

**TÜRKİYE CUMHURİYETİ
ÇUKUROVA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANA BİLİM DALI**

**ATÖLYE TİPİ ÜRETİM ÇİZELGELEMEDE METASEZGİSEL
YAKLAŞIMLAR VE YAPAY SİNİR AĞI UYGULAMASI**

İNCİLAY YILDIZ

YÜKSEK LİSANS TEZİ

ADANA / 2019

**TÜRKİYE CUMHURİYETİ
ÇUKUROVA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANA BİLİM DALI**

**ATÖLYE TİPİ ÜRETİM ÇİZELGELEMEDE METASEZGİSEL
YAKLAŞIMLAR VE YAPAY SİNİR AĞI UYGULAMASI**

İNCİLAY YILDIZ

Danışman: Prof. Dr. Selçuk ÇOLAK

Jüri Üyesi: Dr. Öğr. Üyesi Mert DEMİRCİOĞLU

Jüri Üyesi: Dr. Öğr. Üyesi Fatih ABUT

YÜKSEK LİSANS TEZİ

ADANA / 2019

ukurova niversitesi Sosyal Bilimler Enstits Mdrlgne;

Bu alıřma, jrimiz tarafından İřletme Ana Bilim Dalında YKSEK LİSANS TEZİ olarak kabul edilmiřtir.

Bařkan: Prof. Dr. Seluk OLAK
(Danıřman)

ye: Dr. gr. yesi Mert DEMİRCİOđLU

ye: Dr. gr. yesi Fatih ABUT

ONAY

Yukarıdaki imzaların, adı geen ğretim elemanlarına ait olduklarını onaylarım.

.../.../2019

Enstit Mdr

NOT: Bu tezde kullanılan ve bařka kaynaktan yapılan bildiriřlerin, izelge, řekil ve fotođrafların kaynak gsterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunu'ndaki hkmlere tabidir.

ETİK BEYANI

Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde ve ortaya çıkan sonuçlarda herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim. / / 2019

İMZA

İncilay YILDIZ

ÖZET

ATÖLYE TİPİ ÜRETİM ÇİZELGELEMEDE METASEZGİSEL YAKLAŞIMLAR VE YAPAY SİNİR AĞI UYGULAMASI

İNCİLAY YILDIZ

Yüksek Lisans Tezi, İşletme Ana Bilim Dalı

Danışman: Prof. Dr. Selçuk Çolak

Ağustos 2019, 86 sayfa

Üretim çizelgeleme, üretim planlaması çalışmaları içinde ileriye yönelik kısa periyotlu planlar olarak kabul edildiğinden etkin bir çizelgeleme ve kontrolün üstünlükleri ve üretim sürecine katkıları çoktur. Kaynakların etkin kullanımı verimliliği yükseltirken müşteriler açısından da siparişlerin zamanında karşılanmasını sağlar. En basit çizelgeleme sistemi bile karmaşık bir çözüm yapısına sahiptir. Uzun tedarik süreleri de talebin doğru olarak tahmin edilmesini güçleştirmektedir. Bu tür yönetilmesi zor üretim süreçleri için çizelgeleme problemlerinin etkin şekilde çözümlenmesi büyük önem arz etmektedir.

Atölye tipi çizelgeleme problemleri (ATÇP), NP-zor problemler sınıfında yer alan kombinatoriyal problemlerdendir. Bu tip problemlerde kısıtlar arttıkça çözüm uzayı sonsuza doğru gitmeye başlar ve kesin optimum çözümü bulmak gittikçe zorlaşır. Bu nedenle son yıllarda bu tip problemleri çözmek için metasezgisel algoritmalarından yararlanılmaktadır.

Bu tez çalışmasında ATÇP için kullanılan metasezgisel yöntemler ve literatür çalışmaları açıklanmıştır. Ayrıca çizelgeleme için C# programlama dili kullanılarak yapay sinir ağı (YSA) algortiması geliştirilmiştir. Taillard karşılaştırma problemlerinin atölye tipi üretim verileri kullanılarak, geliştirilen YSA yöntemi ile toplam işlem süresi (makespan) hesaplanmıştır. Elde edilen sonuçlar literatürde bulunan sonuçlar ile karşılaştırılmış ve algoritmanın etkinliği test edilmiştir.

Anahtar Kelimeler: Atölye Tipi Üretim Sistemi, Çizelgeleme Problemleri, Metasezgisel Yöntemler, Yapay Sinir Ağı

ABSTRACT**JOB SHOP SCHEDULING WITH METAHEURISTIC APPROACHES AND
ARTIFICIAL NEURAL NETWORK APPLICATION****İNCİLAY YILDIZ****Master of Business Administration****Supervisor: Prof.Dr. Selçuk Çolak****August 2019, 86 pages**

Since production scheduling is considered as short-term plans for the future in production planning, the advantages of effective scheduling and control and its contribution to the production process are numerous. Efficient use of resources improves productivity and ensures that orders are met on time for customers. Even the simplest scheduling system has a complex solution structure. Long lead times also make it difficult to estimate the demand accurately. It is very important to solve scheduling problems effectively for such difficult-to-manage production processes.

Job shop scheduling problems are among the combinatorial problems in the NP-hard problems class. As constraints increase in such problems, the solution space starts to go to infinity and it becomes increasingly difficult to find the exact optimum solution. For this reason, in recent years, metaheuristic algorithms have been used to solve such problems.

In this thesis, metaheuristic methods and literature studies used for JSS are explained. In addition, an artificial neural network algorithm has been developed by using the C# programming language for scheduling. Total processing time (i.e. makespan) was calculated by using the ANN method on Taillard Job Shop benchmark instances. The results were compared with the ones reported in the literature.

Key Words: Job Shop Production, Scheduling Problems, Metaheuristic Methods, Artificial Neural Network

ÖNSÖZ

Birçok akademik ve idari görevi bulunan tez danışmanım Sayın Prof. Dr. Selçuk ÇOLAK'a yoğun çalışmaları arasında bana zamanını ayırarak yardımcı ve destek olduğu için teşekkürlerimi sunarım.

Araştırmanın uygulama kısmı sürecinde verdiği yardım ve destek için Sayın Öğr. Gör. Abdülvahap SAYGIN'a teşekkür ederim.

Tüm eğitim hayatım boyunca maddi ve manevi desteğini benden esirgemeyen sevgili annem Cavidan KURTULDU'ya çalışmayı gerçekleştirdiğim dönem içerisinde gösterdiği sabır ve özveriden dolayı minnettarım.

Ayrıca baş öğretmen Mustafa Kemal ATATÜRK başta olmak üzere, ilk öğretimden bugüne kadar üzerimde emeği olan değerli öğretmenlerim ve akademisyenlerime, tez savunma jürisinde bulunan değerli akademisyenler Dr. Öğr. Üyesi Mert DEMİRCİOĞLU ve Dr. Öğr. Üyesi Fatih ABUT'a teşekkürlerimi sunmayı bir borç bilirim.

İncilay YILDIZ

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
KISALTMALAR	x
TABLolar LİSTESİ	xi
ŞEKİLLER LİSTESİ	xii

BÖLÜM I

GİRİŞ

1.1. Problem.....	1
1.2. Çalışmanın Amacı	2
1.3. Çalışmanın Önemi	2
1.4. Çalışmanın Yöntemi.....	3
1.5. Çalışmanın Planı.....	3

BÖLÜM II

ÜRETİM ÇİZELGELEME

2.1. Üretim Sistemleri, Yönetimi ve Tarihçesi	5
2.2. Üretim Çizelgeleme.....	6
2.2.1.Gantt Şemaları	7
2.3.Üretim Çizelgeleme Problemleri	7
2.3.1. Atölyeye Geliş Şekline Göre Sınıflandırma	8
2.3.1.1. Statik Problemler	8
2.3.1.2. Dinamik Problemler.....	9
2.3.2. İş ve Makine Sayısına Göre Sınıflandırma	9
2.3.2.1. Tek Makine – Çok İş Problemi	9
2.3.2.2. Çok Makine – Çok İş Problemi.....	9
2.4. P ve NP problemler	10
2.4.1. Kombinatoryal Optimizasyon Problemleri.....	10
2.5. Atölye Tipi Üretim Çizelgeleme Problemleri.....	11

2.5.1. Sabit Atölye Tipi	11
2.5.2. Esnek Atölye Tipi	12
2.5.3. Açık Atölye Tipi	12
2.6. Üretim Çizelgelemede Optimizasyon	13
2.7. Üretim Çizelgeleme Problemlerine Çözüm Yaklaşımları.....	13

BÖLÜM III

ATÖLYE TİPİ ÇİZELGELEMEDE SEZGİSEL VE METASEZGİSEL YÖNTEMLER

3.1. Metasezgisel Yöntemler	16
3.1.1. Açgözlü Yaklaşımı (Greedy Approach)	16
3.1.2. Ateşböceği Algoritması (Firefly Algorithm)	16
3.1.3. Tavlama Benzetimi (Simulated Annealing)	18
3.1.4. Değişken Komşu Arama (Variable Neighborhood Search)	19
3.1.5. Genetik Algoritma (Genetic Algorithm)	20
3.1.6. Hibrit Algoritmalar (Hybrid Algorithm)	22
3.1.7. Karınca Kolonileri Optimizasyonu (Ant System)	23
3.1.8. Örnekleme & Kümeleme (Sampling & Clustering)	25
3.1.9. Paralel Kanguru Algoritması (Parallel Kangaroo Algorithm)	26
3.1.10. Parçacık Sürü Optimizasyonu (Particle Swarm Optimization).....	28
3.1.11. Tabu / Yasaklı Arama (Tabu Search)	30
3.1.12. Yapay Bağışıklık Sistemi (Artificial Immune System).....	31
3.2. Atölye Tipi Üretim Çizelgelemede Metasezgisel Optimizasyon Çalışmaları	32

BÖLÜM IV

ATÖLYE TİPİ ÇİZELGELEMEDE YAPAY ZEKÂ UYGULAMALARI

4.1. Yapay Zekâ ve Tarihçesi.....	40
4.2. Makine Öğrenmesi.....	41
4.3. Yapay Sinir Ağlarında Öğrenme.....	42
4.4. Yapay Sinir Ağları ve Tarihçesi.....	44
4.5. Yapay Sinir Ağları Yapısı.....	45
4.6. Atölye Tipi Çizelgeleme Problemi Çözümünde Kullanılan Yapay Sinir Ağı Yaklaşımları ve Çalışmaları	51

BÖLÜM V

ATÖLYE TİPİ ÇİZELGELEMEDE YAPAY SİNİR AĞI İLE UYGULAMA

5.1. Taillard Atölye Tipi Çizelgeleme Problemi	58
5.2. Taillard Veri Seti ile Atölye Tipi Çizelgelemede Yapay Sinir Ağ Uygulaması..	58
5.3. Atölye Tipi Çizelgelemede Taillard Veri Setini Kullanan Çalışmalar.....	62

BÖLÜM VI

SONUÇ

6.1. Çalışmanın Sonuçlarının Literatür Çalışmaları ile Karşılaştırılması	68
--	----

KAYNAKÇA.....	76
----------------------	-----------

ÖZGEÇMİŞ	86
-----------------------	-----------



KISALTMALAR

- YSA:** Yapay Sinir Ağları
ATÇP: Atölye Tipi Çizelgeleme Problemi
P: Bunlara polinomsal zamanda çalışan algoritma
NP: polinomiyal zaman sınırı olmayan problemler
EATÇ: Esnek atölye tipi çizelgeleme
AATÇ: Açık atölye tipi çizelgeleme
AY: Açgözlü Yaklaşımı
AA: Ateşböceği Algoritması
TB: Benzetim Tavlama
DKA: Değişken Komşu Arama
GA: Genetik Algoritma
HA: Hibrit Algoritma
KKO: Karınca Kolonileri Optimizasyonu
ÖK: Örnekleme & Kümeleme
PKA: Paralel Kanguru Algoritması
PSO: Parçacık Sürü Optimizasyonu
TA: Tabu / Yasaklı Arama
YBS: Yapay Bağışıklık Sistemi
BK: Bulanık Küme
EİT: En İyileştirme Teknikleri
YÇT: Yaklaşık Çözüm Teknikleri
BEÇ: Bilinen En iyi Çözüm

TABLOLAR LİSTESİ

	Sayfa
Tablo 1. En İyileme Teknikleri.....	14
Tablo 2. Yaklaşık Çözüm Teknikleri	14
Tablo 3 Metasezgisel Yaklaşımlar	15
Tablo 4 Ateş böceği Algoritması	17
Tablo 5 Bağışıklık Sistemi ile Atölye Tipi Çizelgeleme Sisteminin Karşılaştırması.....	32
Tablo 6 Biyolojik Sinir Hücresi ile Yapay Sinir Ağı Benzerliği	47
Tablo 7 YSA yöntemi ile elde edilen Sonuçların Taillard Sonuçları ile Karşılaştırılması	67
Tablo 8 Son Yıllarda Yapılan Çalışmalar ile Bu Çalışmanın Yöntem Karşılaştırması	68
Tablo 9 Sha ve Hsu'nun 2006 Yılındaki Çalışması ile YSA Sonuçlarının karşılaştırılması	69
Tablo 10 M. Chandrasekaran Ve Diğerlerinin 2007 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması.....	70
Tablo 11 Zhang ve Diğerlerinin 2008 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması	71
Tablo 12 Nasiri ve Kianfar'ın 2010 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması	72
Tablo 13 Gao ve Diğerlerinin 2014 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması	73
Tablo 14 Peng ve Diğerlerinin 2015 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması	74
Tablo 15 Bürgy ve Diğerlerinin 2017 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması	75

ŞEKİLLER LİSTESİ**Sayfa**

Şekil 1 Genel DKA Algoritması	20
Şekil 2 Genetik Algoritmanın Adımları	22
Şekil 3 Akıllı Sistemler.....	23
Şekil 4 Karıncalar Yiyeceğe Ulaşmak İçin En Kısa Yolu Seçme Yöntemleri	25
Şekil 5 Yapay Sinir Ağının FormüselYapısı.....	48
Şekil 6 Yapay Sinir Ağı Yapısı.....	49
Şekil 7 YSA Modelinin Katmanlarında Bulunan Veriler	59
Şekil 8 YSA'nın Matematiksel Modellemesi.....	60



BÖLÜM I

GİRİŞ

Bu tez çalışmasında, Atölye Tipi Çizelgeleme Problemi (ATÇP) araştırılmış, literatür çalışmaları incelenmiştir. Son yıllarda ATÇP çözümü için kullanılan metasezgisel yöntemler araştırılarak, açıklanmıştır. Bu çalışmada geliştirilen Yapay Sinir Ağları (YSA) yöntemi, Taillard'ın atölye tipi çizelgeleme verileri kullanılarak test edilmiştir. Taillard verilerini kullanan literatür çalışmalarının sonuçları incelenmiş ve bu çalışmada elde edilen sonuçlar diğer araştırmacıların elde ettiği sonuçlar ile karşılaştırılmıştır.

1.1.Problem

Üretim, doğadaki hammadde ve malzemelerin insan gereksinimlerine uygun olacak şekilde mal ve hizmetlere değiştirilmesi sürecinde uğraşılan fiziksel, kimyasal, mekanik ve benzeri işlemler topluluğu olarak tanımlanabilir. Günümüz ihtiyaçları karşılamak amacıyla üretim, kavramsal olarak genişleyerek üretim faktörlerinin tedarikten mal ve hizmetlere dönüşmesini kapsayan bir işletme fonksiyonu haline gelmiştir. Üretim sistemi ise türlü öğelerden oluşmaktadır. Bu sistem, çeşitli girdilerden beklenen çıktıların elde edilmesi süreçlerinden oluşmaktadır (Çelikçapa & Şenol, 2015).

Bir operasyonu oluşturan iş parçalarının mevcut makinelerde hangi sıra ile hangi zaman diliminde işleneceğinin zamansal olarak planlanmasına çizelgeleme denir. İyi bir çizelgeleme üretimde verimliliği artırırken, maliyetleri ve zaman kaybını önler. Optimal çizelgeleme mevcut kaynakların en iyi şekilde kullanılması ile sağlanılabilir. Günümüzde teknolojinin gelişmesi ile artan yöntem skalası dikkat çekmektedir. Özellikle metasezgisel yöntemler, daha az maliyetli olması ve kısa sürede optimal çözüme yakın sonuçlar vermesinden dolayı tercih edilmektedir. Fabrikaların günümüz rekabet ortamında doğru yöntem ile etkin çizelgeleme yaparak sağladıkları verim büyük önem kazanmıştır.

ATÇP çözüm sürecinde ürün çeşitliğinin fazla olması ve gelen siparişlerin küçük miktarda olması sebebiyle, bu tür üretim yapan işletmelerde çok yüksek bir planlama konsantrasyonunun gerekliliği ortaya çıkmaktadır. Etkin bir çizelgeleme ile optimal

zamanda işlerin tamamlanması için planlama yapmak mümkündür. ATÇP sürecinde genellikle genel amaçlı makinelerden oluşan bir sistem içinde malzemelerin yavaş üretim aşamalarından geçmesi söz konusudur. Bu nedenle bu tür sistemlerin zaman çizelgelemesi ve üretim kontrolü maliyetli olup aynı zamanda işlevi zordur. Bir atölye tipi üretim sisteminin genel yapısında mevcut işler çizelgelenmiş makinelerin bekleme hattına alınırlar. Bu bekleme süreleri gantt şemasında boş zaman olarak gösterilir. Makinelerde iş bulunmadığı zaman (makine boşa düştüğünde) çizelgelemede öncelik kuralına göre seçilen işler makinelere atanırlar ve işlem bittikten sonra işleneceği diğer makinenin bekleme hattına taşıma aracı ile taşınırlar. İşlerin, üretim işlemleri tamamlandıktan sonra montaj yapılması gerekiyorsa yapılarak ürün oluşturulur (Akkaya & Gökçen, 2006).

1.2.Çalışmanın Amacı

Bu çalışmada ATÇP için optimal veya optimale yakın çözüm geliştirmek amaçlanmıştır. Atölye tipi bir üretim çizelgenirken makinelerde işlerin herhangi bir işlem sırasına izin verilmektedir. Her iş için, işlemler makinelerde belirli bir sırayla işlenmelidir, ancak bu sıra işlere göre farklılık gösterebilmektedir. Bu tez çalışmasında, kısıtlar göz önünde bulundurularak kodlanan YSA yöntemi ile makine sayısı, iş sayısı, işlerin makinede tamamlanma süreleri ve makine sıraları verileri kullanılarak işin toplam tamamlanma süresi minimize edilmeye çalışılmıştır. Optimal tamamlanma süresi ile üretimde en etkin çizelgelemeyi planlamak hedeflenmiştir. Çalışmada geliştirilen YSA yönteminin etkinliğini ölçebilmek amacıyla, yöntem Taillard'ın atölye tipi çizelgeleme verilerine uygulanmıştır. Elde edilen sonuçlar, literatürde bu verileri kullanan çalışmalar ile karşılaştırılmış ve yöntemin etkinliği sorgulanmıştır.

1.3.Çalışmanın Önemi

Üretim çizelgeleme, üretim koşulundaki kaynakların etkin biçimde kullanılarak müşterilerin isteklerini karşılayacak biçimde üretim faaliyetlerinin organize edildiği bir planlamadır. Planlama, bir ya da daha fazla performans ölçütünü göz önünde bulundurarak sisteme gelen işleri tezgâhlara atamak ve bu işlerle ilgili tezgâhlardaki üretim sırasını belirlemek gibi süreçlerden oluşur. Üretim çizelgeleme ise, üretim planının karşılanabilmesi ya da siparişlerin teslimat tarihlerine yetişebilmesi için, işlerin hangi kaynaklar (makine, işgücü, malzeme vb.) tarafından hangi sırayla yapılması gerektiği

belirlenmesidir. Ayrıca çizelgelemede her kaynak için iş sırası ile tahmini başlangıç ve bitiş tarihleri planlanır. Üretim çizelgelemenin endüstriye faydası; şirketlerin fiziki koşullarını fazlalaştırmadan daha çok miktarda üretim yapabilmeleri ve bu yolla verimliliklerini arttırabilmeleri için kullanılan bir yol olmasıdır. Üretim çizelgeleme sayesinde imalat ve hizmet sektörlerinde önemli oranlarda verimlilik artışı sağlanabilmektedir. Dolayısıyla çizelgeleme üretim şirketleri için önemli bir unsur haline gelmiştir.

1.4.Çalışmanın Yöntemi

ATÇP, hizmet ve üretim ortamlarında sık kullanım alanları bulunan atölye tipi üretim sistemlerinde her makinede bir iş işlem görecektir şekilde, işlerin belirli bir rota (makine sırası) ile zamansal olarak sıralanması problemi. İş ve makine sayısı arttıkça problem karmaşıklaşır. Bu nedenle literatürde ATÇP NP zor problem sınıfında gösterilmektedir. Bu tip problemlerde optimum veya optimuma yakın çözümlere ulaşabilmek için genellikle sezgisel ve metasezgisel yöntemler kullanılmaktadır.

Bu çalışmada, Taillard'ın çizelgeleme problemlerine ortak bir karşılaştırma tabanı oluşturmak amacı ile yayımlanmış olduğu veri setinin atölye tipi çizelgeleme kısmına YSA algoritması çözüm yaklaşımı olarak geliştirilmiştir. YSA, yapay zekâ biliminin makine öğrenme yöntemlerinden biri olan ve insan beyin hücresinden esinlenerek geliştirilen bir yöntemidir. YSA algoritması, İntel i7 6850 broadwell LGA 2011v3 işlemcili, MSI TOMAHAWK X99A MS7A54 anakartına sahip, Asus rog strix GTX 1070 Gaming OC 8gb gddr5 256bit ekran kartlı, 16 gb 3300 GHz dual DDR4 ram'e sahip bir bilgisayarda Visual Studio 2019 programında C# dilinde kodlanmıştır.

1.5.Çalışmanın Planı

Tez çalışması altı bölümden oluşmaktadır. Birinci bölüm giriş bölümüdür. Giriş bölümünde problemin tanımı, amacı, önemi, yöntemi ve çalışma planı hakkında bilgiler verilmiştir.

İkinci bölümde üretim sistemleri, yönetimi tanımlanmış ve tarihçesi açıklanmıştır. Sonrasında üretim çizelgeleme tanımlanarak, üretim çizelgeleme problemleri sınıflandırılarak açıklanmıştır. Daha sonra NP problemler ve kombinatoriyal optimizasyon problemlerinden bahsedilmiş tez çalışmamızda kullandığımız çizelgeleme problemi çeşidi olan atölye tipi üretim çizelgeleme problemleri çeşitleri ile açıklanmıştır. Son olarak,

retim izelgelemede optimizasyondan bahsedilerek retim izelgeleme problemlerine zm yaklařımı eřitleri alt bařlıklar halinde aıklanmıřtır.

nc blmde atlye tipi izelgelemede son yıllarda kullanılan metasezgisel yntemler bařlıklar halinde aıklanmıřtır. Bu yntemleri kullanan akademik alıřmalar incelenmiřtir.

Drdnc blmde bu alıřmanın yntemi olan yapay sinir aęlarının aıklanması iin yapay zekâ kullanım alanları ve tarihesi, makine ęrenmesi, YSA tarihesi ve yapısı aıklanmıřtır. Ayrıca literatrde atlye tipi izelgeleme iin kullanılan YSA yaklařımları ve alıřmalarına yer verilmiřtir.

Beřinci blmde Taillard karřılařtırma problemleri veri tabanının atlye tipi izelgeleme verilerini kullanan akademik alıřmalar incelenmiřtir. Ayrıca bu tez alıřamasında kullanılan YSA yntemi algoritması adım adım aıklanmıř ve matematiksel modeli sunulmuřtur.

Altıncı blmde sonu blm yer almaktadır. Sonu kısmında alıřmada elde edilen bulgular aıklanmıř, nerilen YSA yntemi ile elde edilen sonuların literatr sonuları ile karřılařtırılması tablo halinde zetlenmiř ve iyileřme oranları gsterilmiřtir.

BÖLÜM II

ÜRETİM ÇİZELGELEME

2.1. Üretim Sistemleri, Yönetimi ve Tarihçesi

Üretim sistemleri ilk kez 1943 yılında genel bir hesaplama mekanizması olarak önerilmiştir. Sistem literatürde çok fazla gelişme göstererek çeşitli problem topluluğuna uygulanmıştır (Randall & Jonathan, 1975). Üretim sistemlerine dair literatürde birçok tanımlama mevcuttur. Klahr'ın yorumuna göre; bir üretim sistemi, bir bilgi işlem sisteminin belirlenmesi için bir şema olarak düşünülebilir. Her üretim bir durumdan ve bir eylemden oluşan bir dizi yapımdan oluşmaktadır. Ayrıca bu sistem, veri yapılarının bir koleksiyonuna da sahiptir. Üretim sisteminin çalıştığı bilgiyi işleyen, eylemlerin işlediği ve koşulların doğru ya da yanlış olarak belirlenebileceği bilgileri ifade eden veri yapılarıdır (Klahr, 1972). Gever ve diğerleri, bir çalışmada üretim sistemini, mahsuller ya da hizmetler yaratmak için fiziksel unsurları birleştirerek entegre bir biçimde çalışması olarak tanımlanmıştır (Gever ve diğerleri, 2014). Taguchi ise üretim sistemini; “Şirketlerin belli gayelerine erişebilmeleri amacıyla sistemden ya da dış unsurlardan edinilen girdilerin en cazip bileşeni seçilip kullanılarak fiziksel bir çıktıya dönüştürülmesi sürecidir.” şeklinde açıklamıştır (Taguchi ve diğerleri, 1989).

En geniş tanımı ile üretim sistemleri, insan, makine, malzeme, para, miktar, maliyet ve diğer sistem bileşenlerini en işlevsel formda birleştirerek, içinde bulunulan koşul ve kaynaklarla sistemin faydalı ömrü içerisinde, girdi olarak adlandırılan belli üretim etmenlerinin, çıktı olarak adlandırılan tüketim etmenlerine dönüştürülerek mamul ve hizmetlerin oluşturulduğu bir oluşum olarak açıklanmıştır (Gever ve diğerleri, 2014).

Üretim yönetiminde ilk olarak Adam Smith, fabrika sisteminin yaygınlaşmaya başladığı yıllarda üretim ekonomisinin önemine dikkat çekmiştir. 1776 yılında Ülkelerin Zenginliği (The Wealth of Nations) adlı kitabında iş bölümünün önemini vurgulayarak ekonomik açıdan avantaj sağlanabileceğini belirtmiştir. Çıktı/girdi oranına göre hesaplanan verimlilikte iş bölümünün çıktı miktarını arttırıcı yönden etkili olabileceğini açıklamıştır (Smith, 1776).

Adam Taylor ise üretimde teknik problemlerin yanı sıra zaman, işçi seçimi, makineler, işlem sıraları gibi faktörlerin varlığını ortaya koymuştur. Taylor aynı zamanda

bilimsel yönetim ve imalat yönetimi gibi felsefelerin gelişmesine katkı sağlamıştır (Çelikçapa & Şenol, 2015).

Taylor dan sonra Henry Fayol, imalat sorunlarında liderliğin önemi üzerinde durmuştur. Yönetici seçimi konularında çalışmalar yapmıştır. Henry'den sonra ise Taylor'un çizdiği yolu izleyen hareket etüdü olarak bilinen konuyu geliştiren çalışmaları ile Frank ve Lillian Gilberth ikilisi gelmektedir (Çelikçapa & Şenol, 2015).

1930'lu yıllarda matematiksel istatistik yöntemi kullanılmaya başlanmıştır. F.W. Haris, Walter Shewart ve L.H.C. Tippet matematiksel istatistik konularına katkı sağlamışlardır. Haris stok kontrolünde matematik modellerini kullanan ilk kişidir. 1931 yılında endüstride istatistiksel kalite kontrolünü geliştiren kişi Shewart, 1934 yılında gecikmeler, iş ve zaman bakımından örnekleme yöntemi ile standartların geliştirilmesi olarak bilinen örnekleme yöntemini geliştiren kişi ise Tippet olarak bilinmektedir (Çelikçapa & Şenol, 2015).

II. Dünya Savaşı sırasında bir İngiliz generalinin Alman uçaklarını durdurmak için çok yönlü hususlarda uzman bilim adamlarından oluşan bir grup tarafından ortaya çıkartılan ve tarihimizde bir optimizasyon problemi olarak kullanılan yöneylem araştırması geliştirilmiştir. Ayrıca o yıllarda ve sonrasında doğrusal programlama, matematiksel programların genel tasarrufu, bilgisayar, simülasyon yöntemlerinin üretim yöntemlerinde kullanılması yaygınlaşmıştır (Çelikçapa & Şenol, 2015).

Adam Smith ile üretim yönetimi kavramı ilk kez ortaya çıkmış ve bu alandaki gelişmeler hızla birbirini izlemiştir. 20.yy sonlarına doğru üretimde bilgisayarın kullanılması daha da yaygınlaşarak gelişmiştir. Bilgisayar kullanımı ile üretim sürecinde çok karmaşık sistem ve yöntemler uygulanabilir olmuştur (Çelikçapa & Şenol, 2015).

2.2.Üretim Çizelgeleme

Çizelgeleme; seçilen birtakım işleri yapmak için hangi kaynakların, ne zaman ve nasıl kullanılacaklarının tespit edilmesidir. Çizelgelemede üretim, kaynak ve zaman olmak üzere üç unsur kullanılmaktadır. Çizelgeleme planında öncelikle yapılması gereken işler bir grup kaynağa atanmaktadır. Örneğin yapılması gereken n adet iş, işlerin yapılması için gereken m adet kaynak (makine, işlemci, vb.) belirlenmektedir. Seçilen tüm işler tamamlanmak üzere mevcut kaynakların bir kısmından ya da tamamından belli bir sıra ile geçerek işlenmektedir (Akkaya & Gökçen, 2006).

