



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

İŞ AKIŞI ÇİZELGELEME PROBLEMİNİN YAPAY BAĞIŞIKLIK SİSTEMİ İLE OPTİMİZASYONU

Ahmet Haşim YURTTAKAL

YÜKSEK LİSANS TEZİ

ELEKTRONİK-BİLGİSAYAR SİSTEMLERİ EĞİTİMİ ANABİLİM DALI

Haziran-2014
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS
İŞ AKIŞI ÇİZELGELEME PROBLEMİNİN YAPAY BAĞIŞIKLIK SİSTEMİ İLE OPTİMİZASYONU

Ahmet Haşim YURTTAKAL

SELÇUK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRONİK-BİLGİSAYAR SİSTEMLERİ EĞİTİMİ ANABİLİM DALI

Danışman: Doç. Dr. Adem Alpaslan ALTUN

2014, 72 Sayfa

Jüri
Doç.Dr. Cemil SUNGUR
Doç. Dr. Adem Alpaslan ALTUN
Yrd.Doç.Dr. Hasan Erdinç KOÇER

İş akışı çizelgeleme, bir ürünü oluşturan iş parçalarının eldeki tek veya çok sayıda makinelerle hangi sırada ve ne zaman işleneceğinin belirlenmesidir. Kaliteli çizelgeleme planları, istenilen üretim hedeflerine en ekonomik ve en verimli şekilde ulaşabilmenin yoludur.

Bu tez çalışmasında; optimizasyon kavramı, çizelgeleme problemleri, sezgisel yöntemler yerel arama algoritmaları, doğal bağışıklık sistemi ve yapay bağışıklık sistemi anlatılmıştır. Taillard'ın (1990) 110 adet örnek iş akış çizelgeleme problemleri, C# dilinde geliştirilen bir uygulama üzerinde, hem yapay bağışıklık sistemi(YBS) ile hem de yapay bağışıklık sisteminin FRB3 yerel arama algoritmasıyla(YBS&FRB3) melez kullanılmasıyla elde edilen algoritma yardımıyla, çözülmüş sonuçlar incelenmiş daha sonra literatürdeki diğer çözümlerle karşılaştırılmıştır.

İş akışı çizelgeleme problemleri için geliştirilen yeni yöntemin, YBS modelinden daha iyi sonuçlar verdiği görülmüştür. Literatürde test problemleri olarak kullanılan değişik boyutlardaki toplam 110 adet Taillard Problemi 20'şer kere çözülmüştür. Sonuçta önerilen YBS&FRB3 algoritması ile 5 örnek için optimum sonucu elde etmiş ve her örnekte YBS modelinden daha düşük yüzdelik hata oranına ulaşılmıştır.

Anahtar Kelimeler: FRB3 yerel arama algoritması, iş akışı çizelgeleme, sezgisel yöntemler, yapay bağışıklık sistemi

ABSTRACT

MS THESIS

SOLVING PERMUTATION FLOWSHOP SCHEDULING PROBLEM BY ARTIFICIAL IMMUNE SYSTEM

Ahmet Haşim YURTTAKAL

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN ELECTRONIC-COMPUTER SYSTEMS EDUCATION

Advisor: Assoc.Prof.Dr. Adem Alpaslan ALTUN

2014, 72 Pages

Jury

Assoc.Prof.Dr. Cemil SUNGUR
Assoc.Prof.Dr. Adem Alpaslan ALTUN
Asst.Prof.Dr. Hasan Erdinç KOÇER

Scheduling is determinination of proceeding sequence and timing of product workpieces using single or large number of machines. Scheduling plans in good quality is the way to reach desired production goals with the most economically and the most efficiently.

In this thesis; optimization, scheduling problems, heuristic methods, local search algorithms, native immune system and artificial immune system were introduced. Taillard's 110 flowshop scheduling samples were solved based on an application improved by C# using Artificial Immune System (AIS) and the hybrid algorithm (AIS&FRB3) obtained by Artificial Immune System (AIS) and FRB3 Local Search Algorithm than the results were analyzed and compared with ones in literature.

It has been shown that the novel method for flowshop scheduling problems gives better results comparing AIS model. 110 Taillard problems which used as test problems in literature in different dimensions were solved 20 times for each one. Finally, the optimum result for 5 samples by introduced AIS & FRB3 algorithm and the most minimum percentage error rate in each sample were reached.

Keywords: Airtifical Immune Systems, Flow-shop Scheduling Problem, FRB3 Local Search Algorithm, Heuristic Methods

ÖNSÖZ

Çalışmamın her aşamasında büyük minnet borçlu olduğum başta danışmanım Sayın Doç. Dr. Adem Alpaslan ALTUN olmak üzere; yardımlarını esirgemeyen Selçuk Üniversitesi Teknoloji Fakültesi Öğretim Üyesi Sayın Doç. Dr. Mehmet CUNKAŞ'a, Dumlupınar Üniversitesi Mühendislik Mimarlık Fakültesi Öğretim Üyesi Sayın Yrd. Doç. Dr. Doğan AYDIN'a, Bozok Üniversitesi Mühendislik Mimarlık Fakültesi Öğretim Üyesi Sayın Yrd. Doç. Dr. Davut Erdem ŞAHİN'e, Bozok Üniversitesi Öğretim Görevlisi Sayın Murat ALTINDAŞ'a ve son olarak sevgili eşime teşekkürlerimi sunarım.

Ahmet Haşim YURTTAKAL
KONYA-2014

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **ix**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **3**

2.1. İş Akışı Çizelgeleme Problemi ile İlgili Kaynak Araştırması 3
2.2. Yapay Bağışıklık Sistemleri ile İlgili Kaynak Araştırması 7

3. MATERYAL VE YÖNTEM **10**

3.1. Optimizasyon 10
3.2. Üretim Çizelgeleme Problemleri 12
3.2.1. Üretim çizelgelemenin sınıflandırılması 15
3.2.2. Çizelgeleme probleminin çözülebilirliği ve karmaşıklık 17
3.2.3. Üretim çizelgeleme parametreleri 18
3.2.4. Tek makine problemleri 21
3.2.5. Paralel makine problemleri 21
3.2.6. Atölye tipi üretim çizelgeleme problemleri 22
3.2.7. Akış tipi üretim çizelgeleme problemleri 23
3.3. Sezgisel Yöntemler 24
3.3.1. Tabu arama 25
3.3.2. Benzetilmiş tavlama 26
3.3.3. Genetik algoritma 27
3.3.4. Nawaz, Enscore, Ham (NEH) yaklaşımı 28
3.3.5. FRB yaklaşımı 29
3.4. Bağışıklık Sistemi 34
3.4.1. Doğal bağışıklık sistemi 34
3.4.2. Yapay bağışıklık sistemi 44

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA **50**

4.1. Geliştirilen Yazılımda Kullanılan Parametreler 50
4.2. Geliştirilen Yazılımın Uygulanması 51
4.3. İş Akış Çizelgeleme Problemi İçin Yapılan Denemeler 53

5. SONUÇLAR VE ÖNERİLER 62

5.1 Sonuçlar 62
5.2 Öneriler 63

KAYNAKLAR 65

ÖZGEÇMİŞ.....72

SİMGELER VE KISALTMALAR

Simgeler

FRB : Farahmand Rad, Rubén Ruiz, Naser Boroojerdian (2009) tarafından geliştirilen sezgisel algoritma
GA : Genetik Algoritma
NEH : Nawaz Enscore, Ham (1983) tarafından geliştirilen sezgisel algoritma
NP : Non-Polinomial
P : Polinomial
SA : Simulated Annealing(Benzetilmiş Tavlama)
YBS : Yapay Bağışıklık Sistemi
YSA : Yapay Sinir Ağları

Kısaltmalar

Ag : Antijen
Ab_n : Uygunluğu yüksek antikorlar grubu
C : Klonlama
β : Klonal faktör
G : Jenerasyon sayısı
N : Antikor popülasyon boyutu
$f(i)$: Üzerinde çalışılan antikorun uygunluğu
C_{max} : En büyük tamamlanma zamanı
n : iş sayısı
m : makine sayısı
P_{ij} : j işinin i makinesindeki işlem zamanı
W_j : j işinin sitemdeki diğer işlere göre önemi
$Best_{sol}$: Taillard tablolarındaki problemlerin bugüne kadar hesaplanmış en kısa çözüm süresi
Heu_{sol} : Önerilen algoritma içinde hesaplanan toplam işlem süreleri
RPD : Yüzdelik hata oranı

1. GİRİŞ

Son yıllarda işletmeler, artan rekabet ortamına uyum sağlayabilmek için en az miktarda kaynak kullanarak, müşteri gereksinimlerine en hızlı yanıt verebilecek tekniklerle en kaliteli ürün ve hizmet üretmek zorunda kalmıştır. Üretim sistemlerinde bulunan çok fazla değişken parametre ve sürekli değişen çevre koşulları, bu ihtiyaca cevap verme noktasında yavaşlığa ve etkisizleşmeye neden olabilmektedir. Bu yüzden, hızlı değişen müşteri talepleri karşısında özellikle üretim planlarını en çabuk ve en kaliteli oluşturabilen işletmeler rekabette bir adım öne geçmektedir. Bu nedenle, çok değişkenli sistemlerde oldukça etkili olan ve gelişimini hala sürdüren üretimleri en iyileme teknikleri, günümüz işletmeleri tarafından etkin bir şekilde kullanılmaktadır.

Bir işletme için önemli olan unsurlar; müşterinin isteklerini eksiksiz ve zamanında tamamlama, istenen kalitede mal üretme ve en az maliyetle en verimli şekilde çalışmadır. Bu unsurlar, işletmenin ancak üretim öncesi yapmış olduğu planlamalara bağlıdır. Bu planlamalar işin hangi makinede ve hangi sırayla işleme konulması gerektiğine karar verme bir başka deyişle çizelgelemedir. Kaliteli çizelgeleme planları, istenilen üretim hedeflerine en ekonomik ve en verimli şekilde ulaşabilmenin yoludur. Bunun için en küçük işletmeden en büyük işletmeye kadar tüm üreticiler bu planlamaya gerekli önemi vermekte ve bu çizelgelemelere maddi kaynak ayırmaktadırlar.

Teknolojinin gelişmesiyle de üretim hatlarında kullanılan makinelerde tamamen elektronikleşmiştir. Dolayısıyla insan hataları en aza indirgenirken sistemde insan faktörünün azalmasıyla çizelgelemelerinin önemi artmaktadır.

Çizelgeleme problemlerinde iki farklı sınırlamayla karşılaşılmaktadır. Bunlardan ilki makine kapasitelerindeki sınırlamalar, ikincisi ise bazı işlerin işlem sırası üzerindeki sınırlamalardır. Çizelgeleme problemlerinin çözümü, iki gruba ayrılabilen bu sınırlamaların uygun çözümünü ifade eder. Dolayısıyla çizelgeleme problemi atama ve sıralama işlemlerinden oluşur denilebilir (Baker ve Trietsch, 2009).

Üretim çizelgeleme, bir veya birkaç performans ölçütünü optimum seviyeye getirecek şekilde üretim kaynaklarının, zaman bazında çeşitli görevlere atanmasını kapsar (Laha ve Sarin, 2009).

Üretim sistemine gelen siparişler öncelikle, teslim tarihlerini de içeren işlere çevrilmelidir. Genellikle bu işlere ait operasyonların belirli bir sırada atölyedeki bazı makineler üzerinde yapılması gerekir. Buna karşın bu operasyonların, bazı durumlarda bekletilmesi gerekebilir Bunun yanı sıra beklenmeyen atölye olayları da göz önünde

bulundurulması gerekebilir. İşlemlerin kontrolü ve etkinliliği açısından yapılması gereken görevlere ilişkin detaylı bir çizelgenin hazırlanması yararlı olacaktır (Pinedo, 2008).

Optimizasyon bir şeyin daha iyisini gerçekleştirme süreci olarak tanımlanabilir. Çizelgeleme problemleri de bu açıdan optimizasyon problemleridir. Genel olarak optimizasyon metotları 2 gruba ayrılır: Tam çözüm yöntemleri ve sezgisel yöntemler.

Tam çözüm yöntemleri, doğrudan metodlar oarak adlandırılır ve tahmini bir başlangıç çözümü ile araştırmaya başlanır. Optimallik şartları sağlanıncaya dek başlangıç çözümü iteratif olarak geliştirilir. Optimum çözümleri bulmak için çözüm uzayı araştırılır. Sezgisel yöntemler, dolaylı metotlar olarak adlandırılır ve gerek ve yeter şartlar yazılır. Sonra bu şartlar bölgesel minimumluğa aday noktalar için çözülür (Karaboğa, 2011).

Literatürde optimizasyon problemlerinin çözümünde kullanılan birçok sezgisel yöntemler bulunmaktadır. Karınca kolonileri algoritması, tabu arama, tavlama benzetimi, genetik algoritma ve yapay bağışıklık sistemleri kullanılan bir takım sezgisel yöntemlerdir.

Yapay bağışıklık sistemleri doğal bağışıklık sisteminden esinlenmiş olup, doğal sistemin prensipleri yapay bağışıklık sistemi algoritmalarının geliştirilmesinde kullanılmıştır. Bu konuda Farmer ve ark. (1986) tarafından yapılan ve yapay bağışıklık sistemini diğer yapay zekâ yaklaşımları ile ilişkilendiren çalışma öncü çalışma olarak kabul edilmektedir (Akçay, 2009).

Bu tez çalışmasında; hem yapay bağışıklık sistemi (YBS) hem de yapay bağışıklık sistemi FRB3 yerel arama algoritmasıyla (YBS&FRB3) melez kullanılarak, C# dilinde geliştirilen bir uygulama üzerinde, Taillard'ın (1988) 100 adet örnek iş akış çizelgeleme problemleri çözülmüş sonuçlar incelenmiş ve karşılaştırılmıştır.

Tezin ikinci bölümünde çizelgeleme problemleri ve yapay bağışıklık sistemi hakkında literatür taraması verilmiştir.

Tezin üçüncü bölümünde optimizasyon kavramı, çizelgeleme problemleri, yerel arama algoritmaları, doğal bağışıklık sistemi ve yapay bağışıklık sistemi anlatılmıştır.

Dördüncü bölümde iş akışı çizelgeleme probleminin yapay bağışıklık sistemi ile çözümü anlatılmış, elde edilen sonuçlar verilmiş, farklı algoritmalarla elde edilen sonuçların karşılaştırılması yapılmıştır.

Sonuç bölümünde algoritmanın daha iyi sonuçlar üretebilmesi için önerilerde bulunulmuştur.

2. KAYNAK ARAŞTIRMASI

Son yarım asırdır yapay zeka ve üretim çizelgeleme problemleri konusundaki çalışmalar giderek artmaktadır. Bilgisayar teknolojisindeki gelişmelere bağlı olarak incelenen problemlerin karmaşıklığının da arttığı görülmektedir.

Kaynak araştırması İş Akışı Çizelgeleme Problemleri ve Yapay Bağışıklık Sistemi olmak üzere iki ana konu üzerinde yapılmıştır.

2.1. İş Akışı Çizelgeleme Problemi ile İlgili Kaynak Araştırması

Atölyede yapılacak işler aynı rotaya sahipse, yani tüm işler aynı makineleri aynı sırada takip ediyorsa bu ortam akış tipi olarak adlandırılmaktadır. Son yıllarda akış tipi çizelgeleme problemleri üzerine yapılan çalışmalar giderek artmaktadır. Bazı problemlerin optimizasyonunda, algoritmanın yerel arama algoritmalarıyla kuvvetlendirildiği, parametre optimizasyonu yapıldığı veya yeni yaklaşımlar önerildiği görülmüştür. Aşağıda bu çalışmaların bazılarına kısaca yer verilmiştir.

Garey ve ark (1976), makine sayısı 3 ve üzeri olan *(m≥3)* akış tipi problemlerde en kısa uzunluktaki çizelgeyi bulmanın mümkün olmadığını yani problemin NP-tam olduğunu ispatlamıştır. Ayrıca en küçük ortalama akışı sağlayan çizelgeyi bulmanın da ikiden fazla makinen bulunduğu *(m>2)* akış tipi çizelgeleme problemleri için mümkün olmadığını bulmuştur.

Lenstra ve ark (1977), iki makineli akış tipi problemlerinde maksimum gecikmeyi en küçükleyen çizelgeyi bulmanın mümkün olmadığını, çözümün NP-tam olduğunu bulmuştur.

Taillard, 1990'da yapmış olduğu çalışmada, akış tipi çizelgeleme probleminin çözümü için paralel tabu arama algoritmasını önermiş ve sonuçları literatürdeki diğer algoritmalarla karşılaştırmıştır.

Gupta (1988), birbirinin aynı paralel makinelerin olduğu iki aşamalı akış tipi çizelgeleme probleminin NP-tam olduğu ispatlamış ve yaklaşık bir çözüm bulmak için sezgisel bir yöntem geliştirmiştir. Tek bir makinen olduğu özel durum için tasarlanan bu yöntemin en küçük akış süresini bulmadaki başarısını deneysel olarak incelemiştir.

Hoogeven ve ark. (1996), birden fazla makinenin paralel olarak işlem gördüğü esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanının en küçüklenmesini sağlayan çizelgenin bulunmasını araştırmıştır. İki aşamalı akış tipi

problemlerin birinci veya ikinci aşamasında iki paralel makine olduğu ve işlemin parçalara bölünebildiği durumlarda bile problemin NP-Zor olduğunu belirtmiştir.

Engin, (2001)'de akış tipi çizelgeleme problemlerini genetik algoritma ile çözerek parametre optimizasyonu üzerine çalışmıştır. Engin ve Fığlalı 2002'deki çalışmalarında ise uygun çaprazlama operatörünü belirlemiştir. Engin ve Döyen, 2004 yılındaki yapmış oldukları çalışmalarda, klasik akış tipi ve esnek çizelgeleme problemlerini, yapay bağışıklık sistemiyle çözmüş ve parametre optimizasyonu üzerine çalışmıştır. Çalışmalarında, herhangi bir problemin çözümünde optimuma yakın çözüm veren parametre seti, başka bir YBS uygulaması için genelleştirilemeyeceğini belirterek; akış tipi ve esnek akış tipi problemleri için farklı parametre optimizasyonu yapmışlardır. Carlier (1978) tarafından oluşturulan kıyaslama problemleri baz alınmıştır. Akış tipi çizelgeleme problemlerinde optimize edilen parametreler; eleme adımları yüzdelik oranı (A), antikor popülasyon büyüklüğü (B), antikor eleme yüzdelik oranı (C)'dır. Esnek akış tipi çizelgeleme problemlerinde optimize edilen parametreler; antikor popülasyon büyüklüğü (A), antikor eleme yüzdelik oranı (B)'dır.

Cheng ve Ding (2001), tek makine çizelgeleme problemleri üzerinde işlem sürelerini adım adım düşüren bir algoritma geliştirmişlerdir.

Seçme, (2006)'da 5x5 ve 20x20 boyutlu akış tipi çizelgeleme problemlerini yapay sinir ağlarıyla çözmüştür. Günaydın (2008) ise yapay sinir ağlarıyla çözmüş, sonuçları Aldowasian ve Allahverdi'nin (2003) önerdiği Genetik Algoritma (GA) ile karşılaştırmıştır. Güçlü, (2010)'daki çalışmasında karınca kolonisiyle çözmüş, Günaydın'ın sonuçlarıyla karşılaştırmıştır. Günaydın ve Güçlü, literatüre uygun rastsal olarak oluşturulan 192 adet beklemesiz akış tipi çizelgeleme problemi verisi kullanılmıştır. İş sayıları n = 8, 10, 12, 50, 100, 150, 200, 250 olmak üzere 8 farklı iş sayısı olarak belirlenmiştir. Makine sayıları ise m = 2, 3, 5, 8, 10, 15, 20, 25 olarak seçilmiştir. Makinelerin hazırlık süreleri [1,10] arasında rastsal olarak üretilmiştir. İşlem zamanları için; üç çeşit işlem zamanı seçilmiştir. Birinci tip işlem zamanı a; [1,10] arasında, ikinci tip işlem zamanı b; [1,50] arasında ve üçüncü tip işlem zamanı c; [1,100] arasında rastsal olarak üretilmiştir. Bu şekilde (mxnx3) olarak toplam 8x8x3=192 problem rastsal olarak geliştirilmiştir. Keskin, (2010)'daki çalışmasında aynı problem verilerini çok amaçlı melez genetik algoritma yardımıyla çözmüş ve sonuçları karşılaştırmıştır.

Ruiz, Rad ve Boroojerdian tarafından 2007 yılında yapılan çalışmada, akış tipi çizelgeleme problemlerinin çözümü için FRB yerel arama algoritmaları geliştirilmiştir.

Çalışmada öncelikle NEH yerel arama algoritması hakkında bilgi verilmiş ardından FRB1, FRB2, FRB3, FRB4, FRB5 yerel arama algoritmaları önerilmiştir. Problem tüm bahsedilen yerel arama algoritmaları yardımıyla çözülmüş sonuçlar karşılaştırılmıştır. Kurnaz ve Kart, 2010'da iş akış çizelgeleme problemleri üzerine NEH, FRB3 ve FRB4 algoritmalarını karşılaştıran bir çalışma yayınlamıştır.

Chang ve ark. (2007), çok amaçlı akış tipi çizelgeleme problemi üzerinde genlerin iyileştirilmesini sağlayan bir algoritma önermiştir. Önerilen algoritma, sezgisel ve yerel araştırmayı geleneksel genetik algoritma ile birleştirerek sezgisel bir genetik algoritma(SPGA) önermişlerdir.

Çetin (2007), yapmış olduğu yüksek lisans tez çalışmasında, gezgin satıcı problemlerinin optimum sonuçlarının grid yardımıyla elde etmiştir. Grid, farklı coğrafi konumlardaki organizasyonlar aracılığı ile bilgisayarların hesaplama, uygulama, depolama ve ağ kaynaklarının paylaşılması olarak tanımlanabilir. Bilgisayar kaynaklarının ve depolama kapasitelerinin internet üzerinden paylaşımını sağlayan bir servis olarak düşünülebilir.

Saraç ve Sipahioğlu, 2008'de plastik enjeksiyon makinelerinin çizelgelenmesi problemi üzerine çalışmıştır. Enjeksiyon makinelerinin çizelgelenmesi problemi tek kademe paralel makine çizelgeleme problemleri kapsamına girmektedir. Örnek paket programlar, GAMS paket programının çözücüleri kullanılarak çözülmüştür. Eren, 2013'teki çalışmasında tek makine çizelgeleme problemleri üzerine doğrusal olmayan modeller geliştirmiştir.

Gözen, (2008)'deki çalışmasında esnek akış tipi çizelgeleme problemlerini genetik algoritma ve tavlama benzetimi ile çözerek sonuçları karşılaştırmıştır. Ceran (2006) ise genetik algoritma ile veri madenciliğini birlikte kullanarak çözmüştür. Carlier (1978) tarafından oluşturulan kıyaslama problemleri baz alınmıştır.

Cheng, Huang ve Shiau, 2008'de esnek akış tipi çizelgeleme problemi üzerine hibrit bir genetik algoritma önermişlerdir. Önerilen CGA algoritmasına göre, tabu arama algoritması tarafından geliştirilen popülasyonların uygunluk değerleri incelenmektedir.

Çatal, 2009'daki yüksek lisans tez çalışmasında akış tipi çizelgeleme problemlerini, Taillard'ın deney seti üzerinde, karınca kolonisi ve karınca kolonisini FRB3 algoritmasıyla geliştirerek çözümler üretmiş sonuçları literatürdeki diğer çalışmalarla karşılaştırmıştır. Yağmahan ve Yenisey (2006) 20x5, 20x10 ve 20x20 problem hacmindeki 30 test problemi üzerinde karınca kolonisi kullanarak eniyilemiştir.

Pour (2010)'daki çalışmasında n iş, m makineli akış tipi çizelgeleme problemlerini çözmek için yeni bir sezgisel algoritma geliştirmiştir. Sonuçları NEH, CDS ve Palmer algoritmalarıyla karşılaştırmıştır.

Temiz (2010), yapmış olduğu çalışmada çok kriterli permütasyon tipi çizelgeleme problemlerinde; tamamlanma zamanı, toplam akış zamanı ve en büyük tehir zamanı ölçütlerinin eş zamanlı en iyilenmesini, geliştirilmiş benzetilmiş tavlama algoritması yardımıyla elde edilmiştir. Geliştirilen algoritmanın başlangıç sırası NEH sezgiseliyle oluşturulmuştur.

Çevikcan, Durmuşoğlu ve Baskak, 2011'deki makalelerinde, paralel makinelerde sıraya bağımlı hazırlık sürelerine sahip işlerin çizelgelenmesini ele almıştır. Makalede, İş çizelgelemesinde ürün tasarım özelliklerini dikkate alan bir sistematik sunulmuştur. Geliştirilen sistematiğin yol haritası dahilinde sıraya bağımlı hazırlık süreleri toplamını azaltmaya yönelik olarak ürün tasarım özelliklerine bağlı bir sıralama algoritması geliştirilmiş ve ayrıca literatürde yer alan sıralama algoritmaları incelenmiştir. Eren 2008'de, farklı geliş zamanlı öğrenme etkili paralel makineli çizelgeleme problemi üzerine çalışmıştır. Problemin daha büyük boyutlularını çözmek için çizelgelemede çok kullanılan dağıtım kuralları başlangıç çözümü alınarak tabu arama yöntemi geliştirilmiş ve problemin 500 işe kadar çözümleri gerçekleştirilmiştir. Eren ve Güner, 2005'de iki ölçütlü paralel makineli çizelgeleme problemi üzerine çalışmıştır. Ele alınan performans ölçütleri maksimum tamamlanma zamanı ve maksimum erken bitirmedir. Eroğlu, Ermutlu ve Köksal, 2013'de genetik algoritma kullanılarak bölünebilir ve sıra bağımlı hazırlık süreli işler içeren, ilişkisiz paralel makine çizelgeleme probleminde, en büyük tamamlanma zamanının en küçüklenmesi üzerine çalışmıştır.

