  A3:
    for i:=1 to 7 do
      for j:=1 to mak do
        begin
          tablo[i,j]:=0;
        end;

wi:=0; yy:=1; b:=0; t:=0; dd:=1;
for i:=1 to mak do
  for j:=1 to par do
    begin
      a:=0; t1:=0 ;t2:=0;
      for k:=1 to par do
        begin
          if (mat[i,k]=1) and (mat[j,k]=1) then
            begin
              a:=a+1
            end;
          t1:=t1+mat[i,k];
          t2:=t2+mat[j,k];
        end;
      d[i,j]:=1-(a/(t1+t2-a))
    end;
  for i:=1 to mak do
    begin

```



```

ata[i]:=i;
end;
A1:
dx:=0;
for i:=1 to mak do
begin
  x:=ata[i];
  y:=ata[i+1];
  dx:=dx+d[x,y];
end;
if yy=1 then
begin
  yenidx:=dx;
end;

if YY=2 Then
begin
  if dx<=yenidx then
  begin
    for i:=1 to 7 do
    for j:=1 to mak do
    begin
      if t<mak then
      begin
        if tablo[i,j]=ata[j] then
        begin
          t:=t+1;
        end
      else begin t:=0; j:=mak; end;
      end;
    end;
  if t<mak then
  begin
    yenidx:=dx;
    wi:=0;
    for i:=1 to mak do
    begin
      tablo1[1,i]:=ata[i];
    end;
    for i:=1 to 7 do
    for j:=1 to mak do
    begin
      tablo1[i+1,j]:=tablo[i,j];
    end;
    for i:=1 to 7 do
    for j:=1 to mak do
    begin
      tablo[i,j]:=tablo1[i,j];
    end;
  end;
end;

```

```

        end;
    end;
end;
    m:=0;N:=0;
    if wi<k1 then
    begin
        wi:=wi+1;
        repeat

            n:=random(mak)+1;
            m:=random(mak)+1;
            until n<>m;
            for i:=1 to mak do
            begin
                if i=n then
                begin
                    z:=ata[i];
                end;
                if i=m then
                begin
                    w:=ata[i];
                end;
            end;
            ata[n]:=w;
            ata[m]:=z;
        yy:=2;
        goto A1;
    end;
if aa=1 then
begin
    GOTOXY(6+mak,3);
    writeln('MAK~NALAR ~□~N TABU L~STES~ ');
    for i:=1 to 7 do
    for j:=1 to mak do
    begin
        gotoxy((j*3)+2,i+4);
        writeln(tablo[i,j]:2);
        TTT:=(J*3)+6;
    end;
    GOTOXY(TTT,5);
    writeln('EN ~Y~ MAK~NA SIRASI ');
    readkey;
end;
    if aa=1 then
    begin
        for i:=1 to mak do
        begin
            ata[i]:=tablo[1,i];
        end;
    end;

```

```

for i:=1 to mak do
  for j:=1 to par do
    begin
      rr:=ata[i];
      mat2[i,j]:=mat[rr,j];
    end;
  end;
clrscr;
if aa=3 then
begin
gotoxy(6+par,3);
writeln('PARALAR ^N TABU L^STES^ ');
for i:=1 to 7 do
for j:=1 to mak do
begin
gotoxy((j*3)+2,i+4);
writeln(tablo[i,j]:2);
TTT:=(J*3)+6;
end;
GOTOXY(TTT,5);
writeln('EN ^Y^ PAR^A SIRASI ');
readkey;
end;
  if aa=1 then
begin
    For i:=1 to mak do
      for j:=1 to par do
        begin
          mat1[j,i]:=mat[i,j];
        end;
      kk:=mak;
      mak:=par;
      par:=kk;
      for i:=1 to mak do
        for j:=1 to par do
          begin
            mat[i,j]:=mat1[i,j];
          end;
        aa:=3;
        yy:=1;
        goto A3;
      end;

  clrscr;
  readkey;
  for i:=1 to mak do
    begin
      ata[i]:=tablo[1,i];
    end;

```

```
gotoxy(10,4);
writeln('ZŠM MATR~S~');
for i:=1 to mak do
  for j:=1 to par do
    begin
      rr:=ata[i];
      mat3[j,i]:=mat2[j,rr];
    end;
  for i:=1 to mak do
    for j:=1 to par do
      begin
        gotoxy(5+(j*3),5+i);
        writeln(mat3[i,j]:2);
      end;
    end;
  end;
readkey;
end.
```



### Ek B.5. Procedure Çok Boyutlu Ölçeklendirme