Bir işin bir kaynak üzerinde işlenmesinde operasyon denilmektedir. i işinin j kaynağı üzerinde işlenmesine O_{ij} ifadesi verildiğinde çizelgeleme, her bir O_{ij} için bir giriş ve bitiş zamanı keşfetmek işi olarak tanımlanabilir. Üretim Çizelgeleme ise sorunun çıkarılan sınırlarına ilişkili olarak belirli bir amaç fonksiyonunu optimum yapan kombinasyonun bulunması ile alakalıdır. Üretim sistemlerinin çizelgenmesinde amaç, üretim sürecinde stokta bekleme süresini ya da mevcut işlerin toplam tamamlanma süresini (makespan) kısıtları da (tesis maksimum kapasitesi, elverişli makine sayısı vb.) göz önünde bulundurarak minimize etmektir (Akkaya & Gökçen, 2006).

2.2.1. Gantt Şemaları

Gantt şemaları, 20.yy başlarında Henry Gantt tarafından geliştirilmiş olup çizelgeleme sonucu ortaya çıkan işlerin sırasını göstermek için kullanılmaktadır. Aynı zamanda çubuk diyagramı olarak adlandırılan Gantt şemaları yapılması planlanan işlerin zaman içindeki durumunu toplu biçimde göstermektedir. Gantt şemaları yatay eksenle zamanın, dikey eksenle ise çalışmaların zaman içinde ilerleyişi takip etme imkânı sağlamaktadır. Bir projenin alt faaliyetleri, belirli aralıklar ile içereceği zaman dilimine uygun büyüklükte yatay çubuklar olarak çizilerek belirtilmektedir. Tamamlanan faaliyetler ise çizgi şeklinde taranarak gösterilmektedir (Çelikçapa & Şenol, 2015).

Bir projedeki işlerin takibini yapmak amacıyla yaygın olarak kullanılan Gantt çizelgesi ile kritik yolun değerlendirmesi yapılabilir. Üretim çizelgelemede Gantt şemaları kullanarak işler ve makineler kontrol edilebilir. Uygun işe uygun makine rahatlıkla atanabilir. Gantt şemalarında işler zamanın bir işlevi olarak gösterilirken iş başlıkları, süreleri ve bağımlılıkları belirlenir. Bu sayede iş için öngörülen süre hesaplanabilir ve işin durumu takip edilebilir

2.3. Üretim Çizelgeleme Problemleri

Üretim yönetiminde çizelgeleme, planlama süresini azaltan ve tedarik zincirinde değişiklik olması durumunda müdahaleyi kolaylaştıran bir çözüm yöntemidir. Üretim çizelgeleme problemlerinde, işin ne zaman gerekli olduğu, faaliyetlerin zamanlaması, üretime başlama ve tamamlanma zamanları, faaliyetin bölümlerinde kimlerin hangi işi yapacağı ve yeni işler için ne zaman hazır olunacağı belirlenmesi sorunsallarının çözülmesi gerekmektedir. Problemin çözümünde amaç, müşterilerin taleplerine karşılık vermek, tedarik süresini kısaltmak, kapasiteyi etkin kullanabilmektedir. Basit

çizelgeleme problemlerinde programın performansı, akış süresi ve gecikmeye bağlı olarak değişmektedir. Akış süresi, bir işin hazırlık süreleri, üretimler arası geçiş süreleri, makine bozulmalarına harcanan süreler, kaynak tükenmesi ile kaybedilen zaman gibi durumlarda harcanılan zamandır. Gecikme ise işin termin süresi ile termininden sonra teslimi arasındaki süredir (Küçükkoç, 2018).

Üretim çizelgeleme problemlerinde teslim tarihi, tamamlanma süresi ve makine kullanım sırası olmak üzere üç unsur planlanmaktadır. Üretim çizelgelemede teslim tarihi, müşteri siparişlerinin gecikmemesi açısından önemli olduğundan uygun makine, insan gücü ve donanımların en etkin biçimde kullanılarak üretim tamamlanma süresini en aza indirmek amaçlanmaktadır (Küçükkoç, 2018).

Üretim çizelgeleme problemleri kısıtlı kaynakların çeşitli görevlere belirli zamanlarda atandığı kombinatoriyal optimizasyon problemleri arasında yer almaktadır. Dolayısıyla çizelgeleme problemlerinde optimal çözüm bulmak zordur. Bu tür problemlerin çözümlerinde sezgisel ve metasezgisel optimizasyon tekniklerinden yararlanılmaktadır. Çizelgelemede kullanılan bu optimizasyon teknikleri sayesinde şirketler müşteri memnuniyetini ve kaynaklarının verimliliğini artırabilir, bunların yanı sıra maliyetleri de minimize eden planlamalar oluşturabilirler.

2.3.1. Atölyeye Geliş Şekline Göre Sınıflandırma

Atölyeye geliş şekline göre çizelgeleme problemleri statik ve dinamik olarak iki sınıfta incelenebilir. Statik problemler zamandan bağımsız iken dinamik problemler zamana bağlıdır. Günlük yaşamdan bir örnek ile açıklamak gerekirse bir öğrencinin okula gidebilmesi için mevcut olan yolların fazlalığı göz önüne alınırsa en kısa yolun hangi yol olduğu sorgulanması sorunsalında probleme mesafe bağlamından bakıldığında statiktir, zaman açısından bakılacak olursa problem dinamiktir (Yurttakal, 2014).

2.3.1.1. Statik Problemler

Planlama yapılacak dönemin başında yani $t=0$ zamanlama anında, mevcut işlerin tümünün imalata hazır olduğu problem şeklidir. Mevcut işler periyodik olarak işleme konuluyorsa problem statiktir. Statik problemlerde işlem periyodunun sonuna kadar sisteme yeni bir işin girmesi söz konusu olamaz. Çizelge oluşturulduktan sonra gelen işler, bir sonraki periyotta planlanması için bekletilir ve işlemekte olan sisteme dahil edilmez. Statik problemler deterministik problemler olarak da nitelendirilebilir. Çünkü

planlama periyodunun başında gerekli olan tüm veriler kesin olarak belirlidir (Öztemel, 2017).

2.3.1.2. Dinamik Problemler

Dinamik problemlerde sürekli olarak sisteme giren ve çıkan işler vardır. İşlerin belirli olmayan zamanlarda gelmesinden ve işlem sürelerinin değişkenliğinden kaynaklanan stokastik bir yapı vardır. Atölye ortamına herhangi bir anda ya da özel bir durumda yeni iş girdileri olabilir bir başka deyişle çizelgeleme zamanı boyunca üretilecek olan işlerin miktarına sonradan ilave yapılabilecek şekilde problem tanımlanabilir (Yurttakal, 2014).

2.3.2. İş ve Makine Sayısına Göre Sınıflandırma

Genellikle atölye tipi üretim ortamlarında birden fazla işin makine sayısına göre sınıflandırması yapılmaktadır. Bu tür çizelgeleme problemlerini tek makineli ve çok makineli problemler olarak iki başlıkta inceleyebiliriz.

2.3.2.1 Tek Makine-Çok İş Problemi

Aynı makinede yapılacak olan bir grup işin çizelgelemesini yapmak için işlerin sıraya konması gerekmektedir başka bir deyiş ile operasyonların hangi sıra ile mevcut makinede yapılacağına karar verilmesi gerekmektedir. Bu tür problemler saf bir sıralama ile kolaylıkla çözümlenebilir (Öztemel, 2017).

2.3.2.2 Çok Makine-Çok İş Problemi

Birçok operasyonu olan birden fazla işin ortak kaynakları (makinelere) kullanarak yapılması durumunda çizelgeleme işlevi işlerin sıralanması ve operasyonların makinelere atanması sorunsallarına çözüm bulmayı amaçlar. Bu tür problemlerde işlerin hangi makinede hangi sıra ile rotalanacağına karar verilmesi gerekmektedir. Birden çok iş ve makinenin bulunduğu bir atölyede işlerin rotasına göre problemler akış tipi ve sipariş tipi olarak sınıflandırılabilir. Akış tipinde işlerin rotası yani operasyonların makinelerden geçiş sırası aynıdır. Sipariş tipi atölyede ise her iş farklı bir makine rotası izlemektedir (Öztemel, 2017).

2.4. P ve NP problemler

Problemlerin karmaşıklığının temelde iki başlık altında sınıflandığı görülmektedir. Bunlar P sınıfı ve NP sınıfı problemlerdir. P Sınıfı problemlerin işleme zamanı, girdi verilerinin boyutuna bir polinom türünden bağlıdır. Bunlara polinomsal zamanda çalışan algoritma denilmektedir. Bu tür problemlerin, kompleks oluşu polinomsal olması ile ilişkilidir. Kısaca “P” şeklinde gösterilmektedir. NP sınıfı problemlerde sorgulanan sorun ve girilen verinin boyutuna polinom derecesinde bağımlı sürelerde işlenebilecek çözüm algoritmaları bulunmamaktadır. Bu tür problemlerin cevabı tahmin edilebiliyorsa tahminin doğruluğunu sınamak için veri büyüklüğüne polinom mertebesinde bağımlı sürelerde çalışabilecek algoritmalar kullanılmaktadır. Bu tür problemlere NP (non-deterministic polynomial time) sınıfı problemler adı verilir ve “NP” şeklinde gösterilmektedir. Eğer NP sınıfındaki her problem polinomiyal zamanda, bir P problemine dönüştürülebiliyorsa, ilgili P problemine NP-tam problem denir. Eğer P probleminin kendisi de NP sınıfında bir problemse bu durumda P problemine NP-zor problem denir. Kombinatoriyel optimizasyon problemlerinin çoğu NP-zor, polinomiyal zaman sınırı olmayan problemler safına girmektedir. NP problemlerine, polinomiyal algoritma geliştirilememiştir. NP safında bulunan problemler için asıl optimum çözüm yerine yakın çözümler bulunabilmektedir (Reeves, 1995).

2.4.1. Kombinatoriyel Optimizasyon Problemleri

Kombinatoriyel optimizasyon problemlerinde karar verilmesi gereken alternatif yollar ne olursa olsun ya da tanımlanan ve çözüm aranan problem ne olursa olsun genel amaç, sonucu optimize etmektedir. Bu nedenle bu tür problemlerde optimum sonucun elde edilmesi işlemine optimizasyon denilmektedir.

Bir bilgisayar terimi olarak optimizasyon, sistem içerisinde mevcut olan kaynakların (işgücü, zaman, sermaye, oluşumlar, hammaddeler, kapasite, donanım gibi) olabildiğince etkin biçimde tüketilerek belirli amaçlara (maliyet minimize edilmesi, kazancın maksimize edilmesi, kapasite tasarrufunun ve verimliliğin maksimizasyonu gibi) erişmeye yardımcı bir teknik bilgi olarak tanımlanmıştır (Gass, 2000).

Kombinatoriyel optimizasyon problemlerinde projelerdeki faaliyet sayılarının fazlalığı, işgücü ve hammadde gibi unsurlar, faaliyetler arasındaki öncüllük sırası gibi etkenler sebebiyle oldukça karmaşık bir yapıya sahiptir ve büyük bir çoğunluğu çözümü zor problemler olarak nitelendirilmektedir.

Bilgisayar bilimleri, endüstri mühendisliği, yöneylem araştırması ve üretim yönetimi gibi birçok alanda karşımıza çıkan kombinatoriyal optimizasyon problemlerinde optimum sonuç bulunması çok fazla hesaplama zamanı gerektirdiği için kesin çözüm algoritmaları ile problemlerin çözümü pratikte mümkün olmamaktadır. Bu sebeple sezgisel ve metasezgisel yöntemler kullanılarak kısa sürede optimum veya optimuma yakın olduğuna kanaat edinilen sonuçlar elde edilmektedir.

2.5. Atölye Tipi Çizelgeleme Problemleri

Atölye tipi çizelgeleme problemleri (ATÇP) bir NP-zor problem türü olduğundan objektif bir program oluşturmak için mümkün olduğu kadar kısıtlamaları sağlayarak genel toplam zamanı minimize etmek gerekmektedir. ATÇP hem endüstri sektörü hem de hizmet sektörü için önem teşkil etmektedir. İyi bir çizelgeleme yapan şirketler üretim açısından hem parasal hem de zaman olarak kazanç sağlanabilir.

Atölye tipi üretim sisteminde, birçok makine ve iş vardır. Önceden belirlenmiş iş akışını takip eden makinelerde ürün işlenir. İşlerin işlem süreleri bilmekte ve düzenlenebilmektedir. Makine kurulum saatleri işlem sürelerine dahil edilir. Bir makine de belli bir zamanda bir iş yapılabilir. İşin hangi makinede işleneceği konusunda bir öncelik sıralaması mevcuttur (Chiang & Fu, 2006).

Atölye tipi üretim sistemleri, sabit atölye tipi, açık atölye tipi ve esnek atölye tipi şeklinde üç başlıkta toplanabilir.

2.5.1. Sabit Atölye Tipi

Sabit Atölye tipi üretim sistemleri ürüne göre ya da işe göre yerleştirilmiş makinelerden oluşurlar. Bu biçimdeki sistemler çoğunlukla çok çeşitlilikte az miktarda ürün üretmek için kullanılırlar. Sisteme sipariş üzerine girilen iş verileri sistemin uygun bölümünde uygun makinelerde üretim amacına göre dönüşüm işlemine uğrarlar. Üretilen her ürünün, yapılacak olan her işin kendine özgü bir makine rotası vardır. Bu rota sistemde belirlenen zamanda hangi makinenin boş, hangisinin dolu olduğuna göre değişiklik gösterebilmektedir.

Sabit atölye tipi üretim sistemlerinde;

- Sonlu bir n iş kümesi vardır.
- Sonlu m makineleri kümesi vardır.

- Her bir iş bir operasyon zincirinden oluşur.
- Her makine bir seferde en fazla bir işlem yapabilir.
- Her işlem belirli bir makinede belirli bir uzunlukta kesintisiz bir süre boyunca işlenmelidir.
- Her işin makine sırası, belli bir rotası vardır.
- Amaç, operasyonların zaman aralıklarını dikkate alarak makinelerde işlenen işlerin minimum toplam zamana sahip olduğu bir çizelge oluşturmaktır.

2.5.2. Esnek Atölye Tipi

Esnek atölye tipi çizelgeleme (EATÇ) probleminde, iş yollarını belirlemek amacıyla her işlem için, mevcut olanlar arasından hangi makinenin belirlenen operasyonu işlenmesi gerektiğine karar vermek gerekmektedir. EATÇ probleminde işlemlerin, farklı yollarda belirlenen bir setten herhangi bir makinede işlenerek yapılmasına izin verilmektedir.

EATÇ de operasyonların sistemde mevcut makinelerden herhangi birisine atanması ve bu makine takımında işlerin dizilmesi alt problemlerinden oluşmaktadır. EATÇ problemi bir tesiste mevcut birtakım makineler arasında işlem yapılmasına izin verilen bir klasik ATÇP'nin uzantısı olarak düşünülebilir. EATÇ, bir tesisteki makinelere ek bir operasyon atanması nedeniyle ATÇP'den daha zor olduğu bilinmektedir (Singh & Mahapatra, 2016).

2.5.3. Açık Atölye Tipi

Açık Atölye Tipi Çizelgeleme (AATÇ) problemi, her bir işin mevcut makinelerde işlem göreceği olan yalnızca bir işleminin bulunduğu ve işlem sırasının önemsiz olduğu bir ATÇP çeşitidir.

Genellikle benzer ürün gruplarının üretildiği tesislerde görülmektedir. AATÇ, m makinelik bir sette işlenmesi gereken n adet işin bulunduğu, her bir işin farklı bir makinede gerçekleştirilmesi gereken, i adet işlem den oluştuğu bir çizelgeleme problemi olarak açıklanabilir. AATÇ, işlemlerin herhangi bir sıra ile yani takip edilmesi gereken bir işlem sırası olmadan gerçekleştirildiği atölye çizelgelemesidir (Baysal ve diğerleri, 2012).

2.6. Üretim Çizelgelemede Optimizasyon

Optimizasyon, işletmelerde tasarım, çizelgeleme ve maliyetlendirme faaliyetlerinde önemli bir rol oynamaktadır. Kaynakların verimli kullanılmasını, zaman tasarrufunu ve ürün niteliğinin artırılmasını sağlamaktadır. Özellikle devasa sistemlerde, dinamiklerin insan çabası ile optimize edilmesi meşakkatli kesildiğinden, bu tür işlemler algoritmalar oluşturularak bilgisayar programları yardımı ile yapılmaktadır.

2.7. Üretim Çizelgeleme Problemlerine Çözüm Yaklaşımları

Üretim sistemlerinin çizelgeleme problemlerinde kullanılan çözüm yaklaşımları en iyileştirme teknikleri (EİT) ve yaklaşık çözüm teknikleri (YÇT) olarak ikiye ayrılırlar. EİT, en iyi çözümü garanti etmelerine rağmen büyük boyutlu problemlerde çözüme ulaşması çok uzun zaman alan tekniklerdir. YÇT ise daha kısa sürede çözüm sunabilmelerine karşın en iyi çözümü garanti edemezler. YÇT'nin EİT'ye göre çok daha kısa sürede yeterli iyilikte çözümler sunabilmeleri nedeniyle daha fazla tercih edilmektedir (Şevkli, 2005).

EİT, matemaiksel yöntemler ve dal sınır algoritması olarak tablo 1 de gösterildiği gibi iki başlıkta toplanabilir.

Matematikselsel Yöntemler: Küçük problemlere uygulanabilen denklemler ve fonksiyonlardır. Yavaş fakat iyi sonuçlar veren bu yöntemler problem büyüdükçe çözüm süresi üstsel olarak arttığından hafıza kapasite sorunu ile karşılaşmaktadır. Hafıza kapasite sorunu nedeniyle çözüm süreleri uzamakta ya da çözüm mümkün olmamaktadır. Matematikselsel yöntemlerin en sık kullanılanları tam sayılı programlama, karışık tam sayılı programlama ve Lagrangian yöntemidir (Şevkli, 2005).

Dal sınır algoritması: Genel olarak algoritma bir ağacın dallanmasından esinlenilerek oluşturulur. İlk olarak ağaç sadece tek bir düğümden yani kökten ibarettir. Bu düğüm tezgâh işlem sırası belirlemeden kabul edilen çözümleri temsil etmektedir. Kökün ardılları tezgâhların işlem sırasının belirlenmesinde kullanılır. Örneğin aynı tezgâh işlemlerini gösteren $[i,j]$ yayı seçilir ve bu iki düğümden dallanma oluşturulur. Sonrasında üst ve alt sınırlar belirlenir. Üst sınır en iyi çizelgelemeyi temsil ederken, alt sınır en yüksek optimal değeri temsil etmektedir. Çözüm yolunda kritik tezgâh olarak belirlenen en büyük alt sınırlı tezgâhın işlemi ilk sırada olacak şekilde işlem sıraları oluşturulur (Şevkli, 2005).

Tablo 1.
En İyileme Teknikleri

EN İYİLEME TEKNİKLERİ	
1. Matematiksel Yöntemler	2. Dal Sınır Algoritması
1.1. Tam Sayılı Programlama (Integer Programming)	
1.2. Karışık Tamsayı Programlama (Mixed Integer Programming)	
1.3. Lagrangian Yöntemi (Lagrangian Relaxation)	

Kaynak: (Şevkli, 2005)

Tablo 2
Yaklaşık Çözüm Teknikleri

YAKLAŞIK ÇÖZÜM TEKNİKLERİ
1. Değişen Darboğaz Yöntemi (Shifting Bottleneck)
2. Öncelik Dağıtım Kuralları (Priority Dispatch Rule)
3. Sezgisel ve Metasezgisel Yaklaşımlar (Heuristic and Metaheuristic Approaches)
4. Yapay Zekâ (Artificial Intelligence)

Kaynak: (Şevkli, 2005)

YÇT, tablo 2 de belirtilen şekilde 4 başlıkta toplanabilir

Değişen Darboğaz Yöntemi (Shifting Bottleneck): ATÇP’inde çizelgeleme alt problemlere bölünerek ayrıştırılır. Sırlanmamış tezgâhlar (makineler) arasında darboğaz, en büyük çözüm değerine sahip tezgâh olarak belirlenir. Darboğaz tezgâhı ele alınarak alt problemler optimal olarak sıralanır. Önceden belirlenen sıralar korunurken kritik tezgâh sıraları çözülen tezgâh problemlerine göre yerel olarak optimize edilir (Adams ve diğerleri,1988).

Öncelik Dağıtım Kuralları (Priority Dispatch Rule): ATÇP’inde bekleyen işlerin belirlenen yöntemlere göre sıraya atanmasıdır. Öncelik kuralları teslim tarihi, bağımlı işlem sayıları, hazırlık süresi, maliyetler vb. gibi bilgilerin biri ya da daha fazlası dikkate alınarak belirlenir. Öncelik belirleme sırasında amaca yönelik koşullar dikkate alınır (Pinedo, 1991).

Sezgisel ve Metasezgisel Yaklaşımlar (Heuristic and Metaheuristic Approaches): Bu tür yöntemler kısa sürede optimuma yakın sonuçlar elde edebilirler. Bu

nedenle kullanımları büyük boyutlu problemlerde çok daha yaygındır. Hesaplama gücünün iyi olması, dönüştürülebilir yönünün olması, çözüm zamanının matematiksel yöntemlere göre çok daha kısa olması, birçok çeşit problem için kolay bir şekilde uyum sağlaması avantajları olarak gösterilebilir. Dezavantajları ise bu tür yöntemler ile elde edilen sonuçların optimum olduğunun garantisinin olmaması ve optimale yakın çözüm elde edebilmek için birden fazla parametrenin doğru bir biçimde ayarlanması gerekliliği olarak gösterilebilir.

En sık kullanılan metasezgisel yöntemler tablo 3 de sunulmuştur. Bölüm 3 de açıklanmıştır.

Tablo 3
Metasezgisel Yaklaşımlar

METASEZGİSEL YAKLAŞIMLAR
1. Açgözlü Yaklaşımı (Greedy Approach)
2. Ateşböceği Algoritması (Firefly Algorithm)
3. Tavlama Benzetimi (Simulated Annealing)
4. Değişken Komşu Arama (Variable Neighborhood Search)
5. Genetik Algoritma (Genetic Algorithm)
6. Hibrit Algoritmalar (Hybrid Algorithms)
7. Karınca Kolonileri Optimizasyonu (Ant System)
8. Örneklem & Kümeleme (Sampling & Clustering)
9. Paralel Kanguru Algoritması (Parallel Kangaroo Algorithm)
10. Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
11. Tabu / Yasaklı Arama (Tabu Search)
12. Yapay Bağışıklık Sistemi (Artificial Immune System)

Yapay Zekâ (Artificial Intelligence): ATÇP çözümünde yapay zekâ yaklaşımları YSA ve uzman sistemler olarak ikiye ayrılmaktadır. YSA kısaca insan beyninin biyolojik yapısından esinlenilerek oluşturulan insan gibi öğrenme ve sonuç geliştirme yeteneğine sahip olan tekniklerdir. Genel olarak ATÇP’inde kullanılan YSA algoritmaları Hopfield ağları ve Hatayı geriye yayma ağlarıdır. Bölüm 4 de literatür çalışmaları ile birlikte yapay zekâ yöntemleri ve yapay sinir ağları açıklanmıştır.

BÖLÜM III

ATÖLYE TİPİ ÇİZELGELEMEDE METASEZGİSEL YÖNTEMLER

3.1. Metasezgisel Yöntemler

3.1.1. Açgözlü Yaklaşımı (Greedy Approach)

Açgözlü Yaklaşımı (AY), genellikle basit yapılu problemlerde çözümü çok hızlı hesaplamak için kullanılır. AY ile problem çözümünde birkaç alt sorun olabilir. Bu alt sorunların çözümünde lokal olarak en uygun seçim yapılır. AY ile çözüm süreci en iyi sonuca ulaşmak amacıyla yerel olarak en iyi sonuçları seçme stratejisine dayanır. Durumun özelliklerine göre en iyi kararı bulmaya çalışan AY, her zaman “hepsini al”, “en çok/yüksek olanı seç” mantıkları ile hareket etme yöntemidir. AY, NP-zor problemlere tutarlı optimal çözümler sağlayamayabilir. Ancak diğer optimizasyon yöntemlerinden daha hızlıdır ve çoğu zaman optimale yakın sonuçlar verebilir. Özellikle şimdiki kararların gelecek kararları etkilemeyeceği durumlarda ve temel problemin optimum çözümünün, problemin diğer aşamalarının optimum çözümlerinden oluşabildiği koşullarda AY tercih edilmektedir (Takilli, 2012).

3.1.2. Ateşböceği Algoritması (Firefly Algorithm)

Ateş böceği Algoritması (AA), 2008 Yılında Xin-She Yang tarafından geliştirilmiştir. AA, Ateş böceklerinin parlaklığa duyarlı sosyal davranışlarını ele alarak geliştirilmiştir (Yang, 2008).

Ateş böceklerinin tek cins olması ve birbirilerini çekmeleri algoritmanın temelini oluşturmaktadır. Ateş böceklerinin çekiciliği, parlaklıkları ile doğru orantılıdır. Dolayısıyla bir ateş böceği ne kadar parlaksa o kadar çekicidir. Daha az parlak olan ateş böcekleri ise çekici olan ateş böceklerine doğru yönelmektedirler. Parlaklık etkisi uzaklık arttıkça azalacağı için daha uzaktaki ateş böcekleri uzaktaki parlak ateş böceklerinden daha az etkilenmektedir. Algoritmada oluşturulan çözüm kümelerinin her birinin her bir parametresine sırayla geliştirilme formülü uygulanmaktadır. Her bir ateş böceği diğer tüm ateş böceklerini kontrol etmekte ve kendinden parlak olan ateş böceklerine doğru hareket etmektedir. Eğer kendinden daha parlak ateşböceği bulamaz ise rastgele hareket gerçekleştirmektedir (Yang & Xingshi, 2013).

AA yönteminin ateş böceği davranışlarından etkilenecek geliştirilmesi süreci tablo 4 de açıklanmıştır.

Tablo 4
Ateş böceği Algoritması

ESİNLENEN DAVRANIŞLAR	ALGORİTMADA KARŞILIK GELEN EYLEM
1. Ateş böceklerinin rastgele konumları belirlenir. Sonra parlaklıkları hesaplanır.	1. Rastgele çözüm kümeleri oluştur. Her çözüm kümesi parametre sayısı kadar eleman içerir. Oluşturulan çözüm kümelerinin uygunluk değerleri bulunur
2. Ateş böcekleri daha parlak ateş böceklerine doğru hareket eder.	2. Tüm çözüm kümelerine sırayla iyileştirilme formülü uygulanır.
3. Belirlenen tekrar sayısına ulaşıncaya kadar 2. Aşamaya geri dönülür.	3. Maksimum iterasyona ulaşıncaya kadar 2. Aşamaya geri dönülür.

Kaynak: (Pardolos ve diğerleri, 2008)

Ateş böceğinin çekiciliği β olarak adlandırılır, β bitişik ateşböceklerinin gördüğü ışık şiddeti ile orantılıdır.

$$\beta = \beta_0 e^{-\varphi r^2} \quad (1)$$

Ateş böceğin hareketi daha parlak ateş böceğine doğru çekilir. Çekiciliğin varyasyonunu r mesafesiyle tanımlayabiliriz. $r = 0$ 'da çekiciliğin 0 olduğu yer.

$$X_i^{t+1} = X_i^t + \beta_0 e^{-\varphi r_{ij}^2} (X_j^t - X_i^t) + a_t \dot{X}_i^t \quad (2)$$

Esas olarak çözüm çeşitliliği kontrol edilememektedir, bu nedenle yineleme sırasında a_t parametresi ayarlanmaktadır.

$$a_t = a_0^{\delta^t}, \quad 0 < \delta < 1 \quad (3)$$

İterasyon sayacı t değiştirilerek yineleme sırasında a_t parametresi ayarlanabilir. İlk rastgele ölçeklendirme faktörü a_0 parametresidir. Çoğu uygulama için \mathcal{S} ifadesi 0.95 ile 0.97 arasında kullanılabilir. Ortalama ölçeği L parametresiyle ifade edilmektedir. Başlangıçta $a_0 = 0.01L$ şeklinde ayarlanabilir. $a_0 = 0.01$ olduğu durum, rastlantısal yürüyüşlerin hedefe ulaşmak için birkaç adım atması ve yerel sömürüyü birkaç adımda çok fazla zıplamadan dengelemesinden kaynaklanmaktadır. β Parametresi çekiciliği kontrol eder ve parametrik çalışmalar, çoğu uygulama için $\beta_0 = 1$ 'in kullanılabilirliğini gösterir. Bununla birlikte, aynı zamanda L parametresi ölçekleme ile ilgili olmalıdır. Genel olarak, $\varphi = 1/\sqrt{L}$ 'yi ayarlayabiliriz. Ölçekleme varyasyonları önemli değilse, $\varphi = 0$ alınır. Çoğu uygulama için, en iyi popülasyon sayısını 15 ile 100 arasında belirlenmektedir (Yang & Xingshi, 2013).

3.1.3. Tavlama Benzetimi (Simulated Annealing)

Tavlama Benzetimi (TB) algoritması, ilk olarak 1983 yılında Kirkpatrick ve diğerleri tarafından kullanılmıştır. TB arama işlemi sırasında alt çözümlere geçişlere izin vererek local minimumlardan kaçma yeteneğine sahiptir (Kirkpatrick ve diğerleri, 1983).