Wang ve Qing-dao-er-ji, 2012 yılında yapmış oldukları çalışmalarında, atölye tipi çizelgeleme problemlerini HGA ismini verdikleri melez genetik algoritma yardımıyla çözmüştür. Önerilen algoritmada yeni bir çaprazlama ve mutasyon işlemi önerilmiştir.

Chung ve Liao 2013 yılında yapmış oldukları çalışmalarında, iş akışı çizelgeleme problemleri için, IAIS ismini verdikleri melez çözüm yöntemini geliştirmişlerdir. IAIS'de, yapay bağışıklık sistemindeki seleksiyon, klonlama ve hipermutasyon işlemlerine karşılık olarak, somatik rekombinasyon, somatik hipermutasyon ve izotip değiştirme önerilmiştir.

Qiu ve Lau, 2012 yılında yapmış oldukları çalışmada, static atölye tipi çizelgeleme problemlerini, öncelik kuralını baz alarak melez yapay bağışıklık sistemi yardımıyla çözmüştür. Aynı algoritmayı 2013 yılındaki çalışmalarında çok amaçlı dinamik atölye tipi çizelgeleme problemi üzerinde test etmişlerdir.

Lin ve Ying, 2013 yılında yapmış oldukları çalışmada, permütasyon tipi akış çizelgeleme problemlerini, Taillard'ın deney seti üzerinde, yapay bağışıklık sistemi ve benzetilmiş tavlama algoritmaları melez kullanılarak çözülmüştür. Kullanılan melez çözüme revize edilmiş yapay bağışıklık sistemi (RAIS) adı verilmiştir. Elde edilen sonuçları literatürdeki diğer çözümlerle karşılaştırmıştır.

Lakshmi ve Vasantharathna, 2014 yılında yapmış oldukları çalışmada, yenilenebilir enerjilere dikkat çekmiştir. Elektrik üretim şirketlerinin en az maliyetle maksimum üretim gerçekleştirebilmeleri için termal parametreleri, yapay bağışıklık sistemi yardımıyla optimize edebilmeleri üzerine çalışmıştır.

Ruiz ve Pan, 2014 yılında yapmış oldukları çalışmada, beklemesiz akış tipi çizelgeleme problemi için IG (iterated greedy) yerel arama algoritması önermiştir. Bu algoritma, iki aşamalı operatör kullanarak elde edilen C_{max}'ları karşılaştıran bir yerel arama algoritmasıdır.

2.2. Yapay Bağışıklık Sistemleri ile İlgili Kaynak Araştırması

YBS alanında yapılan çalışmalar 1990'lı yılların sonlarına doğru artmaya başlamıştır. Bugüne kadar anormallik tespiti, optimizasyon ve sınıflandırma gibi problemlerde yapılmış çalışmalar söz konusudur. Aşağıda bu çalışmaların bazılarına kısaca yer verilmiştir.

Matzinger (1994), bağışıklık sisteminin antijenleri nasıl tanıdığını anlayabilmek ve mevcut olan anormallikleri açıklamak için bir teori önermiştir. Bu teori, antijenlerin hücre ölümü veya hücresel baskı başlatarak antijen oluşumunu tetiklediğini ileri sürmektedir.

Forrest ve ark. (1994), negatif seçim algoritmasını sunmuşlardır. Bu algoritma, bağışıklık sisteminde bulunan T lenfositlerinin oluşumunu temel alarak self-nonself hücrelerin ayrımını modellemişlerdir. Hesaplama maliyeti açısından matematiksel bir analiz gerçekleştirilmiş ve bilgisayar virüsleri üzerinde algoritmanın nasıl uygulanabileceği gösterilmiştir.

Dasgupta ve Forrest (1995) self-nonself ayırımını yapan negatif seçim algoritmasını kullanarak değişikliği algılama algoritması sunmuşlardır. Sinyal işleme uygulamalarında problem olarak karşımıza çıkan zaman serisi verilerindeki anormallikler tespit edilmeye çalışılmıştır. 1999'da alet hatası tespiti için algoritma önermişlerdir. Önerilen algoritma, torna operasyonları için simülasyon çalışması ile gösterilmiş, alet bozulması durumunda algoritmanın bunu tespit etme performansı belirlenmiştir.

Forrest ve Hofmeyr (1999), yapay bağışıklık sistemlerinin bilgisayar güvenliğinde kullanılabilecek özelliklerini belirtmişlerdir. Yapay bağışıklık sistem, ayrık algılama bağışıklık modeliyle karşılaştırılmıştır. Modele olan benzerliği ve farklılıkları tartışılmıştır.

Kim ve Bentley (1999), yapay bağışıklık sistemiyle, ağ saldırılarını tespit sistemlerini incelemişlerdir. Çalışmada mevcut saldırı tespit sistemlerinden kısaca bahsedilmiştir. Yapay bağışıklık sisteminin, ağ tabanlı saldırı tespit sistemi geliştirmeye olan katkıları sunulmuştur.

De Castro ve Von Zuben (2000) tarafından yapay bağışıklık sistemleri arasında en önemli yeri tutan klonal seçim algoritması ortaya atılmıştır. Ortaya atılan algoritma antijene karşı bağışıklık tepkisinin temel özelliklerini tanımlamak için bağışıklık sistemindeki klonal seçme mekanizması referans alınmıştır. Bu algoritmadaki temel prensip, antijeni tanıyan hücreleri çoğaltmaktır. Çoğaltılan hücreler duyarlılık olgunlaşması sürecine sokulmuştur. Örüntü tanıma ve optimizasyon gibi görevlerde algoritma test edilmiştir.

De Castro ve Von Zuben (2001), veri analizi, tanıma ve sınıflandırma problemlerini çözmek için Artificial Immune Network adını verdikleri yapay bağışıklık ağ modelini geliştirmişlerdir. Bu sistemde amaç B hücreleri veya yapay tanıma toplarının yerine antikor popülasyonunu geliştirmektedir. Bu modele göre antikor popülasyonu rastgele seçilmektedir. Daha sonra klonal seçim ve mutasyon işlemi uygulamaktadır.

Polat (2004) medikal teşhis üzerine çalışmıştır. UCI makine öğrenmesi veri ambarından alınan Breast Cancer veri kümesi üzerinde değerlendirmişlerdir. Kaymaz (2007) protein örüntülerinin hücredeki yerleşim yerlerinin belirlenmesi üzerine çalışmıştır. Akbal (2007) kablosuz ağlarda saldırı tespiti ve güvenlik üzerine çalışırken Sertkaya (2009) bilgisayar güvenliği üzerine çalışmıştır. Kodaz (2007) Yapay Bağışıklık Tanıma Sistemi (YBTS) incelenmiş ve tespit edilen eksiklikleri gidermek

amacıyla Bilgi Kazancı Tabanlı Yapay Bağışıklık Tanıma Sistemi (BK-YBTS) geliştirilmiştir. Özşen (2008) biyomedikal sınıflama problemleri üzerine uygulamıştır. Diri ve Kaban (2008) tarafından Türkçe metinlerde tür ve yazar tanıma üzerine bir deney gerçekleştirilmiştir. Ulutaş (2008) klonal seçim algoritması ile dinamik yerleşim probleminin çözümü üzerine çalışılmıştır. Yapay bağışıklık sistemiyle ilgili pek çok tez çalışmaları da yapılmıştır.

Akçay (2009) çalışmasında akış tipi çizelgeleme problemleri için çok amaçlı yapay bağışıklık sistemi önermiştir. Önerilen algoritma, yapay bağışıklık algoritmasının klonlama aşamasında çözüm uzayını açısal bölümlere ayırmış, çözümlerin ait oldukları açısal bölümlemedeki yoğunluğa göre klon sayısını belirleyerek çözümler arasındaki çeşitliliği sağlamaya çalışmaktadır. Bir sonraki neslin seçim aşamasında ise pareto baskınlık derecesini kullanmıştır.

3. MATERYAL VE YÖNTEM

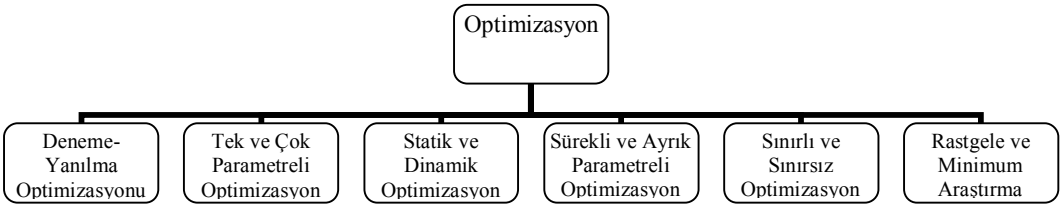
3.1. Optimizasyon

Optimizasyon, bir çözümün daha iyisini bulma süreci olarak tanımlanabilir. Bir mühendis veya bilim adamı yeni bir fikir ortaya koyar. Optimizasyon bu fikrin geliştirilmesine yardımcı olur. Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilir (Karaboğa, 2011).

Bir problemin birden fazla çözümü varsa, en iyi çözümü bulmak gerekir. Aslında "en iyi" tanımlaması izafi kavramdır. Optimum çözüm, problemi formülize eden kişiye göre değişir. Eğitim, fikirler, ideolojiler ve sosyal durumlar gibi optimum çözümü etkileyebilecek bir çok faktör vardır. Bazı problemlerin tam cevabı bulunur. Bazılarının farklı minimum ve maksimum değerleri vardır. İşte tip durumlarda "en iyi" kavramı izafi olabilir (Cunkaş, 2010).

Günlük hayatımız, optimizasyon problemleri ve çözümleri ile doludur. Bununla ilgili birkaç örnek verilebilir. Hem işe zamanında gitmek, hem de uyku miktarını artırmak için ne zaman kalkılmalıdır? İşe giderken en iyi yol neresidir? Bir projeyi yürütürken ilk olarak nereden başlanmalıdır? Bir şeyin uzunluğunu kısaltmak, ağırlığını azaltmak, bir ürünün cazibesini artırmak ve maliyetini düşürmek, en az miktarda kaynak kullanarak, müşteri gereksinimlerine en hızlı yanıt verebilecek tekniklerle en kaliteli ürün ve hizmet üretmek optimizasyon tekniğinin ilgi alanı içerisindedir.

Optimizasyon algoritmaları birkaç grupta ele alınabilir. Şekil 3.1'de optimizasyon türleri gösterilmiştir. Fakat kesin hatlarıyla tüm sınıflandırmanın bu türlerden oluştuğu söylenemez (Broyden, 1965).



Şekil 3.1. Optimizasyon türleri

Deneme-yanılma optimizasyonu; işlem hakkında çok fazla bilgi sahibi olmadan sadece çıkışı etkileyen parametrelerin ayarlanmasıdır. Örneğin televizyonda antenin hangi eğiminde en iyi görüntü ve ses elde edileceği, deneme yanılma yoluyla ayarlanır.

Tek ve çok parametreli optimizasyon; sadece tek parametre varsa, optimizasyon tek parametrelidir. Birden fazla parametreye sahipse çok parametreli optimizasyondur. Parametre sayısı artarsa, optimizasyonun zorluk derecesi de artar.

Statik ve dinamik optimizasyon; statik optimizasyon zamandan bağımsızdır, dinamik optimizasyon ise zamana bağlıdır. Örneğin; bir öğrencinin okula gitmesi için birçok yol olduğunu kabul edilirse en iyi yolun hangisi olduğu sorgulanabilir. Mesafe açısından bakılacak olursa problem statiktir. Zaman açısından bakılacak olursa problem dinamiktir. Değişkenlerin çokluğu nedeniyle problem, pek de basit değildir. En kısa yol, her zaman en hızlı yol olmayabilir.

Sürekli ve ayrık parametreli optimizasyon; sürekli parametreler sonsuz değer alırken, ayrık parametreler sınırlı değerler alır. Ayrık parametreli optimizasyon kombinatoriyel bir optimizasyon olarak da adlandırılabilir ve ayrık niceliklerin optimal olarak düzenlenmesi, gruplanması, sıraya konulması veya seçilmesi problemleridir. Örneğin yapılacak işler bir liste halinde verilmiştir. Bu işlerin yapılması bir birinden bağımsız olduğundan ayrık parametreli düşünülebilir Bir çizgide f(x)'in minimum değerini bulmaya çalışmak ise sürekli parametreli optimizasyon olarak tanımlanır.

Sınırlı ve sınırsız optimizasyon: sınırlı optimizasyon, parametreleri bir değer aralığında tanımlanır. Sınırsız optimizasyonda ise parametreler her hangi bir değerde olabilir. Değişkenlerin sınırları kaldırılarak sınırlı parametreler sınırsız parametrelere çevrilirler. Çoğu nümerik optimizasyon rutinleri sınırsız parametrelerle çalışırlar.

Sınırlı optimizasyon, lineer denklemler ve lineer sınırlarla parametreleri optimize ettiği zaman, program lineer program olarak adlandırılır. Sınırlar ve maliyet denklemleri nonlineer ise, program da nonlineer programlama problemi olarak adlandırılır (Karaboğa, 2011).

Rasgele ve klasik optimizasyon: Bazı algoritmalar parametrelerin başlangıç değerlerini ayarlayarak uygunluk değerlerini minimize etmeye çalışır. Bu araştırma tekniği, hızlı olmakla beraber lokal minimumlara ulaşabilir. Bunlar nümerik metotlara dayanan klasik optimizasyon algoritmalarıdır. Bir parametreden hareketle diğer parametreyi tespit etmek, bazı deterministik adımlarla gerçekleştirilmektedir. Diğer

taraftan rasgele metotlar; parametrelerin optimum çözümünü bulmada ihtimal hesaplarını kullanırlar. Bu metotlar yavaş olmakla birlikte global minimumu bulmada daha başarılıdırlar.

3.2. Üretim Çizelgeleme Problemleri

Günlük hayatımızda çizelgeleme kelimesinden daha çok çizelge kelimesiyle karşılaşırız. Otobüs hareket saatleri çizelgesi, ders programı çizelgesi, sınav programı çizelgesi örnek olarak verilebilir. Çizelge, işlerin ne zaman gerçekleşmesinin hedeflendiğini gösterir.

Çizelgeleme ise; çizelgenin oluşturulması süreci olarak tanımlanabilir. Çizelgelemenin iki aşamadan oluştuğu söylenebilir. Önce faaliyetlerin hangi sıra ile gerçekleştirileceğinin tespiti yapılır. Sonra her bir faaliyetin başlama zamanı hatta tamamlanma zamanı belirlenir (Baker ve Trietsch, 2009).

Çizelgeleme, üretim sanayi ve hizmet sektörlerinde rutin olarak yapılan karar verme sürecidir. Faaliyetlerin hangi sırada olacağına ve ne zaman başlayıp ne zaman biteceğine karar verilmektedir. Belirli bir zaman için, kaynakların işlere atanmasını ve bunu yaparken de belirli amaçların en iyilenmesini hedeflemektedir (Pinedo, 2012).

İstenilen miktarda, istenilen kalitede, istenilen zamanda ve en makul maliyetle üretim yapmak için iyi organize edilmiş bir çizelgelemeye ihtiyaç vardır. Çizelgelemede en önemli amaç, bütün işlerin mümkün olan en kısa sürede tamamlanmasıdır (Mcnaughton, 1959).

Üretim çizelgeleme, bir ürünü oluşturan iş parçalarının eldeki tek veya çok sayıda makinelerde hangi sırada ve ne zaman işleneceğinin belirlenmesidir (Saraç ve Sipahioğlu, 2008).

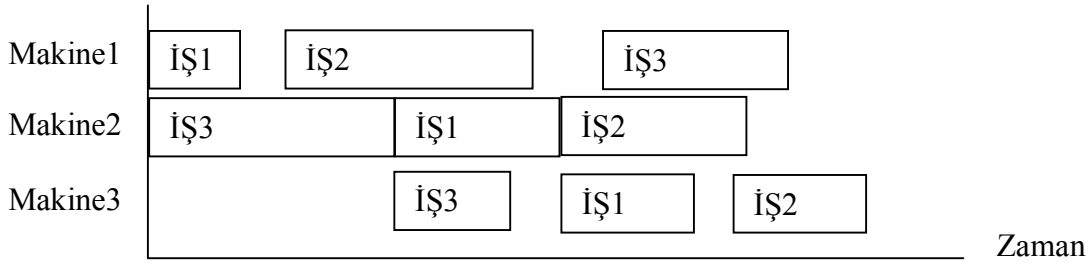
Üretim çizelgelemede hangi makinelerin hangi işleri hangi sıralama ile ne zaman yapacağı belirlenir. Kullanılan makine sayısı ve parça başına iş sayısına göre üretim çizelgeleme sınıflandırılabilir.

Çizelgeleme literatüründe; en basit tek makineli problemden en genel m makineli probleme kadar çok çeşitli modeller bulunmaktadır. Ancak birçok temel araştırmacı, temel çizelgeleme problemlerinde French (1981)'in belirttiği aşağıdaki varsayımları kullanarak tanımlamaktadır (Engin, 2001):

Her operasyon bir bütündür. Bir iş, farklı operasyonlardan meydana gelir. Ancak o işe ait iki operasyon aynı anda gerçekleştirilmez. Operasyonlar bölünemez. Bir makinede bir operasyon başlatıldıktan sonra tamamlanıncaya kadar başka operasyon aynı makinede başlatılamaz. Her iş, her biri bir makinede olmak üzere m adet farklı operasyona sahiptir. Çizelgeleme sırasında herhangi bir iş iptal edilemez. İşlem zamanları çizelgeden bağımsızdır. Makinelerin hazırlık zamanı sıralamadan bağımsızdır. Bir makinenin hazırlık süresinin o makinede en son işlenmiş iş ile bağlantısı yoktur. Her tip makineden yalnız bir adet vardır. Bir işin işlenmesi için makineler arasında bir seçim yapılabilmesi söz konusu değildir. Makinelerin boş kalmalarına izin verilebilir. Çizelgeleme boyunca makineler işler durumdadır. Bozulmaları veya bakım için durdurulmaları söz konusu değildir. Teknolojik kısıtlar önceden belirlenmiş ve sabittir. İş sayısı, makine sayısı, işlerin işlem zamanları, makinelerin hazırlık zamanları ve diğer tüm parametreler önceden biliniyor ve sabittir.

Yukarıda belirtilen varsayımlardan bazıları çözülmesi istenen çizelgeleme probleminin özelliğine göre dikkate alınmayabilir.

Bir iş yerinde n adet farklı iş ve bunların işlem göreceği m adet farklı makine olduğunu kabul edelim. Her makinede bir anda en çok bir iş yapılabilmekte ve her iş bir anda en çok bir makinede işlem görebilmektedir. Üretim çizelgeleme, her bir işin bir makineye ya da birden fazla makineye farklı zamanlarda işlem görmek üzere atanmasıdır. Bu atamanın şematik gösteriminde Gantt şeması kullanılır. Şekil 3.1'de 3 makinede 3 işin çizelgelenmesi görüntülenmektedir. Bir işin bir makinede işlemi tamamlandıktan sonra diğer makinede işlem görebildiğine dikkat edilmelidir (Brucker, 2007).



Şekil 3.2. 3 makine 3 işin Gantt şemasında gösterimi

Problem: Çizelge 3.1'de tek tek her işin her makinede tek başına kaç dakikada işlendiği verilmiştir. İşlerin makinelerden ayrılış sürelerini bularak gantt şeması üzerinde gösteriniz.

Çizelge 3.1. 3x3 şeklinde oluşturulmuş örnek problem

Q	1. Makine	2. Makine	3. Makine
1. İş	3 dk	5 dk	2 dk
2. İş	9 dk	6 dk	2 dk
3. İş	8 dk	7 dk	5 dk

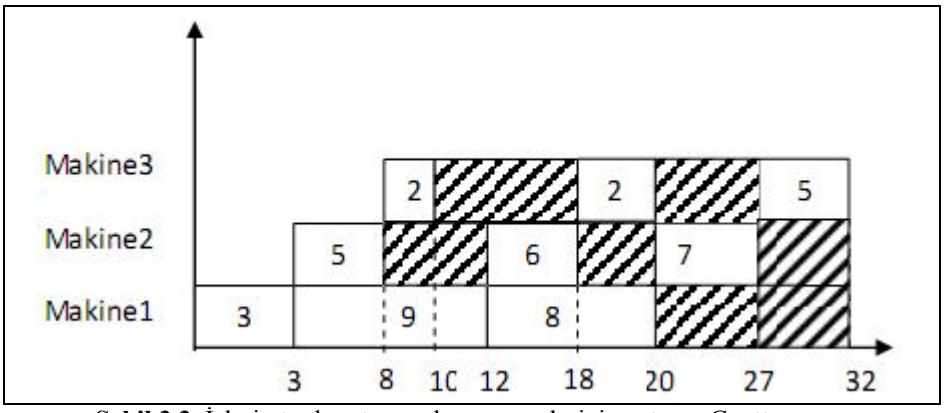
Çözüm: Kod bloğunda her makine işi alırken, kendi işini ne kadar sürede işlemiş (q[i-1, j]) ve alacağı işin bir önceki makinede işi kaç dakikada bitmiş (q[i, j-1]) kontrol edilir ve bu sürelerden büyük olanı seçilir, üzerine de makinenin o andaki işi işleme süresi eklenir. Böylece işin önceki makinelerle birlikte o makinedeki toplam işlenme süresi bulunur.

Çizelge 3.2'de işler 1. iş, 2. iş, 3. iş sırasında işlendiğinde makinelerden kaçıncı dakikalarda ayrıldıkları gösterilmiştir. En son 3. Makinede 3. iş işlendiğinde tüm işler tamamlanmış olur.

Çizelge 3.2. 3x3 örnek problemindeki işlerin makinelerden ayrılış süreleri

Q	1. Makine	2. Makine	3. Makine
1. İş	3 dk	8 dk	10 dk
2. İş	12 dk	18 dk	20 dk
3. İş	20 dk	27 dk	32 dk

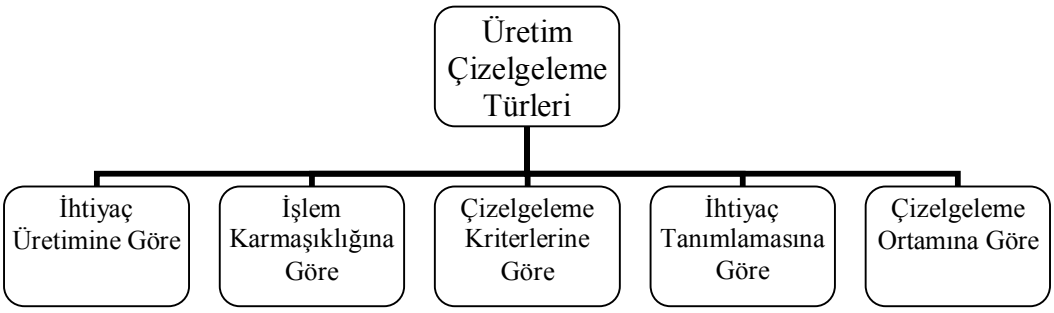
Ardından elde edilen değerlere göre gantt şeması oluşturulur. Şekil 3.3'te işlerin toplam tamamlanma süreleri gantt şeması üzerinde gösterilmiştir.



Şekil 3.3. İşlerin toplam tamamlanma sürelerini gösteren Gantt şeması

3.2.1. Üretim çizelgelemenin sınıflandırılması

Üretim çizelgelemenin boyutları ihtiyaç üretimine göre, işlem karmaşıklığına göre, çizelgeleme kriterlerine göre, ihtiyaç tanımlamasına göre ve çizelgeleme ortamına göre sınıflandırılmaktadır. Şekil 3.4'de bu sınıflandırmanın detayları görülmektedir.



Şekil 3.4. Üretim çizelgelemenin sınıflandırılması

İhtiyaç üretimine göre çizelgelemede, ihtiyaçlar açık ve kapalı atölyede üretilmektedir.

Açık atölye tipi ihtiyaç üretiminde, ihtiyaçlar doğrudan doğruya müşteri siparişleri doğrultusunda üretilirken, *kapalı atölye tipi ihtiyaç üretiminde* ise ihtiyaçlar stoktan karşılanmaktadır.

İşlem karmaşıklığına göre çizelgeleme 3 sınıfa ayrılmaktadır. Tek kademe tek makine problemleri, tek kademe paralel makine problemleri, çok kademeli problemler.

Tek kademe, tek makine problemleri, en basit problem biçimidir. Bütün işler sadece tek makinede yapılmaktadır. Bu işlem basittir fakat diğer karmaşık problemlerin temelini oluşturması açısından önemlidir.

Tek kademe, paralel makineler problemleri ise, her bir iş paralel makinelerden birisinde işlenmek üzere yine tek bir işlem kademesi gerekmektedir. Ancak bu problemlerde aynı işi yapan birden fazla makine mevcuttur. Vida üretimin yapıldığı atölye tipi buna örnek gösterilebilir (Çevikcan, Durmuşoğlu ve Baskak, 2011).