```

Uses crt;
Var
  pd_str,md_str,ss:string;
  s:array[1..15,1..15] of real;
  mt,nt,d,x: array[1..15,1..15] of integer;
  ppa,md,pd,N,np: array [1..15] of integer;
  max_s_j,min_np,min_j,parca_sayisi,makine_sayisi:byte;
  p_say,sayac,mnd_sayisi,pnd_sayisi,mnd,pnd,pma,i,j:
byte;
  max_s:real;
procedure step1;
begin
  clrscr;
  write('Parca sayisini giriniz
...:');readln(parca_sayisi);
  write('Makine sayisini giriniz
...:');readln(makine_sayisi);
  for j:=1 to makine_sayisi do
  for i:=1 to parca_sayisi do
  Begin
    Write(j,'. makinelenin ',i,'. parca degerini
giriniz..:');
    readln(x[i,j]);
  End;
end;

procedure step2;
Begin
  for j:=1 to makine_sayisi do
  for i:=1 to parca_sayisi do
    if x[i,j]=1 then n[j]:=n[j]+1;
End;

procedure step3;
Begin
  min_j:=parca_sayisi;
  for j:=1 to makine_sayisi do
  if n[j]<min_j then min_j:=n[j];
end;

procedure step78;forward;

procedure step4;
Begin
  md_str:='';
  md[min_j]:=1;
  str(min_j,ss);

```

```

    md_str:=md_str+ss;
end;

procedure step78;
var
t,np_yedek,p_yedek,sira:array [1..15]of byte;
point,c,say,cntr:byte;
begin
    pma:=min_j;
    cntr:=0;
    for i:=1 to parca_sayisi do
    begin
        np[i]:=0;
        p_yedek[i]:=0;
    end;

    for i:=1 to parca_sayisi do
    if ((x[i,pma]=1)and(ppa[i]=0)) then
    begin
        point:=i;
        cntr:=cntr+1;
        ppa[i]:=1;
        p_yedek[cntr]:=i;
        pd[i]:=1;
        for j:=1 to makine_sayisi do
            if x[i,j]=1 then np[i]:=np[i]+1;
        end;

        if ((cntr<>1)and(cntr<>0)) then
        begin
            for i:=1 to parca_sayisi do np_yedek[i]:=np[i];
            for i:=1 to parca_sayisi-1 do
                for j:=i+1 to parca_sayisi do
                begin
                    if np_yedek[i]>np_yedek[j] then
                    begin
                        c:=np_yedek[i];
                        np_yedek[i]:=np_yedek[j];
                        np_yedek[j]:=c;
                    end;
                end;
            end;
            for i:=parca_sayisi downto 1 do
                t[i]:=np_yedek[i];
                say:=0;
                for i:=1 to parca_sayisi do
                begin
                    for j:=1 to parca_sayisi do
                        if ((t[i]<>0)and(t[i]=np[j])) then
                            begin

```

```

        say:=say+1;
        sira[say]:=j;
    end;
end;
for i:=1 to cntr do
begin
    p_say:=p_say+1;
    if p_say<=parca_sayisi then
    begin
        str(sira[i],ss);
        pd_str:=pd_str+'-'+ss;
    end;
end;
end else
begin
    p_say:=p_say+1;
    if p_say<=parca_sayisi then
    begin
        str(point,ss);
        pd_str:=pd_str+'-'+ss;
    end;
end;
end;
end;

```

```

procedure step9;
begin
    min_np:=makine_sayisi;
    for i:=1 to parca_sayisi do
    if np[i]<min_np then min_np:=np[i];
end;

```

```

procedure step6;
begin
    max_s:=0;
    for j:=1 to makine_sayisi do
    if s[pma,j]>max_s then
    begin
        max_s_j:=j;
        max_s:=s[pma,j]
    end;
    min_j:=max_s_j;
    md[min_j]:=1;
    str(min_j,ss);
    md_str:=md_str+'-'+ss;
end;

```

```

procedure step5;
begin

```

```

for j:=1 to makine_sayisi do
if ((j<>pma)and(md[j]=0)) then
begin
for i:=1 to parca_sayisi do
begin
if ((x[i,j]=1) and (x[i,pma]=1)) then
mt[pma,j]:=mt[pma,j]+1;
if (((x[i,j]=1) and
(x[i,pma]=0))or((x[i,j]=0) and
(x[i,pma]=1)))
then nt[pma,j]:=nt[pma,j]+1;
end;
s[pma,j]:=mt[pma,j]/(sqrt(sqr(mt[pma,j])+sqr(nt[pma,j])))
;
end;
step6;
end;

```

```

procedure step11;
var
son_j:byte;
begin
{
for i:=1 to parca_sayisi do
if pd[i]=0 then pnd_sayisi:=pnd_sayisi+1;}
mnd_sayisi:=0;
for j:=1 to makine_sayisi do
if md[j]=0 then
begin
mnd_sayisi:=mnd_sayisi+1;
son_j:=j;
end;
if mnd_sayisi>1 then step5;
if mnd_sayisi=1 then
begin
min_j:=son_j;
md[min_j]:=1;
str(min_j,ss);
md_str:=md_str+'-'+ss;
end;
end;

```

```

Begin
sayac:=0;
step1;
step2;
step3;
step4;
repeat

```



```
    sayac:=sayac+1;
    step78;
    step9;
    step11;
until sayac=makine_sayisi;
clrscr;
writeln(md_str);
write(pd_str);readkey;
End.
```