TB araştırmasında, rasgele bir başlangıç noktası belirlenir ve algoritma bu noktadan, kullanıcı tarafından önceden belirlenmiş bir uzaklığa bir adım ilerler. Yeni noktadaki amaç fonksiyon değeri başlangıçtaki amaç fonksiyonu değeri ile karşılaştırılır. Yeni değer küçük olup olmadığı belirlenir. Minimizasyon durumunda, amaç fonksiyon değeri azalır otomatik olarak kabul edilir ve kabul edilen konum, araştırmanın devam ettirileceği nokta olarak belirlenir. Algoritmanın sonraki adımı amaç fonksiyonunun yüksek çıkan değerlerinin, Metropolis kriteri ile belirlenen bir olasılık dahilinde kabul edilmesidir. Algoritma ilerledikçe, son çözüme doğru yaklaşıldıkça, adımların uzunluğu da azalmaktadır. Metropolis kriteri kullanıcı tanımlı başlangıç parametrelerinde (T : sıcaklık ve RT : sıcaklık azaltma faktörü), amaç fonksiyonunun yüksek bir değerini kabul etme olasılığını belirlemede kullanılır. Gerçek tavlama sürecine paralel olarak T azaldıkça yüksek değerleri kabul etme olasılığı da azalır. T_i her NT iterasyonundan sonra fonksiyonun değerlendirilmesinin i 'nci iterasyonu olmak üzere, $T_{i+1} = RT \times T_i$ fonksiyonunu kullanılarak azaltılır. NT , sıcaklık azaltmaları arasındaki iterasyon sayısını gösteren hafızadaki değerdir (Sexton ve diğerleri, 1999).

Teorik olarak tavlama benzetiminin çalışma prensibi, Markov zincirlerinin modellenmesi ile elde edilmektedir. Markov zincirinin algoritma adımları aşağıdaki gibi sıralanabilir;

1. Rassal olarak ya da seçilmiş bir çözüm kümesini, başlangıç çözümü olarak kabul et.
2. Başlangıç çözümüne en iyi çözüm olarak bir değere ata.
3. Başlangıç çözüm değerinin maliyet fonksiyonunu hesapla.
4. Başlangıç sıcaklığı belirle.
5. Başlangıç sıcaklığı bir değere ata.
6. Durma kriterleri sağlanmadı ise aşağıdaki işlemleri gerçekleştir.
 - a. Markov zinciri uzunluğunu belirle.
 - b. Başlangıç sıcaklığını azalt ve altıncı maddeye geri dön.
7. En iyi çözüm çizelgesini oluştur ve dur.

Tavlama benzetimi, fiziksel tavlama süreci ile ayrık optimizasyon problemlerinin benzeşiminden ortaya çıkartılmış bir yöntem olarak kullanılmaktadır (Şevkli, 2005).

3.1.4. Değişken Komşu Arama (Variable Neighborhood Search)

Değişken Komşu Arama (DKM) Yöntemi, Mladenovic ve Hanşen tarafından 1997'de ortaya konulmuştur. Bu metasezgisel yöntem, zamanla optimizasyon problemlerinin çözümünde kullanılmak üzere geliştirilmiştir. DKM yönteminin amacı, arama içerisinde kullanılan komşuluğun sistematik olarak değiştirilmesidir. Dolayısıyla DKM yaklaşımı, komşuluk yapıları tabanlı bir yaklaşım olup yerel iyilerde takılıp kalınmasını önlemek için, komşuluk değişimini kullanmaktadır. DKM yönteminde komşuluk yapısının kullanılmasının nedenleri; Mevcut komşuluk yapılarından yerel olarak en iyi değer, diğer komşuluk değerleri için en iyi olmak zorunda değildir. Genel minimum değer ise bütün komşuluk yapıları için minimum noktadır. Ayrıca birçok problem için, yerel en iyi bir veya birden fazla komşuluk için birbirine kısmen yakındır. Komşuluk değişiminde, eğer x' , optimizasyon kriterine göre x değerinden daha iyi ise, sistemin yeni durumu x' haline getirilir ve geçerli komşuluk olarak ilk komşuluk belirlenir. Aksi halde sistemin geçerli durumu x kalır ve bir sonraki komşuluğa geçilir (Hanşen ve diğerleri, 2010).

Genel olarak bir DKA algoritmasının kodları şekil 1 de gösterilmiştir.

```

procedure DKA
  x ← İlkDurumaGetir
  k ← 1 // Komşuluk İndexi
  while not DurmaKoşulu do
    x' ← Salla(x,k)
    x'' ← YerelArama(x')
    if (x'' < x) then
      x ← x''
      k ← 1
    else
      k ← k + 1
      if (k = kmaks) then
        k ← 1
    end-if
  end-if
end-while
end-procedure

```

Şekil 1.Genel DKA Algoritması

Kaynak: (Roshanaei ve diğerleri, 2009)

3.1.5. Genetik Algoritma (Genetic Algorithm)

Evrimsel hesaplama ilk olarak I.Rechenberg tarafından “Evrin Stratejileri” eserinde tanıtılmıştır (Rechenberg,1973). Genetik Algoritma (GA) yöntemi ilk olarak J Darwin’in evrim teorisinden esinlenerek ve evrimsel hesaplama tekniği kullanılarak bilgisayar yardımıyla John Holland tarafından 1975 yılında oluşturulmuştur (Holland,1975). Holland’ın öğrencisi D.E.Golldberg 1989 yılında bu konuda çıkardığı kitabıyla genetik algoritmaların çeşitli konularda kullanılabileceğini göstermiştir (Goldberg, 1989). 1992 yılında John Koza genetik algoritmayı kullanarak genetik programlamayı geliştirmiştir (Koza, 1992).

GA, kombinatoryal problemlerin optimizasyonuna odaklanan bir yöntemdir. GA yöntemi, geleneksel yöntemlerle çözümü zor ya da neredeyse imkânsız olan ve çözüm uzayının çok büyük olduğu problemlerin çözümünde kullanılmaktadır. Optimizasyon alalındaki problemlerin çözüm uzayının taranması geleneksel metotlarla çok uzun sürebilmektedir. GA ile kısa sürede optimuma yakın ve kabul edilebilir sonuçlar bulunabilmektedir (Bolat ve diğerleri, 2004).

GA yöntemi ile çözüm ararken ilk olarak hesaba alınan problem için rastgele bir miktar kromozomlu popülasyon oluşturulmaktadır. Sonrasında popülasyondaki her bir

kromozoma yönelik uygunluk fonksiyonu oranlanmaktadır. Yeni bir popülasyon elde edilinceye kadar uygunluk durumuna göre popülasyondan iki adet kromozom, çaprazlanmak amacıyla seçilmektedir. Seçilmiş kromozomlar, en iyi bireyleri elde etmek amacıyla çaprazlanmaktadır. Çaprazlama yöntemi ile yeni oluşacak popülasyonun atalarından farklı olması sağlanmaktadır. Kromozom üzerindeki bazı DNA sıra ve düzen yerleri değiştirilerek belirli mutasyon ölçeğine göre başkalaştırma yapılabilir. Yeni bir popülasyon oluşuncaya kadar bu süreç tekrarlanır. Yeni popülasyon kabul edildikten sonra ise, oluşturulan yeni popülasyon eskileriyle yer değiştirilir. Amaçlanan uygunluk seviyesi elde edildiğinde program bitirilir ve popülasyondaki en iyi çözüm kabul edilir (Bolat ve diğerleri, 2004).

Genetik algoritmalar yöntemi özetle; çözülmek istenilen problem için rastgele bir n kromozomlu popülasyon oluşturulur. Sonrasında popülasyonda bulunan kromozomların her biri için $f(x)$ uygunluk fonksiyonu hesaplanır. Yeni bir popülasyon oluşuncaya kadar seleksiyon, çaprazlama ve mutasyon adımları tekrar edilir (Bolat ve diğerleri, 2004).

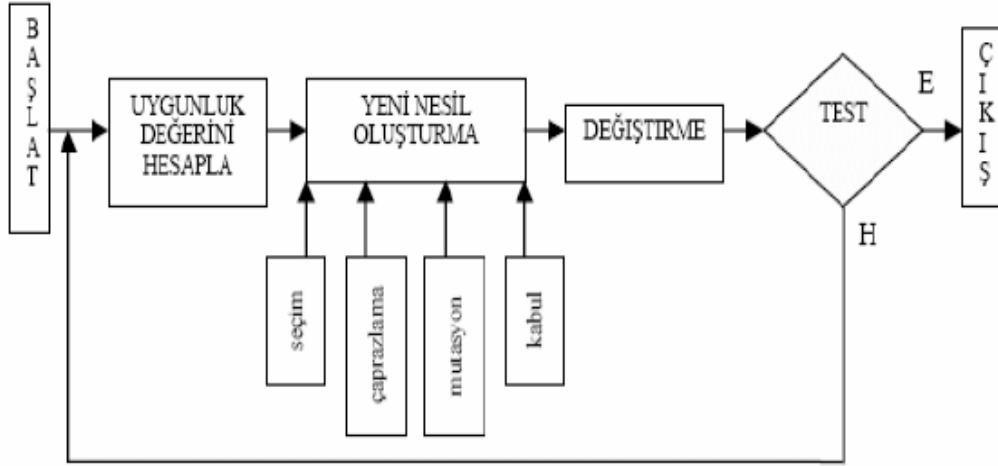
Seleksiyon: Belirlenen problemin çözümü için uygun olacak şekilde popülasyondan iki tane kromozom, çaprazlanmak (crossover) için seçilir (çoğullama – (reproduction)). Seçim yapılırken uygunluk derecesi yüksek olan kromozomun kullanılma ihtimali daha olasıdır.

Çaprazlama: Yeni bireyler oluşturmak amacı ile seçilmiş olan ebeveyn kromozomlar kullanılır. Seçim yapılırken çaprazlama oranı göz önünde tutulur. Çaprazlama uygulanmasının amacı ise yeni bireylerin, ataları ile genetik açıdan benzerliğini azaltmaktır.

Mutasyon: Kromozom üzerindeki bulunan DNA dizilerinin bazılarının buldukları konum belirlenen bir mutasyon oranına göre değiştirilir. Elde edilen yeni popülasyon eski kromozomlar ile yer değiştirilir. Hedeflenen uygunluk değerine ulaşıldığında program durdurulur ve popülasyondaki en iyi çözüm alınarak kayıt edilir.

Genetik algoritmalar, evrimsel hesaplama tekniğinin bir parçasını oluşturmakta ve çoğunlukla geleneksel yöntemlerle çözümü zor ya da neredeyse imkânsız olan problemlerin çözümünde tercih edilmektedir. Mühendislik alanında, daha çok optimizasyon için kullanılmakta ve diğer klasik yöntemlere göre daha iyi sonuç verdiği gözlemlenmektedir (Bolat ve diğerleri, 2004).

Genetik Algoritmanın adımları şekil 2 de açıklanmıştır.



Şekil 2. Genetik Algoritmanın Adımları

Kaynak: (Bolat ve diğerleri, 2004).

3.1.6. Hibrit Algoritmalar (Hybrid Algorithm)

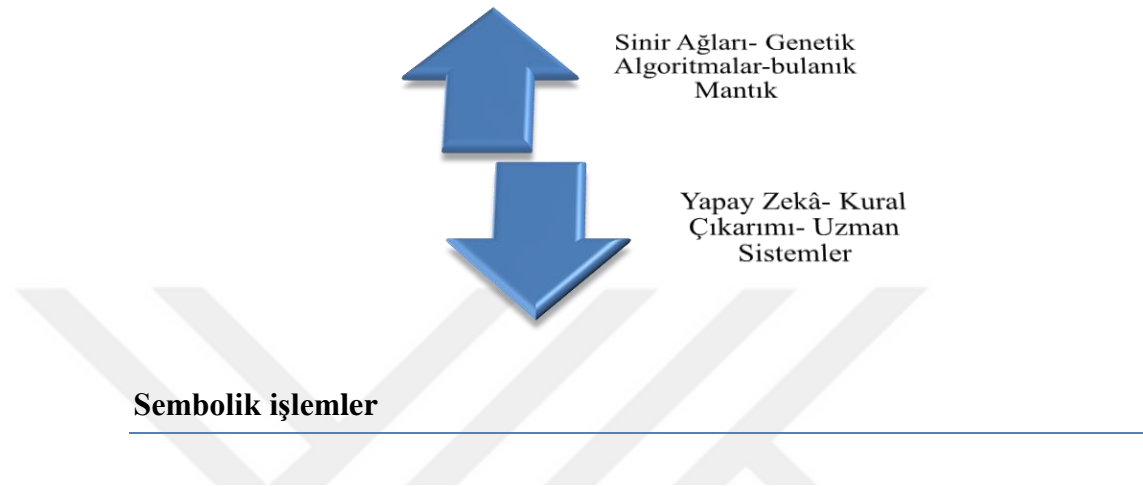
Problem çözümünde kullanılan akıllı teknolojiler; yapay zekâ, uzman sistemler, sinir ağları, bulanık sistemler, kural işleme ve genetik algoritmaların değişik tekniklerini içermektedir. Bu akıllı teknolojiler iki ucu sonsuz bir doğru olarak düşünülebilir. Doğrunun bir ucunda, tahmin, uygunluk işleme yapısına sahip sinir ağları, diğer ucunda ise uzman sistemler gibi bilgi işleme özelliği taşıyan teknolojiler eklenebilir. Genetik algoritmalar, Bulanık sistemler sinir ağlarına yakın konumlandırılırken, Kural işleme sistemi uzman sistemlere yakın düşünülmelidir (De Souza, 1997).

Hibrit akıllı sistemler, en az bir tanesi akıllı teknoloji olan birkaç teknolojinin entegre edilmesi ile oluşturulmuş problem çözüm sistemleridir. Hibrit sistemlerin özellikle tercih edilmesinin nedeni; tek bir yaklaşımın gereksinim duyduğu varsayımların tam olarak karşılanamadığı problemleri, etkin olarak çözmek için, değişik teknolojilerin gücünü birleştiren bir sistemi ortaya koyabilmektir. Teknoloji birleşmesiyle oluşan hibrit sistemler, çizelgeleme sistemini ya da üretim sistemini, öğrenme ve uygulama özelliklerinden dolayı etkin bir şekilde modellemede kullanılabilir. Hibrit bir sistem oluşturmak için en azından bir akıllı teknoloji ile diğer teknolojileri birleştirmek gereklidir ve kullanılacak sistemleri her birinin gücü diğerinin zayıf yönlerini telafi edecek şekilde karıştırılıp birleştirilerek oluşturulmalıdır (Dağlı, 1994).

Hibrit akıllı sistemler, en az bir tanesi akıllı teknoloji olan birkaç teknolojinin bütünleştirilmesi ile oluşturulmuş problem-çözüm sistemleridir.

Akıllı sistemler Şekil 3 de gösterilmiştir.

Altsembolik İşlemler



Şekil 3. Akıllı Sistemler

Bu tür sistemlerin nasıl entegre edilebileceğine dört biçimde örnek verebiliriz;

- 1.Sinir ağları + Uzman sistemler
- 2.Genetik Algoritmalar + Sinir Ağları
- 3.Sinir Ağları + Simülasyon
- 4.Matematiksel Programlama + Sinir Ağları

Bu durumlar aşağıdaki gibi oluşturulabilir;

- 1.Patern-eşleştirme için bir Uzman Sistem içerisine gömülmüş Sinir Ağı
- 2.Sinir Ağı 'nın ağırlıklarını modifiye etmek için genetik algoritmayı kullanma
- 3.Adaptif davranışı modellemek için bir ayrık-olay simülasyonu içerisine gömülmüş sinir ağı
- 4.Sinir Ağı'nın ağırlıklarını modifiye etmek için doğrusal olmayan programlamayı kullanma (De Souza, 1997).

3.1.7.Karınca Kolonileri Optimizasyonu (Ant System)

Karınca Kolonileri Optimizasyonu (KKO) algoritması ilk olarak Dorigo ve arkadaşları tarafından gezgin satıcı ve kuadratik atama problemleri gibi zor

kombinatoriyel optimizasyon problemlerinin çözümü için metasezgisel bir yaklaşım olarak önerilmiştir. Algoritmada kullanılan yapay karıncalar, doğal karıncaların yiyecek arama sürecinde kullandıkları en kısa yolu bulma yöntemi baz alınarak programlanmaktadır. Yapay karıncaların, optimizasyon problemine iyi sonuçları elde edebilmesi için izlerle oluşturdukları yol haritası ve bu süreçteki birbirleri arasındaki iletişim kullanılmaktadır. Karınca sistemi ile arama uzayı içerisinde optimum sonucu arayan yapay karıncalar arasında iletişim sağlanarak, arama esnasında yapılan iyi seçimlerden diğer yapay karıncalarında haberdar olması sağlanmaktadır. Böylece çok sayıda karınca kullanılan bir rastgele aramadan, çok daha iyi sonuçlar elde edilmesi mümkün kılınabilmektedir (Darigo ve diğerleri 1997).

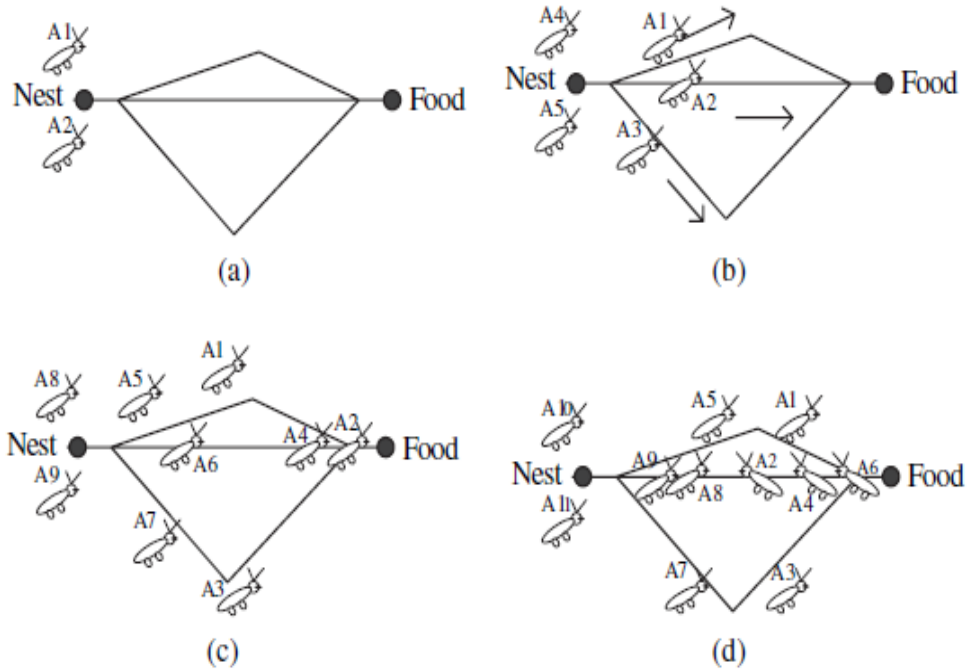
Karınca sisteminin en önemli süreci, izlerin takibi yönetimidir. Standart bir karınca sisteminde izler yeni çözümler oluşturmak için rehber görevi görmektedir. Çözüm yolu üzerinde bulunan izler, çözümün kalitesine bağlı olarak sistem tarafından güncellenir. Sistemdeki tüm karıncaların güncelleme yapabildiği, en iyi karıncanın güncelleme yapabildiği ya da güncellemenin eşik değerleri arasında sınırlandırıldığı güncelleme süreçleri ile farklı tipte arama mekanizmaları oluşturulmaktadır. Oluşturulan arama mekanizmalarının iz yapıları, iyi çözümler üretmede yardımcı olmaktadır (Gross ve diğerleri, 1989).

Zhang ve diğerlerinin 2006 yılında atölye tipi üretim çizelgeleme için kullandıkları KKO yönteminde “gerçek karıncalar yiyeceğe ulaşmak için en kısa yolu nasıl bulurlar?” sorusundan esinlenilmiştir.

Şekil 4 üzerinde cevap olasılıkları dört biçimde yorumlanmıştır.

- Karıncalar bir karar noktasına varırlar.
- Karıncalar rastgele üç yol seçerler
- Karıncalar sabit bir hızla hareket ettikleri için, ortadaki en kısa yolu seçen karıncalar, diğer iki yolu seçenlerden daha hızlı yiyeceğe ulaşırlar.
- Karıncalar, orta yolda daha yüksek bir oranda toplanır ve daha fazla karınca bu yolu seçer.

En kısa yolu bulmak için sunulan cevaplar şekil 4 de açıklanmıştır.



Şekil 4. Karıncaların Yiyeceğe Ulaşmak İçin En Kısa Yolu Seçme Yöntemleri

Kaynak: (Zhang ve diğerleri 2006)

3.1.8.Örnekleme & Kümeleme (Sampling & Clustering)

Örnekleme & Kümeleme (ÖK) metodu Boender ve diğerleri tarafından 1982 yılında önerilmiştir. Genellikle S çözüm uzayı içerisindeki birçok başlangıç noktası aynı lokal optimuma ulaşmaktadır. ÖK metasezgiselinin en önemli özelliği aynı lokal optimum sonuca ulaşan başlangıç noktalarını kümeleyerek hesaplama zamanını veya hesaplama eforunu azaltmaktır. Başarılı bir kümeleme yapıldığı takdirde aynı optimum değerine ulaşacak birçok başlangıç noktası için tekrarlama yapılmasına gerek kalmamaktadır (Boender ve diğerleri, 1982).

Kombinatoryal problemlerin çözümünde kullanılacak bir ÖK metodunda aşağıdaki ayrık kümeler kullanılır:

1. Lokal optimum için aramanın başlatılabileceği noktalar kümesi (Y) olarak, Lokal optimumlar kümesi ise (Y^*) olarak adlandırılır. Arama neticesinde Y^* kümesine ulaşan Y^\wedge başlangıç noktaları kümesi olarak kabul edilir.
2. Y kümesi elemanlarına aday noktalar adı verilir. ($U =$ birleşim ifadesi) Y^* $U \cap Y^\wedge$ kümesi elemanlarına tohum noktalar (seed points) denir.

3. ÖK ile, işlem basamaklarında ilk önce S uzayında rastgele yeni K başlangıç noktaları eklenir ve ardından sonuçlar Y kümesine eklenir, Y kümesindeki en kötü amaç fonksiyon değerine sahip elemanları γ oranını kullanarak kümeden çıkartır
4. son olarak uygun bir kümeleme (clustering) kuralı ile Y kümesindeki noktaları mümkün olduğunca tohum kümesi etrafında kümelenirler (Colomi ve diğerleri, 1996).

3.1.9. Paralel Kanguru Algoritması (Parallel Kangaroo Algorithm)

Paralel Kanguru Algoritması (PKA), zıplayarak hareket eden kangurulardan esinlenilerek geliştirilmiş bir yöntemdir. PKA, literatürde “Pollard’ın Kangurusu”, “Pollard’ın Rho algoritması” veya “Pollard’ın Lambda Algoritması” olarak bilinmektedir. PKA’sı ilk kez 1978 yılında Pollard tarafından sunulmuştur (Kökçam & Engin, 2010).

Arama tekniği kullanan rassal kökenli bir yakınsama metodu olan PKA, bir fonksiyonu minimize etmek amacı ile tekrarlı süreçte çalıştırılarak uygulanmaktadır. İterasyonlar sonucunda iyileşme olmuyorsa, zıplama tekniği kullanılarak yerel minimumun çekim alanından uzaklaşmaya çalışılır. Genelde kısa süren bu süreçte elde edilen çözümlerden daha iyi bir çözüme ulaşılması beklenmemektedir. Durma kriteri, maksimum iterasyon sayısı veya amaç fonksiyonunun alt sınırı olarak belirlenmektedir (Durmaz, 2011).

PKA yönteminde aynı zaman dilimi içinde birbirinden bağımsız zıplayan iki kanguru bulunmaktadır. Evcil kangurular arama lokasyonunu küçültme amacı ile küçük adımlarla zıplama gerçekleştirirler. Vahşi kangurular ise daha büyük adımlarla arama yaparak farklı çözüm bölgelerine ulaşmaya çalışırlar. Çözüm sürecinde vahşi ve evcil kangurular hareketlerine aynı zamanda, farklı başlangıç sıralamaları ile başlarlar ve hedef değere ulaşınca ya da maksimum iterasyon sayısına ulaşınca kadar zıplamaya devam ederler. Her bir zıplama sürecinin ardından amaç fonksiyonu tekrar hesaplanır. Ulaşılan sonuç geçmiş değerlerden daha iyiye yeni bir sıralama üzerinden sürece devam edilir. Maksimum iterasyon sayısına ulaşıldığında, hangi kangurunun ulaştığı sonuç daha iyi değerdeseyse, diğer kanguru da sıradaki iterasyon grubuna o dizilim değeri ile başlar. Bu süreç, hedef değere ya da maksimum iterasyon sayısına ulaşılınca kadar devam etmektedir (Durmaz, 2011).

ATÇP çözümü için önerilen PKA'nın işlem adımları aşağıda özetlenmiştir;

1.Parametrelerin Seçimi

- İterasyon Sayısı
- İterasyon Grubu Sayısı
- Evcil ve Vahşi Kangurular İçin Bağımsız Adım Aralıkları
- Evcil ve Vahşi Kanguru İçin Başlangıç İterasyonları
- Zıplama Yönteminin Seçilmesi

2.Algoritmanın İşleyişi

- Vahşi kanguru başlangıç iterasyonu için amaç fonksiyonu değeri hesaplanır.
- Vahşi kanguru, adım büyüklüğünü, kendisi için belirlenmiş aralık içinden rastsal olarak seçer ve belirli kurallara göre zıplar.
- Vahşi kanguru için amaç fonksiyon değeri tekrar hesaplanır, ulaşılan değer öncekiyle aynı veya daha iyiyse bu iterasyon üzerinden, değilse önceki iterasyon üzerinden devam edilir.
- İlk üç adım, hedef değere ulaşıncaya kadar, ulaşılmazsa iterasyon sayısına ulaşıncaya kadar tekrarlanır
- İlk dört adım evcil kanguru için kendi parametreleri ile tekrarlanır.
- Böylece bir iterasyon grubu tamamlanır, vahşi ve evcil kangurudan sonucu daha iyi olanın dizilimi, sonraki turda hem vahşi hem evcil kanguru için başlangıç iterasyonu kabul edilir,
- Hedef değere, ulaşılmazsa maksimum tur sayısına ulaşıncaya kadar b, c, d, e, f adımları tekrarlanır. Parametre optimizasyonu sonucunda hesaplanan tamamlanma zamanı (Cmax-makespan), literatürde bu problemler için belirlenen, Bilinen En iyi Çözüm (BEÇ) değerleri ile aşağıda verilen formül ile yüzde sapma (%S) değerleri hesaplanmaktadır.
- $\%S = ((\text{Hesaplanan Tamamlanma Zamanı} - \text{BEÇ}) / \text{BEÇ}) * 100$
(Durmaz, 2011).

3.1.10. Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)

Parçacık Sürü Optimizasyonu (PSO) kuş ve balık sürülerinin iki boyutlu hareketlerinden esinlenilerek, ilk olarak 1995 yılında Kennedy ve Eberhart tarafından yapılan çalışmada tanıtılmıştır (Eberhard & Kennedy, 1995).

PSO algoritması, optimizasyon problemlerini çözmek için uçan kuşların davranışlarını ve bilgi alışverişlerini simüle eden bir arama işlemi gerçekleştirmektedir. Her kuş bir parçacık ve parçacıklardan oluşan popülasyon da sürü olarak adlandırılmıştır. Sürüdeki her bir parçacık kendi deneyimlerine ve diğer parçacıklarla etkileşim yoluyla kazandıkları bilgiye dayanarak hızlarını belirler. Parçacıklar, optimum çözüm arayışındaki optimizasyon probleminin çözüm alanı boyunca belirledikleri hızlarla uçar, kendi pozisyonlarını bir önceki tecrübelerinden yararlanarak bir önceki sürüdeki en iyi pozisyonu doğru ayarlarlar. Böylece uçuş pozisyonları, problem için potansiyel bir çözümü temsil eder (Eberhard & Kennedy, 1995).

PSO algoritması, parçacıklar olarak kavramsallaştırılan rastgele aday çözümlerinin bir popülasyonu ile başlatılır. PSO yönteminde her birey, kendi uçuş deneyimlerine ve arkadaşlarının uçuş deneyimlerine göre dinamik olarak ayarlanan bir hız ile arama alanında uçar.

Her bir birey, p boyutlu arama alanında hacimsiz bir parçacık (bir nokta) olarak ele alınmaktadır. Bu parçacık $X_t = (X_{t1}, X_{t2}, \dots, X_{td})$ ile gösterilir. Parçacık t için pozisyon değişikliğinin hızı $V_t = (V_{t1}, V_{t2}, \dots, V_{td})$ olarak temsil edilir. Her iterasyonda, t parçacığı aşağıdaki iki en iyi değerle güncellenir: P_1 (en iyi p değeri olarak adlandırılır) ve P_g (en iyi g değeri olarak adlandırılır.) $P_1 = (P_{11}, P_{12}, \dots, P_{1d})$ Parçacıkların şimdiye kadar elde ettiği en iyi çözüm (yerel en iyi çözüm) olan $P_g = (P_{g1}, P_{g2}, \dots, P_{gd})$ şimdiye kadar sürünme döneminde elde edilen en iyi çözüm (global en iyi çözüm) olarak adlandırılır.

Her bir parçacık aşağıdaki denklemlere göre yinelemeli olarak güncellenir.

$$V_{td} = W \times V_{td} + C_1 \times \text{rand}(\) \times (P_{td} - X_{td}) + C_2 \times \text{Rand}(\) \times (P_{gd} - X_{td}) \quad (4)$$

$$X_{td} = x_{td} + V_{td} \quad (5)$$

Fornüllerdeki parametrelerden w , atalet ağırlığı olarak adlandırılır. Sürünmenin küresel keşif ve yerel sömürü yetenekleri arasındaki önceki hızın miktarını kontrol etmek için kullanılır. $C1$ ve $C2$ iki pozitif sabittir ve her parçacığı P_l ve P_g pozisyonlarına doğru çeken stokastik hızlanma terimlerinin ağırlığını temsil ederler. $rand()$ ve $Rand()$, $[0, 1]$ aralığında iki rastgele işlemdir. 4. Denklem için, geleneksel PSO yönteminde, hızın güncellenmesi aşağıdaki üç bölümden oluşmaktadır.