Çok kademeli problemler, her bir işin işlem sırasında çok kesin bir öncelik ilişkisinin bulunduğu durumlardır. Her bir iş, makineler grubunda öncelik ilişkisine göre işlenmeyi gerektirir. Çok kademeli problemler, akış tipi ve atölye tipi olmak üzere iki şekilde incelenebilir (Saraç ve Sipahioğlu, 2009).

- *Akış tipi çizelgeleme problemlerinde,* tüm işler aynı işlem sırasıyla aynı makine grubunda işlenir. Diğer bir deyişle, işlerin makinelerdeki işlem sırası ve öncelik ilişkisi aynıdır.
- *Atölye tipi çizelgeleme problemleri,* sınıflandırmada en genel ve en karmaşık olan çizelgeleme problemidir. Belli bir işe ait işlem kademeleri sayısı hakkında hiçbir sınırlama yoktur ve alternatif olarak kabul edilebilecek rotalar mevcuttur. Atölye tipi çizelgelemede, her iş farklı makinelerde işlenmek üzere kendine özgü işlem ve işlem sıralarına sahiptir ve diğer işlerin işlem sıralarından bağımsızdır. Atölye tipi çizelgelemede ana unsur makineler ve bunlar üzerindeki işlerdir

İhtiyaç tanımlama durumlarına göre üretim çizelgeleme deterministik ve stokastik olmak üzere 2 sınıfa ayrılır.

Eğer tüm parametrelerin nümerik değerleri önceden biliniyorsa ve sabitse belirgin *deterministik* olarak tanımlanır. Aksi takdirde; parametrelerin nümerik değerleri tam olarak bilinmiyorsa ve belli bir olasılık dağılımıyla rastgele değişken olursa bu tipteki atölye belirsiz *stokastik* olarak tanımlanır. Gerçek uygulamalarda ve çizelgelemelerde karşılaşılan problemlerin çoğu belirsiz ve dinamiktir. Üretilen çözüm modellerinin çoğu ise belirli ve statik olarak kabul edilir. Böyle kabullerin sebebi ise; statik ve belirli problemleri anlamadan dinamik ve belirsiz problemleri anlamak zordur (Biroğul, 2005).

Üretim çizelgeleme, çizelgeleme ortamına göre ise dinamik ve statik olmak üzere 2 sınıfa ayrılmaktadır.

Statik çizelgeleme ortamında, çizelgeleme zamanı boyunca üretilecek gereksinimlerin miktarı ve buna bağlı olarak atölye ortamına giren işlerin miktarı belirlenir ve sonradan atölye ortamına ek bir iş girişi yapılmaz.

Dinamik çizelgeleme ortamında ise, çizelgeleme zamanı boyunca üretilecek olan gereksinimlerin miktarı ve buna bağlı giren iş miktarına sonradan ilave yapılabilecek şekilde problem tanımlanabilir. Atölye ortamına herhangi bir anda ya da özel bir durumda yeni iş girdileri olabilir (Çatal, 2009).

3.2.2. Çizelgeleme probleminin çözülebilirliği ve karmaşıklık

Çizelgeleme problemleri kombinasyonlu yapıdadır. Örneğin, n adet işin m adet makinede yapılmasının $(n!)^m$ adet farklı çizelgeleme şekli vardır. Küçük n ve m sayıları için problemin çözümü zor görünmeyebilir fakat n ve m değerleri büyüdükçe hesaplama zorlaşmaktadır.

Bir algoritmanın çalışma süresi algoritmanın adımları ile ölçülür. Bu ise $T(n)$ şeklinde bir fonksiyon ile ifade edilebilir. Oluşan fonksiyonun en hızlı büyüyen terimi algoritmanın işlem zamanını gösterir. Bu ise $T(n)=O(g(n))$ şeklinde gösterilir ki $g(n)$ fonksiyonun en hızlı büyüyen terimi $g(n)=n^{100}$ ise $O(n^{100})$ ile çalışma süresi gösterilen bu algoritmaya polinom zamanlı algoritma denir. $g(n)=2^n$ ifadesi ise bu algoritmanın üssel zamanlı algoritma olduğunu gösterir. Üssel zamanlı algoritmalar, polinom zamanlı algoritmalardan daha hızlı büyümektedir (Altındaş, 2011).

Problemlerin karmaşıklığının temelde iki başlık altında sınıflandığı görülmektedir. Bunlar P sınıfı ve NP sınıfı problemlerdir.

P Sınıfı: Bu tip problemlerin çalışma süresi, girilen verinin büyüklüğüne bir polinom cinsinden bağlıdır. Bunlara polinomsal zamanda çalışan algoritma denilmektedir. Bu tip problemlerin, karmaşıklığı polinomsaldır ve kısaca "P" şeklinde gösterilmektedir.

NP Sınıfı: Bu tip problemlerde sorulan soru ve girilen verinin büyüklüğüne polinom mertebesinde bağımlı bir sürede çözüm algoritması bulunmamaktadır. Bu tür problemlerin cevabı tahmin edilebiliyorsa tahminin doğruluğunu sınamak için veri büyüklüğüne polinom mertebesinde bağımlı sürelerde çalışabilecek algoritmalar kullanılmaktadır. İşte bu tip problemlere NP (non-deterministic polinomial time) sınıfı problemler adı verilir ve "NP" şeklinde gösterilmektedir.

Eğer NP sınıfındaki her problem polinomiyal zamanda, bir P problemine dönüştürülebiliyorsa, ilgili P problemine NP-Zor problem denir. Eğer P probleminin kendisi de NP sınıfında bir problemse bu durumda P problemine NP-Tam problem denir (Reeves, 1995).

NP-Tam sınıfındaki her hangi bir problem için polinomiyal zamanlı bir algoritma bulunabilirse, bu sınıftaki tüm problemler için de polinomiyal zamanlı bir algoritma bulunabilir. Bu durumda P=NP olduğu gösterilmiş olacaktır (Kellegöz, 2006).

Kombinatoriyel optimizasyon problemlerinin çoğu NP-tam, polinomiyal zaman sınırı olmayan problemler sınıfına girmektedir. NP problemlerine, polinomiyal

algoritma geliştirilememiştir. NP kapsamında yer alan problemler için asıl optimum çözüm yerine yakın çözümler tercih edilir.

Matematikçiler, NP içerisinde P'ye asla taşınamayacak problemlerin bulunduğunu ortaya koymuşlardır. Bir başka deyişle, çok zor oldukları için bazı problemlere ait polinomsal zaman karmaşıklık fonksiyonuna sahip olan algoritmaların asla bulunamayacağına inanırlar.

3.2.3. Üretim çizelgeleme parametreleri

Herhangi bir üretim çizelgeleme problemini tanımlamada $\alpha \mid \beta \mid \ell$ şeklinde bir gösterim kullanılmaktadır. "α" o probleme ait makine parametresini, "β" iş parametresini ve "ℓ" ise bu problemde hangi amaç fonksiyonun en iyi yapılmaya çalışıldığını göstermektedir (Graham ve ark., 1979).

Çizelgeleme problemlerinde, çizelgelenecek işlerin ve makinelerin önceden bilindiği varsayımı unutulmamalıdır. Çizelgelenecek işlerin sayısı n ve makinelerin sayısı ise m ile ifade edilmektedir. İş ve makine sayıları alt indis olarak kullanıldıklarında; j iş sayısına ve i makine sayısına karşılık gelmektedir. Eğer iş bir dizi işlem veya operasyon gerektiriyorsa *(i,j)* çifti; j işinin i makinesindeki operasyon adımını ifade etmektedir. Çizelgelenecek i işi ile ilgili diğer gerekli bilgiler ise şunlardır (Pinedo, 2012):

Üretim çizelgelemede kullanılan bazı tanımlar ve üçlü gösterimin parametreleri açıklanmıştır. Üretim çizelgeleme problemleri ile ilgili genel tanımlar şunlardır:

İşlem zamanı (p_{ij}); j işinin i makinesindeki işlem zamanını göstermektedir. Eğer j işinin işlem zamanı makineden bağımsız veya sadece tek makinede işlem söz konusu olduğu durumlarda ise p_j ile gösterilir.

Hazır olma zamanı (r_j); j işinin sistemde işlem görmeye başlayabileceği zamanı gösterir.

Teslim tarihi (d_j); j işinin tamamlanmış olarak müşteriye teslim edileceği tarihi gösterir. Teslim tarihinden sonra işi teslim etmek mümkün fakat istenmeyen bir durumdur.

Tamamlanma zamanı (C_j), j işinin bütün işlemleri tamamlandıktan sonra sistemden çıktığı zamandır. Her makine için işlem tamamlanma zamanı C_{ij} şeklinde gösterilir ve j işinin i makinesinde işlem görmesinin bittiği zamanı ifade eder.

Gecikme süresi ($L_j=C_j-d_j$), j işinin tamamlanma zamanının teslim tarihinden farkıdır. Tamamlanma zamanı teslim tarihinden büyükse pozitif değer alır, küçükse negatif değer alır.

Ağırlık (w_j), j işinin sistemdeki diğer işlere göre önemini gösterir. Örneğin, bu ağırlık faktörü işlerin maliyeti olabileceği gibi, işe verilen değer de olabilir

Gecikmişlik göstergesi (U_j); j işinin gecikip gecikmediğini gösterir. J işinin gecikme süresi sıfırdan büyükse göstergenin değeri 1 değilse değeri sıfırdır.

Üretim çizelgeleme problemlerinde karşılaşılan makine parametreleri şunlardır:

Tek makine ($\alpha=1$); işleri yapmak üzere tek bir makine mevcuttur. Her iş aynı makinede işlem görür.

Akış tipi üretim sistemi ($\alpha=F_m$); akış tipi üretim sisteminde birbirine seri olarak bağlı m adet makineyi ifade eder. Tüm işler aynı sırayı izleyerek işlem görürler. Bir makinedeki işlemi tamamlanan iş, sonraki makinede işlem görmek üzere kuyruğa girer ve genellikle ilk giren ilk çıkar kuralına göre işlem görürler. Bu kurala göre işlem gören sistemler, permütasyon akış tipi üretim sistemi olarak nitelendirilir

Paralel makineler ($\alpha=P_m$); birbiriyle aynı özelliklere sahip m adet paralel makineyi ifade eder. P_5 gösterimi sistemde 5 adet paralel makine olduğunu ifade etmektedir. İşler paralel bağlı makinelerin her hangi birinde işlem görebilirler.

Atölye tipi üretim sistemi ($\alpha=J_c$); atölye tipi yerleşimin ve paralel makine sisteminin genel halidir. J_5 gösterimi sistemde 5 adet farklı iş merkezi olduğunu ifade etmektedir. Her bir iş merkezinde belirli sayıda paralel makine bulunmaktadır.

Açık tip üretim sistemi ($\alpha=O_m$); n adet işin m adet makinede ki işlemlerinin herhangi bir sıralama şartı olmaksızın yerine getirilmesidir. Açık tip üretim sisteminde her bir iş m adet makinenin her birinde tekrar işlem görebilir. İşlerin makinelerdeki işlem sıralarına ait bir sınırlama yoktur. Her bir iş için bir sıra hazırlanır ve farklı işler farklı sıralara sahip olabilir. O_5 gösterimi sistemde 5 adet farklı makine olduğunu ifade etmektedir.

Üretim çizelgeleme problemlerinde karşılaşılan iş parametreleri şunlardır:

Hazır olma zamanı (r_j); j işinin sistemde işlem görmeye başlayabileceği zamanı gösterir. Eğer β alanında r_j değeri bulunmuyorsa, bu durumda j işinin herhangi bir zamanda başlayabileceğini ifade eder.

Bir işi bitirmeden başka işe başlama (prmp); işlerin bölünebilirliği ifade eder. Bir makinede işlem görmeye başlayan bir işin yarım bırakılıp yeni bir işe başlanabilmesi durumu olarak tanımlanır. Eğer çizelgeleme problemlerinde işlerin

bölünebilirliğine izin verilmesi isteniyorsa β alanında *prmp* ifadesi bulunur. Aksi takdirde işlerin bir makinedeki işlemleri boyunca bölünmesine izin verilmez.

Öncelik sınırlaması (prec); öncelik sınırlaması tek makine veya paralel makine ortamlarında kullanılır. Bir veya daha fazla işin bir makinede, diğer işler işlem görmeden önce tamamlanması gerektiğini ifade eder. Eğer β alanında *prec* ifadesi yoksa işler öncelik sınırlamasına tabi değildir.

Sıra (prmu); akış tipi üretim çizelgeleme problemlerinde görülen bir parametredir. Makinelerin önünde işlem görmek için kuyrukta bekleyen işlerin ilk giren ilk çıkar kuralına göre işleme alınmasını ifade eder. Bu durum bir sıralama ve permutasyonu ifade eder. Permütasyon tipi akış çizelgeleme problemlerinde kullanılır.

Beklemelere izin verilmeme (nwt); akış tipi üretim çizelgeleme problemlerinde görülen bir parametredir. Birbirini takip eden iki makine arasında işlerin beklemesine izin verilmemesini ifade eder. İşlerin beklemesine izin verilmemesi isteniliyorsa β alanına *nwt* parametresi yazılır.

Tekrar işlem görme (rcrc); atölye tipi üretim çizelgeleme problemlerinde görülür. Bir işin bir makineyi veya iş merkezini birden fazla sayıda ziyaret etmesidir. İşlerin tekrar edilmesi isteniliyorsa β alanına *rcrc* parametresi yazılır.

Üretim çizelgeleme problemlerinde karşılaşılan başlıca amaç parametreleri ise şunlardır:

Maksimum tamamlanma zamanı; (C_{max}) ile ifade edilir. Bir işin en son operasyonunun tamamlandığı zamana tamamlanma zamanı denir.

Maksimum gecikme zamanı; (L_{max}) ile ifade edilir. Bir işin tamamlanmasının teslim tarihinden sapmalarının bütün işler içinde en büyük olanını gösterir. L_j gösterimi, j işinin gecikme zamanıdır. Bunun en küçük değerlere indirgenmesi müşterinin gecikmeden kaynaklanan şikayetlerinin minimize edilmesini sağlar.

Tehir zamanı (T_j); j işinin tamamlanma zamanı teslim tarihinden sonra gerçekleşmesini yani ($L_j>=0$) durumunu ifade eder.

Gecikmiş işlerin sayısı (U_j); j işinin tamamlanma zamanı teslim tarihinden sonra gerçekleşen işlerin sayısının ifade eder.

3.2.4. Tek makine problemleri

Tek makinede çizelgeleme işleminde işlerin sıralanması söz konusu olmaktadır. Daha önceki bölümlerde de açıklandığı üzere işlerin sıralanması *(n!)* alternatif ile gerçekleştirilmektedir. Problem şu şekilde tanımlanabilir:

İşler *(j =1,2,3…,n)* işlem görmek için hazırdır. P_j, w_j, d_j, sırasıyla j işinin işlem zamanını, ağırlığını ve teslim tarihini vermektedir. P_r, w_r, d_r, sırasıyla r. pozisyona atanan işin işlem zamanını, ağırlığını ve teslim tarihini vermektedir. Eğer j işi, r poziyonuna atanırsa işin işlem zamanı Denklem (3.1)'deki gibi tanımlanır.

$$P_{jr} = P_j\left(1 - \frac{\sum_{i=1}^{r-1} P_i}{\sum_{i=1}^{n} P_i}\right)^{\alpha} = P_j\left(1 - \frac{\sum_{i=r}^{n} P_i}{\sum_{i=1}^{n} P_i}\right)^{\alpha} \quad (3.1)$$

$\alpha \geq 1$ öğrenme indisidir. C_r, r pozisyonundaki işin tamamlanma zamanını verir. L_r ve U_r, r pozisyonundaki işin gecikme zamanını ve gecikme olup olmamasını verir.

$$L_r = C_r - d_r \text{ ve } U_r = \begin{cases} 1 & Lr > 0 \\ 0 & Lr \leq 0 \end{cases} \quad (3.2)$$

Bu durum Denklem (3.2)'deki gibi ifade edilir. Maksimum gecikme zamanı, geciken işlerin sayısı, geciken işlerin ağırlık sayıları(önemi) sırasıyla (3.3), (3.4), (3.5)'de aşağıda ifade edilmiştir (Eren, 2012).

$$L_{max} = max_r L_r \quad (3.3)$$

$$n_t = \sum_{r=1}^{n} Ur \quad (3.4)$$

$$n_{wt} = \sum_{r=1}^{n} wrUr \quad (3.5)$$

3.2.5. Paralel makine problemleri

Paralel Makineler, aynı paralel makineler, aynı tip fakat farklı hızlarda paralel makineler veya farklı paralel makineler olmak üzere 3 farklı şekilde karşımıza çıkabilir. Birbirinin aynı olan paralel makinelerde işlem süresi $p_{ij}=p_j$ şeklinde gösterilir. İşlem süresi sadece işe bağlıdır. Tek tip fakat farklı hızlarda paralel makinelerde işlem süresi işin yanında makinenin hızına da bağlı olduğu için $p_{ij}=p_j/s_i$ şeklinde ifade edilebilir. Burada s_i, i makinesinin hızını gösterir. Farklı makinelerde ise işlem süresi hem

makineye hem de işe bağlıdır. Bu yüzden işlem süresi olan $p_{ij}=p_j/s_{ij}$ şeklinde gösterilir. Burada s_{ij}, i makinesinin j işini yapmaktaki hızını ifade eder (Shmoys, 1991).

Makinelerin sayısının, çeşitlerinin ve hızlarının artmasıyla problemlerin karmaşıklığı artar. Bu yüzden birbirinin aynı olan paralel makineler üzerinde durulacaktır. Problem şu şekilde tanımlanabilir:

Üretim ortamına gelen n adet iş sisteme herhangi bir zamanda gelebilir. Paralel makineli sistemde gelen işler *(j = 1,2,..., n)* mevcut paralel makinelerin *(i = 1,2,...,m)* herhangi birinde işlem görebilir. *rj* ve *pj*, j işinin geliş zamanı ve işlem zamanını göstermektedir. Bir işin işlem zamanı öğrenme etkisi olduğunda sıradaki pozisyonun bir fonksiyonu olarak azalır. j işi i. makinede k. pozisyonda çizelgeleniyor ise bu işin işlem zamanı $p_{jik}=p_{ji}k^a$ olarak ifade edilir. Burada $a \leq 0$ olan öğrenme indeksi sabitidir ve öğrenme oranının iki tabanına göre logaritması olarak verilir. C_{ji}, j işinin i. makinede tamamlanma zamanını ifade etmektedir. Makineler özdeş makinelerdir. Bu makinelere işlerin atanması örneği olarak 10 işli 2 makineli bir durum dikkate alındığında $(n_1,n_2)=(9,1);(8,2);(7,3);(6,4)$ *ve* $(5,5)$ olmaktadır. Burada n_1 ve n_2 birinci ve ikinci makinelere atanan işlerin sayısını göstermektedir (Eren, 2008).

3.2.6. Atölye tipi üretim çizelgeleme problemleri

Atölye tipi üretim çizelgeleme problemlerinde, $J_1,J_2,...,J_n$'den oluşan işlerin $M_1,M_2,...,M_n$'den oluşan makinelerde işlem görmesi gerekmektedir. Her işin belli bir işlem sırası olup farklı makinelerde işlem yapılamaz fakat bir sonraki makinede işlem görmek için bekliyor olabilir. Her J_j işinin, $O_{j1},O_{j2},...,O_{jm}$ şeklinde işlem görme sırası mevcut olup, çizelgelemelerde bu sıralama korunmalıdır. Yani O_{21} işlemi 2 nolu işin 1 nolu makinede işlem görmesi zorunluluğunu gösterir. İşlem süresi p_{32} şeklinde gösterilir. C_{max}, son işin tamamlanma zamanına eşit olan çizelge zamanı, ayni zamanda tüm n işin tamamlanması için gerekli süredir.

Atölye tipi üretim çizelgeleme problemlerinde, literatürde yapılan çalışmalarda genelde temel varsayımlar şu şekildedir:

Bütün işler işlenmek ve bütün makineler işleri işlemek için aynı anda hazır durumdadır ve birbirinden bağımsızdır. Bir sınırsız zaman dilimi boyunca bütün işler ve makineler hazır olarak kalacaktır. Bir iş aynı anda yalnızca bir makine tarafından işlenebilir (Erdiller ve Orbak, 2003).

3.2.7. Akış tipi üretim çizelgeleme problemleri

Akış tipi çizelgeleme, birbirinden farklı, m makine ve n işin bulunduğu, her işin m operasyondan oluştuğu, her bir operasyonun farklı makinelerde yapıldığı ve bütün işlerin operasyonlarının aynı sıra ile yapıldığı problemlere denir (Engin, 2001).

Akış tipi üretim çizelgelemede, bütün işlerin birden çok işlem basamağı vardır ve bütün işlerin bu işlem basamaklardan aynı sırada geçmek zorundadır. Diğer bir ifadeyle $J_1,J_2,...,J_n$'e kadar olan işlerin $M_1,M_2,...,M_n$'e kadar olan makinede aynı sırada işlem görmesidir. İşlem süresi $p_{ij,}$ i işinin j makinesindeki işlem zamanını ifade eder ve sabit, pozitif veya sıfır olabilir.

Literatürde yapılan çalışmalarda genelde temel varsayımlar şu şekildedir:

Bir iş herhangi bir anda en fazla bir makinede işlem görebilir. İşlem parçalara bölünemez. Bütün işler işlenmek ve bütün makineler işleri işlemek için aynı anda hazır durumdadır. Hazırlık zamanı kullanılacaksa sıralamadan bağımsızdır. Her iş örneğin önce 1. makine, sonra 3. makine, sonra 4. makine olmak üzere sıralı olarak son makineye kadar sabit bir sırayı takip eder (Stützle, 1998).

3.2.7.1. Permütasyon akış tipi üretim çizelgeleme problemleri

Bu tip iş akış tekniği seri imalat yapan işletmelerde ortaya çıkan problemlere çözüm üretmek için kullanılır. Her bir aşamada işlerin aynı sıralamayı takip ederek bir anda bir makinede işlem gördüğü ve sonra diğer makinede işlem gördüğü çizelgeleme çeşididir. Bu tip üretim atölyesinde makineler bir hat boyunca dizilirler ve işler bu hattı takip ederler. 2 makineli akış tipi üretim çizelgeleme problemi $F_2||C_{max}$ şeklinde gösterilir. Polinom zamanlı çözümü olan tek akış tipi üretim çizelgeleme problemidir. Eğer makine sayısı 2'den fazla olursa, çözüm NP-Zordur. Çözüm için çok sayıda sezgisel yöntem geliştirilmiştir (Döyen, 2004).

Literatürde yapılan çalışmalarda genelde temel varsayımlar şu şekildedir:

Tüm işlerin ve makinelerin t = 0 anında hazır oldukları kabul edilir. Bir iş bir makineye atandığında, o iş tamamlanmadan aynı makinede başka bir işlem yapılmaz. Bir makinede operasyonun başlatılması için işin bir önceki makinede tamamlanmış olması gerekir. Makinelerde bozulma gibi sorunlar göz ardı edilir. Makinelerin

hazırlanma süreleri işlem süresinin içinde kabul edilir. İşlerin makineler arasındaki taşıma süresi ihmal edilir.

3.2.7.2. Esnek akış tipi üretim çizelgeleme problemleri

Klasik akış tipinde her bir işlem aşamasında tek makine yerine birbirinin aynı olan makineler paralel olarak yerleştirilir ve çalıştırılırsa bu yapılan permütasyon tipi akış çizelgelemenin bir genellemesi olup esnek akış tipi çizelgeleme adını alır. Akış tipi çizelgeleme problemleri ile paralel makine problemlerinin özelliklerini birleşimi denilebilir.

Esnek akış tipi sistemde, makineler s tane seri kademeye yerleştirilmişlerdir. $l=1,2,...s$ olmak üzere bir l kademesinde, m_l tane eş makine bulunmaktadır. $j=1,2,...n$ olmak üzere bir j işi, her bir kademedeki makinelerden herhangi birinde yada daha fazlasında işlem görmelidir. Farklı kademelerde j işinin işlem süreleri $p_{1j},p_{2j},...p_{sj}$ ile gösterilir. İşlerin önceliği yoktur, örneğin, bir makinede bir operasyon başladıktan sonra başka bir operasyonun aynı makinede işlem görmeye başlayabilmesi için öncekinin mutlaka bitirilmesi gerekir. Her makinede belli bir anda en fazla bir iş işlem görebilir. Her kademede işlem görmeyi bekleyen işler için ayrılan stok alanı kısıtsız kabul edilir. Amaç, genellikle en son işin sistemden ayrılma zamanını (C_{max}), minimize etmektir. Esnek akış tipi problemler NP-Zor'dur (Gupta, 1988).

Bu yüzden ikiden fazla aşamadan oluşan çok sayıda işten oluşan daha karmaşık esnek akış tipi üretim sistemlerinin çözümünün de NP-Zor olduğu söylenebilir.

3.3. Sezgisel Yöntemler

Cheung (1994), çoğu optimizasyon problemlerinin mümkün olan çözümleri için gerekli hesaplama sayısının problem boyutunun faktöriyeli (n iş ve m makine için $(n!)^m$ adet küme elemanı) kadar olduğundan, optimum çözüm için tam sayma yapılmasının her zaman mümkün olmadığını belirtmiştir (Akyol, 2006).

Yeşilyaprak (2007), birçok çizelgeleme problemi NP-Zor olduğu için tam çözüm yöntemleri çözümü mümkün olmadığını belirtmiştir. Bu yüzden yaklaşık çözüm üretmek için çeşitli sezgisel algoritmalar geliştirilmiştir.