1. $W \times V_{td}$, son hızın mevcut hıza olan etkisini temsil eden “momentum” kısmı olarak adlandırılır.
2. $C1 \times Rand() \times (P_{ld} - X_{td})$, ilkel düşünmeyi kendi başına temsil eden “bilişsel” bir parçasıdır.
3. $C2 \times Rand() \times (P_{gd} - X_{td})$ parçacıklar arasındaki işbirliğini temsil eden “sosyal” parçasıdır (Zhang ve diğerleri, 2009).

PSO algoritmasının geleneksel uygulama süreci aşağıdaki gibi özetlenmiştir:

- 1.Adım: P -boyutlu problem arama alanında rasgele konum X_t ve hızları V_t ile parçacıkların bir sürü başlatılması;
- 2.Adım: Her parçacığı fitness fonksiyonu ile değerlendirilmesi;
- 3.Adım: Parçacıkların fitness değerini partikülün en iyi p ve en iyi g ile karşılaştırarak mevcut partikülün p_{best} ve g_{best} değerini güncelleyin. Mevcut p_{best} değeri eski p_{best} 'den daha iyi ise, p_{best} değerini p değeri olarak geçerli değere eşit olarak ayarlanır ve p -boyutlu konumu p -boyutlu uzayda mevcut konuma eşittir; Aynı şekilde, eğer geçerli g_{best} değeri eski en iyi g değerinden daha iyi, mevcut geni mevcut partikülün fitness değerine sıfırlanır;
- 4.Adım: Parçacıkların konumu ve hızı denklemlere göre güncellenir. Sırasıyla (4) ve (5);
- 5.Adım: Genellikle yeterince iyi bir kondisyon veya belirli bir sayıda nesil olmayan sonlandırma kriteri yerine getirilirse adım 2'ye geçin; aksi halde, adım 6'ya geçin;
6. Adım Çıktı çözümleri (Zhang ve diğerleri, 2009).

Geleneksel PSO, bireylerin ortak bir hedefe ulaşmak için bireyler arasında iş birliği ve rekabet ile evrimleştikleri temel optimizasyon konseptini içermektedir. Arama sürecinde, sürünün her bir parçacığı karşılıklı bilgiyi küresel olarak paylaşır ve diğer meslektaşlarının keşiflerinden ve önceki deneyimlerinden yararlanır. Geleneksel PSO

algoritmasında hız deęiřtirme modeli, optimizasyon konseptinin sadece bir somut uygulamasıdır. Hız deęiřtirme modelinin sınırlamaları nedeniyle, geleneksel PSO'nun modifikasyon olmaksızın kombinatoryal optimizasyon problemlerini ele alması zordur (Zhang ve dięerleri, 2009).

3.1.11. Tabu / Yasaklı Arama (Tabu Search)

Tabu araması (TA) metasezgisel terimi Glover tarafından 1986 da ortaya çıkarılmıştır. TA yaklaşımı problem çözmeye, uyarlanabilir bellek özelliğini ve belirleyici olasılıksal keşif özelliğini kullanarak araştırma yapar. Uyarlanabilir bellek özelliği, çözüm alanını daha az çabayla daha çok verim getirecek şekilde etkili olarak araştırabilen yöntemlerin uygulanmasına izin vermektedir. Belirleyici olasılıksal keşif özelliği, kötü bir stratejik tercihin genellikle iyi bir rastgele seçimden daha fazla bilgi verebileceği varsayımından yararlanır (Glover, 1997).

Bütün komşuların araştırılmasının zor olduğu büyük problemler için aday liste stratejisi önerilmiştir. Fakat optimumdan uzak yeni noktadaki mevcut en iyi hareketin kabul edilmesi aynı deęerin tekrar ziyaret edilmesine neden olduğu için sorun yaratmaktadır. Bu durumu önlemek için belirli bir periyot süresince tekrar ziyaretini önleyen bir yasaklı listesi oluşturulur. Maksimum liste uzunluęuna erişilinceye kadar bu listeye yeni elemanlar eklenir. Maksimuma erişildiğinde, listenin en üstünde yer alan eleman listeden çıkarılır. Yasaklı listesi olarak kabul edilmesi için mevcut hareketlerin hangisinin yasak listesinde olup olmayacağına karar verilir. TA, yinelemeyi önlemek amacıyla ile belirli hareketler için yasaklama listesi oluşturan bir yaklaşımdır (Glover, 1990).

TA yönteminde en iyi çözümü elde etmek için önerilen strateji řu şekilde özetlenebilir; Tabu taramasının uyarlamalı bellek çerçevesinin kilit unsuru, arama yoğunluğu ve çeşitlendirme arasında bir denge oluşturmaktır. Yoğunlaştırma stratejileri, hareket kombinasyonlarını ve tarihsel olarak iyi bulunan çözüm özelliklerini teşvik etmek için seçim kurallarını deęiřtirmeye dayanır. Ayrıca, daha ayrıntılı aramalar için çekici bölgelere bir dönüş başlatabilirler. Dięer taraftan, çeşitlendirme stratejileri ile daha önce üretilen çözümlere dahil edilmeyen yeni nitelikler ve öznelik kombinasyonlarını birleřtirmek amaçlamaktadır. Bu stratejiler, arařtırmayı daha önce incelenenlere benzemeyen bölgelere yönlendirmeyi taahhüt eder (Glover ve dięerleri, 2007).

TA, çizelgeleme problemleri için en verimli yerel arama stratejilerinden biridir. ATÇP çözümü için TA yönteminde kullanılan parametreler aşağıdaki gibi karakterize edilebilir;

Bir çift (v, k) , tabu elemandır, burada v , hareket ettirilmekte olup, k , v harekete geçmeden önce atanan makinedir. T tabu listesidir. $|T|$ tabu durum uzunluğudur, mevcut kritik yolun çalışma sayısına eşittir ve operasyon için mevcut olan alternatif makinelerin sayısı, ATÇP için uygun bir çözümü olarak temsil edilir. $V(s)$, s 'nin komşusudur. $ObjFun(s)$, s 'nin amaç fonksiyonu değeridir. $ObjFun^*$ geçerli minimum değerdir. $CurInterNum$ geçerli yineleme sayısıdır. $MaxInterNum$, tabu aramasının gerçekleştirdiği yineleme sayısıdır. s^* , TA yöntemi ile bulunan en iyi çözümdür (Zhang ve diğerleri, 2009).

Çözümün algoritma kodu aşağıdaki gibi özetlenebilir.

1.Adım: Başlatma;

// MaxIterNum değerini ayarlayın;

// set $CurInterNum = 0$, $T = \emptyset$, $ObjFun^* = ObjFun(s^*)$, $s = s^*$;

2. Adım: $CurInterNum = CurInterNum + 1$, Geçerli komşuları alın Vs);

3. Adım: Mevcut çözümü değerlendirin; // en iyi hamle olsun s' , yeni çözüm s' ve yeni tabu T listesi;

// set $s = s'$, $T = T'$;

4. Adım: $ObjFun(s) < ObjFun^*$ ise, $ObjFun^* = ObjFun(s)$ değerini ayarlayın ve $s^* = s$; başka adım gitmek

5.Adım: $CurInterNum > MaxIterNum$ ise, Adım 2'ye gidin; aksi halde yinelemeyi durdurun ve yeni çözümü s^* olarak değiştirin.

(Zhang ve diğerleri, 2009).

3.1.12. Yapay Bağışıklık Sistemi (Artificial Immune System)

İnsan bağışıklık sistemi, vücudumuzda hastalığa yol açacak bakterilerden, virüslerden ve diğer normal olmayan hücrelerden bizi korumaktadır. Moleküller, hücreler ve organlar topluluğunun karmaşık ilişkilerinin oluşturduğu etkin bir sistemdir. Sistem, hangi antikorun hangi miktarda ve ne zaman üretildiğini kontrol etmektedir. Kendi spesifik hücreleri ile vücuda giren yabancı hücreleri tanıyan ve onları etkisiz hale getiren sistemde temel bağışıklık hücresi lenfosit beyaz kan hücreleridir. Lenfositler T ve B hücreleri olarak iki alt sınıfa ayrılmaktadır. T ve B hücrelerinin yüzeyinde reseptör

(algılayıcı) moleküller bulunmaktadır. Antikorlar tarafından tanınan antijenler daha sonra T ve B hücreleri tarafından yok edilirler (Hart, 2002).

İnsan bağışıklık sisteminin yetenekleri nedeniyle bilim adamları, sistemin çalışma mekanizmasından problem çözme aracı olarak kullanılmak üzere metasezgisel bir yöntem olan YBS algoritmasını kurmuşlardır (Hart, 2002).

YBS yönteminin öncüsü olarak Farmer ve diğerlerinin 1987 de yaptığı çalışma bilinmektedir. Üretim çizelgeleme problemleri için YBS yöntemini kullanan Mori ve diğerlerinin 1997 de, Fukuda ve diğerlerinin 1999 da yaptığı çalışmalar öncü olarak gösterilebilir. Bu çalışmalarda, üretim hattının kontrolü algılayıcı ajanlar kümesi tarafından yapılmaktadır. Algılayıcı ajanlar, insan bağışıklık sistemindeki B hücrelerinden esinlenerek oluşturulmuş ve üretim sistemindeki aksaklıkları tespit etmek için kullanılmıştır (Atay & Kodaz, 2014).

İnsan bağışıklık sistemi ile Atay ve Kodaz'ın çalışmasında geliştirilen yapay bağışıklık sisteminin bileşenlerinin terimsel açıklamaları tablo 5 de karşılaştırılmıştır.

Tablo 5

Bağışıklık Sistemi ile Atölye Tipi Çizelgeleme Sisteminin Karşılaştırması

Bağışıklık Sistemi	Atölye Tipi Çizelgeleme
Antikor	Çizelgeleme
İlgi	Tamamlanma Zamanı
Popülasyon	Zamanlamalar
Hafıza Hücresi	En İyi Antikorlar

Kaynak: (Atay & Kodaz, 2014).

3.2. Atölye Tipi Üretim Çizelgelemede Metasezgisel Optimizasyon Çalışmaları

Gonçalves ve diğerleri, 2005 yılında ATÇP için bir hibrit genetik algoritma sunmuşlardır. Bu çalışmada zamanlamalar, önceliklerin GA tarafından belirlendiği bir kural oluşturulmuşken çizelgeler, parametrelili aktif zamanlamaları üreten bir yöntem kullanılarak oluşturulmuştur. Önerilen yaklaşım, literatürden alınan 43 standart problem örneği üzerinde test edilmiş ve diğer 12 yaklaşımla karşılaştırılmıştır. Hesaplamalı sonuçlar ile önerilen algoritmanın test edilen tüm örneklerde optimal ya da optimale yakın

çözümler ürettiğini gösterilmiştir. Sonuç olarak, önerilen algoritmanın en iyi bilinen çözüme ortalama % 0.39'luk bir sapma ile çözümler ürettiği açıklanmıştır (Gonçalves ve diğerleri, 2005).

Zhang ve diğerleri, 2006 yılında ATÇP'ni optimize etmek için bir KKO sisteminin kullanımıyla ilgili bir inceleme yapmışlardır. Bu sistemin temel özellikleri olumlu geribildirim, dağınık hesaplama, sağlamlık ve yapıcı AY kullanımı olarak açıklanmıştır. KKO yönteminin performansının iyileştirilmesi tartışılmış ve yöntem sayısal deneyleri küçük bir ATÇP için uygulanmıştır. KKO yönteminin optimize edilmiş sonuçları, geleneksel optimizasyon yöntemleri ile karşılaştırılmış ve sonuçlar önerilen KKO'nun etkili bir yöntem olduğunu ve ATÇP için iyi bir çözüm bulabileceğini göstermiştir. Çalışmada ayrıca, KKO yönteminin performansının büyük ölçüde parametre değerlerine ve karıncaların sayısına bağlı olduğu açıklanmıştır. Bunun yanında, parametre değerlerinin ayarlanması, en uygun parametre değerlerinin bulunması çözümleri gereken probleme bağlı olarak zaman aldığını ve tüm problemler için çok amaçlı bir parametre ayarı bulmanın zor olduğu da açıklamıştır (Zhang ve diğerleri, 2006).

Şevkli ve Yenisey, 2006 yılında ATÇP çözümü için bir PSO modeli ve bir DKA yöntemini birlikte kullanmışlardır. Oluşturulan bu hibrit modelin, işlerin toplam tamamlanma zamanı ve performans ölçütü için literatürde yer alan bazı zor test problemleri üzerindeki sonuçları incelenmiştir. Önerilen algoritma modelinin sonuçları, literatürde iyi sonuçlar veren diğer metasezgisel yöntemleri (TB, TA, KKO ve GA) kullanan çalışmaların sonuçlarıyla ortalama bağıl hata değeri dikkate alınarak karşılaştırılmıştır. Yapılan karşılaştırmalı incelemede, önerilen algoritma modelinin genel olarak diğer algoritmalarla eşit ya da daha iyi sonuç verdiği tespit edilmiştir (Şevkli & Yenisey, 2006).

Huang ve Yang, 2008 yılında ATÇP çözümü için yaptıkları çalışmada, stok, atık ve boşta kalma süresini en aza indirmek amacıyla teslim süresini bir zaman noktası yerine bir aralık olarak kabul etmişlerdir. Bu çalışmada, veri testinin dört bölümü olarak test verilerinin oluşturulması, etkililik analizi, sağlamlık analizi ve büyük bir problemin simülasyonu kullanılmıştır. Çalışmanın amacı, ATÇP ile zamanlamayı ele almakta ve zaman penceresine optimale yakın bir çözüm getirmektir. KKO yöntemi yaklaşık çözümleri hızlı bir şekilde tanımlamak için uygulanmıştır. Simülasyon, önerilen algoritmanın etkililiğini ve sağlamlığını kanıtlamak için gerçekleştirilmiştir. Başlangıçta ATÇP için matematiksel bir model oluşturulmakta ve daha sonra veriler simüle

edilmektedir. Çözümler C dilinde kodlanan KKO yöntemi ile elde edilmiş ve verimliliği belirlemek için LINGO 7.0 kullanılarak elde edilen en iyi çözüm ile karşılaştırılmıştır. Test sonuçlarında KKO'nun çözüm süresi, LINGO kullandıktan daha az olduğundan daha verimli olduğunu gözlenmiştir (Huang & Yang, 2008).

Pezzella ve diğerlerinin 2008 yılında yapmış oldukları çalışmada, EATÇ problemi için bir GA sunulmuştur. Sunulan algoritma, başlangıçtaki popülasyonu oluşturmak, yeniden üretmek ve bireyleri seçmek için farklı stratejiler içermektedir. Hesaplamalı sonuçlar, genetik bir çerçevede daha fazla strateji entegrasyonu kullanıldığında diğer genetik algoritmalara nazaran daha iyi sonuçlar elde edildiğini göstermiştir. Ayrıca, elde edilen sonuçların TA yöntemine dayanan etkin algoritmaların sonuçlarıyla da oldukça benzerlik gösterdiği tespit edilmiştir. Bu iki sonuçtan yola çıkılarak, EATÇ problemi çözümü için GA yönteminin etkili olduğu açıklanmıştır (Pezzella ve diğerleri, 2008).

Roshanaei ve diğerlerinin 2009 yılında yapmış oldukları çalışma ATÇP için işlerin toplam tamamlanma sürelerini hesaplamak ve bu süreleri en aza indirmek amacıyla DKA yöntemini tercih etmişlerdir. DKA algoritmaları, programlama problemlerini optimal ya da optimale yakın zamanlamaya kadar çözebilmiştir. DKA, sistematik mahalle arama yapıları ile donatılmış ve yerel arama tabanlı bir algoritma olarak kategorize edilmiştir. Bu çalışmada önerilen DKA yönteminin, çeşitli sistematik yerleştirme komşu arama yapıları aracılığıyla, yerel arama temelli metasezgisel algoritmaların dezavantajlarının azaltıldığı vurgulanmıştır. Literatürdeki algoritmalara karşı önerilen bu algoritmanın etkinliğini değerlendirmek için Taillard'ın kriterlerine dayanan deneysel bir tasarım yapılmıştır. Elde edilen sonuçlar, önerilen algoritmanın diğer iyi bilinen sezgisel ve metasezgisel algoritmalara göre daha yüksek performans gösterdiği açıklanmıştır (Roshanaei ve diğerleri, 2009).

Zhang ve diğerleri, 2009 yılında EATÇ problemini çözmek için bir PSO algoritması ve TA algoritması birleştirilerek elde edilen hibrit yöntem için bir çözüm modeli oluşturmuşlardır. Yerel arama ve global arama şemasını bütünleştiren PSO, yüksek arama verimliliğine sahip bir yöntem olduğu için tercih edilmiştir. TA yöntemi ise kombinatoriyal optimizasyon problemlerinin optimal bir çözüm bulmak için tasarlanmış bir metasezgisel yöntem olduğu için tercih edildiği açıklanmıştır. Hesaplamalı sonuçlar ile önerilen hibrit algoritmanın, özellikle büyük ölçekli problemleri çözmeye etkili bir yaklaşım olduğu açıklanmıştır (Zhang ve diğerleri, 2009).

Asadzadeh ve Zamanifar'ın 2010 yılında yapmış oldukları çalışmada, genetik algoritmaların paralelleştirilmesinin bu algoritmaların performansını arttırmak için kullanılabileceğini önermiştir. Başlangıçtaki popülasyonun yaratılması ve genetik algoritmanın paralel hale getirilmesi bir etkene (popülasyon yoğunluğu, üretim aralığı vb) dayalı şekilde gerçekleştirilmiştir. Önerilen yaklaşımın performansını araştırmak için kıyas örnekleri kullanılmıştır. Elde edilen sonuçlar, çalışmada kullanılan yaklaşımın verimliliği artırdığını göstermiştir. Çalışmada kullanılan paralel etken bazlı yaklaşımın performansı, seriye dayalı yaklaşım yönteminin sonuçları karşılaştırılmış ve paralel yöntemle daha iyi sonuçlar alındığı açıklanmıştır (Asadzadeh & Zamanifar, 2010).

Vela ve diğerlerinin 2010 yılında yapmış oldukları çalışmada, yerel arama ile melezleştirilmiş bir GA yöntemi önerilmiştir. Kullanılan GA yöntemine, geleneksel GA yöntemi genişletilerek farklı kod çözme algoritması eklemiştir. Mattfeld'in çalışmasında yerel arama için kullandığı N1, N2 ve N3 olarak adlandırılan komşu yapıları bu makalede genişletilmiş ve resmileştirmiştir (Mattfeld, 1995). Bu şekilde kurulum sürelerinin varlığında iyileştirici çözümler üretilmiş ve uygulanabilir olmayan bazı iyileştirilmemiş programların atılmasına izin veren koşulları oluşturmuşlardır. İki takım problem üzerinde deneysel bir çalışma yürütülmüştür. Brucker ve Thiele'nin 1996 yılında yapmış oldukları çalışmada önerilen TB kümesi, en yeni yaklaşımları değerlendirmek için kullanılmıştır. TB setindeki deneylerde, ilk üç komşu yakaşım yapısı karşılaştırılmıştır. En iyi tercihin hepsinin aynı anda kullanılması olarak bilinmesine karşın, çalışmada önerilen yöntemin en iyi tercih olduğu gösterilmiştir (Brucker & Thiele, 1996). Literatürde çalışılmış yöntemlerin en iyisi, birinci olarak 2008 yılında yapılan bir çalışmada önerilen dal ve sınır algoritması (Artigues & Feillet, 2008) ve ikinci olarak 2005 yılında yapılan çalışmada önerilen yöntem olduğu açıklanmıştır (Balas ve diğerleri, 2005). Birincisi kesin ve tüm 15 örnek için en iyi alt sınırı vermişken, ikincisi kesin olmayıp beş büyük örnekten dördü için en iyi üst sınırları vermiştir. Bu çalışma da önerilen Hibrit genetik algoritma, 15 örneğin 14'ünde bilinen en iyi çözüme ulaşmış ve 15 örneğin 6'sında, büyük ve orta düzeyde en iyi bilinen çözümleri geliştirmiştir. Artigues ve Feillet'in 2008 yılında yapmış oldukları çalışmadan elde edilen dal ve sınırlama yöntemine kıyasla, hibrit GA, 5 büyük örnekten 4'ü için daha iyi bir ortalama sapma elde etmiştir. Bu nedenle, bu hibrit GA'nın ATÇP çözümü için verimli bir yöntem olduğu sonucuna varılmıştır (Vela ve diğerleri, 2010).

Xing ve diğerleri, 2010 yılında EATÇ probleminin çözümü için Bilgi Tabanlı Bir Karınca Kolonisi Optimizasyonu algoritması önerilmişlerdir. Önerilen algoritma,

KKO modeli ve bilgi modeli arasında etkili bir entegrasyon sağlanmıştır. Ayrıca önerilen KKO algoritmasında oluşturulan bilgi modeli, geleneksel KKO yönteminin optimizasyonundan bazı mevcut bilgileri öğrenir ve daha sonra çözüm araştırmak için mevcut bilgiyi kullanır. Önerilen yöntemin performansı, literatürdeki çalışmalarla kıyaslanarak değerlendirildiğinde, yayınlanmış bazı güncel yaklaşımları geride bıraktığını gösterilmiştir (Xing ve diğerleri, 2010).

Yang ve diğerleri, 2011 yılında Atölye tipi üretim ortamındaki dinamik tesis planlama problemini çözmek için bir model önermişlerdir. Önerilen modelde GA tabanlı bir yaklaşım uygulanmış ve hesaplanan sonuçlar önerilen modelin iyi performans gösterdiğini ortaya koymuştur. Mevcut düzen ve ideal düzen (GA çözümü) arasındaki maliyet farkına dayanan çalışma, maliyet-fayda analizini bir yönetim perspektifi ile kullanmıştır. Bu çalışmada yeniden düzenleme maliyeti, maliyet farkından daha düşükse, mevcut düzeni değiştirmenin iyi bir seçim olabileceği ortaya konulmuştur (Yang ve diğerleri, 2011).

Rajkumar ve diğerlerinin, 2011 yılında yapmış oldukları çalışmada zor kombinatoriyal problemin optimizasyonu için yinelemeli çok aşamalı bir metasezgisel yöntem olan açgözlü randomize uyarlamalı arama yöntemi kullanılmıştır. Bu yöntem iki aşamadan oluşmaktadır. Uygun bir çözümün bulunduğu bir inşaat aşaması ve inşa edilen çözüm mahalinde yerel bir optimumun arandığı bir yerel arama fazı aşaması. Çizelgeleme probleminin kısıtları, her bir işlemin, atanmış bir işlem siparişini takip etmesi ve her bir işlemin atanmış bir makinede işlenmesi gerektiğidir. Bu kısıtlar, kaynak sınırlaması ve makine esnekliği arasında denge kurmak için kullanılmıştır. Çalışmanın hedefleri, markalama, maksimum iş yükü ve toplam iş yükünün en aza indirilmesidir. Önerilen algoritmanın etkinliğini test etmek amacıyla sonuçlar literatürdeki benzer çalışmalarla karşılaştırılmıştır. Elde edilen sonuçlar, önerilen algoritmanın literatürdeki algoritmalarından daha iyi sonuçlar verdiğini göstermiştir (Rajkumar ve diğerleri, 2011).

Meeran ve Morshed, 2012 yılında ATÇP çözmek için GA ve TA yöntemlerini birlikte kullanan hibrit bir model önermişlerdir. GA yönteminin paralel geçişi ve TA yönteminin yerel olarak optimum şekilde kaçınmalarının tamamlayıcı güçlü yanları kullanılmıştır. Bu çalışmada sunulan hibrit yöntem probleme özel bir yöntem olmaktan ziyade, herhangi bir ATÇP için çözüm bulabilen bir sistem olarak geliştirilmiştir. Önerilen hibrit model, otomobil ön cam üreticisi ve çelik fabrikası gibi şirketlerin karşılaştığı gerçek yaşam sorunları için en uygun çözümleri bulabilmektedir. Önerilen yöntemin, literatürde var olan 51 karşılaştırma problemi üzerinde test edildiğinde, bu

problemlerin 48'i için optimum çözümler bulunduğu gözlemlenmiştir. Bu karşılaştırmalar sonucunda önerilen hibrit sistem daha iyi performans gösterdiği açıklanmış ve farklı gerçek hayattaki pratik problemler üzerinde kullanabileceği ifade edilmiştir (Meeran & Morshed, 2012).

Baysal ve diğerlerinin, 2012 yılında yapmış oldukları çalışmada, AATÇ problemini çözmek amacıyla rassal zıplama tekniği ile çalışan PKA yöntemi kullanılmıştır. Önerilen PKA yönteminde, iki ayrı zıplama tekniği, başlangıç dizisinin oluşturulması ve evcil ile vahşi kanguru için adım büyüklükleri parametreleri kullanılmış ve problem gruplarına göre en iyi değerler belirlenmiştir. Literatürde daha önce ulaşılan en iyi değerler ile karşılaştırma yapıldığında önerilen yöntemin daha etkin olduğu belirlenmiştir (Baysal ve diğerleri, 2012).

Zhang ve diğerlerinin 2013 yılında yapmış oldukları çalışmada, belirsiz bilgi işleme yeteneğine sahip sezgisel bulanık kümelerden faydalanarak, belirsiz işlem süreleriyle başa çıkmak için yeni bir yaklaşım önerilmiştir. Bu yaklaşımda, iki sezgisel bulanık küme arasındaki işlemi karşılaştırmak için genelleştirilmiş bir küme ortaya konulmuştur. Algoritmada üç aşamalı süreç kullanılmıştır. İlk olarak, çizelgelemedeki belirsiz bilgiyi ilgili sezgisel bulanık kümelere dönüştürmek için bir dönüşüm inşa edilmiştir. İkinci adımda; uygun olan iki sezgisel bulanık küme arasında yeni bir ekleme işlemi önerilmiştir. Daha sonra iki sezgisel bulanık küme üzerinde yeni bir karşılaştırma operasyonu sunulmuştur. Son olarak ise, operasyon bazlı bir gösterim ve bir GA kromozomu kullanılarak bir çözüm yöntemi oluşturulmuştur. Kromozomda kod çözme için operasyon tabanlı gösterim kullanılarak ve kondisyon değeri, geçiş, mutasyon vb. değerler hesaplanarak etkili bir GA tasarlanmıştır. Önerilen yöntem, literatür çalışmaları ile karşılaştırıldığında önerilen yöntemin mevcut diğer metotlardan önemli ölçüde daha iyi performans gösterdiği açıklanmıştır (Zhang ve diğerleri, 2013).

Karthikeyan ve diğerlerinin, 2014 yılında yapmış oldukları çalışmada EATÇ problemini çözmek için bir hibrit AA algoritması önerilmiştir. Bu çalışmada, üç en aza indirme hedefi vardır. Maksimum tamamlanma süresi, kritik makinenin iş yükü ve tüm makinelerin toplam iş yükü. Aynı zamanda, Hibrit AA algoritmasının; çekicilik, mesafe ve hareket olarak yeni ayrık işlevlere uygun bir şekilde dönüştürülmesini içeren bir önerme kullanılmıştır. Hibrit algoritmada, farklı AA arama doğruluğu ve ateşböcekleri arasında bilgi paylaşımını geliştirmek için AA yöntemi ile, yerel arama yöntemi birleştirilmiştir. Tanınmış kıyaslama örneklerine ilişkin deneysel sonuçlar ve yakın zamanda yayınlanmış diğer algoritmalar ile karşılaştırmalar sonucunda, önerilen hibrit

algoritmanın uygulanabilir ve etkili bir yaklaşım olduğu gözlenmiştir (Karthikeyan ve diğerleri, 2014).

Yazdani ve diğerlerinin, 2015 yılında yapmış oldukları çalışmada, çift kaynak kısıtlı ATÇP için toplam tamamlanma süresini en aza indirmek amaçlanmıştır. Çözüm için karma bir tamsayı doğrusal programlama modeli sunulmuştur. Modele ek olarak, verilen problemi çözmek için iki adet metasezgisel yöntem, Simüle Tavlama ve Titreşim Sönümlenme Optimizasyonu algoritmaları geliştirilmiştir. Her iki algoritmanın çözüm arama yöntemi olarak dört adet komşu arama yapısı kullanmıştır. Önerilen algoritmaların parametrelerini ve operatörlerini ayarlamak için Taguchi parametre tasarım metodu kullanılmıştır. Önerilen algoritmaları değerlendirmek için, oluşturulan veri kümesi ile hesaplama çalışmasını gerçekleştirilmiştir. Hesaplamalı sonuçlar önerilen Titreşim Sönümlenme Optimizasyonu yöntem sonuçlarının Simüle Tavlama sonuçlarına göre istatistiksel olarak daha iyi çalıştığını göstermiştir (Yazdani ve diğerleri, 2015).

Huang ve Süer'in 2015 yılında yapmış oldukları çalışmada, ATÇP için bulanık doyma düzeyine sahip bir dağıtım kuralı tabanlı GA yöntemi önerilmiştir. Çalışmanın amacı, bir üretim sisteminde farklı performans ölçütleri arasındaki çelişkileri uygun bir şekilde getirmek için karar verme platformu geliştirmektir. Çalışmada önerilen yöntem, işin en kısa sürede tamamlanması, ortalama akış süresi, azami gecikme ve toplam geçikmenin en aza indirilmesi amaçlarıyla bir ATÇP çözüm sürecine odaklanmıştır. Çalışmada kullanılan GA parametrelerinin etkisini incelemek için çeşitli deneyler yapılmıştır. Önerilen yöntemin, optimal ya da optimale yakın çözümler bulduğu açıklanmıştır (Huang & Süer, 2015).

Singh ve Mahapatra, 2016 yılında EATÇ problemine çözüm üretmeyi amaçlayan bir çalışma yapmışlardır. Bu çalışmada, mevcut problemi çözmek için oldukça az hesaplama süresi ile optimuma yakın çözümler sunduğu belirtilen PSO yöntemi kullanılmıştır. PSO yöntemi, iterasyon sayısı ilerledikçe hız değerlerinde oluşan azalma nedeniyle bir dezavantaja sahiptir. Bu dezavantajın üstesinden gelmek için geliştirilen PSO yönteminin, global arama kabiliyeti nedeniyle kuantum-davranışlı PSO olduğu açıklanmıştır. GA yönteminde kullanılan mutasyon operatörüyle, erken yakınsamadan kaçınmak ve çözüm çeşitliliğini geliştirmek amaçlanmış ve çalışmanın çözüm çeşitliliği, rastgele sayılar yerine kaotik sayıların (lojistik harita) kullanılmasıyla geliştirilmiştir. Çalışmada kullanılan kaotik sayı, çözüm çeşitliliği sağlarken ve hesaplama yükünü azalmıştır. Önerilen PSO yaklaşım algoritması, (Kacem ve diğerleri, 2002), (Brandimarte, 1993) ve (Dauzere-Peres & Paulli, 1997) çalışmalarındaki veriler ile

karşılaştırılmış ve önerilen yaklaşım ile elde edilen sonuçların literatürdeki çözümlere kıyasla daha iyi ya da aynı olduğu açıklanmıştır (Singh & Mahapatra, 2016).

Kundakcı ve Kulak'ın 2016 yılında yapmış oldukları çalışmada, üretim ortamında kaçınılmaz olarak gerçekleşen rastlantısal iş varışları, makine arızaları ve işlem süresindeki değişiklikler gibi dinamik olaylar, statik ATÇP çözümü için geçmiş çalışmalarda genellikle göz ardı edilmiştir. Bu çalışma da dinamik ATÇP çözümü için Hibrit GA metodolojileri tanıtılmaktadır. İşlerin sayısı, makine sayısı ve farklı dinamik olaylar gibi çeşitli kıyas problemleri oluşturulmuş ve önerilen metodolojilerin performansını değerlendirmek için ayrıntılı sayısal deneyler gerçekleştirilmiştir. Önerilen hibrit yöntem ile, literatürdeki çalışmalara kıyasla daha iyi çözümler elde edildiği açıklanmıştır (Kundakcı & Kulak, 2016).

Nouiri ve diğerlerinin, 2018 yılında yapmış oldukları çalışmada, makine ataması problemi ve işlem sıralama problemi olmak üzere iki zorluğu kapsayan EATÇ problemini çözmek için PSO algoritması kullanılmıştır. Çalışmanın amacı; yalnızca hızlı bir çözüm uygulamak değil, aynı zamanda öngörülemeyen durumlara uygun olacak şekilde kolayca yeniden yapılandırılabilen verimli bir algoritma oluşturmaktır. Kaynakların durumuna ve planlanmamış olaylara göre gerçek zamanlı kararlar alabilen PSO yöntemi için, çoklu faktör sisteminde dağıtan iki çoklu aracı sistemi yaklaşımı önerilmiştir. Deney sonuçları, önerilen algoritmanın gerçek üretimi yönlendirmede etkili bir rol oynayabildiğini göstermektedir (Nouiri ve diğerleri, 2018).

BÖLÜM IV

YAPAY SİNİR AĞLARI YÖNTEMİ

4.1. Yapay Zekâ ve Tarihçesi

İnsan gibi düşünen ve davranan sistemlerin geliştirilmesine yönelik olarak 1950’li yıllardan beri süren yapay zekâ çalışmaları, algoritma olarak insanı taklit etmeye yönelik olduğundan mühendislik, nöroloji ve psikoloji gibi alanlara da kullanılmaktadır. İnsan gibi düşünebilen ve davranabilen sistemlerin geliştirilmesi için yapılan çalışmalar genel olarak yapay zekâ olarak tanımlanmaktadır (Yurtoğlu, 2005).

1943 yılında II. Dünya Savaşı sırasında Kripto analizi yapabilmek amacıyla Alan Turing önderliğinde üretilen elektromanyetik cihazlar sayesinde bilgisayar bilimi ve yapay zekâ ortaya çıkmıştır. Alan Turing, Nöbert Wiener ve John von Neumann gibi matematikçiler ile bilgisayar bilimcilerin katkılarıyla bugünkü bilgisayar bilimlerinin temelleri atılmıştır. Bilgisayar bilimlerinin Nobel Ödülü sayılan Turing Ödülü ve yapay zekâ alanında halen geçerliliğini koruyan Turing Testi’nin adı da Alan Turing’den gelmektedir (Çamoğlu, 2014).

Yapay zekâ alanındaki ilk önemli gelişme 1956 yılında ABD’de düzenlenen Dartmouth Konferansı olarak kabul edilir. Bu konferans LISP yapay zekâ programlama dilinin yaratıcısı da olan ABD’li bilim insanı John McCarthy (1927-2011) tarafından organize edilmiştir. John McCarthy günümüz yapay zekâ teknolojisinin gerçek öncüsü kabul edilmektedir. “yapay zekâ” ifadesi Dartmouth Konferansı’nda ilk olarak John McCarthy tarafından kullanılmıştır (Çamoğlu, 2014).

Yapay zekâ; bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekâsına özgü kapasitelerle donatılmış bilgisayarlardır. Yapay zekâ teknolojisinden en önemli beklenti, insan gibi tutumlar içinde olan sistemler üretebilmektir.

Alan Turing, zeki davranışı, bir sorgulayıcıyı kandırarak kadar bütün bilişsel görevlerde insan düzeyinde performans göstermek olarak tanımlamıştır. Bunu ölçmek için Turing testi olarak bilinen bir test önermiştir. Turing testinde denek, sorgulayıcıyla bir terminal aracılığıyla haberleşmektedir. Eğer sorgulayıcı, denegın insan mı yoksa bir bilgisayar mı olduğunu anlayamazsa denek Turing testini geçmiş sayılır. Testen anlıyoruz ki; bilgisayarda zeki davranışı üreten sürecin, insan beynindeki süreçlerin

modellenmesiyle elde edilebileceği gibi tamamen başka prensiplerden de hareket edilerek üretilmesi mümkündür (Çamoğlu, 2014).

Yapay zekâ, insanın düşünme sisteminden yola çıkarak bilgisayar işlemlerini geliştirmeye çalışmak olarak tanımlanabilir. Özetle; yapay zekâ programlanmış bir bilgisayarın öğrenme teknikleri ile donatılarak düşünmesini sağlama girişimi olarak tanımlanabilir.

Yapay zekâ araştırmaları ile YSA yaklaşımı ortaya çıkmıştır. Yapay zekâ bilim alanının bir alt kademesi olarak görülen YSA teknolojisi öğrenebilen sistem yapılarının temeli olarak kabul edilmektedir. İnsan beyninin temel işlem elemanı olan nöronu (neuron) şekilsel ve işlevsel olarak basit bir şekilde taklit eden YSA, bu metod ile biyolojik sinir sisteminin basit bir simülasyonu olarak oluşturulan programlardır. İnsanoğluna özgü deneyerek (yaşayarak) öğrenme yeteneğini bilgisayar ortamına taşıyabildiği düşünülen YSA, bir bilgisayar sistemine özgü girdi verisinden öğrenme yapabilme ayrıcalığına sahiptir ve bu yolla birçok avantajlar sunmaktadır (Yurtoğlu, 2005).

4.2. Makine Öğrenmesi

Öğrenmek, bilgi edinme ve yetenek edinme olarak iki açıdan değerlendirilebilir. Bilgi edinme teknik, modeller, şemalar ve kurallar şeklinde yapılır. Teknik, metodolojik ve teorik bilginin elde edilmesini kapsamaktadır. Yetenek kazanma ve geliştirmek ise otomatik ve kavramaya dayanan yeteneklerin pratik yapılar adım adım geliştirilmesini kapsamaktadır. Yetenek geliştirme bilgi edinmenin aksine daha düşük kavrama gerektirmektedir. Bilgi edinmenin yapay zekâ araştırmaları alanında olduğu kabul edilirken, yetenek edinmenin ise bilgi edinmenin alt işlemi olduğu kabul edilmektedir (Carbonell, 1987).

Öğrenme işleminin bilgisayar modellemesi, makine öğrenmesi olarak tanımlanabilir. Yapay zekâ araştırmalarının en önemli hedefi düşünebilen bir bilgisayar sistemi yaratmaktır. Makine öğrenmesi, bilgisayar sisteminin kendi kendine anlayabilme ve öğrenebilme kabiliyetini sağlamayı amaçlar. Bilimsel hedefi, alternatif öğrenme mekanizmalarını araştırmak ve bulmaktır. Ayrıca makine öğrenmesi sisteminden, öğrenci ile iletişim kurması ve çalışma verilerinin sisteme uygulanabilmesi beklenmektedir. Sistemin kurulması ve test edilmesiyle bu süreçlerin etkinliği ve limitleri belirlenmektedir (Carbonell, 1987).

4.3. Yapay Sinir Ağlarında Öğrenme

Yapay zekâ uygulamalarının makine öğrenme yöntemlerinden biri olan YSA, biyolojik sinir sistemi taklit edilerek geliştirilmiştir. Biyolojik sinir hücreleri, sinapslar yardımı ile kendi aralarında iletişim kurarlar. Bir YSA işlediği bilgileri aksonlar vasıtasıyla diğer hücrelere gönderir. Bu sistemden esinlenerek geliştirilen yapay sinir hücreleri de dışarıdan gelen bilgileri girdi olarak kullanarak bir toplama fonksiyonu ile toplar. Elde edilen sonuçları aktivasyon fonksiyonundan geçirerek çıktıyı üretip ağın bağlantılarının üzerinden diğer hücrelere gönderir. YSA'nın katmanlarında kullanılan toplama (x ; girdi, w ; ağırlık değeri ise $(F_{net} = \sum_{i=0}^N xi * wi)$) ve çeşitli aktivasyon fonksiyonları (sigmoid, tanh, lineer vb.) bulunmaktadır. YSA sisteminde hücreleri birbirlerine bağlayan paralel olarak dizilmiş sırasıyla girdi katmanı, ara katmanlar ve çıktı katmanı bulunmaktadır (Öztemel, 2003).

YSA sisteminin öğrenme sürecinde, öncelikle örnek değerler ve bu değerlerden elde edilmesi gereken çıktı sonuçları işlenmektedir. YSA, öğrenme süresinde kullandığı girdi değerlerinden (örneklerden) ve elde edilen çıktılarından genellemeler oluşturarak bir çözüm uzayı üretmektedir. Sonrasında benzer girdiler için bu çözüm uzayından sonuçlar üretebilmektedir. Her prototip için girdi değerleri ve o girdiler için ağın üretmesi gereken çıktı sonuçları belirlenmiştir. Sisteme sunulan girdi için ağın ürettiği çıktı ağın beklenen çıktıları ile karşılaştırıldığında değerler arasındaki fark hata değeri olarak kabul edilmektedir. Sistemin genel amacı bu hata değerini minimize etmektir. Hata değeri, ağın ağırlık değerlerine dağıtılarak bir sonraki iterasyonda hatanın azaltılması sağlamaktadır (Öztemel, 2003).

Öğrenme Algoritmaları 3 başlıkta toplanabilir;

1. Danışmanlı Öğrenme: YSA'ya girdi ve çıktı değerlerinin sunulduğu öğrenme biçimidir. YSA'dan elde edilen çıktı değeri ile beklenen sonuç arasındaki sayısal farklılık (hata değeri) minimize edilerek sıfıra mümkün olduğu kadar yaklaştırmak hedefiyle ağırlık değerleri değiştirilir (Kaya ve diğerleri, 2005).
2. Danışmansız öğrenme: YSA'ya yalnızca girdi değerleri verilir. Bu değerlere uygun sonuç değerleri elde edilinceye kadar ağırlık değerleri değiştirilir (Sağiroğlu ve diğerleri, 2003).

3. Takviyeli öğrenme: YSA sisteminin giriş değerlerine karşılık gelecek uygun çıktı değerlerinin alınması için, ağırlıkların optimum olmasına yönelik genetik algoritmalar ya da optimizasyon yöntemleri kullanılır (Sağiroğlu ve diğerleri, 2003).

YSA'nın öğrenmesi istenilen koşul için hazır örnekler kullanılmaktadır. Ağın eğitilmesi için eğitim seti adı verilen örneklerin toplandığı ve ağın test edilmesi için de test setinin kullanılması gerekmektedir. Sistemin öğrenme süreci için eğitim setindeki örnekler birer birer YSA'ya işlenir. Sonrasında test setindeki örnekler kullanılarak ağın performansı ölçülür. Ağa daha önce sunulmamış girdi değerleri karşısındaki başarılı tahmin oranı ise ağın iyi öğrenmesi durumunun başarısını ortaya koymaktadır (Öztemel, 2003).

Ağın Öğrenme Sistemi 8 adımda açıklanabilir;

1. Ağın yapısının belirlenmesi: Öğrenilmesi istenen duruma özgü oluşturulması gereken ağ sisteminin yapısı belirlenir. Kaç tane girdi ünitesi, kaç tane ara katman, her ara katmanda kaç tane işlem elemanı ve kaç tane çıktı elemanı olması gerektiğinin belirlendiği adımdır.
2. Öğrenme parametrelerinin belirlenmesi: Ağın öğrenme katsayısı, işlem unsurlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametrelerin belirlendiği adımdır.
3. Ağırlıkların başlangıç değerlerinin atanması: İşlem unsurlarını birbirlerine bağlayan ağırlık değerlerinin atanması bu adımda yapılmaktadır. Başlangıç değeri çoğunlukla rastgele seçilir ve ağ sistemine uygun değerler öğrenme sırasında sistem tarafından belirlenir.
4. Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi: YSA'nın öğrenmeye başlaması ve öğrenme kuralına uygun olarak ağırlık değerlerini değiştirmesi için ağ sistemine örneklerin belirli bir kural çerçevesinde gösterilmesi adımdır.
5. Öğrenme sırasında ileri hesaplamaların yapılması: YSA'ya atanan girdi değerleri için ağ sisteminin çıktı değerlerini hesapladığı adımdır.

6. Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması: Ağın ürettiği hata değerlerinin hesaplandığı adımdır.
7. Ağırlıkların değiştirilmesi: Geri hesaplama yöntemi kullanılarak elde edilen hata değerinin azalması için ağırlıkların değiştirildiği adımdır.
8. Öğrenmenin tamamlanması: YSA, öğrenme sürecini tamamlayınca kadar yani gerçekleşen ile beklenen çıktılar arasındaki hatalar kabul edilir düzeye ininceye kadar bu süreçler devam ettirilmektedir (Çayıroğlu, 2012).

4.4. Yapay Sinir Ağları ve Tarihçesi

1943 yılında Warren McCulloch adlı bir nörobiyolojist ve Walter Pitts adlı bir istatistikçi, “Sinir Aktivitesindeki Düşüncelere Ait Bir Mantıksal Hesap” başlıklı bir makale yayınlamaya dijital bilgisayarlar alanında önemli gelişmelere zemin oluşturmuşlardır. McCulloch ve Pitts, kendi nöroloji anlayışları çerçevesinde YSA modelleri geliştirmişlerdir. Bu modeller, nöronların çalışma şekilleri hakkında bazı varsayımlarda bulunmuştur (Yurtoğlu 2005).

YSA hakkında ikinci önemli gelişmeyi, Bronx Yüksek Bilim Okulu’ndan Frank Roşenblatt 1958 yılında, doğrusal algılayıcı (perceptron) modeli ve öğrenme kuralı ile yapmıştır. Önerdiği bu model, günümüzün makine öğrenme algoritmasının da temeli olarak kabul edilmektedir (Şen, 2004).

1959’da, Stanford üniversitesinden Bernard Widrow ve Marcian Hoff, nörona benzer basit yapılı unsurlara dayanan ve “adaline” (Adaptive Linear Neuron) olarak adlandırılan bir uyumlu lineer eleman geliştirmişlerdir. Adaline ve iki tabakalı formda olan “madaline” (Multiple Adaline); ses tanıma, karakter tanıma, hava tahmini ve uyumlu kontrol gibi çok çeşitli uygulamalar için kullanılmıştır. Madaline, telefon hatlarında oluşan yankıları yok eden bir uygulanabilir süzgeç olarak kullanılmıştır. Bu özelliği ile gerçek dünya sorunlarına uygulanmış olan ilk sinir ağıdır ve hala kullanımda bulunmaktadır (Elmas, 2003).

1969 yılında, Minsky ve Papert bir kitap yazmış ve YSA’nın temel olarak ilgi çekici konular olmadığını belirterek birçok araştırmacının bu alanda çalışmaktan vazgeçmelerine sebebiyet vermiştir. Çünkü onlar ‘Perceptron’ isimli kitaplarında YSA sisteminin birçok mantık fonksiyonunu gerçekleştiremeyeceğini matematiksel olarak ispat etmişlerdir. Bu durum YSA üzerindeki çalışmalarını bir süre durdurmuştur (Şen, 2004).

Werbos ve diğeri 1974 yılında geri yayımlı öğrenme yöntemini geliştirerek kullanmışlardır. İlerleyen yıllarda bu yöntem ünlenerek diğeri bilim adamları tarafından çeşitli çalışmalarda kullanılmıştır. Geri yayımlı ağlar, günümüzde bilinen ve yaygın olarak kullanılan yapay sinir ağlarıdır (Yurtoğlu, 2005). Fukushima ise el yazısı karakterleri yorumlamak için, adım adım eğitilmiş çok tabakalı YSA oluşturmuştur. Cognitron olarak adlandırılan bu model 1975 yılında yayınlanmıştır (Yurtoğlu 2005).

1982 ve 1984 yıllarında ilgi çeken bir başka gelişme fizikçi Hopfield tarafından sunulmuştur. Hopfield, YSA yönteminin geliştirilebileceği ve özellikle geleneksel bilgisayar programlama ile çözülmesi zor olan problemlere çözüm üretebileceğini göstermiştir. Kendi adıyla anılan bir ağ yapısı mevcuttur ve birçok alana uygulanmıştır (Öztemel, 2003).

1987 yılında yapılan ilk yapay sinir ağları sempozyumundan sonra YSA uygulamaları yaygınlaşmıştır (Elmas 2003). Grossberg ve Carpenter tarafından yapılan çalışmalar, yankı algoritmaları araştıran bir düşünce okulunun temellerini atmıştır. Bu araştırmacılar, temeli biyolojik olarak makul modellere dayanan Adaptive Resonance Theory – Uyumlu Rezonans Teorisi ağlarını geliştirmişlerdir (Yurtoğlu 2005).

YSA, beynin bir işlevini yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. Bir YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşmaktadır. YSA, öğrenme algoritmaları ile öğrenme sürecinden geçtikten sonra, bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip olmaktadır (Saraç, 2004).

YSA sistemi, bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bunun temel sebebi ise insan beyni ile sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.

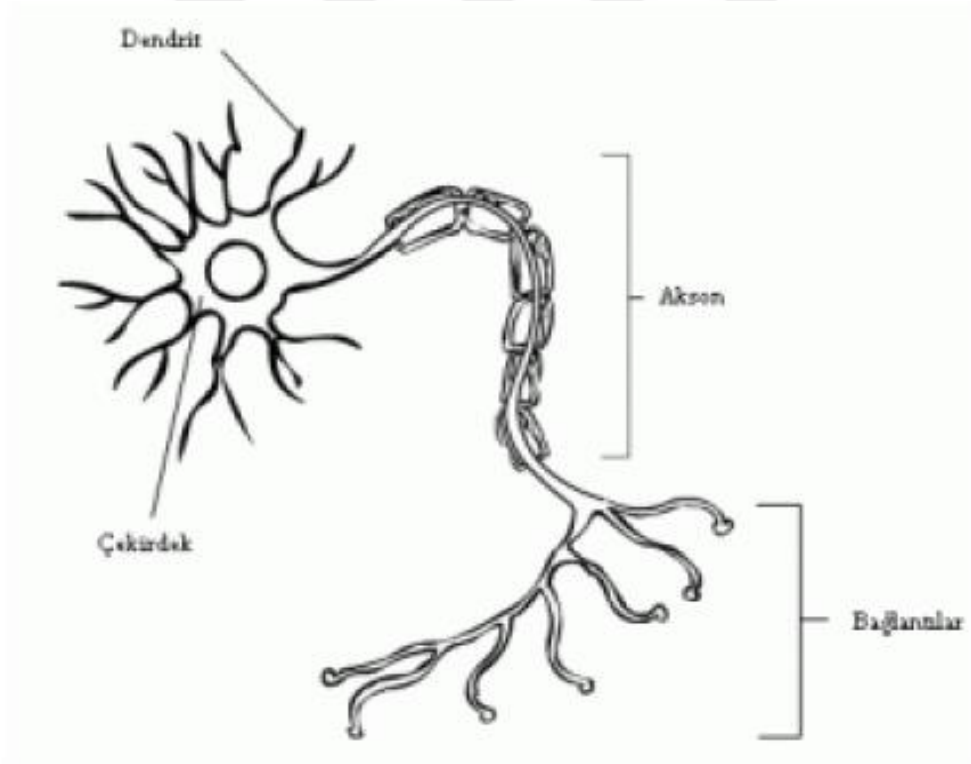
Günümüzde, YSA'larla ilgili araştırmalar yapan çok sayıda bilim adamı ve araştırma grupları vardır. Farklı bilim ve ilgi alanlarında kullanılan YSA, çeşitli teknolojik gelişmelerde rol oynamaya devam etmektedir.

4.5. Yapay Sinir Ağları Yapısı

Yapay sinir ağları biyolojik sinir ağlarının modellenmesidir. Biyolojik sinir sisteminde sinyal iletim süreçleri şu şekilde özetlenebilir. Öncelikle bir insanın beyinde yaklaşık olarak 10 milyar sinir hücresi bulunmaktadır. Bu nöronların birbirleriyle yaptığı bağlantı sayısının yaklaşık 60 trilyon olduğu tahmin edilmektedir. Bu sinirler girdi

bilgilerini duyu organlarından almaktadırlar. Aldıkları bilgiler taşıyıcı sinirler tarafından işlenip bir sonraki sinire aktararak sinyalin merkezi sinir sistemine kadar ulaşmasını sağlanmaktadır. Merkezi sinir sistemi bu sinyalleri alıp yorumlayarak tepki sinyallerini üretmektedir. Bu sinyaller tepkilerin oluşacağı organlara tepki sinirleri vasıtasıyla iletilmektedir. Bu süreç sonucunda duyu organlarından gelen bilgilere karşı, tepki olarak uygun refleksler biyolojik sinir sistemi ile gönderilir (Çayıroğlu, 2012).

Biyolojik sinir sisteminin temel yapı taşı nöronlardır. Nöronların yapısı dört ana bölümden oluşmaktadır; dendrit, akson, çekirdek ve bağlantılar. Dendritlerin sinir hücresinin ucunda bulunan ve ağaç kökü görünümüne sahip bir yapısı vardır. Dendritlerin görevi, bağlı olduğu diğer nöronlardan ya da duyu organlarından gelen sinyalleri çekirdeğe iletmektir. Çekirdek, dendrit tarafından gelen sinyalleri bir araya toplayarak ve aksone iletmektedir. Toplanan bu sinyaller akson tarafından işlenerek nöronun diğer ucunda bulunan bağlantılara gönderilmektedir. Bağlantılar ise yeni üretilen sinyalleri diğer nöronlara iletmektedirler (Çayıroğlu, 2012).



Şekil 4: Sinir Hücresi

Kaynak: (Çayıroğlu, 2012)

Şekil 4.'te bir sinir hücrenin (nöronun) yapısı gösterilmektedir. Bir nöronun yüzlerce, bazen de binlerce dendrit çıkabilir. Bunların uzunluğu genellikle bir milimetreden daha kısadır. Bazıları ise birkaç milimetre uzunluğa ulaşabilir. Sinapslar, bir sinir hücresi çıkışı ile komşu olan bir başka sinir hücresi girişi arasındaki iletişimi sağlayan bağlantılardır. Dendritler, sinapslardan alınan bilgileri gövdeye taşımakla görevlidirler. Hücre gövdesi içerisinde toplanan bilgi uyarma eşiğini aştığında hücre uyarılır ve aksonlar yardımıyla diğer hücrede görülmeye sinyaller gönderilir. Bu yüzden aksonlar taşıma hatları olarak adlandırılır. Aksonlar ve dendritler işlevleri bakımından birbirine benzetilebilir. Akson dendritlere nazaran daha uzundur. Ayrıca daha az sayıda dallanmaya sahip olduklarından dolayı düzgün bir yapıya sahiptirler. Dendritler ise daha düzensiz bir yüzeye ve çok sayıda dallanmaya sahip olduklarından dolayı daha çok doğal bir ağacı andırırlar (Şen 2004).

YSA yapısı, Biyolojik sinir hücresi yapısından esinlenerek şekillendirilmiş beş bölüme ayrılmıştır. Bu bölümlerde nöron girdi elemanı, dendritler toplama fonksiyonu, hücre gövdesi transfer fonksiyonunu, aksonlar çıktı elemanını, sinapslar ağırlıkları temsil etmektedir. Biyolojik sinir hücresi ile yapay sinir ağı benzerliği tablo 6 da özetlenmiştir.

Tablo 6

Biyolojik Sinir Hücresi ile Yapay Sinir Ağı Benzerliği

Biyolojik Sinir Hücresi	Yapay Sinir Ağları
Nöron	Girdi Elemanı
Dendrit	Toplama Fonksiyonu
Hücre Gövdesi	Transfer Fonksiyonu
Aksonlar	Çıktı Elemanı
Sinapslar	Ağırlıklar

Kaynak: (Sağiroğlu ve diğerleri, 2003)

YSA'nın 5 bölümü aşağıdaki gibi açıklanmaktadır.

1. Girdiler (Entries): Girdiler nöronlara gelen verilerdir. Girdiler YSA hücresine başka bir hücreden gelebileceği gibi dışarıdan da gelebilir. Toplanan girdilerden elde edilen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nöron çekirdeğine gönderilir.

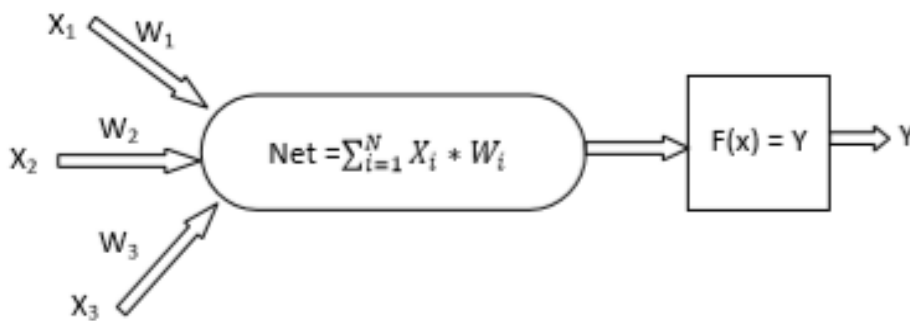
2. **Ağırlıklar (Weight):** YSA hücresine gelen bilgiler girdiler üzerinden çekirdeğe ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılarak çekirdeğe iletilir. Bu sayede girdilerin üretilecek çıktı üzerindeki etkisi ayarlanabilmektedir. Bu ağırlıkların değerleri pozitif, negatif veya sıfır olabilir. Ağırlık değerleri girdi verileri ile çarpılarak toplandığından ağırlığı sıfır olan girdilerin çıktı üzerinde herhangi bir etkisi olmamaktadır.

3. **Toplama Fonksiyonu (Collection Function):** Toplama fonksiyonu bir YSA hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur.

4. **Aktivasyon Fonksiyonu (Activation Function):** Bu fonksiyon hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirlemektedir. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi de kullanıldığından hesaplamaların yavaşlamaması için türevi kolay hesaplanabilir bir fonksiyon seçilmektedir. Günümüzde en yaygın olarak kullanılan “Çok katmanlı algılayıcı” modelinde genel olarak aktivasyon fonksiyonu olarak “Sigmoid fonksiyonu” kullanılır.

5. **Hücrenin Çıktısı (Output Of Cells):** Aktivasyon fonksiyonundan çıkan değer, hücrenin çıktı değeri olmaktadır. Bu değer YSA çıktısı olarak kullanılabilirken tekrardan ağırlık içinde de kullanılabilir. Her hücrenin birden fazla girdisi olmasına rağmen bir tek çıktısı olmaktadır. Bu çıktı istenilen sayıda hücreye bağlanabilir (Öztemel, 2003).

YSA sisteminin formüsel yapısı şekil 5 de gösterilmiştir.



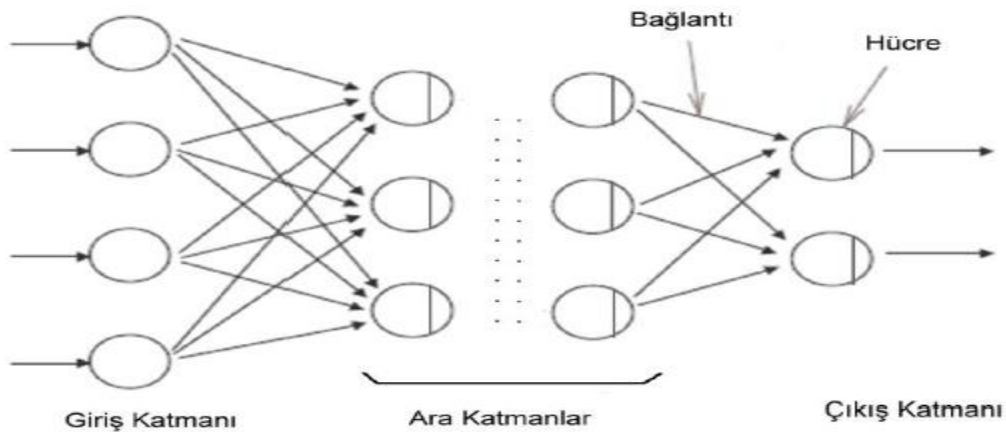
Şekil 5 Yapay Sinir Ağının Formüsel Yapısı

Kaynak: (Çayiroğlu, 2012)

YSA sisteminin 3 katmanı aşağıdaki gibi açıklanmaktadır.