Sezgisel algoritmalar, çözüm uzayında optimum çözüme yakınsaması ispat edilemeyen algoritmalar olarak da adlandırılır. Bu tür algoritmalar yakınsama özelliğine

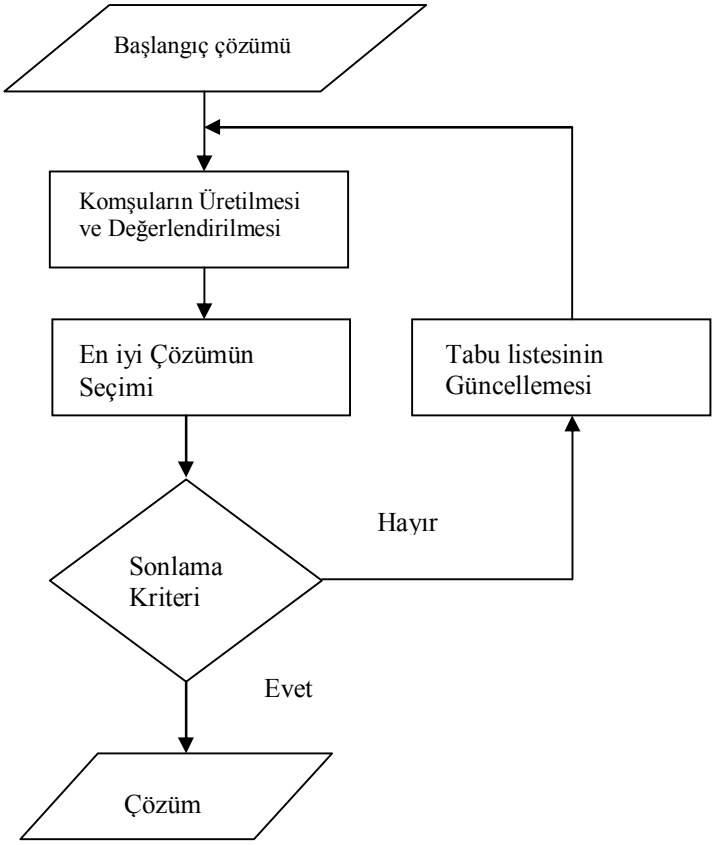
sahiptir, ama kesin çözümü garanti edemezler ve sadece kesin çözüm yakınındaki bir çözümü garanti edebilirler (Karaboğa, 2011).

Literatürde, geliştirilen pek çok sezgisel algoritma mevcuttur. Bu başlık altında Tabu Arama, Benzetilmiş Tavlama, Genetik Algoritma, NEH ve FRB yaklaşımları anlatılmıştır.

3.3.1. Tabu arama

İlk olarak Hansen tarafından önerilmiş olup Glover (1989) tarafından önerilmiş bir sezgisel programlama tekniğidir. Tabu arama yerel optimumlardan kurtulmayı sağlamak amacı ile yerel arama metotlarına yol gösteren sezgisel bir tekniktir. Tabu aramada, kısa dönemli hafıza ve uzun dönemli hafıza olmak üzere iki farklı hafıza yapısı vardır.

Tabu aramada öncelikle başlangıç çözümü oluşturulur. Bu çözüm rastsal olarak oluşturulabileceği gibi herhangi bir algoritma yardımıyla da oluşturulabilir. Mevcut çözümler üzerinde yapılan değişiklikle elde edilebilecek yeni çözümler belirlenerek değerlendirilir. En iyi çözümler seçilir. Tabu yıkma kriterleri, bir hareketin yasak olmasına rağmen yapılabileceği durumları belirtir. En basit ve en sık kullanılan tabu yıkma kriteri, aramada mevcut iterasyona kadar bulunan en iyi çözümden daha iyi bir çözüm veren hareketin tabu olup olmamasına bakılmaksızın yapılmasıdır. Burada yasaklama, sadece tabu listesini güncellemek ve kontrol etmekle gerçekleştirilmektedir. Yasaklama stratejisinde hedef, araştırmanın daha önce görülen çözümlere tekrar dönmesini önlemek ve araştırmayı yeni bölgelere yönlendirmektir. Aday liste, tabu olmaya yakın en iyi hareketin, makul zaman ve çaba ile seçilmesi kuralıdır. En iyi hareket amaç fonksiyonu açısından düşünüldüğünde en iyi amaç fonksiyonu değerini veren harekettir. Örneğin mevcut x çözümün komşu kümesi, bu kümedeki elemanların değerlendirilmesinin hesaplama maliyeti yüksek ise, aday liste kullanımı kaçınılmazdır (Memiş, 2008). Tabu arama algoritmasının genel akış şeması Şekil 3.5'de verilmiştir.



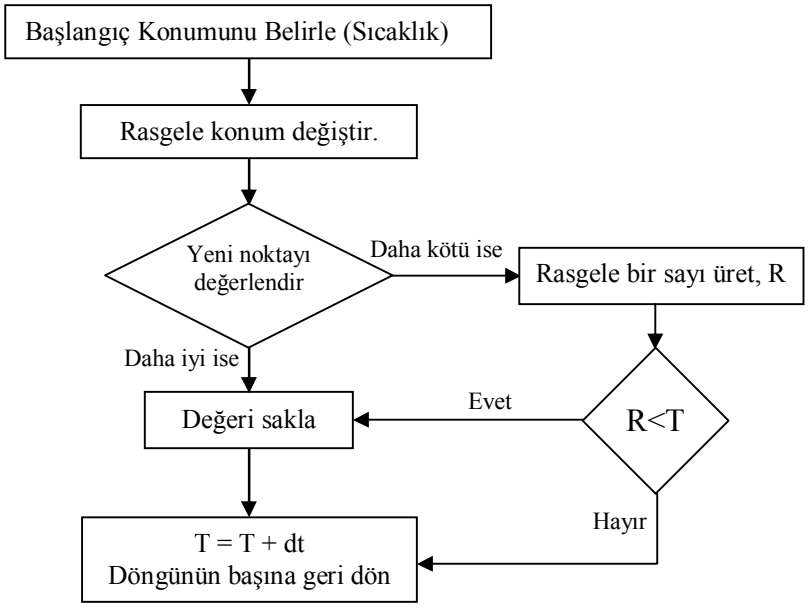
Şekil 3.5. Tabu Arama Algoritması akış şeması

3.3.2. Benzetilmiş tavlama

Benzetilmiş tavlama, Kirkpatrick ve arkadaşları tarafından 1983'te ortaya çıkmıştır. İterasyon geliştirmeye dayalıdır ve katı hal fiziğinden esinlenilerek çıkartılmış sezgisel bir optimizasyon tekniğidir (Ogbu ve Smith, 1990).

Tavlama, katı metalin eritilmesi ve takip eden soğutma işlemi neticesinde uygun kristalleşme sonucunda en iyi kafes yapısını elde etmeyi amaçlayan fiziksel bir işlemdir. Bu işlem süre-since katı metalin serbest enerjisi azaltılmaya çalışılır (Aarts ve Jan, 1989).

Tavlama algoritması çalışma prensibi olarak, yerel arama algoritmalarına benzer. Yerel arama algoritmalarının en büyük dezavantajı, global optimum yerine lokal optimum bulmasıdır. Tavlama benzetimi, bu olumsuzluğu ortadan kaldırır. Kontrol parametrelerinden sıcaklığa bağlı olarak, komşuluk değerlerindeki maliyet fonksiyonu değişimini kabul eder (Engin ve Kaya, 2009). Şekil 3.6'da benzetilmiş tavlama algoritmasının akış şeması verilmiştir.



Şekil 3.6. Benzetilmiş Tavlama Algoritması akış şeması

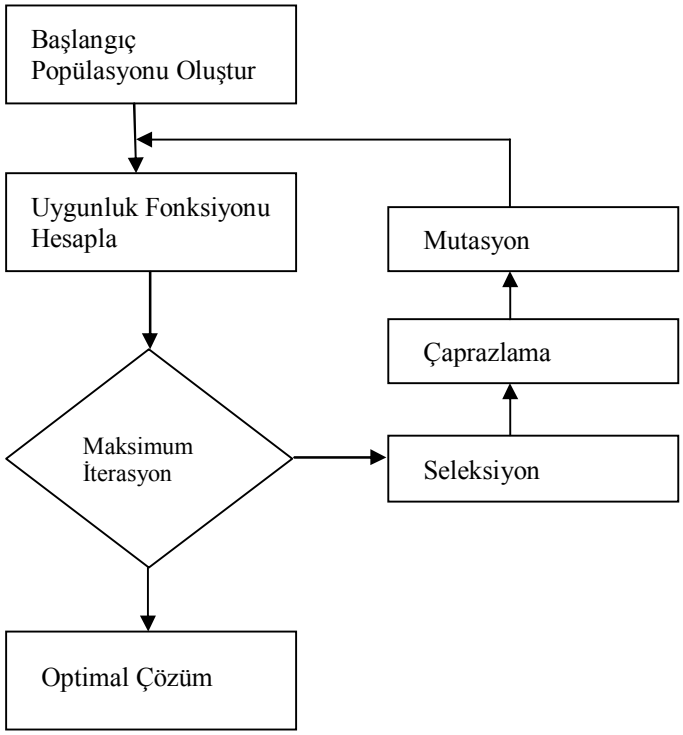
3.3.3. Genetik algoritma

Genetik algoritma'nın temel ilkeleri ilk defa Holland'ın çalışmalarını 1975 yılında "Adaptation in Natural and Artificial Systems" adlı kitabında bir araya getirmesiyle ortaya çıkmıştır. Genetik algoritmalar problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir. Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdır. Her biri çok boyutlu uzay üzerinde bir vektördür (Anonim 2).

Genetik algoritma, diğer sezgisel yöntemlerde olduğu gibi tek bir çözüm geliştirmek yerine, böyle çözümlerden meydana gelen bir küme oluştururlar. Problem için olası pek çok çözümü temsil eden bu küme, genetik algoritma terminolojisinde popülasyon adını alır. Popülasyonlar, kromozom veya birey adı verilen sayı dizilerinden oluşur. Birey içindeki her bir elemana gen adı verilir. Popülasyondaki, bireyler her bir iterasyon sonunda güncellenir.

Popülasyon içindeki her kromozomun veya bireyin, problem için çözüm olup olmayacağına karar veren bir uygunluk fonksiyonu vardır. Uygunluk fonksiyonu yüksek bireylerin seleksiyon ve çaprazlama işlemleri ile çoğalmaları sağlanır. Yeni bireyler üretilirken düşük uygunluk değerine sahip bireyler daha az seçileceğinden bu bireyler bir süre sonra popülasyon dışında bırakılırlar. Uygunluk değeri yüksek olan ne kadar çok birey bir araya gelip, yeni bireyler oluştururursa arama uzayı içerisinde o kadar

iyi bir çalışma alanı elde edilir. Şekil 3.7'de genetik algoritmanın akış şeması gösterilmektedir.



Şekil 3.7. Genetik Algoritma akış şeması

3.3.4. Nawaz, Enscore, Ham (NEH) yaklaşımı

İş akışı çizelgeleme probleminde, zaman kriterine göre yüksek performansa sahip algoritmalardan birisi ve en çok bilineni Nawaz, Enscore ve Ham'in 1983 yılında önerdiği NEH sezgisel yöntemidir. Makine sayısı m, iş sayısı n olan problemlerin çözümü için NEH algoritması 3 temel adımdan oluşur:

1. Adım: Tüm işlerin toplam işlenme süreleri hesaplanır:

$$Pj = \sum_{i=1}^{m} Pij \qquad (3.6)$$

2. Adım: İşler azalan sıraya göre sıralanır.

3. Adım: *j=1, ..., n* olmak üzere j işi alınır ve sıralanmış işler arasındaki tüm olası pozisyonlara yerleştirilerek en iyi sıralama bulunur.

Örnek olarak, sıralanan n iş içerisinden toplam işlem zamanları en yüksek olan iki iş seçilir. Seçilen iki iş sıraya konarak tamamlanma süreleri bulunur. Kısmi olarak bulunan bu sürelerden en küçük olan iş sırası seçilerek bu iki işin birbirlerine olan önceliği belirlenmiş olur. Bu sıra sabitlenerek daha sonraki kullanımlarda da ele alınır. Daha sonra toplam işlem zamanı en büyük olan üçüncü iş seçilir. Bu iş; başa, ortaya

veya sona konarak tekrardan kısmi tamamlama süresi bulunur ve bu süre bir sonraki adımlarda kullanılmak üzere sabitlenir. Her defasında yeni eklenen iş için aynı işlemler uygulanarak bütün işlerin sıralamasının belirlenmesine kadar devam eder (Kurnaz ve Kart, 2010).

3.3.5. FRB yaklaşımı

Rad, Ruiz ve Boroojerdian (2009) tarafından bugüne kadar en iyi çözümleri ve algoritmalar arasında görülen NEH algoritmasının eksikliklerini gidermek ve daha iyi çözümler elde etmek amacıyla geliştirilmiştir.

NEH yönteminin eksikliklerinden birisi, sıralanmış bir işin yerinin, yeni işler geldikçe hep sabit kalmasıdır. NEH tüm işler için tüm pozisyonları denemeden, ilk bulduğu en iyi çözümde işlerin yerini sabitlediğinden dolayı daha iyi çözümleri atlayabilmektedir. Yeni gelen işlerle, yerlerin tekrardan değiştirilmesi bazen daha iyi sonuçlar verebilmektedir. FRB sezgiselleri, NEH'in bu olumsuz yanını düzeltip daha iyi sonuçlar üretmek için geliştirilmiştir. Bu yüzden birbiriyle benzer mantığa sahip olan FRB1 ve FRB2 algoritmaları geliştirilmiştir. Şekil 3.8 ve 3.9'da FRB1 ve FRB2 sezgisellerinin sözde kodu gösterilmektedir.

```
procedure FRB1
        PV:={p11,p21, ...,pm1,p12,p22,...,pm2, ...,p1n,p2n, ...pmn}
        PV'yi azalan sıraya göre sırala
        M':=0          //işlem sürelerinin tutulduğu matris
        π := ∅         //işlem sırası
        for adım:= 1 to (n.m) do
            j := iş (PV[adım])
               if j ∈ π then
                    j işini π 'den çıkar
               endif
            PV[adım] işlem süresini M' matrisine kopyala
            j işi için M' matrisinde π 'nin tüm olası pozisyonlarını dene //Taillard hızlandırıcı
           düşük Cmax değerine sahip j işini π pozisyonuna ekle
        endfor
end
```

Şekil 3.8. FRB1 Yerel Arama Algoritması'nın sözde kodu (Ruiz ve ark., 2007)

Daha önce bahsedildiği üzere NEH algoritmasında, tüm işlerin toplam işlem süreleri hesaplanarak işler azalan bir sırada sıralanmaktadır. FRB1 ve FRB2 algoritmalarında, NEH algoritmasından farklı olarak her bir işin bir makinede işlenme süreleri azalan sıraya göre PV dizisinde sıralanmaktadır. Burada p_{11},1 nolu işin 1 nolu makinede işlenme süresidir.

```
procedure FRB2
        PV:={p11,p21, ...,pm1,p12,p22,...,pm2, ...,p1n,p2n, ...pmn}
        PV'yi azalan sıraya göre sırala
        soniş:=0
        π := ∅
        for adım:= 1 to (n.m) do
              j := iş (PV[adım])
              if soniş≠j then
                 if j ∈ π then
                     j işini π 'den çıkar
                 endif
                 j işi için π 'nin tüm olası pozisyonlarını dene      //Taillard hızlandırıcı
                 düşük Cmax değerine sahip j işini π pozisyonuna ekle
                 soniş:=j
              endif
        endfor
end
```

Şekil 3.9. FRB2 Yerel Arama Algoritması'nın sözde kodu (Ruiz ve ark., 2009)

FRB1 algoritmasında işlem süreleri M matrisine kopyalanmakta, her j bir işi olası tüm pozisyonlara denenerek düşük C_{max} değerine sahip sıraya eklenmektedir. FRB2 algoritmasında ise, matris bulunmamaktadır. Aynı işlemler son iş tamamlana kadar devam etmektedir.

FRB3 yerel arama algoritması, FRB algoritmaları arasında en iyi sonuçları verenidir. Birbiriyle benzer mantığa sahip olan FRB1 ve FRB2 algoritmalarını iyileştirmek amacıyla geliştirilmiştir. FRB3 algoritması aşağıda adım adım gösterilmiştir (Ruiz ve ark., 2009). FRB3 algoritmasının sözde kodu 3.10'da gösterilmektedir.

procedure FRB3

$P_j = \sum_{i=1}^{m} P_{ij}, \forall n \in N$ yi hesapla

p_j yi azalan sıraya göre sırala

$\pi := \varnothing$

for *adım:=* 1 **to** *n* **do**

$j :=$ *iş* (p_j [*adım*])

j işi için π 'nin tüm olası pozisyonlarını dene //Taillard hızlandırıcı

π 'deki *j* işini en düşük C_{max} değerini veren *p* pozisyonuna yerleştir

for *adım2:=*1 **to** *adım* **do**

adım2 pozisyonundaki *h* işini π 'den çıkar

h işi için π 'nin tüm olası pozisyonlarını dene //Taillard hızlandırıcı

π 'deki *h* işini en düşük C_{max} değerini veren pozisyona yerleştir

endfor

endfor

end

Şekil 3.10. FRB3 Yerel Arama Algoritması'nın sözde kodu (Ruiz ve ark., 2009)

FRB3 yerel arama algoritmasının çalışma mantığı adım adım aşağıda verilmiştir.

1. Adım: Tüm işlerin toplam işlenme süreleri Denklem 3.6'ya göre hesaplanır.

2. Adım: İşler azalan sıraya göre sıralanır.

3. Adım: *j=1, ..., n* olmak üzere j işi alınır ve sıralanmış işler arasındaki tüm olası pozisyonlara yerleştirilerek toplam işlem süresinin en az olduğu pozisyon belirlenir ve j işi o pozisyona yerleştirilir.

4. Adım: p_i dizisi içerisinde h işi, toplam işlem süresinin en az olduğu pozisyona kadar olmak üzere sıralanmış işler arasındaki tüm olası pozisyonlara yerleştirilerek toplam işlem süresinin en az olduğu pozisyon belirlenir ve h işi o pozisyona yerleştirilir.

5. Adım: Durma kriteri uygulanır. Tüm işler yerleştirilmişse, işlem sonlanır. Aksi halde Adım3'e geri dönülür.

FRB4 algoritması da FRB3'ün mantığına benzemekle birlikte, FRB3 'ten belirgin bir farklılıklar göstermektedir.

$FRB4_k$ algoritmasını FRB3 algoritmasından ayıran temel farklılık, iç döngüde tüm işlerin yeniden yerleştirilmesi yerine, yeni yerleştirilen işin ön ve arkadan k kadar uzağında olan işler yeniden yerleştirilir. Bunun sebebi ise, yeni yerleştirilen işin çok uzağında bulunan işlerin, yakın olanlara göre daha az etkilenmesidir. Buradaki k sayısı, iç döngüdeki yerleştirme işlemi için olası tüm pozisyonların sayısıdır. Eğer k artarsa işlemci süresi de artar (Kurnaz ve Kart, 2010). Şekil 3.10'da FRB4 yerel arama algoritmasının sözde kodu verilmiştir.

procedure $FRB4_k$

$P_j = \sum_{i=1}^{m} P_{ij}, \forall n \in N$ yi hesapla

p_j yi azalan sıraya göre sırala

$\pi := \varnothing$

for *adım:*= 1 **to** *n* **do**

$j :=$ *iş* (p_j [*adım*])

j işi için π 'nin tüm olası pozisyonlarını dene //Taillard hızlandırıcı

π 'deki j işini en düşük C_{max} değerini veren p pozisyonuna yerleştir

for adım2:=max(1, p-k) **to** min(adım, p+k) **do**

adım2 pozisyonundaki h iş ini π'den çıkar

h işi için π'nin tüm olası pozisyonlarını dene //Taillard hızlandırıcıları

π'deki h işini en düşük C_{max} değerini veren pozisyona yerleştir

endfor

endfor

end

Şekil 3.10. $FRB4_k$ Yerel Arama Algoritması'nın sözde kodu (Ruiz ve ark., 2009)

Ruiz ve ark. (2009), yaptıkları çalışmada NEHT, FRB1-FRB5 sezgisellerini Delphi 2006 ile geliştirilmiş ve tüm test işlemleri bir Pentium IV PC/AT ve 3,2 GHz işlemci hızına sahip 2Gb RAM bellekli bir bilgisayar kullanılarak yürütmüştür. Algoritmaların hata oranlarını karşılaştırmışlardır. Karşılaştırmada kullanılan ölçüt; her bir örnek için bilinen en iyi ya da optimum çözüm üzerine uygulanan relative percentage deviation (RPD)'dır . Bu da Denklem 3.7'de gösterildiği gibi hesaplanır.

$$RPD = \frac{Heu_{sol} - Best_{sol}}{Best_{sol}} \times 100 \qquad (3.7)$$

Burada Heu_{sol} verilen bir örnek için test edilmiş, herhangi bir sezgisel metod tarafından üretilen çözümdür. $Best_{sol}$ ise Aralık 2006 itibariyle Taillard'ın örnekleri için bulunan optimum çözüm yada en düşük bilinen alt sınırdır.

Elde edilen sonuçlar, Çizelge 3.3'de verilmiştir. Çizelgede iyi sonuçlar koyu renkle gösterilirken kötü sonuçlar italik gösterilmiştir. Sonuçlara göre FRB5 algoritmasının optimum sonuçları bulmada diğer sezgisellere göre daha başarılı olduğu görülmüştür. FRB5 algoritması 9 grup problem için en düşük hara oranına ulaşmıştır.

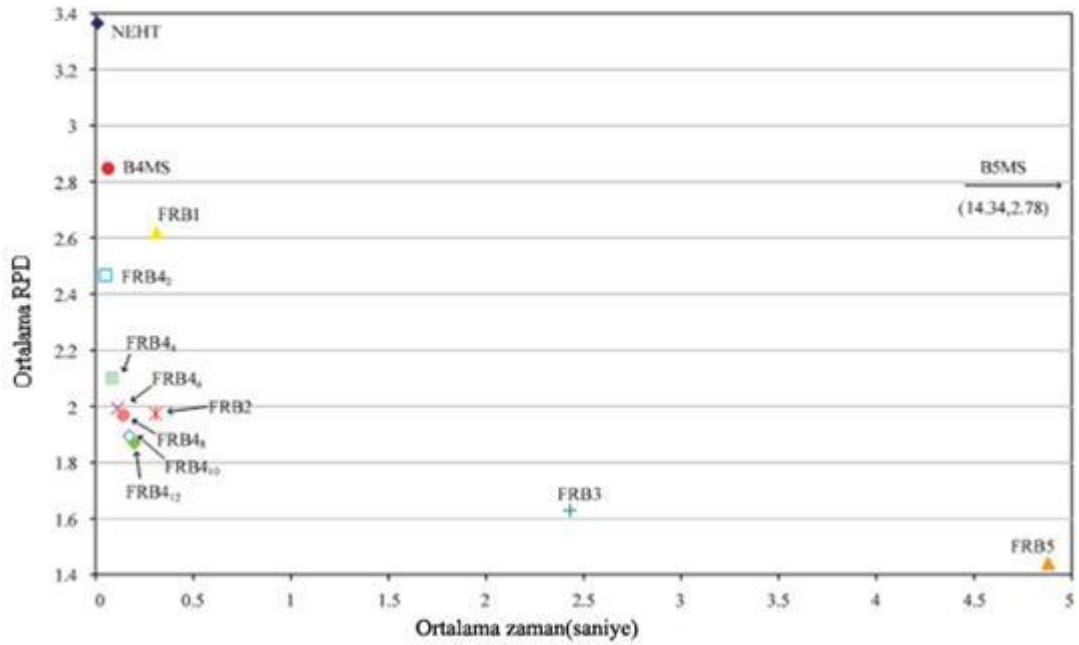
FRB3 algoritması 2 grup problem için en düşük hata oranına ulaşmıştır. Genel ortalamada ise FRB5'den sonra en düşük hata oranına FRB3 algoritması ulaşmıştır.

Çizelge 3.3. NEHT, FRB1-FRB5 algoritmalarının karşılaştırması (Ruiz ve ark., 2009)

Örneğin Boyutu	NEHT	FRB1	FRB2	FRB3	$FRB4_2$	$FRB4_4$	$FRB4_6$	$FRB4_8$	$FRB4_{10}$	$FRB4_{12}$	FRB5
20X5	3,35	*3,57*	1,52	**0,89**	2,02	1,22	1,24	1,14	1,11	1,12	1,08
20X10	*5,02*	3,11	2,54	1,86	3,23	2,7	2,2	2,45	1,8	**1,79**	2,19
20X20	*3,73*	2,34	2,17	2,18	2,49	2,4	2,24	2,26	2,18	2,08	**1,8**
50X5	0,84	*0,91*	0,4	0,22	0,32	0,4	0,29	0,43	0,34	0,3	**0,19**
50X10	*5,12*	4,44	3,1	2,99	4,32	3,11	3,31	2,79	3,45	2,89	**2,25**
50X20	*6,32*	4,2	4,12	**3,38**	5,09	4,4	4,6	4,31	4,15	4,08	3,38
100X5	*0,46*	0,42	0,3	0,21	0,39	0,18	0,18	0,32	0,25	0,28	**0,16**
100X10	*2,13*	1,92	1,34	0,94	1,58	1,38	1,04	1,12	1,17	1,22	**0,8**
100X20	*5,23*	4,16	3,52	2,9	4,31	4	3,74	3,8	3,39	3,73	**2,51**
200X10	*1,43*	1,22	0,89	0,52	0,75	0,67	0,71	0,61	0,73	0,6	**0,38**
200X20	*4,52*	3,46	2,62	2,41	3,49	3,22	2,99	2,99	2,84	2,95	**1,84**
500X20	*2,24*	1,68	1,16	1,06	1,62	1,52	1,39	1,4	1,33	1,4	**0,72**
Ort.	*3,37*	2,62	1,97	1,63	2,47	2,1	1,99	1,97	1,9	1,87	**1,44**

Şekil 3.12'de ise, NEHT sezgiseli ile FRB1-FRB5 algoritmalarının RPD-zaman grafiği verilmiştir. Şekle göre FRB5 optimum değerleri bulmada en başarılısı olmasına rağmen, diğer sezgisellere göre geçen süre diğerlerine göre daha fazladır. FRB3 algoritmasında ise, FRB5 algoritmasındaki geçen süre kadar fazla olmamakla birlikte diğer sezgisellere göre daha iyi sonuçlar verdiği görülmüştür.