1. Giriş Katmanı (Input Layer): Bu katmanda YSA'ya girdi verileri gelmektedir ve çoğunlukla girdi verileri herhangi bir işleme tabi tutulmadan ara katmana iletilmektedir.
2. Ara/Gizli Katmanlar (Intermediate Layer): Bu katmanda girdi katmanından elde edilen veriler kullanılır. YSA sisteminde kaç ara katmanı bulunacağı problemin türüne göre kullanıcı tarafından seçilir. Bazı YSA sistemlerinde ara katman bulunmayabilir. Ara katmanlardaki nöron sayısı, girdi ve çıktı değerlerinin sayısından bağımsızdır. Birden fazla ara katman olan ağlarda ara katmanların kendi aralarındaki hücre sayıları da farklı olabilir. Ara katmanların ve bu katmanlardaki nöronların sayısının artması hesaplama karmaşıklığını ve çözüme ulaşma süresinin arttırmasına sebep olmaktadır. Buna rağmen ara katmanlar, YSA yönteminin daha karmaşık problemlerin çözümünde kullanılabilmesini sağlar.
3. Çıkış Katmanı (Output Layer): Bu katmanda ara katmanlardan gelen bilgiler işlenerek ağın çıktı değerleri üretilir. Geri beslemeli ağlarda bu katmanda üretilen çıktı değerleri kullanılarak ağın yeni ağırlık değerleri hesaplanır (Öztemel, 2003).

YSA yapısı, şekil 6 da gösterilmiştir.



Şekil 6 Yapay Sinir Ağı Yapısı

Kaynak: (Çayiroğlu, 2012)

Ara katman sayısının ve her ara katmandaki oluşum unsuru sayısının kaç olması gerektiğini gösteren belirli bir yöntem yoktur. Dolayısıyla deneme yanılma yolu ile bu sayılar belirlenmektedir. Her katmanda bulunan oluşum elamanları birbirlerine bağlıdır. Bilgi akışı girdi katmanından ara katmana oradan da çıktı katmanına doğrudur (Öztemel, 2003).

YSA içerdiği nöronların birbirine bağlantı şekline göre ileri ve geri beslemeli olarak ikiye ayrılır.

İleri Beslemeli Ağlar: İleri beslemeli ağlarda veriler giriş katmandan çıkış katmana doğru iletilmektedir. Her bir katman yalnızca kendinden sonraki katman ile bağ kurmaktadır. YSA sistemine gelen bilgiler öncelikle giriş katmanına sonrasında sırasıyla ara katmanlardan ve çıkış katmanından işlenerek geçer ve çıktı değerleri elde edilir. İleri beslemeli bir ağda işlemci elemanlar genellikle katmanlara ayrılırlar. Veriler, giriş katmanından çıkış katmanına doğru tek yönlü bağlantılarla iletilir (Yurtoğlu, 2005).

İleri beslemeli ağlara örnek olarak; Çok Katmanlı Algılayıcı (Multi Layer Perseptron) ve Vektör Ölçümlü Öğrenme (Learning Vector Quantization) ağları verilebilir (Yurtoğlu, 2005).

Geri Beslemeli Ağlar: Geri beslemeli ağlarda bir hücrenin çıktısı kendinden sonra gelen hücrenin katmanına girdi olarak verilebileceği gibi kendinden önceki katmanda ya da kendi katmanında bulunan herhangi bir hücreye de girdi olarak aktarılabilir. Bu yapısı ile geri beslemeli ağlar doğrusal olmayan dinamik bir davranış göstermektedir. Geri beslemeli sinir ağı, ağın çıktı değerini ve ara katmanın çıktı değerlerini kullanır. Bu değerlerin giriş birimlerine ya da önceki ara katmanlara geri beslendiği bir ağ yapısına sahiptir. Böylece, giriş değerlerinin hem ileri yönde hem de geri yönde aktarımı yapılmaktadır. Bu tür ağların dinamik hafızaları vardır ve bir andaki çıkış hem o andaki girdiyi hem de önceki girdileri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için bu ağ yapıları uygundur. Ayrıca bu ağlar, çeşitli tipteki zaman-serilerinin tahmininde oldukça başarı sağlamışlardır (Saraç, 2004).

Geri beslemeli ağlara örnek olarak; Hopfield, Kendi kendini organize eden harita (Self Organizing Map), Elman ve Jordan ağları verilebilir (Saraç, 2004).

4.6. Atölye Tipi Çizelgeleme Problemi Çözümünde Kullanılan Yapay Sinir Ağı Yaklaşımları ve Çalışmaları

Hopfield ve Tank'ın 1985 de yapmış oldukları bir çalışmada kullanılan Hopfield Ağı, sinir ağı tabanlı programlama sistemlerine ve işlemsel kuvvetlendiricilerden oluşan elektronik bir devreye benzer. Bu ağ modelleri bu tür donanım bileşenleri kullanılarak simüle edilir ve dinamiksel bir enerji işlevi tarafından yönetilir. Bu çalışmanın amacı, çeşitli kaynak kısıtlamalarına tabi olan, enerji işlev değerindeki işin, toplam tamamlanma süresine dayanan değeri en aza indirmektir (Hopfield & Tank 1985).

Foo ve Takefuji 1988 yılında ATÇP için, sinir ağı tabanlı bir sistem uygulayan ilk kişilerdir. Bu çalışmalarında ay için geliştirilmiş bir matris sistemi kullanarak mn ($mn+1$) nöronlarını, iki boyutlu bir ATÇP çözümü için uygulamışlardır. Yerel yakınsama problemlerinden kaçınmak için modele simüle edilmiş bir tavlama işlemi uygulamışlardır. Kullandıkları bu yöntemde birkaç sınırlama vardır. Bunların en önemlisi, iş sayısının makine sayısından daha fazla olması gerektiği, sadece küçük problemlerin çözülebilir olduğu ve optimal bir çözüm garantisi olmadığıdır. Modellerinde tavlama yöntemi aşırı hesaplama gerektirdiğinden nöronlara ihtiyaç duyulmuştur (Foo & Takefuji,1988).

Foo ve Takefuji bir Tamsayılı Doğrusal Programlama Sinir Ağı oluşturarak önceki çalışmalarını genişletmişlerdir. Oluşturulan sinir ağında enerji işlevi, tüm işlerin başlangıç zamanlarının toplamı ile temsil edilmiştir. Çalışmada tüm işler belirli bir sırada işlenmektedir, hiçbir makine aynı anda iki işi işlemeyecek şekilde doğru işlem sırası takip edilmektedir. Böyle bir yapıyı birleştirerek, nöron ve bağlantı sayısındaki gereksinimi ise sırasıyla $[mn(n+1)]$ ve $(n+1)$ seviyesine düşürülmüştür. Önerilen model ile optimum bir çözüm garanti edilemezken küçük problemler çözülebilmektedir (Foo ve diğerleri, 1995).

Zhou ve diğerlerinin 1991 de yaptıkları çalışmada, Foo ve Takefuji tarafından 1988 yılında kullanılan Tamsayılı Doğrusal Programlama Sinir Ağı modelinin yerel yakınsamasının bazı eksikliklerini gidermek için bir Doğrusal Programlama Sinir Ağı önerilmiştir. Sayısız kontrol değişkenini içeren geleneksel bir tamsayı doğrusal programlama yöntemine olan ihtiyacı önlemek için, doğrusal ağ kullanarak ikinci dereceden bir enerji işlevini kullanmaktan kaçınmışlardır. Bu metodu kullanarak, nöronların ve ara bağlantıların sayısını büyük ölçüde azaltılmış ve 20 iş ile 20 makineye kadar olan problemlerin çözülmesini sağlamışlardır (Zhou ve diğerlerinin 1991).

Chang ve Nam'ın 1993 de yaptıkları çalışmada, küresel optimumda birleşmeye çalışan bir Doğrusal Programlama Sinir Ağı önerilmiştir. Ağın çalışma sistemi Foo ve Takefuji (1988c) ile aynı prensiptedir. Ancak rastgele atanan değişkenleri getirerek tamsayı kısıtlamalarını ortadan kaldırmışlardır. Bununla birlikte çok sayıda kısıtlamanın formüle edilmesi gerektiğinden sadece küçük problemler çözülmektedir. Önerilen modelde sayısız ara bağlantı ve nöron gereklidir ayrıca en uygun çözümü bulma garantisi yoktur (Chang & Nam'ın 1993).

Willems ve Rooda'ın 1994 de yayınladıkları çalışmada, arama alanının ön hesaplama ile azaltıldığı ve eşikler olarak bilinen minimum başlangıç zamanlarının yaratıldığı bir Tamsayılı Doğrusal Programlama Sinir Ağı modeli önerilmiştir. Önerilen Model uygulanabilir bir çözüme hızlı bir şekilde yakınsama, geri bildirim bağlantılarının dahil edilmesiyle desteklenir. Ancak sistemin donanım uygulaması zordur ve sadece küçük problemleri çözebilmektedir (Willems & Rooda'ın 1994).

Foo ve diğerlerinin 1995 de yaptıkları çalışmada ATÇP'ni çözmek için Tank ve Hopfield ağı geliştirilerek ağın ölçekleme özellikleri araştırılmıştır. Özellikle, Tank-Hopfield doğrusal programlama ağı, kademeli fonksiyonların eklenmesiyle geliştirilerek, tamsayı ve doğrusal programlama yöntemi ile de karıştırılarak kullanılmıştır. Doğrusal bir enerji fonksiyonu kullanılarak, ikinci dereceden enerji fonksiyonlarını kullanan çoğu Hopfield ağıyla ilişkili geleneksel sorunlar önlenmiştir. Ayrıca Tank-Hopfield ağındaki nöronların kapsamlı hesaplamalar gerektirmemesi için önerilen yaklaşım işlem elemanları ve dirençli ara bağlantılar açısından da değiştirilmiştir (Foo ve diğerleri, 1995).

Sabuncuoğlu ve Gürgün'nün 1996 da yapmış oldukları çalışmada Hopfield ağları geliştirilerek paralel YSA adı verilen bir ağın tasarımı sunulmuştur. Çalışmada ayrıca işlerin sırasını ve pozisyonunu gösteren bir $n \times n$ nöron matrisi geliştirilmiştir. Tezgâh sayısı m olmak üzere, $m \times n \times n$ büyüklüğünde üç boyutlu bir matris de geliştirilmiştir. Ağ tasarımında aktivasyon fonksiyonu olarak sigmoid kullanılırken, çizelgelenen operasyonların en büyük tamamlanma süresi o ağın enerjisi olarak kullanılmıştır. Çalışmanın sonuç kısmında, önerilen ağın birçok ATÇP için optimum çözümler ürettiği açıklanmıştır (Sabuncuoğlu & Gürgün, 1996).

Yang ve Wang'ın 2000 yılında yayınladıkları çalışmada ATÇP çözmek için Kısıt Memnuniyeti Uyarlanabilir Sinir Ağı (Constraint Satisfaction Adaptive Neural Network) algoritması ile sezgisel algoritmaların birleştirildiği bir yaklaşım sunulmuştur. Bu çalışmada önerilen sinir ağı, problemin sırasını ve kaynak kısıtlamalarını kendi mimarisine kolayca eşleme ve bu kısıtlamaların ihlallerini gidermek için uygun çözümler

elde etme özelliğine sahiptir. Bu Çalışmada kullanılan sinir ağının kolay uyarlanabilir özelliği ve basit bir mimarisi olması diğer sinir ağlarından ayıran olumlu özellikleridir (Yang & Wang, 2000).

Yu ve Liang'ın 2001 yılında yaptıkları çalışmada ATÇP çözmek amacıyla YSA ile GA yöntemleri birlikte kullanılarak hibrit bir çözüm tekniği önerilmiştir. Çalışmada kullanılan Kısıtlamalı Sinir Ağı (Constraint Neural Network) sinir ağının yapısında kısıtlı bir ağ modeli mevcuttur. Bu yöntem, atölye başlangıç zamanlarını ve başlangıç sıralamasını bulmak için kullanılmıştır. Bu çalışma pratik uygulamalarla ve birçok benzetim çalışmalarıyla karşılaştırıldığında önerilen yöntem ile daha etkin sonuçlar elde edildiği açıklanmıştır (Yu ve Liang, 2001).

Özkazanç ve Anagün'ün 2001 yılında yaptıkları çalışmada, ATÇP için işin toplam tamamlanma zamanının en küçüklenmesine dayalı Hopfield sinirsel ağ yaklaşımı önerilmiştir. Kombinatorik yapıdaki bir problemin Hopfield sinirsel ağı ile çözümü beş adımda gerçekleştirilmiştir. İlk olarak, 4 iş 3 makine ile oluşturulmuş örnek problem makine sıraları ve işlem sürelerini gösteren örnek matrislerin üzerinden problemi temsil edecek desen oluşturulmuştur. İkinci olarak, problemin istenilen yönde yakınsamasını sağlayan parametre değerleri seçilerek enerji fonksiyonu belirlenmiştir. Üçüncü adımda, amaç fonksiyonuna ve kısıtlara karşılık gelen terimlerin katsayıları olan ve tamamıyla araştırılan probleme bagh olarak deger alan A, B ve C parametreleri belirlenmiştir. Dördüncü adımda, belirlenen parametrelere uygun değerler verilerek Hopfield sinirsel ağına uygun hareket denklemi belirlenmiştir. Son olarak ele alman test problemlerinin "rassal", "en kısa işlem süresi" ve "toplam işlem süresi en az kalan" olmak üzere üç farklı öncelik kuralı için çözümleri araştırılmıştır. Örnek çizelgeleme probleminin toplam tamamlanma zamanını içeren sonuçlar incelenerek, genel anlamda bir Hopfield sinir ağının nasıl yapılandırılacağı ve ağın hesaplama sürecine, davranışlarına ve performansına nelerin etki ettiği araştırılmıştır (Özkazanç & Anagün, 2001).

Feng ve diğerlerinin 2003 yılında yaptıkları çalışmada, bir üretim faaliyet çizelgelemesinin tasarlanması, geliştirilmesi ve uygulanması için çok katmanlı algılayıcı (Multi-Layered Perceptron) modeli önerilmiştir. Bu YSA modeli kurgulanırken, bir işin işlem sırası ve işlem zamanını bulmak için bir veri tabanından yararlanılmış yerel minimum çözümleri kontrol etmek için ise geriye yayma algoritması kullanılmıştır. Bu çalışmada sunulan üretim faaliyet çizelgeleme sistemi, gerçek bir üretim ortamında da test edilerek ve uygun çözümler elde edildiği açıklanmıştır. Ayrıca sistemin, üretim

faaliyet kontrollerinin, müşteri hizmetleri seviyelerinin ve yeteneklerinin geliştirilmesine yardımcı olduğu açıklanmıştır (Feng ve diğerleri, 2003).

Akkaya ve Gökçen 2006 yılında ATÇP için benzetim ve YSA tekniklerini kullanan bir sistem modeli geliştirilmiştir. Makine seçimi, malzeme taşıma sistemi seçimi ve sistemde kullanılacak öncelik kuralı seçimi için geliştirilen modelde geriye yayılma algoritması YSA modeli olarak kullanılmıştır. Ağın eğitmek ve örneklemeleri elde etmek için bilgisayar ortamında benzetim tekniği kullanılmıştır. Modellenen ağın önerdiği sonuçların benzetimi sonunda, performans kriterlerinin beklenen değerlerinden sapma miktarları hesaplanmış ve karar verme işlemi çizelgeler halinde verilmiştir. Performans ölçümünde ortalama akış zamanı, ortalama gecikme, maksimum tamamlanma zamanı ve makine merkezleri kullanım oranları kriter olarak kullanılmıştır (Akkaya & Gökçen, 2006).

El Bouri ve Shah 2006 yılında bir çalışma gerçekleştirmişlerdir. Bu çalışma ATÇP çözümünde, her bir makine için yerel olarak uygulanacak gönderim kurallarını seçen akıllı bir sistemi araştırmıştır. Rastgele oluşturulan problemler, beş makinede üç farklı gönderim kuralının optimal permütasyonları kullanılarak programlanmıştır. Bir YSA yöntemi geliştirilmiş ve bu ağ, iş sırasının istatistiksel bir karakterizasyonu ile kullanılacak olan gönderim kurallarının en iyi kombinasyonu arasında ilişki kurmak üzere eğitilmiştir. Bir kez eğitildikten sonra, YSA her makinede kullanılacak bir gönderme kuralını yeni problemler için tavsiye edebilmektedir. İki ağ, iş istasyonunda yapılan iş miktarını ve ortalama akış süresini en aza indirmek amacıyla ayrı ayrı eğitilmiştir. Test sonuçları, eğitilmiş ağların önerdiği gönderme kuralları kombinasyonlarının, her iki amaç için, tüm makinelerde ortak olan tek bir kuralın kullanılmasından daha iyi sonuçlar verdiğini göstermiştir (El-Bouri & Shah, 2006).

Weckman ve diğerleri 2008 yılında atölye tipi üretim sistemini programlamak için YSA ve GA yöntemlerini birleştiren bir sistem kullanmışlardır. Bu hibrit akıllı sistemde GA, bilinen bir kıyaslama problemine en uygun zamanlamaları oluşturmak için kullanılmıştır. Her optimal çözümde, bir işin planlanma sürecini içeren bilgiyi kullanarak bir karar elde edilir. Her karar, sevkiyat ihtiyacına (örn, en kısa işlem süresi) göre sınıflara ayrılmış bir dizi iş özelliğinin (örn, işlem süresi) bir fonksiyonu olarak modellenmiştir. Bu çalışmaya göre, YSA çizelgelemesi diğer yöntemlerle karşılaştırıldığında, problemin üzerinde daha iyi performans göstermesi YSA tabanlı planlamanın fizibilitesini kanıtlamaktadır (Weckman ve diğerleri, 2008).

Chaudhuri ve Kajal 2010 yılında, makine öğrenim paradigmasının çeşitli bileşenlerini kullanarak ATÇP çözmek için yeni bir bilgi tabanlı yaklaşım sunmuşlardır. GA yönteminin çoklu optimal çözümler sağlama yeteneği, iyi çözümlerin bilgi tabanını oluşturmak için kullanılmış ve Kaba Bulanık Çok Katmanlı Sinir Ağı Modeli (Rough Fuzzy Multi Layer Perception Neural Network Model) mevcut bilgi tabanında başarılı bir şekilde eğitilmiştir. Geliştirilen sinir ağı, zamanlayıcının genelleme yeteneklerini değerlendirmek için 10 rastgele oluşturulmuş 6x6 matris senaryosundan oluşan bir test problem seti kullanılmıştır. Ağ programlayıcı ile elde edilen sonuçlar, farklı bir makine öğrenme metodolojisi ile geliştirilen diğer programlayıcılarla karşılaştırarak bir değerlendirme yapılmıştır. Değerlendirme sonucunda, bu çalışmada geliştirilen YSA zamanlayıcısının, GA yöntemine en yakın ortalama sonuçlara sahip olduğu ve ATÇP çözümü için kayda değer bir performans sağladığı açılanmıştır (Chaudhuri & Kajal, 2010).

Zandieh ve Adibi 2010 yılında rastgele iş girişleri ve makine arızaları ile dinamik ATÇP için DKA yöntemine dayanan bir programlama yöntemi önermişlerdir. Zamanlama yönteminin etkinliğini ve verimliliğini arttırmak için, geri yayılma hatası öğrenme algoritmasına sahip bir YSA kullanılmıştır. Ayrıca DKA yöntemi, parametreleri problem durumuna göre herhangi bir yeniden programlama noktasında güncellemek için kullanılmıştır. Önerilen yöntem, literatürde yaygın olarak kullanılan bazı sevk kuralları ile karşılaştırılmıştır. Sonuçlar, çeşitli atölye koşullarında önerilen yöntemin daha yüksek verimliliğe sahip olduğu göstermiştir (Zandieh & Adibi 2010).

Yang ve diğerlerinin 2011 yılında yaptıkları çalışmada ATÇP için geliştirilmiş bir kısıtlama memnuniyeti uyarlamalı sinir ağı sunulmaktadır. Bu YSA, ATÇP için kısıtlama koşulları baz alınarak uygulanmıştır. Yapısı ve nöron bağlantıları, çözme işlemi sırasında ortaya çıkan gerçek zamanlı kısıtlama durumlarına göre uyarlanabilir. Bir araya getirdiği farklılığı, yakınsama gücünü artırmak, yakınlaşmasını hızlandırmak ve üretilen çözümlerin kalitesini arttırmak için, sinir ağı içerisinde bütünleştirilmiştir. ATÇP için bir dizi kıyaslamaya dayanan deneysel bir çalışma yapılmıştır. Geliştirilmiş Kısıtlama Memnuniyeti Uyarlamalı Sinir Ağının, hesaplama süresi ve üretildiği zamanlamaların kalitesi açısından orijinal kısıtlama tatmin edici uyarlamalı sinir ağını geride bıraktığını göstermektedir (Yang ve diğerleri, 2011).

Azadeh ve diğerleri 2012 yılında, ATÇP çözümü için, bilgisayar simülasyonu ve YSA kullanarak, her bir makine için bir dizi kuraldan en uygun sevkiyat kuralını seçmek amacıyla stokastik bir algoritma önerilmişlerdir. Önerilen algoritmanın, ATÇP

için önceki çalışmalara üç önemli şekilde katkıda bulunduğu açıklanmaktadır; Birincisi, bilgisayar simülasyonu ve YSA yöntemi, gönderim kurallarını seçmesi için ilk kez önerilmiştir. İkincisi, stokastik ortamdaki makineler için özdeş olmayan sevk kuralları dikkate alınmıştır. Üçüncüsü ise önerilen algoritma, tüm olası çözümleri değerlendirdiğinden, optimal çözümü bulabilmektedir. Önerilen algoritmanın performansı, bilgisayar simülasyon yöntemleri ile karşılaştırılmıştır. Beş ve altı makineli işyerleri için kapsamlı hesaplama sonuçları, literatürdeki çalışmalarla karşılaştırılarak önerilen algoritmanın üstünlüğü açıklanmıştır (Azadeh ve diğerleri, 2012).

Fnaiech ve diğerleri 2012 yılında, ATÇP çözmek için yeni bir Hopfield sinir ağı modeli önermişlerdir. Fakat bu çalışmada makine bakım süreleri de hesaba katılmıştır. Çalışmanın amacı, tamamlanma süresini en aza indirmek için dizi kısıtlamalarını, kaynak kısıtlamalarını ve erişilebilirlik kısıtlarını göz önünde bulundurarak ayarlanan farklı başlangıç zamanlarını belirlemektir. Deneysel örnekler, bu çalışmada önerilen modelinin küçük ve büyük problemler için iyi sonuçlar verdiğini göstermiştir (Fnaiech ve diğerleri, 2012).

Sache 2014 yılında, sistemdeki işlerin toplam ağırlıklandırılmış tamamlanma süresini buluşsal zaman yöntemini kullanarak en aza indirmeyi amaçlayan bir çalışma yapmıştır. Makul sürede yüksek kalitede yaklaşık bir çözüm veren Hopfield sinir ağı yöntem olarak kullanılmıştır. Kullanılan yöntemin temel avantajı, optimal çözümünü elde etmek için arama hızını arttırmaktır. Ayrıca bu yöntemin, iki iş ve üç makine veri kümesi problemleri için simülasyon sonuçları, çözünürlük hızı, çözümün kalitesi ve hesaplama süresinin azaltılmaması bakımından verimli olduğu gösterilmektedir. Bu çalışma da önerilen algoritmanın büyük boyutlu örneklere çok hızlı bir şekilde yüksek kaliteli çözümler bulabileceği simülasyon sonuçları ile açıklanmıştır (Sache, 2014).

Teymourifar ve Öztürk'ün, 2018 yılında yapmış oldukları çalışmada, esnek ATÇP için uygun komşular oluşturmak amacıyla bir hibrit yöntem önerilmiştir. Önerilen yaklaşımda, öncelikle, bir başlangıç çözümü üretilir ve daha sonra sol boşluk buluşsallık kullanılarak, yol boşlukları giderilir. Kritik yol ve kritik blok kavramlarına dayanarak, elde edilen çözüm için altı komşu yaklaşımı inşa edilir. Her bir komşunun yaratılmasından sonra, bir sinir ağı çalıştırılır ve sinir ağları problemin sınırlarını kontrol eder. Elde edilen komşu uygunsa, kaydedilir. Eğer uygun değilse, sinir ağı onu uygun bir çözüme dönüştürmeye çalışır. Aksi takdirde, yinelemelerin belirli bir sayısından sonra, herhangi bir uygun komşu oluşturulamazsa sinir ağı yinelemeyi durdurur. Daha sonra bu adımlar diğer oluşturulan komşular için tekrarlanır. Bu kısıtlamaya dayalı süreç, etkili ve

çeşitli bir arama sağlar. Önerilen yaklaşım ile literatürdeki yöntemler arasındaki karşılaştırma, önerilen yaklaşımın daha iyi komşular oluşturduğunu göstermiştir (Teymourifar & Öztürk, 2018).



BÖLÜM V

ATÖLYE TİPİ ÇİZELGELEMEDE YAPAY SİNİR AĞI İLE UYGULAMA

1.1. Taillard'ın Atölye Tipi Çizelgeleme Problemi

ATÇP literatürde çeşitli yöntemler kullanılarak çözümlenmeye çalışılmıştır. Taillard'ın 1993 yılında yayımlamış olduğu “Benchmarks for Basic Scheduling Problems” adlı çalışmasında üretim çizelgeleme problem tipleri ele alınarak araştırmacıların karşılaştırma yapabilmelerine olanak sağlayan problem setleri veri tabanı oluşturmuştur (Taillard, 1993).

Bu tez çalışmasının uygulama sürecinde Taillard'ın veri setinde yer alan 15 iş ve 15 makine, 20 iş ve 20 makine, 30 iş ve 15 makine boyutlu işlemlerin çizelgelenmesinde en kısa toplam tamamlanma zamanını (makespan) bulmak için Visual Studio 2019 C# programı ile YSA algoritması kodlanmıştır. Taillard'ın çalışmasında paylaşılan atölye tipi çizelgeleme verileri kullanılarak elde edilen sonuçlar, bu verileri kullanarak yapılan akademik yayınların sonuçları ile karşılaştırılmıştır.

ATÇP, belirli sayıda işin, belirli sayıda makinelerde, her işin belli bir rotası olacak şekilde, operasyonel sürecin planlanması problemidir. Bu tip bir sistemi çizelgeleyebilmek için öncelikle modelini oluşturmak gerekmektedir. Bir makine, operatör ya da iş istasyonu bir operasyonu tamamlayınca bir sonraki adımda ne yapılacağı konusunda bir karar verilmesi gerekmektedir. Bu karar bir çizelgeleme kuralına göre alınır. Bu çalışmada, en kısa işlem süresi olana öncelik vermek, en uzun işlem süresi olana öncelik vermek, ilk gelen ilk hizmet alır, son gelen ilk hizmet alır ve rastgele atama olmak üzere beş çizelgeleme yöntemi kullanılmıştır.

1.2. Taillard'ın Veri Seti ile Atölye Tipi Çizelgelemede Yapay Sinir Ağı Uygulaması

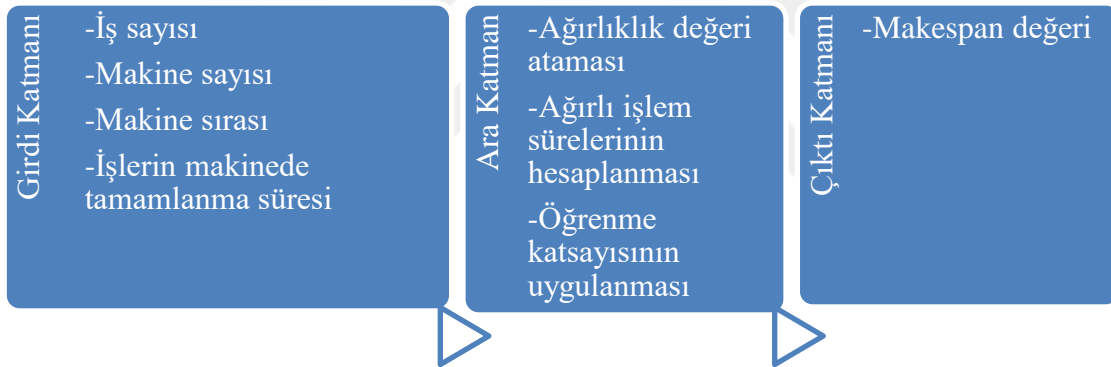
Bu tez çalışmasında önerilen YSA modeli üç katmandan oluşmaktadır.

Giriş katmanında girdi verileri kullanılmıştır. Girdi verileri iş sayısı, makine sayısı, makine sıraları ve işlerin makinede tamamlanma süresi olmak üzere dört başlıktan oluşmaktadır.

YSA modelin çıkış katmanında işlerin toplam tamamlanma süresi (makespan) verileri elde edilmektedir.

YSA modelinin ara katmanında; her operasyonla ilişkili bir ağırlık faktörü tanımlanarak işlem süreleri (makespan) yerine ağırlıklı işlem süreleri kullanılmaktadır. Bu çalışmada genellikle en kısa işlem süresi olana öncelik verildiğinde daha iyi sonuçlar elde edilmiştir. Bu çizelgeleme baz alındığında algoritmada ilk ağırlık değeri 1 olarak atanmaktadır. Sonrasında ise ağırlık değeri rastgele atanarak makespan ile çarpılır. Çarpımdan elde edilen sonuçlar küçükten büyüğe doğru sıralanır. Sonrasında bu sıra takip edilerek işler belirlenmiş rotaları dahilinde makinelere atanır. Tüm makinelere işler atandıktan sonra ilk elde edilen sonuç en iyi sonuç olarak kayıt edilir. Bu işlemler ağırlıklar değiştirilerek diğer iterasyonlarda tekrarlanır. 1000 iterasyona ulaşıncaya kadar her iterasyon sonucu en iyi sonuç ile karşılaştırılır ve çıktı verisinde iyileşme olduğunda en iyi değer güncellenir. Bu çalışmada genellikle en iyi sonuçlar 1000 iterasyonda elde edilmiştir.

Bu çalışmada kullanılan YSA modelinin katmanlarında bulunan veriler şekil 7 de açıklanmıştır.



Şekil7: YSA Modelinin Katmanlarında Bulunan Veriler

Öğrenme Katsayısı (a) ile ağırlıklar güçlendirilir. Öğrenme Katsayısı; her yinelemede ağırlık değişim derecesini kontrol eden parametrelerden biridir. Aramanın ayrıntı düzeyi a parametresi kullanılarak kontrol edilebilir. Öğrenme katsayısı rastgele şekilde, 0.01 ile 0.99 değerleri arasında seçilmektedir. Bu çalışmada, 0.05'lik bir oranın tüm iterasyonlar için ortalama bir değer olarak çalıştığı belirlenmiştir. 10 iterasyon sonrasında sonuç değişmediğinde öğrenme katsayısını 0,01 attırarak sonuç almaya devam edilmektedir.

1000 iterasyona ulaşana kadar program çalıştırılır. İterasyonlar sonucunda ilk sonuçtan daha iyi bir makespan değeri elde edildiğinde, bu sonuç en iyi olarak kabul

edilerek ilk kayıt edilen sonuç ile değiştirilir. Belirli sayıda yineleme ile program test edilir. Belirlenen iterasyon içindeki en iyi ağırlıklar kaydedilerek, en iyi işlem süresi görüntülenir.

Parametreler;

- n İş Kümesi {1, 2, ... n}
- m Makineleri m Seti {1, 2, ... m}
- $Morij$ Makine Sırası
- Pij Operasyonun İşlem Süresi
- Wij Operasyon İşlemi ile İlişkili Ağırlık
- $WPij$ Ağırlıklı İşlem Süresi
- k Yineleme Sayısı
- a Öğrenme Katsayısı
- BMS En İyi Toplam İşlem Süresi (Makespan)
- $BWij$ En İyi Ağırlıklar

YSA yönteminin matematiksel modellemesi şekil 8 de açıklanmıştır.

$$\begin{aligned}
 & n > 0, m > 0, i > 0, j > 0, \\
 & \underline{P_{ij}} > 0, \underline{Mor_{ij}} > 0, 0 < \underline{w_{ij}} \leq 1, \underline{WP_{ij}} > 0 \\
 & \sum_{i=0, j=0}^{i=n, i=m} P_{ij} \\
 & \sum_{i=0, j=0}^{i=n, i=m} W_{ij} * P_{ij} \\
 W_{ij_{k+1}} = & \begin{cases} RND > 0.05, W_{ij_k} + RND * a * P_{ij} \\ RND \leq 0.05, W_{ij_k} - RND * a * P_{ij} \end{cases} \\
 & \text{Amaç: } \underline{\min} BMS
 \end{aligned}$$

Şekil 8: YSA'nın Matematiksel Modellemesi

Yaklaşımımızı algoritma şeklinde açıklayalım;

1. Adım: İşlerin makinelere yerleşimi
 - Her bir makinede toplam iş süreleri hesaplanır (Pij)
 - En kısa işlem süresi olana öncelik verilerek yapılan çizelgeleme de süreler

küçükten büyüğe sıralanır. Küçük olana öncelik verilir.

- En uzun işlem süresi olana öncelik verilerek yapılan çizelgeleme de süreler büyükten küçüğe sıralanır. Büyük olana öncelik verilir.
- İlk gelenin ilk hizmet aldığı çizelgelemede makine sırası önde olan ilk hizmet alır.
- Son gelenin ilk hizmet aldığı çizelgelemede makine sırası son olan ilk hizmet alır.
- İşler makinelere rastgele atanır.

2. Adım: Ağırlıklı işlem sürelerinin hesaplanması

- İlk ağırlık değeri 0.01 ile 1.00 arasında rastgele üretilir (RND).
- Makespan değeri ile üretilen ilk ağırlık değeri çarpılır. Sonuç ağırlıklı işlem süresi olarak kabul edilir.
- $(WP_{ij} = W_{ij} * P_{ij})$ formülü ile ağırlıklı işlem süresi hesaplanır (Agarwal ve diğerleri, 2006).

3. Adım: Makine sırasına göre işleri yerleştirme

- Ağırlıklı işlem süreleri en kısa zamandan en uzun zamana sıralanır.
- Çakışma olan zaman dilimlerine boş zaman (idle time) eklenir.
- Belirlenen makine sırasına göre işler uygun zaman dilimlerine yerleştirilir.

4. Adım: Öğrenme Stratejini Uygulama ve Ağırlıkları Değiştirme

- İlk ağırlık 1 alınır. 2. ağırlık ve sonrası random seçilir.
- 10 kez aynı sonuç elde edilirse öğrenme katsayısı 0.01 arttırılır
- Eğer $RND > 0.5$ ise $(W_{ij})_{k+1} = (W_{ij})_k + RND * a * P_{ij}$.
- Eğer $RND \leq 0.5$ ise $(W_{ij})_{k+1} = (W_{ij})_k - RND * a * P_{ij}$.

Formülleri ile ağırlıklar değiştirilir. (Agarwal ve diğerleri, 2006)

5. Adım: Makespan değerinin hesaplanması

- İlk elde edilen sonuç en iyi sonuç olarak kayıt edilir belli bir iterasyona (genellikle 1000 iterasyon) ulaşana kadar ağırlıklı işlem süreleri ile tekrarlanır. (1000 iterasyonda genellikle 700 tane sonuç elde edilebilmektedir.)

- İterasyonlar sonucunda ilk sonuçtan daha iyi sonuç elde edildiğinde, bu sonuç en iyi sonuç olarak kabul edilerek kayıt yenilenir.
 - Tüm iterasyonlar tamamlandığında en iyi sonuç çıktı değeri olarak sunulur. En iyi sonucun mevcut ağırlıkları da en iyi ağırlıklar olarak kaydedilir. (BWij ve BMS)
6. Adım: Çözümü görüntüle
- (BMS) ve (BWij) değerleri alınır ve çıktı olarak sunulur.

1.3. Atölye Tipi Çizelgelemede Taillard Veri Setini Kullanan Çalışmalar

D.Y. Sha ve C.Y. Hsu 2006 yılında ATÇP için PSO yöntemini öneren ve bu yöntemin kalitesini arttırmak için TA yönteminden faydalanan bir çalışma yapmışlardır. Bu çalışmada elde edilen sonuçlar, değiştirilmiş PSO algoritmasının, orijinal tasarımdan daha iyi performans gösterdiğini açıklanmıştır. PSO ile TA yöntemlerini barındıran bir hibrit algoritmanın diğer geleneksel metasezgisel yöntemlerden daha iyi olduğu vurgulanmıştır. Taillard test problemlerine uygulanan bu hibrit yöntem, temel olarak ilk çözümünden etkilenecek şekilde çözüm üretmektedir. PSO yöntemindeki sürecin temel amacı, TA yöntemine iyi ve çeşitli başlangıç çözümleri sağlamaktır. Bu hibrit yönteminin hem TA hem de PSO yöntemlerinden daha iyi performans gösterdiğini ve ortalama farkının PSO'dan %0.356 daha düşük olduğunu çalışma sonuçları ile ortaya konulmuştur. Hibrit yöntemin maksimum hesaplama süresinin 3 ile 103 sn arasında olduğu ve hesaplama süresinin %99'unun hibrit yöntemin yerel arama sürecine harcandığı bilgisi verilmiştir. Yineleme sayısını azaltarak, yerel bir arama prosedürü uygulayan parçacıkların hesaplama süresi azaltılmadığından, bu hibrit yöntem ile yapılan hesaplamalar 103.sn deki yinelemeden sonra da sonlandırılmıştır (Sha & Hsu, 2006).

M. Chandrasekaran ve diğerleri 2007 yılında ATÇP için toplam işlem tamamlanma süresini minimize etmek amacıyla hesaplamalı YBS algoritması yöntemini öneren bir çalışma yapmışlardır. Bu çalışmada önerilen YBS algoritmasının, Balas ve Vazacopoulos'un en iyi vites değiştirici tıkanıklığı prosedürü ile birlikte TA kaydırması yönlendirme prosedüründen daha iyi sonuçlar veren etkili bir algoritma olduğu gösterilmiştir. Bağışıklık sisteminin ilkelerine dayanan verimli YBS yöntemi, iki prensip ile kullanılmıştır. (Bağışıklık sisteminin ilkeleri: Klonal seçim prensibi ve Benzeşim olgunlaşma prensibi) Klonal seçim prensibinde, her program (antikor), bu antikorun benzeşim değerini ifade eden bir üretim değerine sahiptir. Her programın benzeşim değeri

benzeşim fonksiyonundan hesaplanır. Benzeşme işlevi, benzeşim = $1 / \text{makepan}$ olarak tanımlanmıştır. Klonal seçim prensibinde ise test mutasyon ve ikili değişim mutasyonu olmak üzere 2 yöntem kullanılmıştır. Ters mutasyon: Bir dizi s için, i ve j 'nin dizilerinde rastgele iki pozisyon seçilmesine izin verir. i ve j pozisyonları arasındaki iş sırasını ters çevirerek bir s komşusu elde edilir. Mutasyona uğramış dizinin tamamlayıcı değeri (ters mutasyondan sonra), orijinal diziden daha küçükse, mutasyona uğramış olan, orijinal olanın yerine depolanır. Aksi takdirde, sıra tekrar rastgele çiftli değişim mutasyonu ile mutasyona uğratılır. İkili akıllı değişim mutasyonunu ise; s komşusu aynı yöntem ile elde edildikten sonra mutasyona uğramış dizinin toplam işlem süresi değeri orijinal dizininkinden küçükse, mutasyona uğramış olan orijinalin yerine saklanır. Algoritmanın iki mutasyon prosedüründen sonra daha iyi bir sekans bulamadığı durumda, orijinal sekansı (oluşturulan klon) depolanmıştır. Reseptör düzenleme klonlama ve mutasyon işlemlerinden sonra, antikor popülasyonundaki antikorların yüzdesi elimine edilir ve rastgele oluşturulmuş antikorlar onlarla değiştirilir. Bu mekanizma, toplam arama alanındaki yeni arama bölgelerine karşılık gelen yeni programlar bulmayı sağlamaktadır. YBS yöntemi ile elde edilen bulgular, aynı problemleri test eden literatürdeki çalışmalarla karşılaştırıldığında önerilen YBS algoritmasının daha etkili bir problem çözme tekniği olduğunu açıklanmıştır (Chandrasekaran ve diğerleri, 2007).

2008 yılında C.Y. Zhang ve diğerleri tarafından yapılan çalışmada, son yıllarda etkili bir algoritmik yaklaşım olarak kullanılmakta olan TA yöntemi, ATÇP için önerilmiştir. Ancak, TA yaklaşımı ile bulunan çözümlerin kalitesi ilk çözüme bağlıdır. Bu sorunun üstesinden gelmek için simüle edilmiş tavlama ve TA stratejisini birleştiren sezgisel tarama yaklaşımı geliştirilmiştir. Önerilen hibrit yaklaşımın ana ilkesi, çözüm uzayı içindeki seçkin çözümleri bulmak için simüle edilmiş tavlamanın kullanılmasıdır. Bu hibrit algoritma taillard veri setlerinde test edilmiş ve diğer yaklaşımlarla karşılaştırılmıştır. Hesaplama sonuçları, önerilen algoritmanın makul hesaplama zamanlarında yüksek kaliteli çözümler elde edebileceğini göstermektedir. Önerilen hibrit algoritmada, iyi çözümleri bulmak için simüle edilmiş tavlama prosedürünü kullanan güçlü bir çeşitlendirme stratejisi, çekirdek TA ile donatılmıştır. Çekirdek TA, yoğunlaştırılmış aramayı çözüm alanının diğer bölgelerini keşfetmeye yönlendirir. Daha doğrusu, rastgele bir başlangıç çözümünden başlayarak, hibrit algoritmanın çekirdeği TA prosedürünü yürütür ve arama prosedüründe simüle edilmiş tavlama tarafından bulunan yeterince iyi çözümler izlenir. Simüle edilmiş tavlama prosedürü ile bulunan bu iyi çözümler, elit çözümleri yığını L 'de depolanır. Her yeni iyi çözümler, keşfedildiğinde çözümleri

yığımları L'nin üzerine eklenir. Daha sonra, bu tür çözümler, önceden belirlenmiş sayıda yinelemede yoğunlaştırılmış bir aramanın gerçekleştirildiği yeni yerleşik çözümler olarak L yığından çıkartılabilir. Uygun sıcaklık T göz önüne bulundurulmalı, elit çözümler istifindeki çözümler tükenmemelidir. Algoritma, toplam yineleme sayısına ulaştığında sona erer ve elde edilen çözümün optimal olduğu kabul edilir (Zhang ve diğerleri, 2008).

M.M. Nasiri ve F. Kianfar 2010 yılında yaptıkları çalışmada, TA ve yola yeniden bağlanma ile karmaşılaştırılmış bir dağılım araştırması algoritması kullanılarak çizelgeleme problemine optimal makespan değerleri bulunmuştur. Dağılım araştırması, var olanları sistematik bir şekilde birleştirerek yeni çözümler üreten evrimsel bir yöntemdir. Bu yöntemin ilk adımında, bir başlangıç popülasyonu üretilmektedir. İkinci adımda, referans seti oluşturulmuştur, eski düşük makepanlı b1 solüsyonları içeren 1.setin ve yüksek çeşitlilikteki solüsyonları içeren 2.setin çözümlerine referans çözümler denilmiştir. Her biri iki referans çözümü içeren alt kümeler oluşturulmuş ve her bir alt kümenin iki çözümü birleştirip yola yeniden bağlanma algoritması kullanılarak yeni bir çözüm üretilmiştir. TA yöntemi çözümlere uygulanır ve elde edilen çözümler popülasyona eklenir. Taillard veri setini kullanan bu çalışma ile optimal sonuçlar elde edilmiştir (Nasiri & Kianfar, 2010).

H. Gao ve diğerlerinin 2014 yılında yapmış oldukları çalışmada, ATÇP için hibrit metasezgisel bir yöntem önerilmektedir. Bu yöntem, geliştirilmiş PSO ve bir TA algoritmasından yararlanmaktadır. Bu çalışmada, değerli bir bölgeyi tamamen taramaya dayalı olarak, araştırma yeteneğini arttırmak için PSO yönteminde bir denge stratejisi getirilmiştir. Geliştirilmiş PSO, küresel alanda daha iyi bir araştırma yapmak için TA yöntemine çeşitli başlangıç çözümleri sağlamaktadır. Ayrıca ATÇP için daha iyi sonuçlar elde etmek amacıyla yeni bir yerel arama stratejisi sunulmuştur. Sürekli bir uzayda, bir çözümü ayrı bir program çözümüne bağlamak için kodlama ve kod çözme şeması ile geliştirilmiş olan PSO ve TA yöntemi, daha iyi çözüm arayışlarını yoğunlaştırmak için tasarlanmıştır. Taillard ATÇP veri seti kullanan bu çalışmada önerilen PSO ve TA hibrit algoritmasının etkin bir yöntem olduğunu gösterilmiştir (Gao ve diğerlerinin, 2014).

B. Peng ve diğerlerinin 2015 yılında yapmış oldukları çalışmada, ATÇP için TA ve yeniden yol değiştirme (path relinking) yöntemlerini kullanan hibrit bir algoritma önerilmiştir. Bu algoritma, mesafelere dayalı bir yol çözümü yapım prosedürü olarak çözümlerin belirlenmesi için özel bir mekanizma referans çözümü gibi bir dizi ayırt edici özelliğe sahiptir. Ayrıca bu yöntem, bir başlangıç çözümlerinden, bir rehber çözümlere kurulan yörünge üzerinde optimal çözümler üretmek için kullanılan bir yol yeniden

bağlama yöntemi ve üretilen optimal çözümü geliştiren bir TA prosedürü arasında sürekli olarak çalıştırılmaktadır. Önerilen hibrit yöntemde, ilk popülasyon şu şekilde oluşturulur: Sıfırdan başlayarak, rastgele uygulanabilir bir çözüm üretilir ve daha sonra iyileştirme yöntemi kullanarak çözüm optimize edilir. En uygun çözüm bulunursa veya belirli bir TA yineleme için en iyi objektif değer iyileştirilmediyse, TA sonuçlandırılır. Önerilen hibrit yöntem, Taillard'ın ATÇP veri setine uygulanmıştır ve elde edilen sonuçlar yöntemin etkinliği göstermektedir (Peng ve diğerlerinin 2015).

R. Bürgy nin 2017 de yapmış olduğu çalışmada, ATÇP için uygulanabilecek DKA yöntemi önerilmiştir. Uygulanabilir komşu çözümler, bir işin verilen bir çözümden çıkarılması ve bir komşu konumuna yeniden yerleştirilmesiyle üretilir. Bir anlamda bu komşu nesil, TA yöntemine dahil edilir. Son yinelemelerinin girişlerini depolamak için bir tabu listesi L kullanılır. Başlangıçta, L listesi boştur. Bir yinelemede, komşu çözümlerden bir S değeri seçilir. Bu seçilen S değerinin, L'ye eklenilip eklenilmeyeceği, formüsel seçimler yapılarak değerlendirilir. Son olarak, en düşük objektif değere sahip bir komşu seçilmektedir. Mevcut çözümü potansiyel olarak iyileştiren komşular üretmek için, kritik yaylar konsepti kullanılmıştır. Herhangi bir S seçimi için, bir işin üretim sürecinin en son tamamlanma süresi olan makespan hedefi ile, bir işlemi temsil eden bir düğüme giden en uzun yol mantığı düşünülmüş çözüm rotası kritik yol olarak adlandırılmıştır. Kritik bir yolun ayrık yaylarına kritik yaylar denilmiştir. Genellikle sadece en son tamamlanma süresine sahip olan bir işlem, objektif değere katkıda bulunmaktadır, ancak bazı (alt kümeler) işlemlerin belirli bir seçim için nesnel değeri belirlediği belirtilmiştir. Taillard'ın veri setine uygulanan DKA ve TA hibrit yöntemi ile elde edilen sonuçların etkin olduğu açıklanmıştır (Bürgy, 2017).

BÖLÜM VI

SONUÇ

Atölye tipi üretimde ürün çeşitliğinin fazla olması, gelen siparişlerin küçük miktarda olması yüksek planlama konsantrasyon gerektirmektedir. Kaynakların verimli kullanılması zamansal ve finansal tasarruf için önemlidir. Ürün niteliğinin artırılmasını ve verimli kaynak kullanımı için çizelgemenin etkinliği büyük önem arz etmektedir. İşlerin ve makinelerin sayısı arttıkça ATÇP çözmek zorlaşmakta ve matematiksel yöntemler gibi EİT yöntemleri yetersiz kalmaktadır. İyi bir çizelgeleme yapmak için son yıllarda metasezgisel yöntemlerden faydaniılmaktadır. Bu tür yöntemlerle hızlı sonuç alınabilmekte ve optimal ya da optimale yakın çözümler elde edilebilmektedir.

Bu tez çalışmasında, ATÇP için çözüm aranmıştır. Üretim çizelgeleme ve optimizasyon amacı ile kullanılan yöntemler araştırılarak, çözüm yaklaşımları sınıflandırılıp açıklanmıştır. Çözüm yaklaşımlarından son yıllarda kullanılan metasezgisel yöntemler araştırılarak literatür çalışmaları ile birlikte açıklanmıştır. Çalışmanın uygulama sürecinde kullanılan YSA yöntemi tarihçesi, yapısı, katmanları ve algoritması anlatılmıştır. ATÇP çözümünde YSA yöntemini kullanan çalışmalar ve yöntem modellemeleri incelenmiştir. Taillard'ın 1993 yılında yapmış olduğu üretim çizelgeleme problemlerine yönelik çalışmanın atölye tipi çizelgeleme verilerine bu çalışmada geliştirilen YSA yöntemi uygulanmıştır. Ayrıca son yıllarda Taillard verilerini kullanan çalışmalar ve sonuçları incelenerek bu çalışmada elde edilen sonuçlar ile karşılaştırılmıştır. Taillard'ın 15 iş ve 15 makine, 20 iş ve 20 makine problemlerinde elde edilen en iyi sonuçlar ile (Taillard, 1993) bu çalışma da geliştirdiğimiz YSA yöntemi ile alınan sonuçlar karşılaştırıldığında yöntemimizin daha iyi sonuçlar verdiği gözlemlenmiştir. 30 iş ve 15 makine probleminde ise optimal sonuçlara yakın sonuçlar elde edilmiştir. Literatür çalışmaları incelendiğinde genellikle hibrit yöntemler kullanarak ve yüksek iterasyon sayısı ile optimal çözümler elde edildiği gözlemlenmiştir. Bu çalışmada önerdiğimiz YSA yöntemi literatürde kullanılan yöntemlere göre, kolay adapte olabilen, kısa sürede optimal ya da optimale yakın hızlı çözümler üreten, etkin ve pratik bir yöntem olduğu için tercih edilebilir. Özellikle küçük ölçekli ATÇP için YSA yöntemi ile hızlı ve optimal sonuçlar elde edilmektedir.

Taillard'ın üretim çizelgeleme problemlerine yönelik çalışmasının atölye tipi çizelgelemede 15 iş ve 15 makine, 20 iş ve 20 makine, 30 iş ve 15 makine verilerine göre

elde edilen sonuçlar (Taillard, 1993) ile bu tez çalışmasında kullanılan YSA yöntemi ile 1000 iterasyonda elde edilen sonuçlar tablo 7 de sunulmuş ve karşılaştırılmıştır.

Tablolarda iyileşme oranları hesaplanırken 2019 YSA ile elde edilen sonuçtan literatür çalışmanın sonucu çıkartıp (farkı alınır) elde edilen fark 100'e bölünerek yüzde oranı hesaplanmıştır. Pozitif iyileşme oranı en iyi sonuca yüzde kaç yaklaşıldığını, negatif iyileşme oranı ise yüzde kaç daha iyi sonuç elde edildiğini göstermektedir.

Tablo 7

YSA Yöntemi ile Elde Edilen Sonuçların Taillard'ın Sonuçlarıyla Karşılaştırılması

	YSA (2019)	SÜRE	TAİLLARD	İYİLEŞME ORANI %
TA01 15 · 15	1235	01:01:06	1231	0,04
TA02 15 · 15	1244	01:01:01	1244	0
TA03 15 · 15	1218	01:04:24	1222	-0,04
TA04 15 · 15	1217	01:00:55	1181	0,36
TA05 15 · 15	1224	01:00:57	1233	-0,09
TA06 15 · 15	1238	01:01:07	1243	-0,05
TA07 15 · 15	1261	01:01:02	1228	0,33
TA08 15 · 15	1432	00:58:06	1220	2,12
TA09 15 · 15	1398	00:57:08	1282	1,16
TA10 15 · 15	1398	00:56:40	1259	1,39
TA11 20 · 20	1658	01:02:47	1663	-0,05
TA12 20 · 20	1636	01:03:07	1626	0,1
TA13 20 · 20	1559	01:02:24	1574	-0,15
TA14 20 · 20	1654	01:02:54	1660	-0,06
TA15 20 · 20	1691	01:03:16	1598	0,93
TA16 20 · 20	1662	01:03:09	1657	0,05
TA17 20 · 20	1705	01:02:29	1704	0,01
TA18 20 · 20	1620	01:22:20	1626	-0,06
TA19 20 · 20	1682	01:22:26	1629	0,53
TA20 20 · 20	1599	01:21:56	1614	-0,15
TA21 30 · 15	2597	01:21:36	1770	8,27
TA22 30 · 15	2529	01:21:42	1841	6,88
TA23 30 · 15	2579	01:21:55	1832	7,47
TA24 30 · 15	2693	01:22:01	1851	8,42
TA25 30 · 15	2375	01:22:44	2007	3,68
TA26 30 · 15	2460	01:21:43	1844	6,16
TA27 30 · 15	2672	01:21:27	1815	8,57
TA28 30 · 15	2499	01:21:59	1700	7,99
TA29 30 · 15	2341	01:21:52	1811	5,3
TA30 30 · 15	2656	01:21:41	1720	9,36

6.1. Çalışmanın Sonuçlarının Literatür ile Karşılaştırılması

Tablo 8

Son Yillarda Yapılan Çalışmalar ile Bu Çalışmanın Yöntem Karşılaştırması

	Çalışmanın Yöntemi	İterasyon sayısı	Çalışmanın Yılı
D.Y. Sha ve C.Y. Hsu	PSO ve TA	1000	2006
M. Chandrasekaran ve Diğerleri 2007	YBS	1500	2007
C.Zhang ve Diğerleri	TB ve TA	10.000	2008
M.M.Nasiri ve F.Kianfar	Yol değiştirme ve dağıtım araştırması algoritmaları ile geliştirilmiş bir TA yöntemi	5.000.000-10.000.000	2010
Hao Gao ve Diğerleri	PSO ve TA	100	2014
Bo Peng ve Diğerleri	Yol değiştirme algoritması ile zenginleştirilmiş TA yöntemi	500-12.500	2015
Reinhard Bürky	DKA ve TA	İyileşme olmadığında iterasyon sonlandırılır	2017
İncilay Yıldız (bu tez çalışması)	YSA	1000	2019

Bu tez çalışmasında tablo 9 da açıklanan literatür çalışmalarından daha basit bir algoritma kullanılarak ve daha az iterasyon sayısı ile optimal sonuçlara yakın çözümler üretilmiştir. Çizelgelemede zamanlama açısından avantajlı olan YSA yöntemi 30 iş ve makine kapasiteli problemlere kısa zamanda optimal sonuçlar üretebilmektedir. İterasyon sayısı arttırılarak ve ağın öğrenme süreci geliştirilerek yüksek makine ve iş sayısına sahip problemler için de iyi sonuçlar elde edilebilir. Ayrıca önerilen YSA yöntemi

algoritmasının kısıtları düzenlenerek diğer çizelgeleme problem türlerine de kolaylıkla uygulanabilir.

Tablo 9 da Sha ve Hsu'nun 2006 yılındaki çalışması ile bu çalışmada önerilen YSA yöntemi ile elde edilen sonuçlar karşılaştırılmıştır.

Tablo 9

Sha ve Hsu'nun 2006 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması

	2006 Sha & Hsu	İyileşme Oranı %	2019 YSA
TA01 15 · 15	1231	0,04	1235
TA02 15 · 15	1244	0	1244
TA03 15 · 15	1218	0	1218
TA04 15 · 15	1175	0,42	1217
TA05 15 · 15	1224	0	1224
TA06 15 · 15	1238	0	1238
TA07 15 · 15	1228	0,33	1261
TA08 15 · 15	1217	2,15	1432
TA09 15 · 15	1274	1,24	1398
TA10 15 · 15	1249	1,49	1398
TA11 20 · 20	1658	0	1658
TA12 20 · 20	1614	0,22	1636
TA13 20 · 20	1559	0	1559
TA14 20 · 20	1654	0	1654
TA15 20 · 20	1616	0,75	1691
TA16 20 · 20	1662	0	1662
TA17 20 · 20	1690	0,15	1705
TA18 20 · 20	1617	0,03	1620
TA19 20 · 20	1634	0,48	1682
TA20 20 · 20	1589	0,1	1599
TA21 30 · 15	1766	8,31	2597
TA22 30 · 15	1823	7,06	2529
TA23 30 · 15	1818	7,61	2579
TA24 30 · 15	1844	8,49	2693
TA25 30 · 15	2007	3,68	2375
TA26 30 · 15	1825	6,35	2460
TA27 30 · 15	1795	8,77	2672
TA28 30 · 15	1681	8,18	2499
TA29 30 · 15	1796	5,45	2341
TA30 30 · 15	1698	9,58	2656

Tablo 10 da Chandrasekaran ve diğerlerinin 2007 yılındaki çalışması ile bu çalışmada önerilen YSA yöntemi ile elde edilen sonuçlar karşılaştırılmıştır.

Tablo 10

Chandrasekaran ve Diğerlerinin 2007 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması

	2007 Chandrasekaran ve diğerlerinin	İyileşme Oranı %	2019 YSA
TA01 15 · 15	1231	0,04	1235
TA02 15 · 15	1244	0	1244
TA03 15 · 15	1206	0,12	1218
TA04 15 · 15	1170	0,47	1217
TA05 15 · 15	1215	0,09	1224
TA06 15 · 15	1210	0,28	1238
TA07 15 · 15	1223	0,38	1261
TA08 15 · 15	1187	2,45	1432
TA09 15 · 15	1297	1,01	1398
TA10 15 · 15	1241	1,57	1398
TA11 20 · 20	1649	0,09	1658
TA12 20 · 20	1627	0,09	1636
TA13 20 · 20	1556	0,03	1559
TA14 20 · 20	1624	0,3	1654
TA15 20 · 20	1580	1,11	1691
TA16 20 · 20	1672	-0,1	1662
TA17 20 · 20	1688	0,17	1705
TA18 20 · 20	1602	0,18	1620
TA19 20 · 20	1583	0,99	1682
TA20 20 · 20	1573	0,26	1599
TA21 30 · 15	1764	8,33	2597
TA22 30 · 15	1824	7,05	2529
TA23 30 · 15	1829	7,5	2579
TA24 30 · 15	1841	8,52	2693
TA25 30 · 15	2009	3,66	2375
TA26 30 · 15	1825	6,35	2460
TA27 30 · 15	1796	8,76	2672
TA28 30 · 15	1699	8	2499
TA29 30 · 15	1803	5,38	2341
TA30 30 · 15	1684	9,72	2656

Tablo 11 de Zhang ve diğerlerinin 2008 yılındaki çalışması ile bu çalışmada önerilen YSA yöntemi ile elde edilen sonuçlar karşılaştırılmıştır.