FRB3 algoritmasının bulduğu sonuçları ve geçen süre göz önüne alınarak, yapılan çalışmada Yapay Bağışık Sisteminin FRB3 algoritması ile iyileştirilmesi sağlanmıştır.



Şekil 3.12. NEHT,FRB1-FRB5 algoritmalarının RPD-zaman grafiği (Ruiz ve ark., 2009)

Tez çalışmasında Yapay Bağışıklık Sistemi ile birlikte FRB3 yerel arama algoritması melez bir şekilde kullanılmıştır. Başlangıç popülasyonu ve klonlanacak antikorlar FRB3 algoritmasıyla iyileştirilmiştir.

3.4. Bağışıklık Sistemi

Son yıllarda kombinatoriyel (ayrık) problemlerin çözümünde modern sezgisel yöntemler olarak nitelendirebileceğimiz, doğadan ve biyolojik sistemlerden referans alınarak türetilmiş yöntemler uygulanmaktadır. Genetik algoritmalar, karınca kolonileri optimizasyonu, tavlama benzetimi, tabu arama, yapay sinir ağları ve son yıllarda üzerinde çalışılan yapay bağışıklık sistemleri (Artificial Immune Systems) bu yöntemlere örnek olarak verilebilir (Akçay, 2009).

Bu bölümde Doğal Bağışıklık Sistemi ile Yapay Bağışıklık Sistemleri ve algoritmaları hakkında genel bir bilgi verilmiştir.

3.4.1. Doğal bağışıklık sistemi

Bağışıklık deyimi Latince kökenli olan “immunis” sözcüğünden alınmıştır. Eski Roma’da askerlikten muaf (korunmuş) asillere denilen “immunitas” sözcüğünden gelmektedir. Günümüzde "bağışıklık bilimi" olarak bilinen immunoloji, bilimsel

çalışmalarının oldukça önemli alanlarını oluşturmaktadır. Geçirilen bir enfeksiyon hastalığının vücuda kazandırdığı direnci anlatmak amacıyla tıpta kullanılmaktadır.

İnsan vücudu sürekli olarak dış çevreden gelen mikropların, virüslerin, bakterilerin v.b. yabancı moleküllerin saldırısına uğramaktadır. Bağışıklık sistemi, bir canlıdaki hastalıklara karşı vücudu koruma görevini yerine getiren, patojenleri ve tümör hücrelerini tanıyıp onları yok eden işleyişlerin toplamıdır. Sistem, canlı vücudunda geniş bir çeşitlilikte, virüslerden parazitlik solucanlara, vücuda giren veya vücutla temasta bulunan her yabancı maddeye kadar tarama yapar ve onları, canlının sağlıklı vücut hücrelerinden ve dokularından ayırt eder. Bağışıklık sistemi, çok benzer özellikteki maddeleri bile birbirinden ayırabilir. örneğin; bir aminoasidi farklı olan proteinleri bile birbirinden ayırabilecek özelliğe sahiptir (Anonim 1).

Bağışıklık sistemi, insan vücuduna dış ortamdan gelen mikroorganizmaların vücuda zarar vermesini engellemek için gerekli işlemleri gerçekleştirerek vücudun savunma mekanizması olarak görev yapar. Gerçekleştirilen bu işlemlerin hepsine birden "bağışıklık cevabı" adı verilir. Bağışıklık cevabının sonunda vücut mikroorganizmayı yenmeyi başardıysa bu mikroorganizmaya duyarlı hücreler hafıza hücreleri olarak saklanır ve ileride aynı ya da benzer mikroorganizmalar tekrar vücuda girmek istediğinde onlara karşı daha hızlı ve güçlü bir bağışıklık cevabının oluşmasında kullanılırlar. Vücudun bir mikroorganizmaya karşı "bağışıklık kazanması", bu hafıza hücrelerinin oluşumudur.

Birbiri ile ilişkili iki bağışıklık sistemi vardır. Bunlar doğal (innate) bağışıklık sistemi ve kazanılmış (adaptif) bağışıklık sistemidir (Karaboğa, 2011).

- *Doğal bağışıklık sistemi:* Vücut belirli mikropları tanıyabilme kabiliyeti ile doğar ve bu mikropları karşılaştıklarında tahrip edebilir. Doğal bağışıklık sisteminin önemli elemanlarından birisi komplement olarak adlandırılan kan proteinleri sınıfıdır. Bu eleman antikor aktivitesine yardım etme kabiliyetine sahiptir.
- *Kazanılmış bağışıklık sistemi:* Bir mikrop ile daha önce hiç karşılaşılmasa bile vücuda bu mikroba karşı koyma kabiliyetini kazandırır. Sistem iki tip lenfosit(B ve T hücreleri) üzerine klonal olarak dağıtılmış antijen reseptörlerini kullanır. Bu antijen reseptörleri, rasgele işlemle gen segmentleri bir araya getirilerek üretilir. Adaptif bağışıklık cevabı belirli özellikli reseptörleri gösteren lenfositlerin klonal seçme işlemine dayalıdır.

Doğuştan ve kazanılmış bağışıklıkların her ikisi de kendinden olan ve kendinden olmayan moleküllerin ayrımı prensibine bağımlıdır. İmmünolojide kendinden

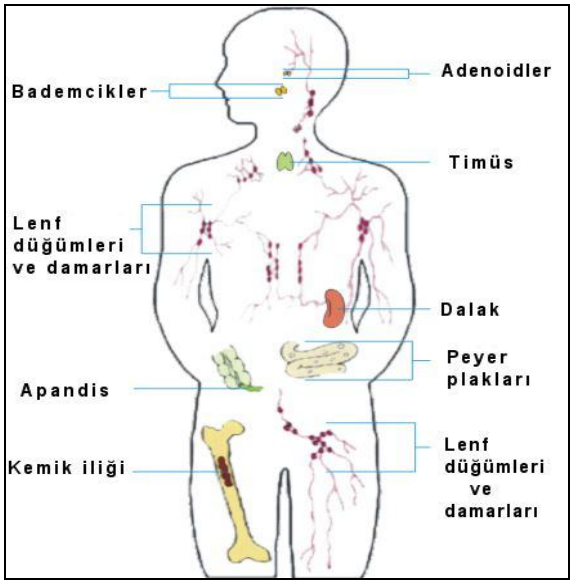
olan moleküller, bir canlının vücudunda bulunan ve bağışıklık sistemince yabancı moleküllerden ayrılabilen bileşenlerdir. Kendinden olmayanlar ise yabancı moleküller olarak tanımlanabilir. Kendinden olmayan moleküllerin bir sınıfı antijenler olarak bilinir ve özgül bağışıklık almaçlarına bağlanıp bir bağışıklık cevabının oluşmasına neden olan maddeler olarak tanımlanabilirler (Anonim 1).

3.4.1.1. Bağışıklık sisteminin ana bileşenleri

Bağışıklık cevabının oluşmasından organizmadaki belli başlı bazı sistemler, organlar ve hücreler sorumludur. Bağışıklık sisteminin organları, lenfoid organlar olarak bilinir. Çünkü bu organlar, bağışıklık sisteminin temel elemanları olan lökositlerin ve lenfositlerin üremesi büyümesi ve gelişmesi ile ilgilidir. Lenfoid organlar santral (birincil) ve periferik (ikincil) olmak üzere ikiye ayrılır.

Santral (birincil) organlar, yeni lenfositlerin antijene bağımlı olmaksızın otonom olarak yapıldıkları ve bağışık tepki oluşturma yeteneği kazandırdıkları yerlerdir. Periferik (ikincil) lenfoid organlar ise, lenfositlerin antijenik uyarılara tepki gösterdikleri yerlerdir (Abbas, Lichtman ve Pober, 1994).

Bağışıklık sisteminin başlıca organları Şekil 3.13'de gösterilmiştir.



Şekil 3.13. Bağışıklık Sisteminin Organları (Anonim 1)

Lenf Düğümleri: Vücudun birçok bölgesinde gruplar halinde bulunur. Boyun, koltuk altı, kasıklarda olduğu gibi yüzeyde bulunan lenf düğümleri kolaylıkla fark edilebilir. Ancak göğüs ve karın boşluğunda da çok sayıda lenf düğümü mevcuttur. Bunların başlıca görevi vücuda giren yabancı maddelere karşı bir süzgeç oluşturarak, mikropların vücuda yayılımlarını engellemek ya da geciktirmektir. Düğümler içinde bağışıklık sistemine ait sayısız hücre bulunmakta, bu hücreler insana zarar verebilecek maddelerin geçişine engel olmaya çalışmaktadırlar. Bu mücadele sırasında lenf bezeleri şişerek elle ya da gözle fark edilebilecek boyutlara ulaşabilmektedir. Bademciklerimiz de birer lenf düğümüdür. Bakteriler ya da virüslerle yoğun bir biçimde savaştığında, bademciklerimiz şişer ve iltihaplanır (Anonim 1).

Dalak: Sol böbreğin arka bölümünde yer alır. Kırmızı kan hücreleri ve bağışıklık sistemin beyaz kan hücreleri için depo olarak görev yapar, aynı zamanda kandaki yabancı maddelerin büyük bir kısmını süzer (Anonim 1).

Timus: Göğüs boşluğu içinde yer alan iki parçadan oluşan bir organdır. Lenfosit, T lenfosit veya sadece "T hücreleri" timüste büyür, eğitilir, olgunlaşır ve bağışıklık sisteminde üstlendikleri görevleri yerine getirmek üzere yeniden kana karışırlar (Anonim 1).

Kemik İliği: Kemiklerin ortasında bulunan yağlı ve gözeli bir dokudur. Bağışıklık sisteminde çok önemli işlevleri olan akyuvarlar da dâhil olmak üzere bütün kan hücrelerinin yapım yeridir (Anonim 1).

Apandist ve peyer plakaları: Sindirim sistemini korumak için görevlendirilmiş bağışıklık hücrelerini ihtiva eden özel lenf düğümleridir (Karaboğa, 2011).

Tonsiller ve adenoids: Solunum sistemini işgalcilere karşı koruyan bağışıklık hücrelerini ihtiva eden özel lenf düğümleridir (Karaboğa, 2011).

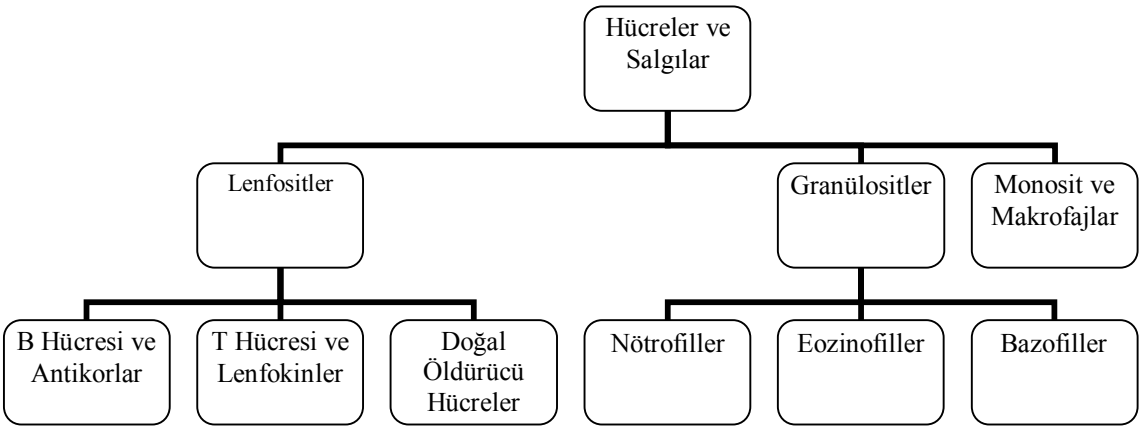
Akyuvarlar (Lökositler): Bağışıklık sistemimizin en önemli savaşçıları ve immünolojik savunmanın temel elemanlarıdır. Eğer bu sistem geçilirse hastalık dediğimiz durum ortaya çıkar. Lökositler damar içinde dolanırken, tehlike sinyallerini aldıkları bölgelerde damardan ayrılıp bakteri ve ölü doku gibi yabancı cisimlerin etrafını sarabilirler. Lökositler plazma kaynaklı kan proteinleriyle birlikte organizmanın bütünlüğünü sağlamakta askeri güç gibi görev yaparlar (De Castro ve Von Zuben, 2001).

3.4.1.2. Bağışıklık hücreleri

Bağışıklık sistemi kemik iliğinde üretilen çok sayıda hücrelerden oluşmaktadır. Bu hücrelerin çoğu burada olgunlaştıktan sonra, kanda ve lenfatik damarlarda dolaşır. Bunlardan bazıları genel savunmadan sorumludur. Bazıları da özel patojenleri karşılamak için eğitilir. Bu hücreler şunlardır:

- Granülositler
- Lenfositler
- Monosit ve Makrofajlar

Şekil 3.14'de bağışıklık hücrelerinin yapısal sınıflandırılması gösterilmektedir.



Şekil 3.14. Bağışıklık hücrelerinin sınıflandırılması

3.4.1.2.1. Lenfositler

Bağışıklık sisteminde büyük bir sorumluluğa sahip küçük hücrelerdir. Akyuvarların % 30-40'ı lenfositlerdir. İki temel tipi vardır: B ve T lenfosidi. B lenfosidi, aktivasyona bağlı olarak antikor üretim kabiliyetine sahip plazma hücrelerine dönüşür.

T Lenfositleri ve Lenfokinler

Timus içinde olgunlaştıkları için bu ismi almaktadır. Diğer hücre hareketlerinin regülasyonunu sağlar ve enfekte olmuş hücrelere saldırırlar. T lenfositleri üç sınıfa ayrılabilir: yardımcı (helper) Th lenfosit, öldürücü (cytotoxic) Tc lenfosit ve baskılayıcı (suppressor) T hücreleri (Karaboğa, 2011).

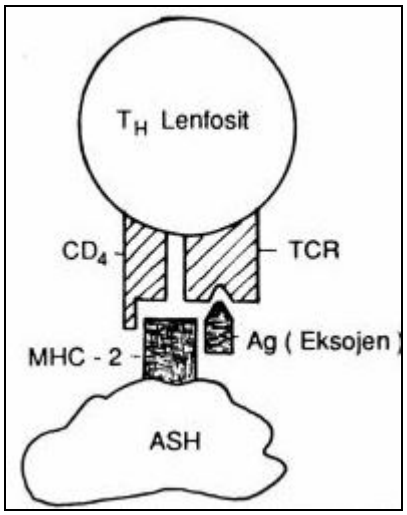
Th hücreleri, B hücrelerinin, diğer T hücrelerinin, makrofajların ve doğal öldürücü hücrelerin aktivasyonu için gereklidir. Bunlar aynı zamanda CD4 veya T4 hücreleri olarak da bilinir.

Öldürücü T hücreleri, kanser hücreleri, virüsler, mikrobik işgalcileri elimine etme kabiliyetine sahiptirler.

Baskılayıcı T lenfositleri, bağışıklık cevabının devamında hayati değere sahiptir. Bazen CD8 hücreleri olarak da adlandırılan bu hücreler diğer bağışıklık diğer bağışıklık hücrelerinin hareketini engellerler. Bunların hareketi olmadan bağışıklık kesinlikle kontrolü kaybeder. Bu da oto bağışıklık hastalıklarına ve alerjik reaksiyonlara sebep olur.

Antijenlerin büyük çoğunluğu (genelde protein olanlar), önce antijen sunucu hücre (ASH) dediğimiz bazı özel hücreler tarafından alınıp hazırlandıktan sonra etkinlik kazanırlar. ASH niteliği taşıyan hücreler; monosit, makrofaj, dendritik hücreler, glia hücreleri, derideki langerhans hücreleridir. B lenfositler de özel koşullarda antijen sunarlar. ASH'ların ortak özelliği yüzeylerinde MHC-II denen doku antijenlerine sahip olmalarıdır.

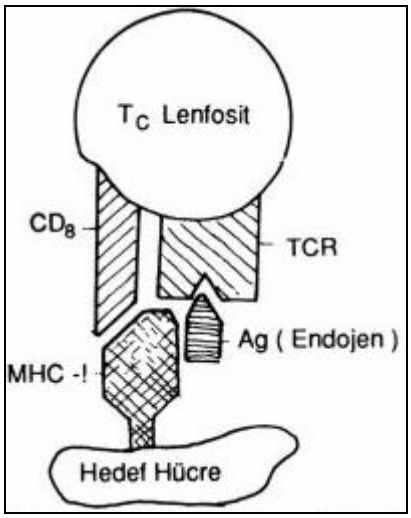
Organizmaya dışardan giren protein yapısındaki antijenler (eksojen Ag) önce ASH tarafından yakalanır. Hücre yüzeyinde MHC-II molekülü içinde sunulan antijen, bu haliyle Th lenfositleri tarafından tanınabilir. Tanınma olayında ASH'daki MHC-II ile Th hücresindeki CD4 yüzey molekülleri, Antijen ile de TCR arasında etkileşme olur.
Th lenfositler uyarıldıktan sonra aktive olurlar ve diğer immün sistem hücrelerini de uyararak hücresel ve hümoral bağışık yanıt oluşmasını sağlarlar (Serter, 1996b). Şekil 3.15'de Th Lenfositi gösterilmiştir.



Şekil 3.15. Th Lenfositi (Serter, 1996b)

Organizmada doğrudan hücre içinde sentezlenen yabancı antijenler (Endojen Ag) de bulunabilir. Örneğin içinde virüs üreyen hücreler veya tümörleşmiş hücrelerde olduğu gibi. Bu durumda virüs veya tümör antijenleri hücre içindeki MHC-1 molekülleri ile birleştirilerek yüzeye sevk edilirler. MHC-I doku uygunluk antijenleridir ve tüm çekirdekli hücrelerimizde mevcuttur.

Hücre yüzeyinde MHC-1 molekülü içinde sunulan bu antijenler ise Tc lenfositlerce tanınabilir. Çünkü MHC-1 ile Tc yüzeyindeki CD8 molekülü ve Antijen ile TCR arasında etkileşme meydana gelir. Uyarılan Tc lenfositler özgül olarak bağlandığı bu hedef hücreleri öldürürler (Serter, 1996b). Şekil 3.16'da Tc Lenfositi gösterilmiştir.



Şekil 3.16. Tc Lenfositi (Serter, 1996b)

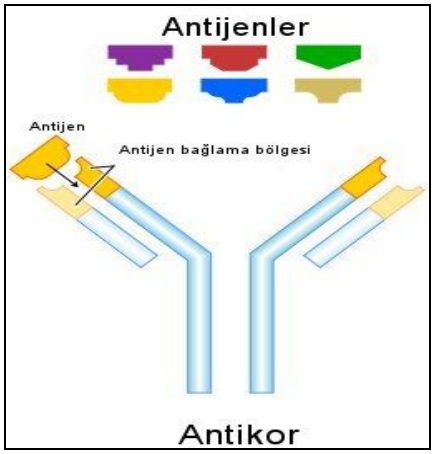
T hücreleri cytokine veya lenfokin olarak bilinen maddeleri salgılamak suretiyle çalışır. Bu maddeler hücresel büyümeyi, aktivasyonu ve regülasyonu geliştirirler. Ayrıca hedef hücreleri öldürür ve makrofajları uyarırlar (Karaboğa, 2011).

B Lenfosidi ve Antikorlar

Antikor bağışıklık sisteminin en önemli elemanlarından birisidir. Her B hücresi özel bir antikor üretmek için programlanmıştır. B lenfositler, her antijene (vücuda yabancı olan ve antikor üretimine neden olan maddeler ya da canlılar) özgü antikor üretirler. Vücuda herhangi bir antijen girdiğinde, B lenfositler çoğalır ve milyonlarca antikor üretirler. 6 aylık bebekken vücudumuz antikor üretmeye başlar. Antikorlar başka özel bir proteini tanıyan ve ona bağlanan proteinlerdir. Antikorların üretilmesi ve

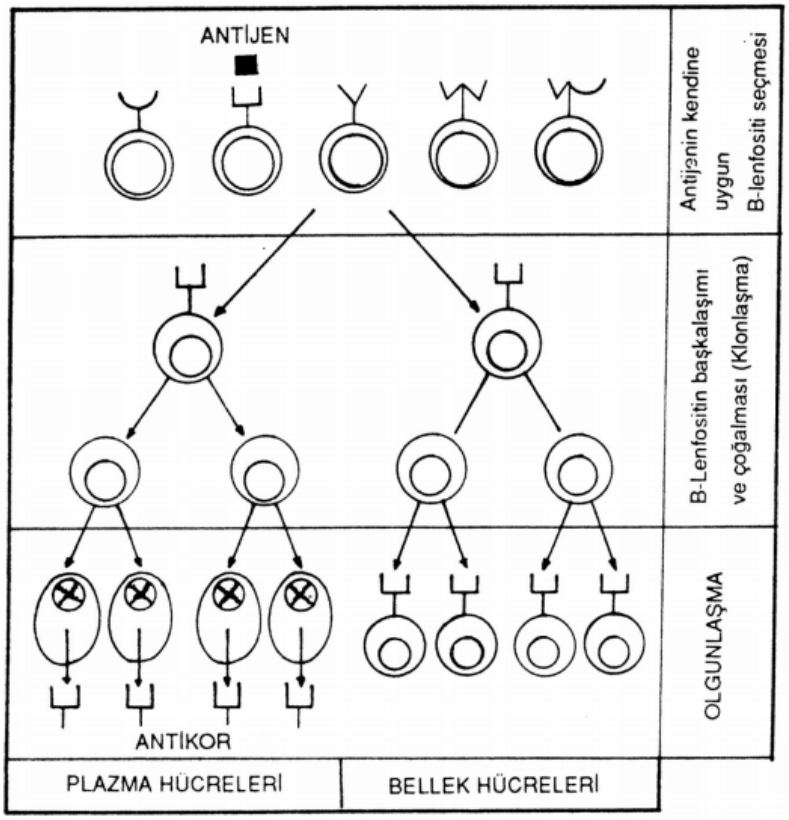
bağlanması, genellikle bağlanılan maddenin öldürülerek yok edilmesi için diğer hücrelerin uyarılmasının bir sonucudur (Anonim 1).

Antikorlar, antijen niteliği taşıyan bakteri virüs ya da zehirli maddelere tepki veren Y biçiminde protein moleküllerdir. Şekil 3.17'de örnek bir antikor gösterilmiştir.



Şekil 3.17. Antikor

Y biçimindeki antikorların kısa kollarının uç kısmında antijenlere bağlanabilmelerini sağlayan özel bölgeler bulunur. Herhangi bir antijene bağlanmış olan antikorlar, ya onların hareketine engel olur ya da bağışıklık sisteminde rol alan başka proteinlerin, hormonların ve makrofajların devreye girmesi için işaret verirler. Bu antikorlar kişiyi ikinci kez aynı mikrop ile hasta olmaktan korurlar (Serter, 1996a). Şekil 3.18'de B lenfositlerinin antijenle karşılaşmaları ve sonrasında gelişen olaylar gösterilmiştir.



Şekil 3.18. B lenfositlerin antijenle karşılaşması ve gelişen olaylar (Serter, 1996a)

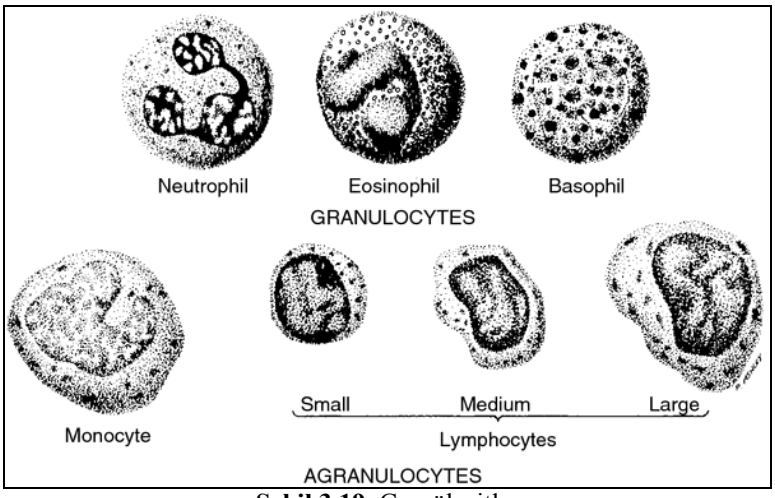
Doğal Öldürücü Hücreler

Öldürücü lenfositlerin bir diğer sınıfıdır. Öldürücü T hücreleri gibi güçlü kimyasallarla dolu granülleri içerirler. Doğal öldürücü oldukları için özel bir antijeni tanımaya ihtiyacı yoktur. Özellikle tümörlere saldırırlar ve çok sayıda enfekte edici mikroplara karşı savunma yaparlar. Ayrıca bağışıklık regülasyonuna katkıda bulunur ve lenfokinlerin büyük miktarını salgılarlar.

Doğal Öldürücü Hücreler de kemik iliğinde yapılmaktadır Bu hücreler kan kemik iliği ve dalakta bulunurlar. Lenfositlerin ayrı bir dalından kaynaklandıkları kabul edilmektedir. Bunlar büyük görünümlü lenfositlerdir En önemli görevleri, tümör hücrelerini ve virüs taşıyan hücreleri öldürmektir. Virüs taşıyan hücreleri öldürdükleri halde normal hücrelere zarar vermezler (Karaboğa, 2011).

3.4.1.2.2. Granülositler

Akyuvarların % 50-60'ı granülositlerdir. Bunlar da, nötrofiller, eozinofiller ve bazofiller olarak üç sınıfa ayrılırlar (Anonim 1). Şekil 3.19'da granülositler gösterilmiştir.



Şekil 3.19. Granülositler

Nötrofiller

Kemik iliğimiz hergün trilyonlarca nötrofil üreterek kan dolaşımına bırakır ama yaşam süreleri kısadır ki bu genelde birgünden azdır. Nötrofiller kan dolaşımına girdikten sonra hızla dokuların doğru hareket ederler. Elimize kıymık battığında ya da elimiz kesildiğinde nötrofiller hızla bu bölgeye göç ederler ve salgıladıkları enzimler, deterjan ve çamaşır suyu etkilerine benzer hidrojen peroksit veya diğer kimyasal maddelerle, karşılaştıkları bakterileri veya yabancı molekülleri öldürürler.

Eozinofiller

Deride ve akciğerde bulunan parazitlere odaklanır. Nötrofillerle bağlantılıdır. Bir patojence etkinleştirildiklerinde, parazitlere karşı savunmada ve (astım gibi) alerjik reaksiyonlarda önemli rolü olan histamini salarlar.

Bazofiller

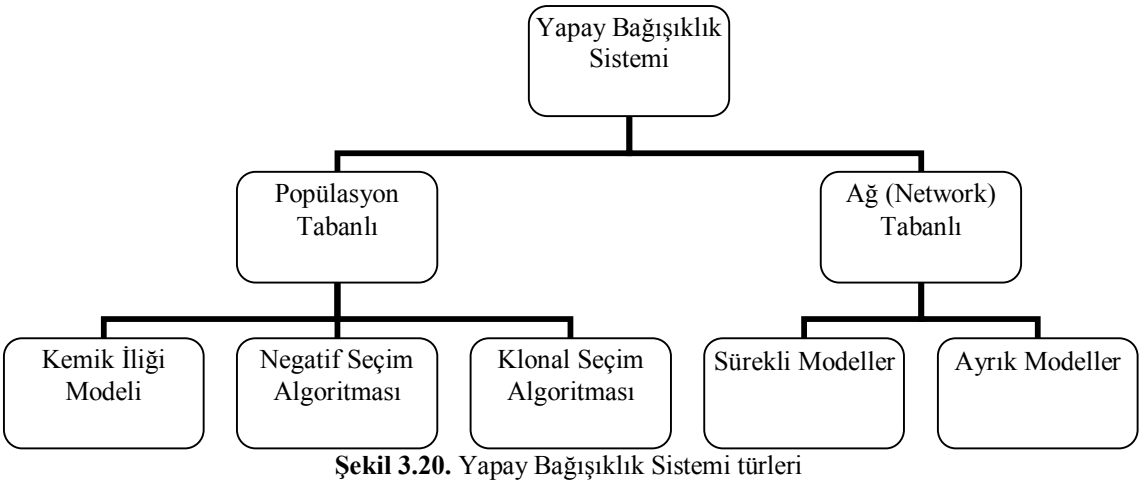
Histamin denilen özel proteinler taşıdıklarından iltihaba sebep olmalarından ötürü önemlidirler. Bu hücreler alerjik hastalıkların gelişmesinden sorumludurlar.

3.4.1.2.3. Monosit ve Makrofajlar

Akyuvarların %7 kadarını monositler oluşturur. Kemik iliğinde yapılıp kan dolaşımına geçen monositler 12 saat içinde dokulara girerler ve makrofajlara dönüşürler. Monosit ve makrofajlar, bağışıklık sistemimizin en önemli hücreleridir. Her dokunun kendine özgü makrofajları vardır. Örneğin; akciğerlerdeki makrofajlar duman ve toz gibi yabancı parçacıkları, bakterileri ve virüsleri temizlerler. Makrofajlar, büyük boyutlu hücrelerdir. En önemli becerisi sindirme ve hazmetme yani fagositoz yapabilmesidir. Fagositoz, bağışıklığın en önemli öğelerinden biridir. Çünkü enfeksiyona karşı çabuk ve çoğunlukla da kesin bir koruma sağlar. Makrofajların önemli bir görevi de; ölmüş nötrofilleri temizlemektir (Anonim 1).

3.4.2. Yapay bağışıklık sistemi

Yapay bağışıklık sisteminde, çeşitli doğal bağışıklık mekanizmalarından referans alan algoritmalar modellenmiş ve algoritmaları çıkarılmıştır. Şu an uygulanabilir genel algoritmalar, negatif ve pozitif seçim algoritmaları, klonal seçim algoritması ve bağışık ağ modelleridir. Literatürde geliştirilen bu algoritmaları, uygulama alanlarına ve yapılarına göre Şekil 3.20'de gösterildiği gibi kategorize etmek mümkündür (Şahan, 2004).



Şekil 3.20. Yapay Bağışıklık Sistemi türleri

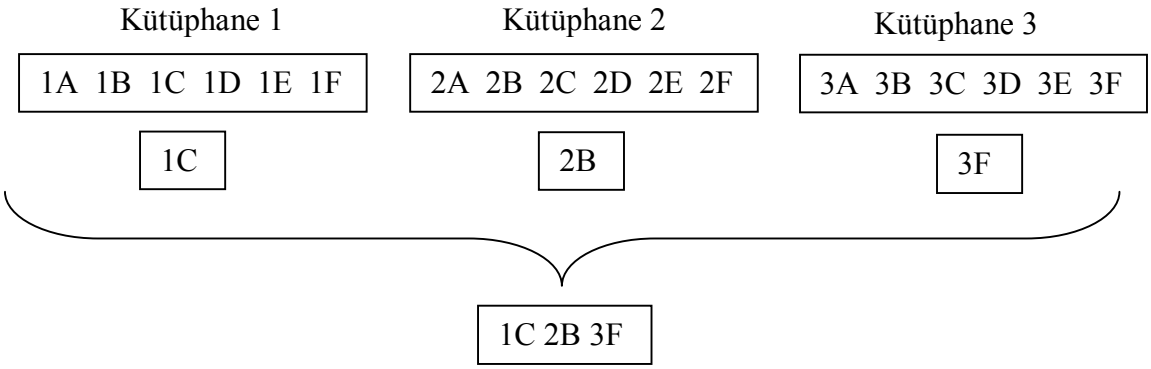
Yapay bağışıklık sistemi algoritmaları temelde 2 kategoride toplanmaktadır Bunlar popülasyon tabanlı ve ağ tabanlı YBS algoritmalarıdır. Popülasyon tabanlı algoritmalar da kendi içerisinde klonal seçim tabanlı, kemik iliği modelleri ve negatif

seçim tabanlı olmak üzere üç gruba ayrılmaktadırlar. Ağ tabanlı algoritmalar ise sürekli modeller ve ayrık modeller olarak iki gruba ayrılmaktadır. Çalışmanın temelini klonal seçim algoritması oluşturmaktadır.

3.4.2.1. Kemik iliği modeli

Kemik iliği, lenfositler dâhil tüm kan hücrelerinin üretiminden sorumludur. Kemik iliği algoritmaları, yapay bağışıklık sistemleri içinde popülasyon tabanlı algoritmalardan birisidir.

Antikorların birbirine bağlı olarak hafıza saklandığı yere gen kütüphanesi adı verilir. Kemik iliğinden antikorları üretmek için kullanılır. En karmaşık ve biyolojik olarak en çekici olan kemik iliği modelleri, bağışıklık hücre ve moleküllerinin yeniden düzenlenmesiyle yâda geliştirilmesiyle oluşturulan gen kütüphanelerinin kullanımını gerektirir. Bir antikor molekülünün üretimi farklı gen bileşenlerinin birbirine bağlanmasıyla meydana gelir. Şekil 3.21'de gen kütüphanesinden örnek bir antikor oluşumu gösterilmiştir (Cunkaş, 2010).



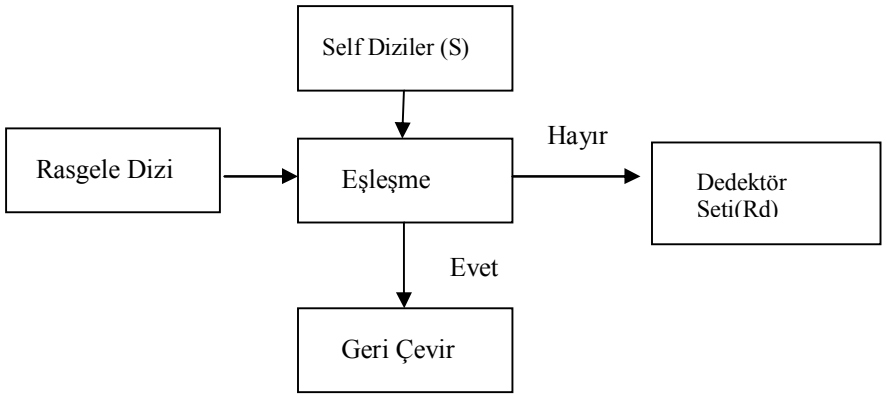
Şekil 3.21. Gen kütüphanesindeki bir antikor molekülünün oluşumu

3.4.2.2. Negatif seçim algoritması

Forrest ve ark. tarafından 1994 yılında oluşturulmuştur. T hücrelerinin timüsteki gelişimleri esnasında, antijenleri tanıyabilen algılayıcıları içeren T hücreleri, negatif seçim denen bir süreç ile T hücreleri repertuarından silinirler.

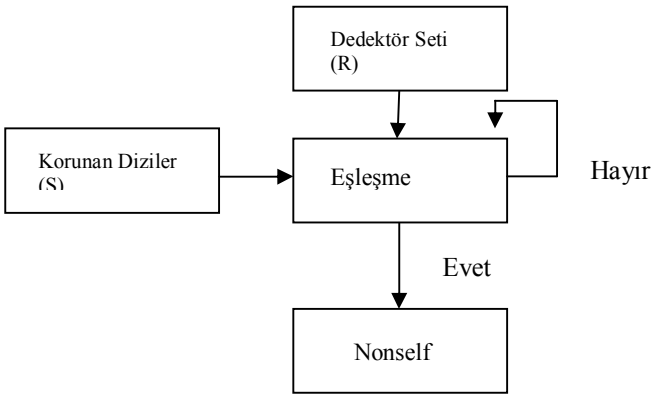
Negatif seçim algoritması iki fazda çalışır. İlk fazda detektörlerden oluşan bir set üretilir. Şekil 3.22'de ilk faz gösterilmiştir. Öncelikle, korunacak modellerin kümesi belirlenir ve self diziler olarak adlandırılır. Negatif seçim algoritmasına dayanarak, self diziler kümesine ait olmayan elemanları tanımakla sorumlu bir detektörler seti

oluşturulur. Rasgele dizilerle oluşturulan kümedeki elemanlarla self diziler kümesindeki elemanlar karşılaştırılır. Bir eşleşme olursa, örneğin, self diziler kümesinin bir elemanı rasgele oluşturulan diziler kümesinin bir elemanı tarafından tanınırsa, self diziler kümesinin bu elemanını atılır. Tersi durumda, self diziler kümesinin bu elemanını algılayıcı(detektör) kümesi içinde saklanır (Hofmeyer ve Forest, 1996).



Şekil 3.22. Negatif Seçim Algoritması ilk faz çalışma şekli

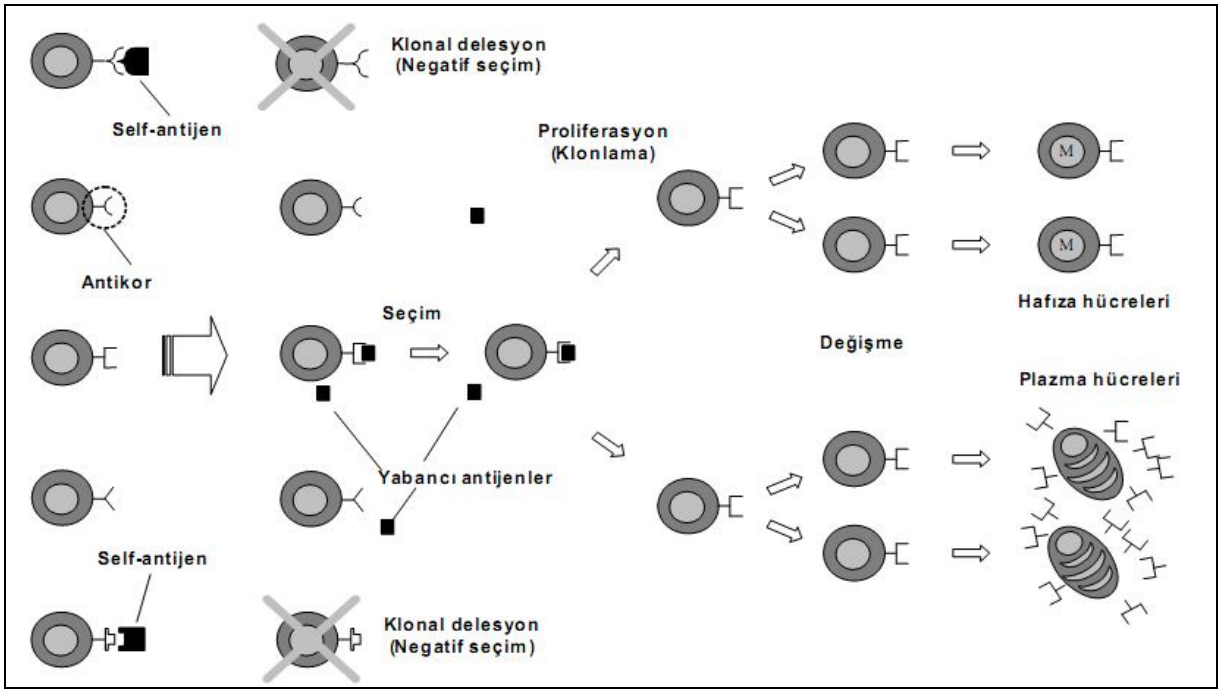
Şekil 3.23'de ikinci faz gösterilmiştir. İkinci fazda korunan veri oluşturulan detektörler ile karşılaştırılarak gözlemlenir. Algılayıcı kümesini oluşturduktan sonra, algoritmanın diğer aşaması tanınmayan modellerin varlığı için sistemi izlemekten oluşur. Bu durumda bir korunan diziler kümesi oluşturulur. Bu küme, self diziler ve diğer yeni modellerden veya tamamıyla yeni bir kümeden meydana gelir. Tanınmayan modellerine karşılık gelen, algılayıcı kümesinin tüm elemanları için, koruyucu diziler kümesinin bir elemanını tanıyıp tanıyamadığı (eşleşme) kontrol edilir. Eşleşme var ise, o zaman bir nonself dizi tanınmıştır ve reaksiyon başlamalıdır (Akbal, 2005).



Şekil 3.23. Negatif Seçim Algoritması ikinci faz çalışma şekli

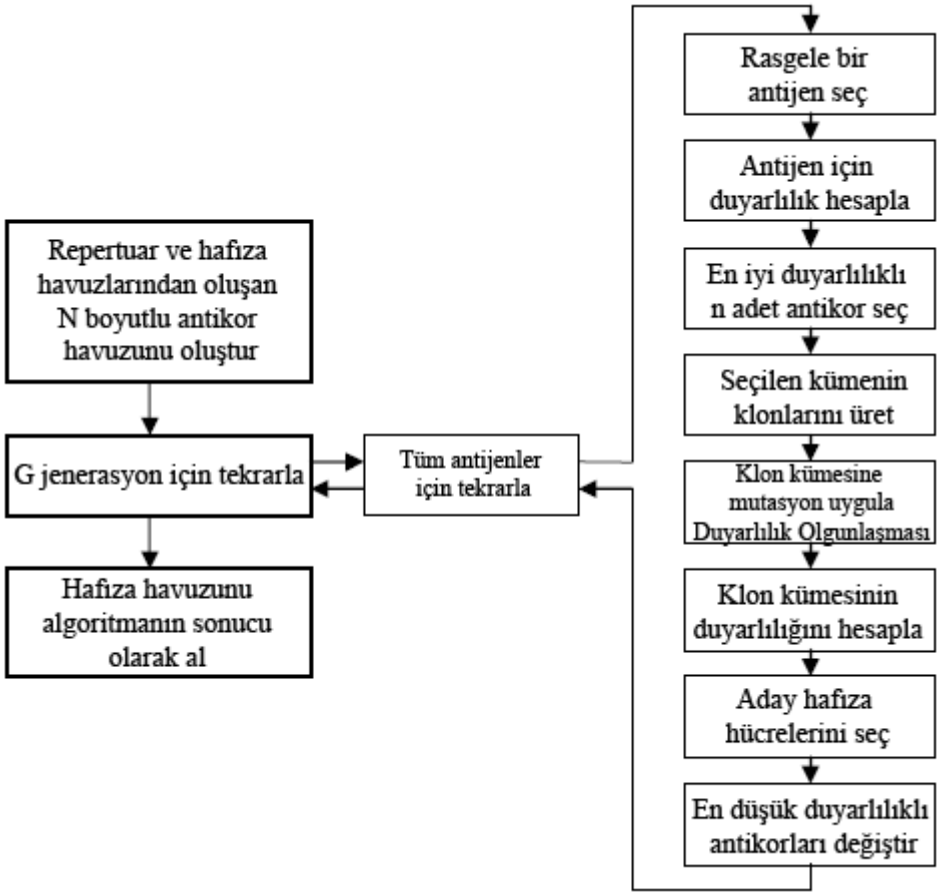
3.4.2.3. Klonal seçim algoritması

Klonal seçim algoritması, negatif seçimin rolünü tamamlayıcı özelliktedir. Antijenler, bir B lenfosidi tarafından tanındığında, nasıl bir bağışıklık tepkisi verileceğini açıklamak için kullanılmaktadır (Ada ve Nossal, 1987). Şekil 3.24'de klonal seçim prensibi gösterilmektedir.



Şekil 3.24. Klonal seçim prensibi (Polat, 2004)

Vücutta bir antijen tespit edildiği zaman, kemik iliğinden B lenfositleri salgılanır ve antikor üretilir. Bu antikorlar, antijenlerle birleştiği zaman Th hücrelerinden gelen lenfokinlerin de etkisi ile B lenfosidini uyarır. B lenfosidi uyarma ile birlikte çoğalır ve plazma hücreleri denilen terminal antikor salgılayan hücrelere dönüşür. Lenfositler çoğalma ve plazma hücrelerine dönüşmenin dışında uzun ömürlü hafıza hücrelerine dönüşebilirler. Hafıza hücreleri kan, lenf, dokular boyunca dolaşırlar. İkinci bir antijenik durumla karşılaşıldığında yüksek benzerlikte antikorlar üretebilirler (De Castro ve Von Zuben, 2000). Şekil 3.25' de Konal Seçim Algoritmasının akış diyagramı görülmektedir.



Şekil 3.25. Klonal Seçim Algoritması akış şeması

Algoritmanın çalışması şu şekildedir:

1.Adım: Giriş kümesindeki antikor sayısı opsiyoneldir. Öncelikle *P* Popülasyonu oluşturulur. Bu popülasyon, bellek hücreleri *(M)* ve geri kalan popülasyondan *(Pr)* oluşmaktadır *(P=Pr+M)*.

2.Adım: Duyarlılık kriterine bağlı olarak popülasyondaki en iyi n tane antikor seçilir ve Pn popülasyonu oluşturulur.

3.Adım: Seçilen bu en iyi antikorlar, antijene olan duyarlılıklarına bağlı olarak klonlanırlar. Klonlama işlemi Denklem 3.8'de verilmiştir.

$$C = Round(\frac{\beta \times s(Ab)}{i}) \tag{3.8}$$

Verilen formüle göre her bir antikor için hesaplanmıştır. Böylece her bir antikordan kaç adet klonlanacağı belirlenir. Burada;

β: dışarıdan girilen bir değişken,

s(Ab): Pn popülasyonundaki antikor sayısı,

i: Klonlanacak antikorun P_n popülasyonu içindeki sıra numarasıdır.

Duyarlılığı yüksek olan antikorların oluşturacağı klon sayısı yüksek, duyarlılığı düşük olan antikorların klon sayısı ise düşük olacaktır. Böylelikle bir *C* kümesi oluşturulmuş olur.

4.Adım: *C* kümesi mutasyon işlemine tabi tutulur. Mutasyon işleminde de duyarlılık ile ters orantı söz konusudur. İşlem Denklem 3.9'da verilen formüle göre gerçekleştirilir.

$$Round\left[\left(e^{\frac{-10.f(i)}{\max(f)}}\right)xn\right] \tag{3.9}$$

Verilen formül, klonlanan her bir antikor için hesaplanır. Burada *f(i)*; üzerinde çalışılan antikorun uygunluğu olmaktadır. 10 sabit bir sayı olup n, problemdeki değişken sayısıdır. *max(f)* ise hesaplanan uygunluk fonksiyonu değerlerinin en büyüğüdür. Formül sonucu bulunan değer;

- 0 ise, mutasyon gerçekleşmez.
- 1 ise, değişkenlerden sadece birisi mutasyona uğratılacaktır. Mutasyon işlemi, rastgele sayı üretimi sonucunda ilgili değişkenin verilen sınırlar dahilinde yeniden rastgele üretilmesiyle gerçekleştirilmiştir.
- 2 ise değişkenlerden her ikisi de hiper mutasyona uğratılacaktır. Hipermutasyon işlemi, değişkenlerin verilen sınırlar dahilinde yeniden rastgele üretilmesiyle gerçekleştirilir.

Mutasyondan sonra *C** kümesi oluşmaktadır.

5.Adım: Klonlanmış, mutasyona uğramış ve uygunluk fonksiyonları hesaplanmış antikorlardan oluşan büyük boyutlu bir küme vardır. Klonlama ve mutasyondan sonra gelişmiş antikorları popülasyona eklemek için tekrar bir seçme işlemi yapılır ve *M* bellek kümesi oluşturulur. Bu seçme işleminden sonra *P* popülasyonundaki bazı antikorlar, yerlerini *C** kümesindeki bazı antikorlara bırakabilir.

6.Adım: Popülasyonda çeşitliliği sağlamak amacıyla *P* popülasyonundaki *d* tane antikor yeni üretilen antikorlar ile yer değiştirir. Düşük duyarlılıklı antikorların yer değiştirme ihtimali daha yüksektir.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu tez çalışmasında, seri üretim yapan işletmelerde kullanılan akış tipi çizelgeleme problemlerinin çözümü için, yapay bağışıklık sistemi kullanılmıştır. Ayrıca YBS de ilk adımda oluşturulan sıralama işlemleri için rastgele bir dizilim oluşturmak yerine permütasyon tipi iş akış çizelgeleme problemlerinde oldukça yüksek başarıya sahip olan FRB3 algoritması kullanılmıştır. Daha sonraki adımlarda oluşturulan çözümleri iyileştirmek için yine FRB3 algoritmasından yararlanılmıştır.

Problemlerin çözümünde kullanılan, YBS algoritmasını içeren program Microsoft Visual Studio 2008 programında C# programlama dili ile nesne yönelimli programlamaya uygun olarak kodlanmıştır. Problem çözümleri Intel Core 2 Duo CPU (T5800) 2.0 GHz işlemcili, 2 GB Ram bulunan bir bilgisayarda yapılmıştır.

Permütasyon tipi iş akış çizelgeleme problemlerinde, tüm işlerin ve tüm makinelerin t = 0 anında hazır oldukları kabul edilir. Bir iş bir makineye atandığında, bitirilmeden aynı makinede başka bir iş yapılamaz. Makinelerin hazırlanma süreleri işlem süresinin içindedir ve işlerin makineler arasındaki taşıma süresi ihmal edilir.

4.1. Geliştirilen Yazılımda Kullanılan Parametreler

Klonal Seçim Algoritması nispeten düşük bir karmaşıklığa sahiptir ve az sayıda kullanıcı tanımlı parametresi vardır. Bunlar; antikor popülasyonun boyutu, hafıza popülasyonunun boyutu, seçim havuzunun boyutu, kalan yer değiştirme boyutu, klonal faktör ve jenerasyon sayısı olarak sıralanabilir.

Antikor popülasyonun boyutu (N): Uygulama tarafından kullanılacak olan toplam antikor sayısını belirtir. Hafıza ve kalan antikorlardan oluşan sıralı antikor havuzudur. Hafıza antikor havuzunu α kabul edersek, hafıza havuzunun boyutu $\alpha \leq N$ olarak tanımlanmalıdır. Böylece kalan antikor havuzunun boyutu $r = N - \alpha$ şeklinde hesaplayabiliriz.