Tablo 11

Zhang ve Diğerlerinin 2008 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması

	2008	İyileşme Oranı %	2019
	Zhang ve diğerleri		YSA
TA01 15 · 15	1231	0,04	1235
TA02 15 · 15	1244	0	1244
TA03 15 · 15	1218	0	1218
TA04 15 · 15	1175	0,42	1217
TA05 15 · 15	1224	0	1224
TA06 15 · 15	1238	0	1238
TA07 15 · 15	1228	0,33	1261
TA08 15 · 15	1217	2,15	1432
TA09 15 · 15	1274	1,24	1398
TA10 15 · 15	1241	1,57	1398
TA11 20 · 20	1644	0,14	1658
TA12 20 · 20	1600	0,36	1636
TA13 20 · 20	1560	-0,01	1559
TA14 20 · 20	1646	0,08	1654
TA15 20 · 20	1597	0,94	1691
TA16 20 · 20	1647	0,15	1662
TA17 20 · 20	1680	0,25	1705
TA18 20 · 20	1603	0,17	1620
TA19 20 · 20	1627	0,55	1682
TA20 20 · 20	1584	0,15	1599
TA21 30 · 15	1764	8,33	2597
TA22 30 · 15	1795	7,34	2529
TA23 30 · 15	1796	7,83	2579
TA24 30 · 15	1831	8,62	2693
TA25 30 · 15	2007	3,68	2375
TA26 30 · 15	1819	6,41	2460
TA27 30 · 15	1778	8,94	2672
TA28 30 · 15	1673	8,26	2499
TA29 30 · 15	1795	5,46	2341
TA30 30 · 15	1676	9,8	2656

Tablo 12 de Nasiri ve Kianfar'ın 2010 yılındaki çalışması ile bu çalışmada önerilen YSA yöntemi ile elde edilen sonuçlar karşılaştırılmıştır.

Tablo 12

Nasiri ve Kianfar'ın 2010 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması

	2010 Nasiri & Kianfar	İyileşme Oranı %	2019 YSA
TA01 15 · 15	1249	-0,14	1235
TA02 15 · 15	1157	0,87	1244
TA03 15 · 15	1087	1,31	1218
TA04 15 · 15	1131	0,86	1217
TA05 15 · 15	1161	0,63	1224
TA06 15 · 15	1162	0,76	1238
TA07 15 · 15	1100	1,61	1261
TA08 15 · 15	1086	3,46	1432
TA09 15 · 15	1139	2,59	1398
TA10 15 · 15	1076	3,22	1398
TA11 20 · 20	1452	2,06	1658
TA12 20 · 20	1365	2,71	1636
TA13 20 · 20	1332	2,27	1559
TA14 20 · 20	1550	1,04	1654
TA15 20 · 20	1519	1,72	1691
TA16 20 · 20	1403	2,59	1662
TA17 20 · 20	1583	1,22	1705
TA18 20 · 20	1485	1,35	1620
TA19 20 · 20	1517	1,65	1682
TA20 20 · 20	1398	2,01	1599
TA21 30 · 15	1780	8,17	2597
TA22 30 · 15	1829	7,00	2529
TA23 30 · 15	1733	8,46	2579
TA24 30 · 15	1849	8,44	2693
TA25 30 · 15	1745	6,30	2375
TA26 30 · 15	1780	6,80	2460
TA27 30 · 15	1774	8,98	2672
TA28 30 · 15	1674	8,25	2499
TA29 30 · 15	1686	6,55	2341
TA30 30 · 15	1642	10,14	2656

Tablo 13 de Gao ve diğerlerinin 2014 yılındaki çalışması ile bu çalışmada önerilen YSA yöntemi ile elde edilen sonuçlar karşılaştırılmıştır.

Tablo 13

Gao ve Diğerlerinin 2014 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması

	2014 Gao ve diğerleri	İyileşme Oranı %	2019 YSA
TA01 15 · 15	1231	0,04	1235
TA02 15 · 15	1244	0	1244
TA03 15 · 15	1218	0	1218
TA04 15 · 15	1175	0,42	1217
TA05 15 · 15	1224	0	1224
TA06 15 · 15	1238	0	1238
TA07 15 · 15	1228	0,33	1261
TA08 15 · 15	1217	2,15	1432
TA09 15 · 15	1274	1,24	1398
TA10 15 · 15	1241	1,57	1398
TA11 20 · 20	1647	0,11	1658
TA12 20 · 20	1600	0,36	1636
TA13 20 · 20	1557	0,02	1559
TA14 20 · 20	1647	0,07	1654
TA15 20 · 20	1597	0,94	1691
TA16 20 · 20	1651	0,11	1662
TA17 20 · 20	1686	0,19	1705
TA18 20 · 20	1617	0,03	1620
TA19 20 · 20	1625	0,57	1682
TA20 20 · 20	1584	0,15	1599
TA21 30 · 15	1764	8,33	2597
TA22 30 · 15	1817	7,12	2529
TA23 30 · 15	1795	7,84	2579
TA24 30 · 15	1830	8,63	2693
TA25 30 · 15	2007	3,68	2375
TA26 30 · 15	1819	6,41	2460
TA27 30 · 15	1791	8,81	2672
TA28 30 · 15	1674	8,25	2499
TA29 30 · 15	1795	5,46	2341
TA30 30 · 15	1686	9,7	2656

Tablo 14 de Peng ve diğerlerinin 2015 yılındaki çalışması ile bu çalışmada önerilen YSA yöntemi ile elde edilen sonuçlar karşılaştırılmıştır.

Tablo 14

Peng ve Diğerlerinin 2015 Yılındaki Çalışması ile YSA Sonuçlarının Karşılaştırılması

	2015	İyileşme Oranı %	2019
	Peng ve diğerleri		YSA
TA01 15 · 15	1231	0,04	1235
TA02 15 · 15	1244	0	1244
TA03 15 · 15	1218	0	1218
TA04 15 · 15	1175	0,42	1217
TA05 15 · 15	1224	0	1224
TA06 15 · 15	1238	0	1238
TA07 15 · 15	1228	0,33	1261
TA08 15 · 15	1217	2,15	1432
TA09 15 · 15	1274	1,24	1398
TA10 15 · 15	1241	1,57	1398
TA11 20 · 20	1644	0,14	1658
TA12 20 · 20	1600	0,36	1636
TA13 20 · 20	1557	0,02	1559
TA14 20 · 20	1645	0,09	1654
TA15 20 · 20	1595	0,96	1691
TA16 20 · 20	1647	0,15	1662
TA17 20 · 20	1680	0,25	1705
TA18 20 · 20	1613	0,07	1620
TA19 20 · 20	1625	0,57	1682
TA20 20 · 20	1584	0,15	1599
TA21 30 · 15	1764	8,33	2597
TA22 30 · 15	1787	7,42	2529
TA23 30 · 15	1791	7,88	2579
TA24 30 · 15	1829	8,64	2693
TA25 30 · 15	2007	3,68	2375
TA26 30 · 15	1819	6,41	2460
TA27 30 · 15	1771	9,01	2672
TA28 30 · 15	1673	8,26	2499
TA29 30 · 15	1795	5,46	2341
TA30 30 · 15	1671	9,85	2656

Tablo 15 de Brgy'nin 2017 yılındaki alıřması ile bu alıřmada nerilen YSA yntemi ile elde edilen sonular karřılařtırılmıřtır.

Tablo 15

Brgy'nin 2017 Yılındaki alıřması ile YSA Sonularının Karřılařtırılması

	2017 Brgy	İyileřme Oranı %	2019 YSA
TA01 15 · 15	1232	0,03	1235
TA02 15 · 15	1245	-0,01	1244
TA03 15 · 15	1223	-0,05	1218
TA04 15 · 15	1177	0,4	1217
TA05 15 · 15	1232	-0,08	1224
TA06 15 · 15	1242	-0,04	1238
TA07 15 · 15	1228	0,33	1261
TA08 15 · 15	1218	2,14	1432
TA09 15 · 15	1283	1,15	1398
TA10 15 · 15	1245	1,53	1398
TA11 20 · 20	1660	-0,02	1658
TA12 20 · 20	1619	0,17	1636
TA13 20 · 20	1566	-0,07	1559
TA14 20 · 20	1661	-0,07	1654
TA15 20 · 20	1604	0,87	1691
TA16 20 · 20	1659	0,03	1662
TA17 20 · 20	1694	0,11	1705
TA18 20 · 20	1624	-0,04	1620
TA19 20 · 20	1633	0,49	1682
TA20 20 · 20	1603	-0,04	1599
TA21 30 · 15	1766	8,31	2597
TA22 30 · 15	1829	7	2529
TA23 30 · 15	1818	7,61	2579
TA24 30 · 15	1843	8,5	2693
TA25 30 · 15	2007	3,68	2375
TA26 30 · 15	1828	6,32	2460
TA27 30 · 15	1800	8,72	2672
TA28 30 · 15	1688	8,11	2499
TA29 30 · 15	1805	5,36	2341
TA30 30 · 15	1706	9,5	2656

KAYNAKÇA

- Adams, J.B., E., Zawack D. (1998) The Shifting Bottleneck Procedure For The Job Shop Scheduling, *Management Science*, 34, 391-401.
- Agarwal A., Colak S., Eryarsoy E., (2006) Improvement heuristic for the flow-shop scheduling problem: An adaptive-learning approach, *European Journal of Operational Research* 169, 801–815
- Akilli, B. Greedy Yaklaşımı, (2012) <http://buketakilli.blogspot.com/>, Erişim Tarihi 11 11 2018
- Akkaya, G., ve Gökçen, T., (2006) Job-Shop Scheduling Design With Artificial Neural Networks, *Journal Of Engineering And Natural Sciences, Sigma 4*, 121-130
- Artigues, C., Feillet, D:(2008) A branch and bound method for the job-shop problem with sequence-dependent setup times. *Annals of Operations Research* 159(1), 135–159
- Asadzadeh, L., Zamanifar, K., (2010) An agent-based parallel approach for the job shop scheduling problem with genetic algorithms, *Mathematical and Computer Modelling* 52, 1957_1965
- Atay, Y., Kodaz, H., (2014) Optimization of job shop scheduling problems using modified clonal selection algorithm, *Turkish Journal of Electrical Engineering & Computer Sciences* 22, 1528 – 1539
- Azadeh A. Negahban & M. Moghaddam (2012) A hybrid computer simulation-artificial neural network algorithm for optimisation of dispatching rule selection in stochastic job shop scheduling problems, *International Journal of Production Research*, 50:2, 551-566,
- Balas, E., Simonetti, N., Vazacopoulos, A.:(2005) Job shop scheduling with set-up times, deadlines and precedence constraints. *In: Proceedings of MISTA*
- Baysal M.E., Durmaz T., Sarucan A., Engin O., (2012) Açık Atölye Tipi Çizelgeleme Problemlerinin Paralel Kanguru Algoritması ile Çözümü, *Journal Of The Faculty Of Engineering And Architecture Of Gazi University*, 27 855-864
- Boender, C.G.E., Rinnooy Kan A.H.G., &Timmer, G.T (1982) A stochastic method for global optimization, *Math programming*, 12, 125-140

- Bolat, B., Erol, K.O., ve İmrak C.E., (2004) “Genetic Algorithms In Engineering Applications The Function Of Operations”, *Yıldız Teknik Üniversitesi, İstanbul*, 264-272
- Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, 41(3), 157–183.
- Brucker, P., Thiele, O.(1996) A branch and bound method for the general-job shop problem with sequencedependent setup times. *Oper. Res. Spektrum* 18, 145–161
- Buffa, E.S., *Journal of Operations Management*, 1980
- Bürgy, R., (2017) A neighborhood for complex job shop scheduling problems with regular objectives, *Springer Science+Business Media New York*, 20, 391–422
- Carbonell, 1987 machine learning, encyclopedia of artificial intelligence, wiley interscience
- Chandrasekaran M., Asokan, P., Kumanan, S., Balamurugan, T., Nickolas, S., (2006) Solving job shop scheduling problems using artificial immune system, *The International Journal of Advanced Manufacturing Technology*, 31, 580–593
- Chang, S. H., And Nam, B. H., 1993, Linear Programming Neural Networks For Job-Shop Scheduling. *Proceedings Of The Ijcn'n'93 International Joint Conference On Neural Networks*, Vol. 2, Nagoya, Japan, 25-29 Oct., Pp. 1557-1560.
- Chang-Lin Yang & Shan-Ping Chuang & Tsung-Shing Hsu, (2011) A genetic algorithm for dynamic facility planning in job shop manufacturing, *The International Journal of Advanced Manufacturing Technology*, 52:303–309
- Chaudhuri, A., Kajal De, K., (2010) Job Scheduling Problem Using Rough Fuzzy Multilayer Perception Neural Networks *Journal of Artificial Intelligence: Theory and Application* Vol.1-2, 4-19
- Chiang, T.C., ve Fu, L.C., (2006) Using Dispatching Rules for Job Shop Scheduling with Due Date-based Objectives *Proceedings, IEEE International Conference on Robotics and Automation* Orlando, Florida
- Colomi, Dorigo, Maffioli, Righini, Trubian, (1996) “Heuristics From Nature For Hard COP Problems”, *International Transactions In Operational Research* 3, *Elsevier Science*, 1- 21

- Çamoğlu, D., (2014) Alan Turing ve yapay zekâ, <https://bilimvegelecek.com.tr/index.php/2014/12/01/alan-turing-ve-yapay-zeka/> , Erişim Tarihi 01.05.2019
- Çayıroğlu, İ., (2012) Yapay Sinir Ağları, Mühendislik Fakültesi Karabük Üniversitesi. <http://www.ibrahimcayiroglu.com/dokumanlar/ilerialgoritmaanalizi/ilerialgoritmaanaliz-i-5.hafta-yapaysiniraglari.pdf> Erişim Tarihi 01 07 2017
- Çelikçapa, O.F., & Şenol G., (2015), Üretim Sistemleri, *Ekin Basım Yayın Dağıtım*, 1-11
- Dağlı, C., (1994) Artificial Neural Networks for Intelligent Manufacturing, *Chapman & Hall*, London, 71-72-73
- Dauzere-Peres, S., & Paulli, J. (1997). An integrated approach for modeling and solving the general multi-processor job shop scheduling problem using tabu search. *Annals of Operation Research*, 70, 281–306.
- De Souza, R. ve Zhao, Z.Y., (1997) A case for intelligent representation of dynamic resources in simulation, *International Journal of Production Research*, 35, 3289-3302
- Dorigo, M., Gambardella, L., (1997) Ant colonies for the traveling salesman problem. *BioSystems*, 43; 73-81.
- Durmaz, T., (2011) Açık Atölye Çizelgeleme Problemlerinin Paralel Kanguru Algoritması ile Çözümü, Yüksek Lisans Tezi, *Selçuk Üniversitesi, Fen Bilimleri Enstitüsü*
- Eberhart, R., & Kennedy, J. 1995. A new optimizer using particle swarm theory, *Micro machine and human science*, 39-43
- El-Bouri, A., ve Shah, P., (2006) A neural network for dispatching rule selection in a job shop, *The International Journal of Advanced Manufacturing Technology* 31, 342–349
- Elmas, C. 2003. Yapay Sinir Ağları, *Seçkin Yayıncılık*, Ankara
- Feng, S., Li, L., Cen, L, Huang, J., (2003) Using MLP networks to design a production scheduling system, *Computers & Operations Research* 30, 821–832
- Foo, P.S., Takefuji, Y., (1888) Stochastic neural networks for solving job shop scheduling: Part 1 problem representation and Part 2 architecture and simulations, *Proceeding of the IEEE International Conference on Neural Network*

- Foo, S. Y., Takefuji, Y., Szu, H., (1995) "Scaling Properties of Neural Networks for Job Shop Scheduling", *Neurocomputing*, 8,79 – 91
- Gao, H., Kwong S., Fan, B., Wang, R.,(2014) A Hybrid Particle-Swarm Tabu Search Algorithm for Solving Job Shop Scheduling Problems, *IEEE Transactions On Industrial Informatics*, 2044-2054
- Gass, S.I., (2000) Making Decisions with Precision, Business Week http://www.businessweek.com/archives/2000/b3705139.arc.htm?campaign_id=search#B3705139), Erişim tarihi: 03 01 2017.
- Gever, B., Güneş, B.N., Gülel, D., Ataseven, E.B., Saaticioğlu, Ö., ve Bebek, Z.S., (2014) Bir Üretim İşletmesinin Verileri Kullanılarak Çarpımsal Deney Tasarımı ile Taguchi Yaklaşımının Karşılaştırılması, *Endüstri Mühendisliği Dergisi* 1, 2-26
- Glover F, Laguna M, Marti R, (2007) Principles of tabu search, *Handbook of Approximation Algorithms and Metaheuristics*, Chapter 23, 1-10
- Glover, F., (1990) Tabu Search, Part II, *ORSA Journal on Computing*, vol. 2, no. 1, 4-32.
- Glover, L., (1997) Tabu Search, *Kluwer Academic Publishers*, 12-25, Boston, 1997.
- Goldberg, D.A., (1989) Genetic Algorithms in Search, Optimisation and Machine Learning, *Addison- Wesley Publ.*
- Gonçalves, J.F., Jose de, J., Mendes, M., Reşende, M.G.C., (2005) A hybrid genetic algorithm for the job shop scheduling problem, *European Journal of Operational Research* 167, 77–95
- Goss S., Aron S., Deneubourg J.L., ve Pasteels J.M., (1989) Self-organized Shortcuts in the Argentine Ant, *Naturwissenschaften* 76, 579–581
- Hanşen P., Mladenovic. N., Brimber J., Jos'e A. Moreno P'erez, (2010) "Variable Neighborhood Search", *Handbook of Metaheuristics*, 57-97
- Hart, E., (2002) Immunology as a metaphor for computational information processing: Fact or Fiction Doctoral Dissertation, Artificial Intelligence Applications Institute, Division of Informatics, *University of Edinburgh*,

- Holland, J.H., (1975) *Adaptation in Natural and Artificial Systems*”, University of Michigan Press, *Ann Arbor*
- Hopfield, J. J. ve Tank, D. W. (1985) “Neural Computation of Decision in Optimization Problems”, *Biological Cybernetics*, 52:141-152, Huang, J., Süer, G.A., (2015) A dispatching rule-based genetic algorithm for multi-objective job shop scheduling using fuzzy satisfaction levels, *Computers & Industrial Engineering* 86, 29–42
- Huang, R., ve Yang, C., (2008) Ant colony system for job shop scheduling with time Windows, *The International Journal of Advanced Manufacturing Technology*, 39,151–157
- Kacem, I., Hammadi, S., & Borne, P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 32(1), 1–13.
- Karthikeyan, S., Asokan, P., Nickolas, S., Page, T., (2014) A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems, *The International Journal of Advanced Manufacturing Technology*, 1567–1579
- Kaya, D., Oktay, S., Engin, O. 2005, Kalite Kontrol Problemlerinin Cozumunde Yapay Sinir Ağlarının Kullanımı, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 21 (1-2), 92-107
- Kirkpatrick, S., Gelatt Jr. C.D., Vecchi, M.P., Optimization by Simulated Annealing, *Science*, 220, 671-680
- Klahr, D., (1972) Visual Information Processing, *Proceedings of the Eighth Annual Carnegie Symposium on Cognition*, Held at the Carnegie-Mellon University, Pittsburgh, Pennsylvania
- Koza, J.R., (1992) *Genetic Programming: on the Programming of Computers by Means of Natural Selection* , *MIT Press*,
- Kundakcı, N., Kulak, O., (2016) Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem, *Computers & Industrial Engineering* 96, 31–51
- Küçükkoç İ., (2018) ENM4129 Çizelgeleme, Balıkesir Üniversitesi, Endüstri Mühendisliği Bölümü <http://ikucukkoc.baun.edu.tr/lectures/EMM4129/EMM4129-S1.pdf> ,Erişim tarihi: 30 09 2018.

- Li-Ning Xing, Ying-Wu Chen, Peng Wang, Qing-Song Zhao, Jian Xiong, (2010) A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems, *Applied Soft Computing* 10, 888–896
- Manas Ranjan Singh, S.S. Mahapatra, (2016) A quantum behaved particle swarm optimization for flexible job shop scheduling, *Computers & Industrial Engineering* 93, 36-44
- Mattfeld, D.(1995) Evolutionary Search and the Job Shop. Investigations on Genetic Algorithms for Production Scheduling. *Springer*, Berlin
- Meeran, S., Morshed, M.S., (2012) A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study, *Journal of Intelligent Manufacturing* 23, 1063–1078
- Nasiri, M.M., Kianfar, F., (2011) A hybrid scatter search for the partial job shop scheduling problem, *The International Journal of Advanced Manufacturing Technology* 52,:1031–1038
- Nouiri, M., Bekrar, A., Jemai, A., Niar, S., Ammari, A.C., (2018) An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem, *Journal of Intelligent Manufacturing* 29, 603–615
- Özkazanç, Ü.A., Anagün, A.S., (2001) Atölye Ortamında İşlerin Çizelgelenmesinde Toplam Tamamlanma Zamanının Enküçüklenmesine Dayalı Bir Sinirsel Ağ Yaklaşımı, *Bilim ve Teknoloji Dergisi*, 2(1), 125-136
- Öztemel E., (2017) Bees Algorithm for multi-mode, resource-constrained project scheduling in molding industry, *Computers & Industrial Engineering*, 187-196
- Öztemel, E., (2003) Yapay Sinir Ağları, *Papatya Yayıncılık*, İstanbul
- Panos M. Pardalos, Oleg V. Shylo, Alkis Vazacopoulos. (2008) "Solving job shop scheduling problems utilizing the properties of backbone and “big valley”", *Computational Optimization and Applications* 47, 61–76
- Peng, B., Lü, Z., ve Cheng, T:C.E., (2015) A tabu search/path relinking algorithm to solve the job shop scheduling problem, *Computers & Operations Research* 53, 154–164
- Pezzella, F., Morganti, G., Ciaschetti, G., (2008) A genetic algorithm for the Flexible Job-shop Scheduling Problem, *Computers & Operations Research* 35, 3202 – 3212

- Pinedo, M.,1991. Scheduling Theory, Algorithms and Systems, *Operations Research Proceedings*, 35-42
- Rajkumar, M., Asokan, P., Anilkumar, N.,ve Page, T., (2011) A Grasp algorithm for flexible job-shop scheduling problem with limited resource constraints, *International Journal of Production Research*, 49:8, 2409-2423
- Randall, D., ve Jonathan, K., (1975) An Overview of Production Systems, *Stanford Universitesi, Bilgisayar Bilimleri*
- Rechenberg, I., (1973) Evolutionstrategie (Evolution Strategy), *Frommann-Holzboog, Stuttgart*,
- Reeves, C.R., 1995, Modern Heuristic Techniques for Combinatorial Problems, *McGraw-Hill Book Company*, UK.
- Roshanaei, V., Naderi, B., Jolai, F., Khalili, M., (2009) A variable neighborhood search for job shop scheduling with set-up times to minimize makespan, *Future Generation Computer Systems* 25, 654 – 661
- Sabuncuoglu, İ. ve Gurgun, B., (1996) A neural network model for scheduling problems, *European Journal of Operational Research* 93 288-299
- Sache, R.G., 2014 Neural Network for Solving Job-Shop Scheduling Problem, *IOSR Journal of Computer Engineering (IOSR-JCE)*, Ver. VIII PP 18-25
- Sağiroğlu, S., Besdok, E., Erler, M. (2003). Muhendislikte Yapay Zeka Uygulamaları:1 Yapay Sinir Ağları, *Ufuk Yayıncılık, Kayseri*
- Sarac, T. 2004. Yapay Sinir Ağları Seminer Projesi, *Gazi Universitesi Endustri Muhendisliđi Bolumu*, Ankara
- Sexton, R.S., Dorsey, R.E., Johnson J.D., (1999) Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing, *European Journal of Operational Research*, 589-601
- Sha, D.Y., ve Hsu, C.Y., (2006) A hybrid particle swarm optimization for job shop scheduling problem, *Computers & Industrial Engineering*, 51, 791–808

- Singh, M.R., ve Mahapatra, S.S., (2016) A quantum behaved particle swarm optimization for flexible job shop scheduling, *Computers & Industrial Engineering* 93, 36–44
- Smith, A., (1776) The wealth of nations, *British Library Cataloging in Publication Data*.
- Şen, Z. 2004. Yapay Sinir Ağları İkeleri, *Su Vakfı Yayınları*, İstanbul
- Şevkli, M. ve Yenisey, M., (2006) Atölye tipi çizelgeleme problemleri için parçacık sürü optimizasyonu yöntemi, *İTÜ İşletme Fakültesi, Endüstri Mühendisliği Bölümü*, 34357, Maçka, İstanbul
- T.M Willems & J.E Rod, (1993) “Neural nets for job shop scheduling will they do the job?” *Proc. IFAC 12th World Conference.*, 3, 53–56.
- Taguchi, G., Elsayed, E. A., Hsiang, T., (1989) Quality Engineering in Production Systems, *McGraw-Hill*, New York, 173
- Taillard, E., (1993) Benchmarks For Basic Scheduling Problems, *European Journal Of Operational Research* 64, 278-285
- Taillard,E., (1993) Scheduling Instances, <http://mistic.heigvd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>
Erişim tarihi 01 05 2019
- Teymourifar, A., ve Ozturk, G., (2018) A Neural Network-Based Hybrid Method To Generate Feasible Neighbors For Flexible Job Shop Scheduling Problem, *Universal Journal Of Applied Mathematics* 6(1), 1-16
- Vela R.C., Varela, R., González, M.A., (2010) Local search and genetic algorithm for the job shop scheduling problem with sequence dependent setup times, *Journal of Heuristics*, 16, 139–165
- Weckman, G.R., ve Ganduri, C.V., (2008) ve Koonce, D.A., A neural network job-shop scheduler, *Journal of Intelligent Manufacturing*, 19,191–201
- Willems, T. M., ve Rooda, J. E., (1994) Neural Networks For Job-Shop Scheduling, *Control Eng. Practice*, 2(1), 31-39.
- Xin-She Yang ve He Xingshi, (2013) Firely Algorithm Recent Advances and Applications, *Int. J. Swarm Intelligence*, 1, 36--50

- Yang, C., Chuang, S., Hsu, T., (2011) A genetic algorithm for dynamic facility planning in job shop manufacturing, *The International Journal of Advanced Manufacturing Technology*, 52:303–309
- Yang, S. ve Wang D., (2000) “Constraint Satisfaction Adaptive Neural Network and Heuristics Combined Approaches for Generalized Job-Shop Scheduling”, *IEEE Transactions on Neural Networks*, 11 (2): 474 – 486
- Yang, X. S. (2008) Nature-Inspired Metaheuristic Algorithms, *Luniver Press*, UK, 81-84
- Yu, H. ve Liang, W., (2001) “Neural Network and Genetic Algorithm-Based Hybrid Approach To Expanded Job-Shop Scheduling”, *Computers & Industrial Engineering*, 39 (3-4): 337- 356
- Yurtoğlu, H., (2005) Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi: Bazı Makroekonomik Değişkenler İçin Türkiye Örneği, *DPT – Uzmanlık Tezleri Yayın No: 2683*
- Yurttakal, A.H., (2014) İş Akışı Çizelgeleme Probleminin Yapay Bağışıklık Sistemi ile Optimizasyonu, Elektronik-Bilgisayar Sistemleri Eğitimi Anabilim Dalı, Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya*
- Zandieh M. ve Adibi M.A., (2010) Dynamic job shop scheduling using variable neighbourhood search, *International Journal of Production Research*, 48:8, 2449-2458
- Zhang, C.Y., Li, P., Rao, Y., Guan, Z., (2008) A very fast TS/SA algorithm for the job shop scheduling problem, *Computers & Operations Research* 35, 282 -294
- Zhang, G., Shao, X., Li, P., Gao, Li., (2009) An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, *Computers & Industrial Engineering* 56, 1309–1318
- Zhang, J., Hu, X., Tan X., Zhong, J.H., Huang, Q., (2006) Implementation of an Ant Colony Optimization technique for job shop scheduling problem, *Transactions of the Institute of Measurement and Control* 28, 1, 93-108
- Zhang, J., Hu, X., Tan, X., Zhong, J.H., Huang, Q., (2006) Implementation of an Ant Colony Optimization technique for job shop scheduling problem, *Transactions of the Institute of Measurement and Control* 28, 1, 93-108

Zhang, X., Deng, Y., Chan, F.T.S., Xu P., Mahadevan S., Hu Y., (2013) IFSJSP: A novel methodology for the Job-Shop Scheduling Problem based on intuitionistic fuzzy sets, *International Journal of Production Research*, 51:17, 5100-5119

Zhou, D. N., Cherkassky, V., Baldwin, T. R., Olson D. E., (1991), A neural network approach to job-shop scheduling, *IEEE Transactions on Neural Network*, 2(1), 175-179



ÖZGEÇMİŞ

Yazar, 1991 yılında Bursa ilinde doğdu. İlkokul eğitimini Bursa'da Ortaokul ve Lise eğitimini Adana'da tamamladı. 2009 yılında liseden mezun olarak Lefke Avrupa Üniversitesi, Bilgisayar ve Enformasyon Sistemleri Lisans bölümünü kazandı. Hazırlık okulu ile birlikte %100 burslu olarak 2014 yılında mezun oldu. Yaklaşık 1 yıl ücretli öğretmenlik yaptıktan sonra 2015-2017 yılları arasında Defacto Perakende Ticaret A.Ş firmasında Mağaza Yöneticisi olarak görev aldı. Aynı zamanda 2015-2017 yılları arasında Çukurova Üniversitesi Elektronik Haberleşme Teknolojileri önlisans eğitimi aldı. 2018-2019 yılları arasında Adana Alpaslan Türkeş Bilim ve Teknoloji Üniversitesi yürütücülüğünde SODES projesi idari personeli olarak çalıştı. Ayrıca 2018 yılından itibaren Çukurova Üniversitesi Endüstri Mühendisliği bölümünde 2.Lisans eğitimine devam etmektedir.