Seçim havuzunun boyutu (n): Her bir antijenin gösteriminde klonlamak için antikor popülasyonundan alınan en yüksek duyarlılıklı antikorların toplam sayısını belirtir. Seçim havuzunun boyutu $n \leq N$ olarak tanımlanmalıdır. Seçim havuzunun boyutunun yüksek olması popülasyonun çeşitliliğini artırabilir.

Kalan yer değiştirme boyutu (d): Her bir antijen uygulamasında üretilen rastgele antikorlar ile en düşük duyarlılıklı antikorların toplam sayısını belirtir. Burada kalan yer değiştirme boyutu $d \leq r$ olmalıdır. Kalan antikor havuzunun boyutunu r temsil etmektedir. Uygulamanın çalışmasına ek bir çeşitlilik getirmek için kullanılmıştır. Değeri sıfıra çekilerek iptal edilebilir.

Klonal Faktör (β): Seçilen antikorlarda oluşturulan klon sayısı için bir ölçekleme faktörünü gösterir. Genelde *β (0,1]* aralığında alınır. Bu değer ne kadar düşerse, algoritma tarafından yerel alanda o kadar daha fazla arama işlemi gerçekleştirilir.

Jenerasyon sayısı (G): Yürütülecek algoritmanın iterasyon sayısını gösterir. Bir iterasyonda algoritma bir tur çalışır. Bu parametre öğrenme miktarını kontrol eder. Jenerasyon sayısını yükseltmek, sistemde problemin aşırı öğrenilmesiyle sonuçlanabilir veya yerel bir optimumda takılı kalabilir.

4.2. Geliştirilen Yazılımın Uygulanması

Uygulamada dosya aç butonuyla harici dosyadan txt formatında hazırlanmış olan deney setleri yüklenmektedir. Tur sayısı, antikor sayısı, hafıza havuzu, seçim havuzu ve beta değerleri girilerek YBS&FRB3 modeli ile veya yalnız YBS modeliyle çalışabilmektedir.

Eğer yerel arama algoritması seçeneği seçilirse, başlangıç popülasyonu FRB3 yerel arama algoritması yardımıyla iyileştirilerek, klonlama ve hipermutasyon işlemine geçilmektedir. Ardından klonlanan antikorlar da FRB3 yerel arama algoritmasıyla iyileştirilerek, sonlanma kriteri devam edene dek algoritma devam etmektedir.

Seçilen antikorların her birinden oluşturulan klonların sayısı, uygunluk değerleriyle orantılıdır. Bunu başarmak için öncelikle seçilen antikorlar, uygunluk değerlerine göre artan düzende sıralanır. Daha sonra sıralı liste kullanılarak her bir antikor için oluşturulan klonların sayısı Denklem 4.1'e göre hesaplanır

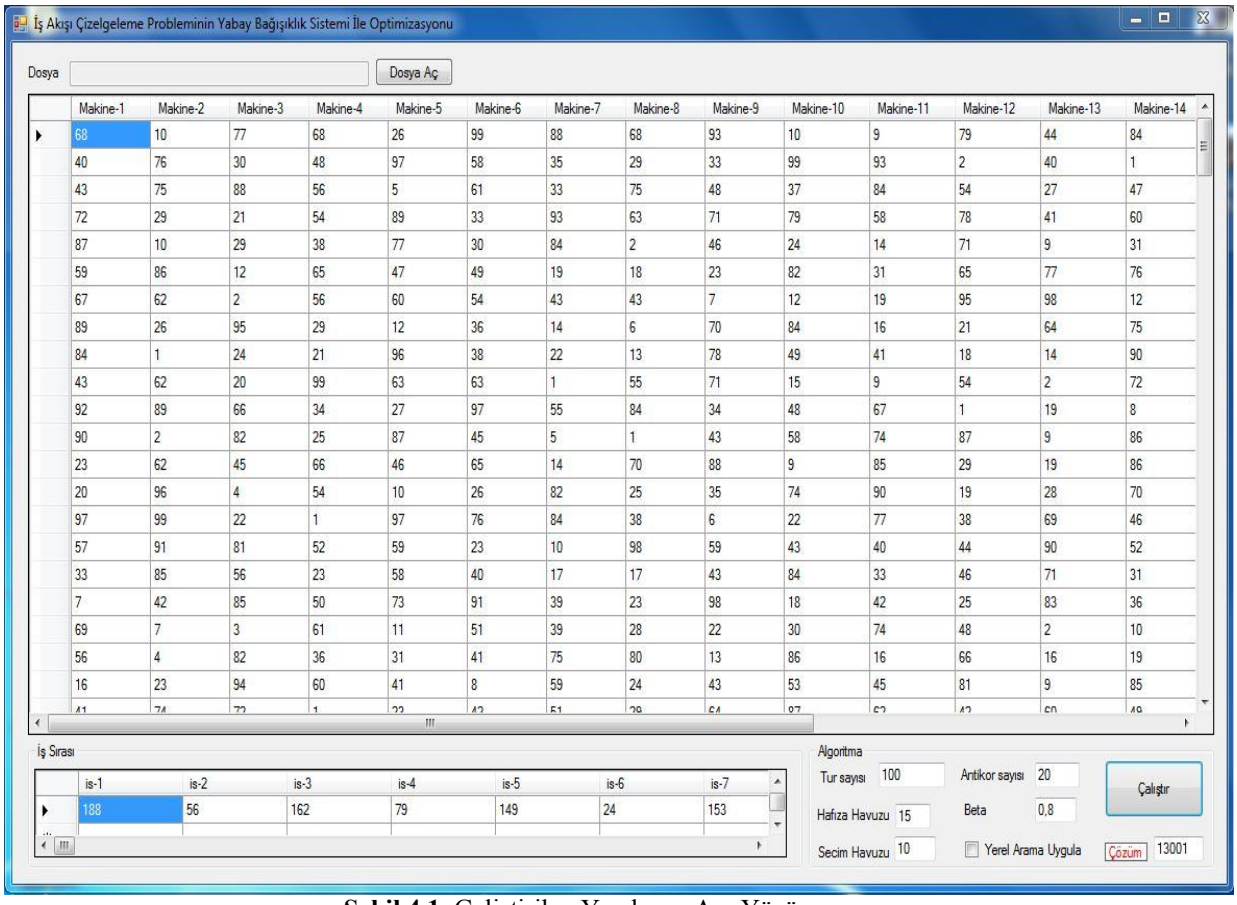
$$Klon\ sayısı = \left[\frac{\beta x N}{i} + 0{,}5\right] \tag{4.1}$$

Ardından hipermutasyon işlemine geçilir. Uygulamada kullanılan hipermutasyon işleminde Denklem 4.2'de, verilen formül ile antikor içindeki parametrelerden kaç tanesinin mutasyona uğratılacağına karar verilir. Daha sonra, seçilen parametreler uygun aralıklarda rastgele üretilecek yeni parametre değerleri ile yer değiştirilir.

$$Round\left[\left(e^{\frac{-10.f(i)}{\max(f)}}\right)xn\right] \tag{4.2}$$

Burada $f(i)$ üzerinde çalışılan antikorun uygunluğudur. 10 sabittir. n problemdeki değişken sayısıdır. $max(f)$ hesaplanan fonksiyon değerlerinden en büyüğüdür.

Bu işlemler durma kriteri sağlanana kadar devam etmektedir. Geliştirilen yazılımın ara yüzü Şekil 4.1'de görülmektedir.



Şekil 4.1. Geliştirilen Yazılımın Ara Yüzü

4.3. İş Akış Çizelgeleme Problemi İçin Yapılan Denemeler

YBS algoritması modelinin test edilmesi amacıyla, literatürde yer alan, Eric Taillard tarafından hazırlanmış 100 adet örnek problem üzerinde 20'şer deneme yapılmıştır. Taillard tarafından oluşturulan tablolarda 20, 50, 100, 200, 500 iş ve 5, 10, 20 adet makine kullanılmıştır. Örnek problemlerin gösterimi (iş sayısı*makine sayısı) şeklinde gösterilmiştir.

Simülasyon çalışmalarında kullanılan YBS parametrelerinin değerleri Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1. Uygulamada kullanılan parametre değerleri

	YBS	YBS&FRB3
Tur Sayısı	100	10
Antikor Sayısı	20	20
Hafıza Havuzu	15	15
Seçim Havuzu	10	10
Beta(Klonal Faktör)	0,8	0,8

Çözülen problemlerdeki iş sayıları (n), makine sayıları (m), hem YBS algoritmasıyla hem de FRB3 Yerel Arama Algoritmasıyla birlikte melez çalıştırılmıştır. Her bir problem için 20'şer çözümden sonra bulunan ortalama sonuçlar, Taillard'ın tablolarında yer alan en iyi sonuçlarla kıyaslanarak hesaplanmış ortalama sapma değeri yüzdesi (yüzde fark) yer almaktadır. Yüzde fark değeri Denklem 4.3 bağıntısı ile hesaplanmıştır.

$$RPD = \frac{Heu_{sol} - Best_{sol}}{Best_{sol}} \times 100 \qquad (4.3)$$

$Best_{sol}$: Taillard tablolarındaki problemlerin bugüne kadar hesaplanmış en kısa çözüm süresi.

Heu_{sol}: İşlerin makinelerde işlendikten sonra geçen, önerilen YBS&FRB3 algoritması içinde hesaplanan toplam işlem süreleridir.

Taillard'ın örnek problemlerine göre oluşturulmuş en iyi çözümlere göre hesaplanan ve iş ve makine sayısına göre değişim gösteren yüzde fark değerleri aşağıdaki çizelgelerde gösterilmiştir.

Çizelge 4.2. 20x5(20 iş 5 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

20*5	**$Best_{sol}$**	**YBS**		**YBS&FRB3**	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai01	1278	1319,2	3,22	1278	0,00
Tai02	1359	1398,5	2,91	1359,55	0,04
Tai03	1081	1238,6	14,58	1093,4	1,15
Tai04	1293	1454,35	12,48	1298,4	0,42
Tai05	1235	1342,7	8,72	1239,1	0,33
Tai06	1195	1326,55	11,01	1195,85	0,07
Tai07	1234	1321,55	7,09	1243,95	0,81
Tai08	1206	1348,2	11,79	1207,35	0,11
Tai09	1230	1371,35	11,49	1233,65	0,30
Tai10	1108	1241,55	12,05	1110,85	0,26

Çizelge 4.2'ye göre; 20*5 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 20*5*Tai01 probleminde elde edilmiş olup YBS&FRB3 modeli en iyi çözümü elde etmiştir. Enyüksek hata oranına sahip sonuç ise 20*5*Tai03 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 14,58'tir.

Çizelge 4.3. 20x10(20 iş 10 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

20*10	**$Best_{sol}$**	**YBS**		**YBS&FRB3**	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai11	1582	1812,1	14,54	1589,8	0,49
Tai12	1659	1873,85	12,95	1672,95	0,84
Tai13	1496	1711,7	14,42	1509,35	0,89
Tai14	1377	1590,1	15,48	1386,7	0,70
Tai15	1419	1652,2	16,43	1427,6	0,61
Tai16	1397	1603,35	14,77	1410,9	0,99
Tai17	1484	1681	13,27	1487	0,20
Tai18	1538	1758,8	14,36	1555,05	1,11
Tai19	1593	1771,15	11,18	1606,65	0,86
Tai20	1591	1798,05	13,01	1606,9	1,00

Çizelge 4.3'e göre; 20*10 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 20*10*Tai17 probleminde YBS&FRB3 modeli ile elde edilmiş olup yüzde fark = 0,20'dir. En yüksek hata oranına sahip sonuç ise 20*10*Tai15 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 16,43'tür.

Çizelge 4.4. 20x20(20 iş 20 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

20*20	**$Best_{sol}$**	**YBS**		**YBS&FRB3**	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai21	2297	2533,5	10,30	2307,75	0,47
Tai22	2099	2353,35	12,12	2104,25	0,25
Tai23	2326	2562,95	10,19	2345,6	0,84
Tai24	2223	2477	11,43	2229,5	0,29
Tai25	2291	2545,9	11,13	2306,15	0,66
Tai26	2226	2543,4	14,26	2239,9	0,62
Tai27	2273	2467,55	8,56	2281,7	0,38
Tai28	2200	2542,25	15,56	2224,05	1,09
Tai29	2237	2527,15	12,97	2243,3	0,28
Tai30	2178	2468,9	13,36	2188,35	0,48

Çizelge 4.4'e göre; 20*20 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 20*20*Tai22 probleminde YBS&FRB3 modeli ile elde edilmiş olup yüzde fark = 0,25'tir. En yüksek hata oranına sahip sonuç ise 20*20*Tai28 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 15,56'dır.

Çizelge 4.5. 50x5(50 iş 5 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

50*5	**$Best_{sol}$**	**YBS**		**YBS&FRB3**	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai31	2724	2877,55	5,64	2724	0,00
Tai32	2834	3055,1	7,80	2840,15	0,22
Tai33	2621	2822	7,67	2622	0,04
Tai34	2751	2985,5	8,52	2760,55	0,35
Tai35	2863	3033,3	5,95	2864	0,03
Tai36	2829	3036,8	7,35	2830,85	0,07
Tai37	2725	2935,65	7,73	2729,8	0,18
Tai38	2683	2914,45	8,63	2684,25	0,05
Tai39	2552	2753,05	7,88	2561	0,35
Tai40	2782	2964,25	6,55	2782	0,00

Çizelge 4.5'e göre; 50*5 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 50*5*Tai31 ve 50*5*Tai40 problemlerinde YBS&FRB3 modeli ile elde edilmiş olup optimum sonuçları elde etmiştir. En yüksek hata oranına sahip sonuç ise 50*5*Tai38 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 8,63'tür.

Çizelge 4.6. 50x10(50 iş 10 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

50*10	**$Best_{sol}$**	**YBS**		**YBS&FRB3**	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai41	2991	3522,75	17,78	3059,2	2,28
Tai42	2867	3405,7	18,79	2937,25	2,45
Tai43	2839	3410,85	20,14	2906,2	2,37
Tai44	3063	3544,1	15,71	3087,9	0,81
Tai45	2976	3513,25	18,05	3044,9	2,32
Tai46	3006	3512,8	16,86	3065,3	1,97
Tai47	3093	3554,5	14,92	3151,1	1,88
Tai48	3037	3481,05	14,62	3065,9	0,95
Tai49	2897	3431,55	18,45	2949	1,79
Tai50	3065	3531,3	15,21	3135,1	2,29

Çizelge 4.6'ya göre; 50*10 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 50*10*Tai44 probleminde YBS&FRB3 modeli ile elde edilmiş olup yüzde fark = 0,81'dir. En yüksek hata oranına sahip sonuç ise 50*10*Tai43 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 20,14'tür.

Çizelge 4.7. 50x20(50 iş 20 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

50*20	**$Best_{sol}$**	**YBS**		**YBS&FRB3**	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai51	3771	4547	20,58	3954,3	4,86
Tai52	3668	4411,45	20,27	3794,7	3,45
Tai53	3591	4345,75	21,02	3744,9	4,29
Tai54	3635	4413,25	21,41	3814,5	4,94
Tai55	3553	4385,95	23,44	3710,7	4,44
Tai56	3667	4390,25	19,72	3758,7	2,50
Tai57	3672	4443,5	21,01	3803	3,57
Tai58	3627	4434,75	22,27	3792,2	4,55
Tai59	3645	4425,55	21,41	3831,2	5,11
Tai60	3696	4490,55	21,50	3844,7	4,02

Çizelge 4.7'e göre; 50*20 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 50*20*Tai56 probleminde YBS&FRB3 modeli ile elde edilmiş olup yüzde fark = 2,50'dir. En yüksek hata oranına sahip sonuç ise 50*20*Tai55 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 23,44'tür.

Çizelge 4.8. 100x5(100 iş 5 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

100*5	$Best_{sol}$	YBS		YBS&FRB3	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai61	5493	5754	4,75	5493	0,00
Tai62	5268	5567,9	5,69	5284	0,30
Tai63	5175	5491,55	6,12	5180,6	0,11
Tai64	5014	5307,2	5,85	5018	0,08
Tai65	5250	5565,75	6,01	5250,5	0,01
Tai66	5135	5425,65	5,66	5135	0,00
Tai67	5246	5537,15	5,55	5247	0,02
Tai68	5094	5447,55	6,94	5101	0,14
Tai69	5448	5746,4	5,48	5454	0,11
Tai70	5322	5670,7	6,55	5328	0,11

Çizelge 4.8'e göre; 100*5 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 100*5*Tai61 ve 100*5*Tai66 problemlerinde YBS&FRB3 modeli ile elde edilmiş olup optimum sonuca ulaşılmıştır. En yüksek hata oranına sahip sonuç ise 100*5*Tai68 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 6,94'tür.

Çizelge 4.9. 100x10(100 iş 10 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

100*10	$Best_{sol}$	YBS		YBS&FRB3	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai71	5770	6517,3	12,95	5800	0,52
Tai72	5349	6125,8	14,52	5382	0,62
Tai73	5676	6329,75	11,52	5679	0,05
Tai74	5781	6571,2	13,67	5860	1,37
Tai75	5467	6254,95	14,41	5523	1,02
Tai76	5303	6033,2	13,77	5312	0,17
Tai77	5595	6220,05	11,17	5617	0,39
Tai78	5617	6296,7	12,10	5677	1,07
Tai79	5871	6503,1	10,77	5949	1,33
Tai80	5845	6457,75	10,48	5881	0,62

Çizelge 4.9'a göre; 100*10 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 100*10*Tai73 probleminde YBS&FRB3 modeli ile elde edilmiş olup yüzde fark = 0,05'tir. En yüksek hata oranına sahip sonuç ise 100*10*Tai72 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 14,52'dir.

Çizelge 4.10. 100x20(100 iş 20 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

100*20	$Best_{sol}$	YBS		YBS&FRB3	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai81	6106	7391,65	21,06	6393	4,70
Tai82	6183	7418,2	19,98	6333	2,43
Tai83	6252	7421,5	18,71	6415	2,61
Tai84	6254	7391,7	18,19	6387	2,13
Tai85	6262	7453,35	19,03	6458	3,13
Tai86	6302	7498,3	18,98	6519	3,44
Tai87	6184	7515,8	21,54	6421	3,83
Tai88	6315	7562,5	19,75	6589	4,34
Tai89	6204	7462	20,28	6437	3,76
Tai90	6404	7543,1	17,79	6564	2,50

Çizelge 4.10'a göre; 100*20 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 100*20*Tai84 probleminde YBS&FRB3 modeli ile elde edilmiş olup yüzde fark = 2,13'tür. En yüksek hata oranına sahip sonuç ise 100*20*Tai87 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 21,54'tür.

Çizelge 4.11. 200x10(200 iş 10 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

200*10	$Best_{sol}$	YBS		YBS&FRB3	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai91	10862	11773,9	8,40	10885	0,21
Tai92	10480	11692,35	11,57	10566	0,82
Tai93	10922	11919,65	9,13	11017	0,87
Tai94	10889	11703,75	7,48	10893	0,04
Tai95	10524	11705,4	11,23	10564	0,38
Tai96	10329	11453,5	10,89	10394	0,63
Tai97	10854	11927,25	9,89	10913	0,54
Tai98	10730	11859,65	10,53	10785	0,51
Tai99	10438	11601	11,14	10479	0,39
Tai100	10675	11748,5	10,06	10727	0,49

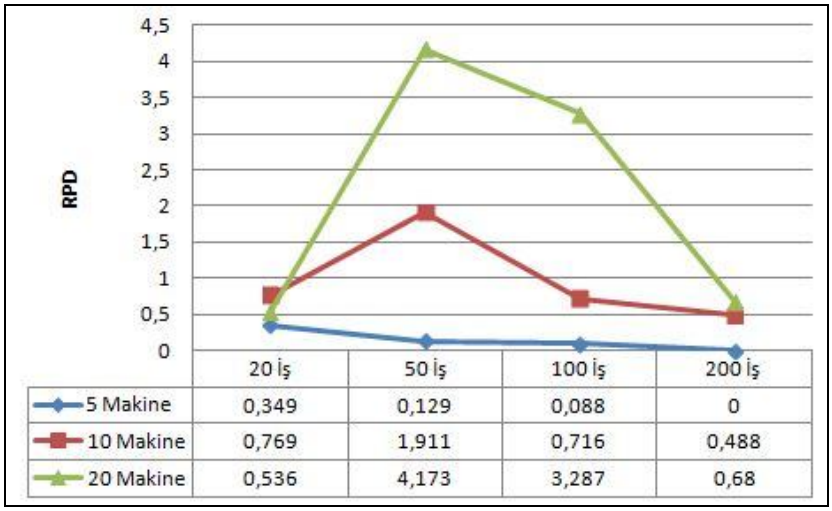
Çizelge 4.11'e göre; 200*10 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 200*10*Tai94 probleminde YBS&FRB3 modeli ile elde edilmiş olup yüzde fark = 0,04'tür. En yüksek hata oranına sahip sonuç ise 200*10*Tai92 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 11,57'dir

Çizelge 4.12. 200x20(200 iş 20 makine) için elde edilen sonuçlar ve yüzdelik fark değerleri

200*20	$Best_{sol}$	YBS		YBS&FRB3	
		Heu_{sol}	RPD	Heu_{sol}	RPD
Tai101	11152	13039,25	16,92	11165	0,12
Tai102	11143	13216,7	18,61	11175	0,29
Tai103	11281	13289,6	17,81	11293	0,11
Tai104	11275	13201,2	17,08	11325	0,44
Tai105	11529	13158	14,13	11640	0,96
Tai106	11176	13198,3	18,10	11247	0,64
Tai107	11337	13309,15	17,40	11459	1,08
Tai108	11301	13186,05	16,68	11468	1,48
Tai109	11145	13114	17,67	11270	1,12
Tai110	11284	13193,25	16,92	11347	0,56

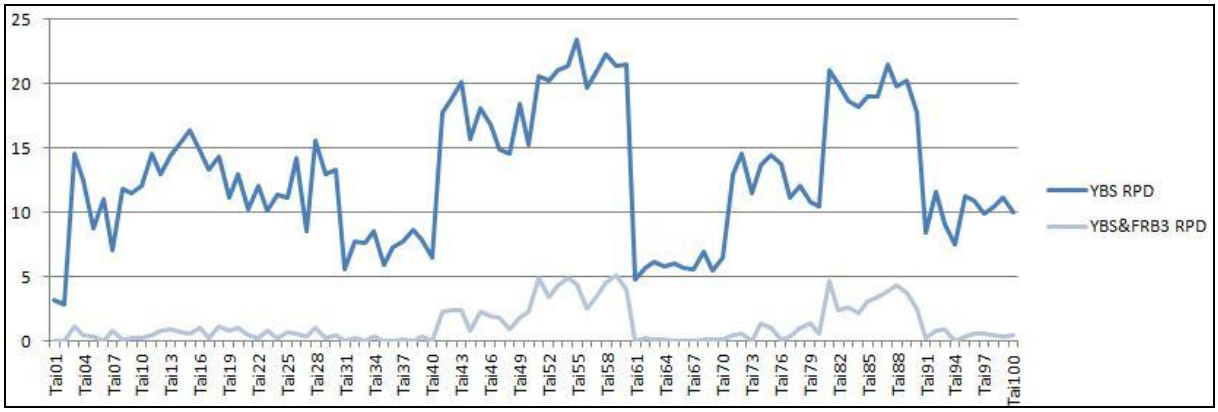
Çizelge 4.12'ye göre; 200*20 için elde edilen sonuçlar arasında en düşük hata oranına sahip sonuç 200*20*Tai103 probleminde YBS&FRB3 modeli ile elde edilmiş olup yüzde fark = 0,11'dir. En yüksek hata oranına sahip sonuç ise 20*20*Tai102 probleminde YBS modeli ile elde edilmiş olup yüzde fark = 18,61'dir.

Önerilen algoritmanın iş makine bazında hata oranlarını gösteren grafik, Şekil 4.2'de gösterilmiştir.



Şekil 4.2. YBS&FRB3 melez algoritmanın iş makine bazında hata oranı

Şekil 4.3'de YBS ile elde edilen sonuçlarla, YBS&FRB3 melez algoritması ile elde edilen sonuçların RPD değerlerini karşılaştıran grafik verilmiştir. Önerilen melez algoritmada sonuçlar, YBS algoritmasına göre daha iyi sonuçlar vermiştir.



Şekil 4.3. YBS ile YBS&FRB3 algoritmalarının hata oranlarının karşılaştırılması

Çizelge 4.13'te önerilen melez algoritma, literatürdeki diğer sonuçlarla karşılaştırılmıştır. Koyu renkle gösterilenler hata oranı en düşük değerlerdir. İtalik olarak gösterilenler hata oranı en yüksek olan değerlerdir.

Çizelge 4.13. YBS&FRB3 melez algoritmanın literatürdeki diğer sonuçlarla karşılaştırılması

	YBS&FRB3	**ACA[1]**	**Palmer[2]**	**GA[3]**	**SA[4]**	**KDS[5]**	**MMAS[6]**	**PACO[7]**
n/m	RPD	RPD	RPD	RPD	RPD	RPD	RPD	RPD
20/5	**0,35**	1,19	*10,81*	1,61	1,27	1,46	0,41	0,70
20/10	0,77	1,70	*15,27*	2,29	1,71	2,02	**0,59**	0,84
20/20	0,54	1,60	*16,34*	1,95	0,86	1,10	**0,41**	0,72
50/5	0,13	0,43	*5,23*	0,45	0,78	0,79	0,14	**0,09**
50/10	1,91	1,89	*13,48*	2,28	1,98	3,21	2,19	**0,75**
50/20	4,17	2,71	*15,46*	3,44	2,86	3,90	2,47	**1,85**
100/5	0,09	0,22	*2,38*	0,23	0,56	0,76	0,19	**0,07**
100/10	0,72	1,22	*9,08*	1,25	1,33	2,69	0,93	**0,40**
100/20	3,29	2,22	*13,24*	2,91	2,32	3,98	2,23	**0,99**
200/10	**0,49**	0,64	*4,75*	0,50	0,83	3,81	X	X
200/20	**0,68**	1,30	*6,72*	1,35	1,74	6,07	X	X
Ort	1,19	1,42	*10,25*	1,66	1,48	2,71	1,06	**0,71**

[1] Ant Colony Algorithm (Karınca Kolonisi Algoritması) (Ying ve Liao, 2004)
[2] (Palmer, 1965)
[3] Genetik Algoritma(Ying ve Liao, 2004)
[4] Simulated Annealing (Tavlama Benzetimi) (Ying ve Liao, 2004)
[5] Komşu Düğüm Seçimi (Ahmadizar, Barzinpour ve Arkat, 2007)
[6] Min-Max Karınca Sistemi (Rajendran ve Ziegler, 2004)
[7] Paralel Karınca Sistemi (Rajendran ve Ziegler, 2004)

Önerilen melez algoritma; 20x5, 200x10 ve 200x20 örneklerinde en iyi sonucu elde ederken diğer örneklerde bazı algoritmaların gerisinde kalmıştır. Genel ortalamada ise, KDS, SA, GA, Palmer ve ACO algoritmalarından daha iyi sonuç verirken, PACO ve MMAS algoritmalarına yetişememiştir. Fakat MMAS ve PACO algoritmalarında 200x10 ve 200x20 problemleri çözülmemiştir.

Sonuç olarak, önerilen melez algoritma düşük parametrelerle çalıştırılmasına rağmen hem Yapay Bağışıklık Sistemi'ne göre daha iyi sonuçlar elde etmekte hem de literatürdeki diğer çözümlere göre daha iyi sonuçlara ulaşmaktadır.

5. SONUÇLAR VE ÖNERİLER

Yapılan yüksek lisans tez çalışmasında bir çok kombinatoriyel optimizasyon problemlerinde kullanılan yapay bağışıklık sistemleri kullanılmıştır. Yapay bağışıklık sistemleri, yapay sinir ağları ve genetik algoritmalara benzer şekilde, bazı doğal sistemlerden esinlenerek türetilmiş ve birçok alanda uygulaması bulunmaktadır.

Çalışmada öncelikle optimizasyon kavramı üzerinde durulmuş ardından çizelgeleme problemleri hakkında detaylı bilgi verilmiştir. Daha sonra çizelgeleme problemlerinde kullanılan bazı sezgisel yöntemlerden bahsedilmiştir. Bu sezgisel yöntemlerden yapay bağışıklık sistemi üzerinde durulmuştur.

Yapay bağışıklık sisteminin bazı eksiklikleri analiz edilerek, performansı artırmak amacı ile yapay bağışıklık sistemi, FRB3 algoritmasıyla iyileştirilmiş ve yeni bir yöntem önerilmiştir. Önerilen bu yeni yönteme YBS&FRB3 adı verilmiştir.

Uygulama alanı olarak permütasyon akış tipi üretim çizelgeleme problemleri incelenmiştir. Çalışılan problem NP-Zor tipte bir problemdir. Bu problemleri çözmek için geliştirilmiş bir yöntem bulunmamakta, daha çok sezgisel yöntemler kullanılmaktadır. Problem çözümünde kullanılan yöntemler çözüm uzayı içinde iyi çözümler arama mantığıyla çalışan sezgisel yöntemlerdir.

5.1 Sonuçlar

Problem çözümü için hem YBS hem de YBS&FRB3 hibrit yöntem kullanılmıştır. Elde edilen sonuçlar karşılaştırılmıştır. Çözülen permütasyon akış tipi üretim çizelgeleme problemlerinde geliştirilen yeni yöntemin, yapay bağışıklık sisteminden daha iyi sonuçlar verdiği görülmüştür.

Daha geniş bir ifadeyle literatürde test problemleri olarak kullanılan değişik boyutlardaki toplam 110 adet Taillard Problemi 20'şer kere çözülmüştür. Önerilen YBS&FRB3 algoritması ile 5 örnek için optimum sonuca ulaşılmış ve her örnekte YBS modelinden daha düşük yüzdelik hata oranına ulaşılmıştır. Bu çalışma ile yapay bağışıklık sisteminin FRB3 algoritmasıyla melez kullanılmasıyla daha başarılı sonuçlar elde ettiği gözlenmiştir. Ayrıca literatürdeki diğer sonuçlarla karşılaştırılmış, iyi sonuçlar verdiği görülmüştür.

Yapay bağışıklık sistemiyle 100 iterasyonda ve yüksek parametre değerleriyle denemeler yapılırken, geliştirilen melez algoritmada 10 iterasyonda ve düşük değerlerde çalıştırılmıştır. Buna rağmen daha iyi sonuçlar elde edildiği görülmüştür.

Parametre değerleri daha yüksek tutulduğu takdirde daha iyi sonuçlar elde edilebilir. Fakat çözüme ulaşma süresi uzamaktadır. Örneğin 20x5 boyutundaki problemlerin tur sayısını 10, antikor sayısını 20, hafıza havuzunu 15, seçim havuzunu 10, beta değerini 0.8 kabul edilerek 20 kez çalıştırılıp ortalamasının alınması 27sn sürerken; tur sayısını 100, antikor sayısını 200, hafıza havuzunu 150, seçim havuzunu 100, beta değerini 0.8 kabul edilerek bir kez çalıştırılmasında çözüme ulaşılan süre ortalama 1 dakika 32 saniye sürmektedir.

Yine problemin boyutunun büyümesi problemin çözüm süresini artırmaktadır. Örneğin 100x10 boyutundaki problemlerin yüksek parametrelerle çalıştırılmasında çözüme ulaşılan süre ortalama 1 gün 5 saat 57 dakika 47 saniye sürmektedir. Bu süreler bilgisayarın konfigürasyon özelliklerine göre de değişiklik gösterebilmektedir.

Genel olarak makine sayısı düşük olan problem guruplarında daha fazla optimum sonuç elde edilmekte veya daha düşük hata oranlarına ulaşılmaktadır. Makine sayısı arttıkça hata oranı da artmaktadır.

5.2 Öneriler

Çalışmada işletmelerde sıkça kullanılan permütasyon akış tipi üretim çizelgeleme problemi ele alınmıştır. Permütasyon akış tipi üretim çizelgeleme uygulamaları gerçek hayatta sıkça görülmektedir. Gerçek hayattaki üretim çizelgeleme sistemlerinde genellikle makineler eş değildir. Aynı işlemleri yapabilen fakat farklı işlem sürelerine sahip makineler kullanılabilir. Bundan sonraki çalışma olarak gerçek fabrika ortamındaki bir üretim çizelgeleme problemi çözmek için algoritma geliştirilebilir.

Önerilen yeni algoritmada olasılık belirleme ve seçim süreçlerinin iyileştirilmesi veya başka tekniklerle birleştirilmesi ile daha kaliteli çözüm veren algoritmalar elde edilebilir. Benzer çalışma diğer üretim ortamları(atölye tipi, esnek üretim, hücresel üretim, montaj hattı gibi) için önerilen algoritma üzerinde yapılabilir.

Daha yüksek parametrelerle çalıştırılarak daha iyi sonuçlar elde edilebilir. Geliştirilen melez algoritma üzerinde parametre optimizasyonu üzerinde çalışılabilir.

Farklı yerel arama algoritmalarla melez kullanılabileceği gibi klonlama ve mutasyon işlemlerinin iyilenmesiyle daha kaliteli sonuçlar elde edilebilir. Lin ve Ying (2013) önerdikleri RAIS (Revised Artificial Immune System) algoritmasıyla kullanılabileceği gibi Chung ve Liao (2013) önerdikleri IAIS algoritmasıyla birlikte kullanılarak da daha iyi sonuçlar elde edilebilir.

KAYNAKLAR

Aarts, E., Jan, K., 1989, Simulated annealing and boltzman machines*, John Wiley Sons pres*, England.

Abbas, A.K., Lichtman, A.H., Pober, J.S., 1994, Cellular and molecular immunology, *WB Saunders*, Philadelphia, 225-244.

Ada, G.L., Nossal, G., 1987, The clonal-selection theory, *Scientific American*, 257.2, 62.

Ahmadizar, F., Barzinpour, F., Arkat, J., 2007, Solving permutation flow shop sequencing using ant colony optimization, *Industrial Engineering and Engineering Management, IEEE International Conference on*. IEEE, 753-757

Akbal, E., 2007, Kablosuz Ağlarda Yapay Bağışıklık Sistemi Kullanılarak Saldırı Tespiti ve Güvenlik, Yüksek Lisans Tezi, *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elazığ

Aldowaisan, T., Allahverdi, A., 2003, New heuristics for no-wait flowshops to minimize makespan, *Computers & Operations Research*, 30.8,1219-1231.

Altındaş, M., 2011, Üretim çizelgeleme ve bir uygulama, Yüksek Lisans Tezi, Bozok Üniversitesi Sosyal Bilimler Enstitüsü, Yozgat.

Anonim1, 2014, Bağışıklık Sistemi, http://tr.wikipedia.org/wiki/Bağışıklık_sistemi [Ziyaret Tarihi: 25 Mayıs 2014]

Anonim2, 2014, Genetik Algoritma, http://tr.wikipedia.org/wiki/Genetik_algoritma [Ziyaret Tarihi: 25 Mayıs 2014]

Baker, K. R., Trietsch, D. 2009, Principles of Sequencing and Scheduling, *John Wiley*, New Jersey

Biroğul, S., 2005, Genetik Algoritma Yaklaşımıyla Atölye Çizelgeleme, Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara.

Broyden, C.G., 1965, A Class of Methods for Solving Nonlinear Simultaneous Equations, *American Mathematical Society*, Vol. 19, No. 92, 577-593

Brucker, P., 2007, Scheduling algorithms, *Springer*, Vol. 3, Berlin.

Carlier, J., 1978, Ordonnancements a contraintes disjonctives, *RAIRO-Operations Research-Recherche Opérationnelle*, 12.4, 333-350.

Chang, P.C., Chen, S.H., Liu, C.H., 2007, Sub-population genetic algorithm with mining gene structures for multiobjective flowshop scheduling problems, *Expert systems with applications*, 33.3, 762-771.

Cheng, T. C. E., Ding, Q., 2001, Single machine scheduling with step-deteriorating processing times, *European Journal of Operational Research,* 134.3, 623-630.

Chung, T.P., Liao, C.J., 2013, An immunoglobulin-based artificial immune system for solving the hybrid flow shop problem, *Applied Soft Computing,* 13.8, 3729-3736.

Cunkaş, M., 2010, 8061011007 Genetik Algoritmalar ve Uygulamaları, Selçuk Üniversitesi, Konya.

Cunkaş, M., 2010, Zeki Optimizasyon Algoritmaları, Sağ, T., Saraçoğlu, E., Selçuk Üniversitesi, Konya.

Çatal, E., 2009, Karınca Kolonileri Algoritması İle İş Akış Problemlerinin Optimizasyonu, Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli

Çetin, M., 2007, Gezgin satıcı örnek problemlerinin optimum sonuçlarının grid aracılığı ile hesaplanması, Yüksek Lisans Tezi, *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü*, Balıkesir

Çevikcan, E., Durmuşoğlu, M.B., Baskak, M., Paralel makinelerde ürün tasarım özellikleri ile iş çizelgelemenin bütünleştirilmesi, *Endüstri Mühendisliği Dergisi*, İstanbul, 22.4, 13-34

Dasgupta D., Forrest S., 1995, Tool Breakage Detection in Milling Operations using a Negative-Selection Algorithm, Technical Report CS95-5, Department of Computer Science, University of New Mexico.

De Castro L.N., Von Zuben F.J., 2000, The Clonal Selection Algorithm with Engineering Applications, *In The Proceedings of Genetic and Evolutionary Computation Conference (GECCO), Workshop on Artificial Immune Systems and Their Applications*, Las Vegas, USA, 36-37.

De Castro, L.N., Von Zuben F.J., 2001, aiNet: an artificial immune network for data analysis, *Data mining: a heuristic approach*, 231-259.

Engin, O., 2001, Akış tipi çizelgeleme problemlerinin genetik algoritma ile çözüm performansının artırılmasında parametre optimizasyonu, Doktora Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul

Engin, O., Döyen, A., 2004, A new approach to solve hybrid flow shop scheduling problems by artificial immune system, *Future Generation Computer Systems*, 20.6, 1083-1095.

Erdiller, A., Orbak, A.Y., 2003, Atölye Tipi Çizelgeleme ve Operasyon Sıralama Problemine Genetik Algoritma Tabanlı Bir Yaklaşım, *Proceedings of the 3rd National Manufacturing Research Symposium, ÜAS'2003*, 133-140

Eren, T., 2008, Farklı Geliş zamanlı Öğrenme Etkili Paralel Makineli Çizelgeleme Problemi, *Havacılık Ve Uzay Teknolojileri Dergisi*, Kırıkkale, 3.4, 37-46

Eren, T., 2013, Tek Makineli Çizelgelemede Genel Öğrenme Fonksiyonları: Optimal Çözümler, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, Denizli, 19.2, 76-80

Eren, T., Güner, E., 2005, İki ölçütlü paralel makineli çizelgeleme problemi: maksimum tamamlanma zamanı ve maksimum erken bitirme, *İstanbul Ticaret Üniversitesi V. Ulusal Üretim Araştırmaları Sempozyumu*, İstanbul, 219-223

Eroğlu, D.Y., Özmutlu, H.C., Köksal, S.A., 2013, Bölünebilir ve sıra bağımlı hazırlık süreli işler içeren, ilişkisiz paralel makine çizelgeleme problemi için genetik algoritma-dokuma tezgâhı çizelgeleme, *Refereed Research Tekstil Ve Konfeksiyon*, Bursa, 24.1, 66-73

Farmer, J.D., Packard. N.H., Perelson, A.S., 1986, The Immune System, Adaptation And Machine Learning, *Physica 22D*.

Fığlalı, A., Engin O., 2011, Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma Yardımı ile Çözümünde Uygun Çaprazlama Operatörünün Belirlenmesi, *Doğuş Üniversitesi Dergisi*, 3.2, 27-35.

Forrest S., Perelson A.S., Allen L., Cherukuri R., 1994, Self-Nonself Discrimination in A Computer, *In Proceedings of The IEEE Computer Society Symposium on Research in Security and Privacy Heldin Oakland*, California, May 16-18, 1994, 202-212. IEEE Comp. Society Press.

French, S. 1981, Sequencing and Scheduling, An Introduction to the Mathematics of the Job Shop, *Ellis Harwood Press*, England.

Garey, M.R., Johnson, D.S., Sethi, R., 1976, The complexity of flowshop and jobshop scheduling, *Mathematics of operations research*, 1.2, 117-129.

Glover, F., 1989, Tabu Search - Part I, *ORSA Journal on Computing, Summer*, Vol.1, No.3, 190-206

Gözen, Ş., 2007, Bulanık esnek akış tipi çok prosesli çizelgeleme problemlerinin genetik algoritma ve tavlama benzetimi ile çözümü, Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya

Graham, R.L., 1979, Optimization and approximation in deterministic sequencing and scheduling: a survey, *Annals of discrete mathematics*, 5, 287-326.

Gupta, J.N.D., 1988, Two-stage, hybrid flowshop scheduling problem., *Journal of the Operational Research Society*, 39.4, 359-364.

Güçlü, A., 2010, Beklemesiz Akış Tipi İş Çizelgeleme Problemlerinin Karınca Kolonileri Algoritması İle Çözümü, Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya

Günaydın, C., 2008, Beklemesiz Akış Tipi Çizelgeleme Problemlerinin Çözümünde Yapay Sinir Ağları Yaklaşımı, Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya

Hofmeyr S.A., Forrest S., 1999, Immunity by Design: An Artificial Immune System, *In The Proceedings of The Genetic and Evolutionary Computation Conference (GECCO)*, Orlando, Florida, USA, 1289-1296.

Holland, J.H., 1975, Adaption in natural and artificial systems.

Hoogeveen, J. A., Lenstra, J.K., Veltman, B., 1996, Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard, *European Journal of Operational Research*, 89.1, 172-175.

Kaban, Z., Diri, B., 2008, Yapay Bağışıklık Sistemleri ile Türkçe Metinlerde Tür ve Yazar Tanıma, *IEEE*, İstanbul

Karaboğa, D., 2011, Yapay Zeka Optimizasyon Algoritmaları, Nobel Kitap Dağıtım A.Ş., Ankara 1-222

Kaymaz, E.D., 2007, Yapay Bağışıklık Sistemi tabanlı K-Nn Sınıflandırma Algoritması İle Protein Örüntülerinin Hücredeki Yerleşim Yerlerinin Belirlenmesi, Yüksek Lisans Tezi, *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elazığ

Kellegöz, T., 2006, Toplam Geç Bitirme Zamanının En Küçüklenmesi Performans Ölçütlü Permütasyon Akış Tipi Çizelgeleme Problemlerinin Çözümünde Genetik Algoritma Yaklaşımı, Yüksek Lisans Tezi, *Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü*, Kırıkkale

Keskin, K., 2010, Beklemesiz Akış Tipi Çizelgeleme Problemlerinin Çok Amaçlı Genetik Algoritma İle Çözümü, Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya

Kim J., Bentley P., 1999, Negative Selection and Niching by An Artificial Immune System for Network Intrusion Detection, *In The Proceedings of International Conference on Genetic and EvolutionaryComputation Conference (GECCO)*, Morgan-Kaufmann, Orlando, Florida, USA, 149-158.

Kirkpatrick, S., Vecchi M.P., 1983, Optimization by simmulated annealing, *science*, 220.4598, 671-680.

Kodaz, H., 2007, Bilgi Kazancı Tabanlı Yapay Bağışıklık Tanıma Sistemi, Doktora Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya

Kurnaz, M.S., Kart, Ö., 2010, İş Akış Çizelgeleme Problemi Üzerinde NEH, FRB3 ve FRB4 Sezgisellerinin Karşılaştırılması, *Muğla Üniversitesi Akademik Bilişim'10 - XII. Akademik Bilişim Konferansı*, Muğla, 625-630

Laha, D., Sarin S.C., 2009, A heuristic to minimize total flow time in permutation flow shop, *Omega*, 37.3, 734-739.

Lakshmi, K., Vasantharathna, S.,2014, Gencos wind–thermal scheduling problem using Artificial Immune System Algorithm, *International Journal of Electrical Power & Energy Systems*, 54, 112-122.

Lenstra, J.K., Kan, A.H.G.R., Brucker, P., 1977, Complexity of machine scheduling problems, *Annals of discrete mathematics*, 1, 343-362.

Lin, S.W., Ying, K.C. 2013, Minimizing makespan in a blocking flowshop using a revised artificial immune system algorithm, *Omega*, 41.2, 383-389.

Matzinger P., 1994, Tolerance, Danger, and The Extended Family, *Annual Review of Immunology*, Vol. 12, 991-1045

McNaughton, R., 1959, Scheduling with deadlines and loss functions, *Management Science*, 6.1, 1-12.

Memiş, G., 2008, Yarı otomatik ders programı sistemi, Yüksek Lisans Tezi, *Başkent Üniversitesi Fen Bilimleri Enstitüsü*, Ankara.

Ogbu,F.A.,Smith,D.K., 1990, The application of simulated annealing algorithm to the solution of the n/m/cmax flow-shop problem, *Computers Operations Research*, 17, 3, 243-253, Great Britain.

Özşen, S., 2008, Biyomedikal Sınıflama Problemleri İçin Problem-Tabanlı Bir Yapay Bağışıklık Sisteminin Geliştirilmesi Ve Biyomedikal Sınıflama Problemlerine Uygulanması, Doktora Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya

Palmer, D. S., 1965, Sequencing jobs through a multi-stage process in the minimum total time--a quick method of obtaining a near optimum, *Operational Research Quarterly*, Vol. 16, No. 1, 101-107.

Pan, Q.K., Ruiz, R., 2014, An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem, *Omega*, 44, 41-50.

Pinedo, M. 2012, Scheduling: theory, algorithms, and systems, Springer, New York.

Polat, K., 2004, Özellik Seçme(FS) ile Yapay Bağışıklık Tanıma Sistemi(AIRS) kullanılarak medikal teşhise gidiş, Yüksek Lisans Tezi, *Selçuk Üniversitesi, Fen Bilimleri Enstitüsü*, Konya.

Qing-dao-er-ji, R., Wang, Y., 2012, A new hybrid genetic algorithm for job shop scheduling problem, *Computers & Operations Research,* 39.10, 2291-2299.

Qiu, X., Lau H.Y.K., 2013, An AIS-based hybrid algorithm with PDRs for multi-objective dynamic online job shop scheduling problem, *Applied Soft Computing*, 13.3, 1340-1351.

Qiu, X., Lau, H.Y.K., 2012, An AIS-based hybrid algorithm for static job shop scheduling problem, *Journal of Intelligent Manufacturing*, 1-15.

Rad, S. F., Ruiz, R., Boroojerdian, N., 2009, New high performing heuristics for minimizing makespan in permutation flowshops, *OMEGA, The International Journal of Management Science*, 331- 345.

Rajendran, C., Ziegler, H., 2004, Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs, *European Journal of Operational Research*, 155.2, 426-438.

Reeves, C.R., 1995, Modern Heuristic Techniques for Combinatorial Problems, *McGraw-Hill Book Company*, UK.

Ruiz, R., Maroto, C., Alcaraz, J., 2006, Two new robust genetic algorithms for the flowshop scheduling problem, *OMEGA, The International Journal of Management Science*, 34, 461–76.

Saraç, T., Sipahioğlu, A., 2008, Plastik enjeksiyon makinelerinin çizelgelenmesi problemi, *Endüstri Mühendisliği Dergisi*, Eskişehir, 20.2, 2-14

Seçme, G., 2006, Akış tipi çizelgeleme problemlerinin yapay sinir ağlarıyla modellenmesi, Yüksek Lisans Tezi, *Erciyes Üniversitesi Sosyal Bilimler Enstitüsü*, Kayseri

Serter, N., 1996a, Mikrobiyoloji, Akşit, F., Akgün, Y., Kiraz N., *T.C. Anadolu Üniversitesi Yayınları,* Eskişehir, Ünite 10, 147-162

Serter, N., 1996b, Mikrobiyoloji, Akşit, F., Akgün, Y., Kiraz N., *T.C. Anadolu Üniversitesi Yayınları,* Eskişehir, Ünite 12, 183-205

Sertkaya, C., 2009, Bilgisayar güvenliğinde yapay bağışıklık sisteminin kullanılması, Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya

Shiau, D.F., Cheng, S.C., Huang, Y.M., 2008, Proportionate flexible flow shop scheduling via a hybrid constructive genetic algorithm, *Expert Systems with Applications*, 34.2, 1133-1143.

Shmoys, D.B., Wein J, Williamson D.P., 1995, Scheduling parallel machines on-line, *SIAM Journal on Computing,* 24.6, 1313-1331.

Stützle, T., 1998, An ant approach to the flow shop problem, *In Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, Vol. 3, 1560-1564.

Şahan, S., 2004, Yapay bağışıklıkta yeni algoritmalar, Yüksek Lisans Tezi, *Selçuk Üniversitesi, Fen Bilimleri Enstitüsü*, Konya.

Taillard, E., 1990, Some efficient heu-ristic methods for the flow shop sequencing problem, *European Journal of Operational Research,* 47, 67–74.

Ulutaş, B., 2008, Dinamik Yerleşim Probleminin Çözümü için bir Klonal Seçim Algoritması ve Uygulamaları, Doktora Tezi, *Eskişehir Osman Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Eskişehir.

Yağmahan, B., Yenisey, M.M., 2006, Akış tipi çizelgeleme problemi için KKE parametre eniyileme, *İstanbul Teknik Üniversitesi Dergisi*, İstanbul, 5.2, 133-141

Yeşilyaprak, E., 2007, Application Of Q-Learning Algorithm To Bicriteria Dynamic Scheduling Problem, Yüksek Lisans Tezi, *Dokuz Eylül University Graduate School Of Natural And Applied Sciences, İzmir*

Ying, K.C., Liao, C.J., 2004, An ant colony system for permutation flow-shop sequencing, *Computers & Operations Research*, 31.5, 791-801.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ahmet Haşim YURTTAKAL
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Yalvaç-26.02.1986
Telefon : 5064995453
Faks : -
e-mail : ahyurttakal@hotmail.com

EĞİTİM

Derece		Adı, İlçe, İl	Bitirme Yılı
Lise	:	**Milli Piyango Anadolu Lisesi, Merkez, Isparta**	**2004**
Üniversite	:	**Süleyman Demirel Üni. T.E.F., Merkez, Isparta**	**2008**
Yüksek Lisans	:	**-**	
Doktora	:	**-**	

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2010-2012	**Artvin Çoruh Üniversitesi**	**Öğretim Görevlisi**
2012-2014	**Bozok Üniversitesi**	**Öğretim Görevlisi**

UZMANLIK ALANI

C#, ASP.NET, SQL

YABANCI DİLLER

İngilizce

YAYINLAR

-